The 7th APSCE International Conference
on Computational Thinking
and STEM Education 2023

CTE-STEM

2023.6.7-6.9

Zhongli, Taiwan

Conference Proceedings

CTE-STEM

The 7th APSCE International
Conference on Computational Thinking
and STEM Education 2023

Programme Handbook of International Conference on Computational Thinking and STEM Education 2023 (CTE-STEM 2023)

7th– 9th June 2023

NATIONAL CENTRAL UNIVERSITY, TAIWAN

Organized by
Asia-Pacific Society for Computers in Education (APSCE)

Publication of Asia Pacific Society for Computers in Education
ISSN 2664-5661

ABOUT THE NATIONAL CENTRAL UNIVERSITY (NCU), TAIWAN

National Central University (NCU) is an university with long-standing traditions in Taiwan. Founded in Nanjing in 1915, NCU was the leading academic center in the Southeast China; the phrase "North the Peking University; South the Central University" at that time revealed the significance of NCU. NCU was later re-established in Taiwan in 1962 and started the development on the basis of Geophysics. After our endeavor over 50 years, NCU has expanded its school size and had great achievements both in academic and research development. NCU is now one of the leading universities in Taiwan.

In addition to constantly making breakthroughs in academic fields, NCU puts emphasis on holistic education, providing students with a platform for self-learning and the realization of their creativity. We also conduct industry-university collaboration and interaction to enable students to engage themselves in autonomous learning, to equip them with professional skills, and to help them develop multiple interests during their study at NCU.

NCU is a campus of tranquility with numerous pine trees scattering over the place. On the wave of global green economy, NCU builds an environment-friendly green campus based on our solid foundations of sustainable development. Meanwhile, the first Kunqu Museum in Taiwan will be open to the public in 2017. The museum symbolizes the abundant resources in humanities and also builds a bridge between the humanities and sciences for the faculty and students.

"Sincerity in knowledge; simplicity in life" is the motto of NCU, and it is the spirit that we expect all our students to keep in mind for a lifetime. In addition to achieving outstanding performances in their professions, students should be able to think reflexively, fit into the society maturely and considerately, and keep the down-to-earth attitude of NCU. We aim to lead our faculty and students to create a learning environment with both strong humanistic concerns and academic research on this beautiful and culturally-rich campus, to provide students with global vision and diverse learning experiences, and to become one of the world's top universities with unique characteristics.

For more information, please visit: https://www.ncu.edu.tw/en/index.php

Asia-Pacific Society for Computers in Education

The **Asia-Pacific Society for Computers in Education (APSCE)** was formed on 1 January 2004. It is an independent academic society whose broad objective is to promote the conduct and communication of scientific research related to all aspects of the use of computers in education, especially within the Asia-Pacific.

The specific objectives of APSCE are:

- To promote the conduct and dissemination of research employing the use of computing technologies in education within the Asia-Pacific region and internationally.
- To encourage and support the academic activities of researchers in member countries and to nurture a vibrant research community of younger as well as more experienced researchers.
- To enhance international awareness of research conducted by researchers in member countries.
- To obtain greater representation of active researchers from the Asia-Pacific region in committees of related leading academic and professional organizations and the editorial boards of reputable journals.
- To organize and hold the International Conference on Computers in Education (ICCE) conference series in member countries.
- To engage in other appropriate academic and professional activities including but not limited to the setting up of Special Interest Groups (SIGs) and the publication of a Society newsletter and a Society journal.

For more information, please visit: https://new.apsce.net/

Preface

The 7th APSCE International Conference on Computational Thinking and STEM Education 2023 (CTE-STEM 2023) is organized by the Asia-Pacific Society for Computers in Education (APSCE). CTE-STEM 2023 is hosted by the National Central University, Taiwan (NCU). This conference continues from the success of the previous six international Computational Thinking conferences organised by the Education University of Hong Kong (EdUHK), CoolThink@JC in Hong Kong, National Institute of Education, Nanyang Technological University (NIE/NTU) and LDE-CEL at the TU Delft in the Netherlands. In this conference, we invite Computational Thinking and STEM researchers and practitioners to share their findings, processes and outcomes in the context of computing education or computational thinking.

CTE-STEM 2023 is a forum for worldwide sharing of ideas as well as dissemination of findings and outcomes on the implementation of computational thinking and STEM development. The conference will comprise keynote speeches, invited speeches, panel discussions, workshops and paper presentations. All accepted papers will be published in ISSN-coded proceedings. The International Teachers Forum is organized for teaching practitioners to share their practices in teaching Computational Thinking, Computing and STEM in the classroom. We believe bringing all these would create enriching experiences for educators and researchers to share, learn and innovate approaches to learning through Computational Thinking and STEM education.

On behalf of APSCE, NCU and the Conference Organizing Committee, we would like to thank all the invited panelists, the keynote and invited speakers, as well as paper presenters for their contribution to the success of CTE-STEM 2023.

We sincerely hope all of you will enjoy and be inspired from participating in and attending CTE-STEM 2023.

Conference Chair
Weiqin Chen, Oslo and Akershus University, Norway
International Program / Local Organizing Committee Chair
Ju-Ling Shih, National Central University, Taiwan
International Program Committee Co-Chair
Tak-Wai Chan, National Central University, Taiwan
Siu Cheung Kong, Education University of Hong Kong, Hong Kong
Looi Chee Kit, Education University of Hong Kong, Hong Kong

Main Theme and Sub-themes

“Computational Thinking and STEM Education” is the main theme of CTE-STEM 2023 which aims to keep abreast of the latest development of how to facilitate students’ computational thinking abilities and STEM development, in the context of computing education or computational thinking. The conference also aims to disseminate findings and outcomes on the implementation of CT development in school and STEM education. There are 19 sub-themes under CTE-STEM 2023, namely:

- Computational Thinking & Coding Education in K-12
- Computational Thinking & Unplugged Activities in K-12
- Computational Thinking & Subject Learning & Teaching in K-12
- Computational Thinking & Teacher Development
- Computational Thinking & IoT
- Computational Thinking & STEM/STEAM Education
- Computational Thinking & Data Science
- Computational Thinking & Artificial Intelligence Education
- Computational Thinking Development in Higher Education
- Computational Thinking & Special Education Needs
- Computational Thinking & Evaluation
- Computational Thinking & Non-formal Learning
- Computational Thinking & Psychological Studies
- Computational Thinking in Educational Policy
- STEM Learning in the Classroom
- STEM Activities in Informal Contexts
- STEM Education Policies
- STEM Pedagogies and Curriculum
- STEM Teacher Education and Professional Development

Paper Submissions to CTE-STEM 2023

The conference received a total of 37 submissions (16 full papers, 15 short papers and 6 poster papers) by 107 authors from 17 countries/regions (see Table 1)

Country/ Region	No. of Authors	Country/Region	No. of Authors
Belgium	1	Netherlands	9
Germany	1	New Zealand	1
Greece	1	Singapore	4
Hong Kong	21	South Korea	1
India	5	Spain	2
Indonesia	1	Sri Lanka	4
Israel	3	Taiwan	25
Japan	8	United States	14
Malaysia	6	Total	106

The International Programme Committee (IPC) is formed by 51 members and 7 co-chairs worldwide. Each paper with author identification anonymous was reviewed by at least three IPC Members or co-chairs. Meta-reviewers then made recommendation on the acceptance of papers based on IPC Members' reviews. With the comprehensive review process, 26 accepted papers are presented (8 full papers, 12 short papers and 6 poster papers) at the conference.

Keynote Speakers

Prof. Yasmin Kafai
Teaching, Learning, and Leadership Division –
University of Pennsylvania
Title: <u>Preparing the Next Generation of Computational Thinkers</u>

Abstract:

During the last decade, national initiatives around the world have introduced computing into K-12 education under the umbrella of computational thinking. While initial efforts have focused on computational thinking's relevance for college and career readiness, more recent efforts also include creative expression, social justice, and critical inquiry, leading to a reevaluation of what it means for learners to be computationally-literate in the 21st century. Currently, three framings for promoting computational thinking in K-12 education have been proposed, emphasizing either (1) skill and competency building, (2) creative expression and participation, or (3) social justice and reflection. While each of these emphases is valuable and needed, their narrow focus can obscure important issues and miss critical transformational opportunities for empowering students as competent, creative, and critical agents. In this talk, I suggest that these framings should be seen as complimentary and suggest a move towards computational literacies, thereby historicizing and situating computer science with respect to broader educational concerns and providing new directions for how schools can help students to actively participate in designing their digital futures.

Biography:

Yasmin Kafai is the Lori and Michael Milken President's Distinguished Professor at the University of Pennsylvania. She is a researcher and developer of tools, communities, and materials to promote computational participation, crafting, and creativity across K-16. Her book monographs include "Connected Code: Why Children Need to Learn Programming" (The MIT Press, 2014) and editions such as the upcoming "Constructionism in Context: The Art, Theory, and Practice of Learning Designs" (2019, The MIT Press). She co-authored the 2010 "National Educational Technology Plan" for the US Department of Education and the 2018 "Priming the Computer Science Teacher Education Pump" reports. Kafai earned a doctorate in education from Harvard University while working with Seymour Papert at the MIT Media Lab. She is an elected fellow of the American Educational Research Association and the International Society for the Learning Sciences.

Keynote Speakers

Prof. Aman Yadav
Educational Psychology & Educational Technology –
Michigan State University
Title: Computational Thinking in the Classroom: Teachers' Implementation Approaches across a Spectrum

Abstract:

Since Wing re-popularized Computational Thinking (CT) to bring computational tools and practices in primary and secondary school, researchers and educators have implemented CT multiple ways across number of disciplines. In this talk, Dr. Yadav will discuss what the goals of CT should be and the opportunities and pitfalls to integrate CT into content areas. Specifically, he will discuss how teachers see the relevance of CT to support their pedagogical goals and how they take up computational thinking within their instruction. The talk will draw upon several projects that have focused on supporting teachers to integrate computational thinking at the primary (ages 5-10) and middle school (ages 11-14). Dr. Yadav will use classroom examples to highlight teachers' implementation of CT and pedagogical tensions that emerge between CT and disciplinary practices. In addition, Dr. Yadav will also discuss the importance of connecting computational learning experiences to students' backgrounds, experiences, and interests rather than teaching CT isolated from students' lives.

Biography:

Dr. Aman Yadav is a Lappan-Phillips Professor of Computing Education in the College of Education and College of Natural Science at Michigan State University with extensive experience in research, evaluation, and teacher professional development. His areas of expertise include computer science education, problem-based learning, and online learning. His research and teaching focus on improving student experiences and outcomes in computer science and engineering at the K-16 level. His recently co-edited book, Computational Thinking in Education: A Pedagogical Perspective tackles how to integrate computational thinking, coding, and subject matter in relevant and meaningful ways. His work has been published in several leading journals, including ACM Transactions on Computing Education, Journal of Research in Science Teaching, Journal of Engineering Education, and Communications of the ACM. Twitter (@yadavaman), website (http://www.amanyadav.org)

Keynote Speakers

Prof. Pasi Silander
Department of Teacher Education - University of Helsinki
Title: New Perspectives to AI and Computational Thinking Education via Phenomenon-based STEAM-projects: The Necessity of New Praxis for Epistemic Fluency

Abstract:

Modern societies rely heavily on advanced technologies, such as artificial intelligence (AI) and data analytics. In order to understand the role of automatic decision making and machine learning (ML) e.g. in social media, in economics, or in hybrid influencing, students will need computational thinking skills focused on AI.

Computational thinking focused on understanding the role of artificial intelligence and machine learning cannot adequately be learned through traditional methods used in schools. Therefore, there is an urgent need to rethink and redesign computer thinking education in K-12. Phenomenon-based learning is one of the most promising new pedagogical approaches and is widely used in schools in Finland. Phenomenon-based learning has been successfully implemented, for instance, in STEAM (science, technology, engineering, arts, math) education and in co-invention projects.

Computational thinking is not only important for computing, but it is also a highly generalized cognitive skill needed for critical thinking, media literacy, and knowledge production, as well as for comprehending ethical issues related to data-driven society and various aspects of AI and its ethically sustainable use.

The utilization of computational thinking in K-12 education is anchored in our conceptions of emerging digital technology, theories of learning, and technology-mediated practices of learning and teaching. It appears to us that computational thinking focused on AI requires a new level of epistemic fluency, interconnecting abstract and real-life phenomena by learners and teachers. When considering pedagogical applications of computational thinking in K-12 education, it is not enough to address mere programming or coding. The focus should be on modelling and understanding real-world phenomena by designing, creating, and utilizing abstractions and by creating algorithms, simulations and utilizing principles of machine learning. In addition, the focus of learning should be on systemic thinking, as in system theories or system design.

The major challenge of the K-12 educational system globally is to help students develop critical thinking skills and creative capabilities, especially related to understanding artificial intelligence and machine learning. In the digital world in which we live, computational thinking skills are a prerequisite for critical thinking and ensuring democracy.

Biography:
Pasi Silander, an educational futurist, is a computer scientist and an expert on digitalization. He also has a background in learning psychology and in pedagogy. He has worked long-time as a researcher and developer of eLearning, and he has created new innovative concepts, pedagogical models and design methods that are widely used in the Finnish education system. He is one of the original developers of phenomenon-based learning and teaching methods (PhenoBL).

The objects of research and development have included pedagogical leadership, digital transformation in education, phenomenon-based learning, STEM, innovative digital learning environments, as well as AI in education and Learning Analytics (LA). The research and development have taken place both in the business and public sector as well as in the research sector. In addition, he has authored many books focused on digital transformation in education and how to create the school of the future.

Silander has led the digitalization process of the Helsinki City school district (including around 120 schools), a systemic development process of new digital learning and teaching culture. Currently he is leading the development artificial intelligent and learning analytics for education.
Website: www.phenomenaleducation.info

Keynote Speakers

Prof. Chun-Yen Chang
Science Education Center - National Taiwan Normal University

Title: Bilingual STEM Education for Global Competitiveness in New Asia

Abstract:

To forge Taiwan's global competitiveness, the government released a "Bilingual 2030" policy, which encompasses the needs of accelerating bilingual higher education, optimizing bilingual conditions for primary and secondary schools, developing digital learning, and expanding affordable English proficiency tests. Funded and fueled by the Bilingual 2030 policy, a network project collaborating with University College London, Institute of Education (UCL-IOE) has been conducted by NTNU to establish a series of academic exchange activities. The primary theme of this project is "STEM education for global citizenship," focusing on improving students' computational thinking, mathematical understanding, and language acquisition. Previous studies by UCL-IOE have revealed that well-constructed STEM programs (ScratchMaths & Cornerstone Maths) can utilize the strengths of computational training to benefit specific mathematical concepts such as algebra, geometry, and ratio. To seek promising curriculum units suitable for adaption to a bilingual context, we operated a sequence of computer-based tasks derived from the ScratchMaths curriculum developed by U.C.L. Knowledge Lab. We explore whether dynamic mathematical technology, teacher materials, and professional development will enable teachers and students to grasp interdisciplinary knowledge and achieve deep learning. The intervention significantly improved students' computational knowledge (programming) and mathematical concepts (geometry). Students' motivations for learning STEM in the bilingual context were also fostered. The initial study's main implication demonstrates the potential of bilingual STEM tasks to provide a scaffolding for learners to exploit multilingual resources to consolidate mathematics and technological concepts and reinforce interdisciplinary interactions. In the future, our group will endeavor to develop sustainable bilingual STEM curriculum units with the consideration of more pedagogical elements, from boosting digital learning procedures to engaging teachers in professional development.

Biography:
Dr. Chang, a science education scholar at heart, currently serves at the National Taiwan Normal University (NTNU) as Chair Professor, Director of Science Education Center (NTNU), Professor of the Graduate Institute of Science Education and the Department of Earth Sciences (NTNU). Over the past few years, he has likewise been honored as a Visiting Professor at the Education University of Hong Kong as well as at Paris 8 University. His major research interests include science education, e-Learning, interdisciplinary science learning, and science communication.

Dr. Chang has authored and co-authored more than 150 articles, of which more than 125 are indexed in the Science/Social Science Citation Index (SCI/SSCI) database. He now is the Editor-in-Chief of three journals: (1) Eurasia Journal of Mathematics, Science and Technology Education; (2) European Journal of Mathematics and Science Education; (3) Educational Innovations and Emerging Technologies, as well as on the Editorial Board of three SSCI-level journals: (1) Studies in Science Education (science education); (2) Learning, Media & Technology (learning technology); (3) Journal of Science Education and Technology (science education & technology).

In February 2013, Dr. Chang's catechol-O-methyltransferase (COMT) study was privileged in a report by the New York Times Sunday Magazine and in the news featured on the Association of Psychological Science website. In 2019, the CouldClassRoom (C.C.R.) mobile system he and his research team developed was selected as an exemplar institution in the 2019 EDUCAUSE Horizon Report. For more information, see here.

Conference Organization

CTE-STEM Steering Committee

Siu Cheung Kong, Education University of Hong Kong, Hong Kong
Marcus Specht, Leiden Delft Erasmus, the Netherlands
Ronghuai Huang, Beijing Normal University, China
Hiroaki Ogata, Kyoto University, Japan
Weiqin Chen, Oslo and Akershus University, Norway
Jon Mason, Charles Darwin University, Australia

Conference Chair

Weiqin Chen, Oslo and Akershus University, Norway

International Program / Local Organizing Committee Chair

Ju-Ling Shih, National Central University, Taiwan

International Program Committee Co-Chair

Tak-Wai Chan, National Central University, Taiwan
Siu Cheung Kong, Education University of Hong Kong, Hong Kong
Looi Chee Kit, Education University of Hong Kong, Hong Kong

Local Organizing Committee Co-Chair

Ying-Tien Wu, National Central University, Taiwan
Tsung-Yen Chuang, National University of Tainan, Taiwan

Teacher Forum Committee Chair

Chang-Yen Liao, National Central University, Taiwan

International Programme Committee Members

AIVALOGLOU Efthimia	Delft University of Technology
BIST Ankur	Govind ballabh pant university of agri. and tech.
BOTICKI Ivica	FER UNIZG
BRINDA Torsten	University of Duisburg-Essen
CHANG Chi-Cheng	National Taiwan Normal University
CHANG Li-Chieh	National Central University
CHANG Shao-Chen	Yuan Ze University
CHEN Guang	Beijing Normal University

CHEN Ming-Puu	National Taiwan Normal University
CHENG Shu-Chen	Southern Taiwan University of Science and Technology
CHEON Jongpil Cheon	Texas Tech University
CHIANG Tosti H. C.	National Taiwan Normal University
CHOI Hyungshin	Chuncheon National University of Education
DAGIENE Valentina	Vilnius university Institute of Mathematics
DAVENPORT James H.	University of Bath
HAN Sungwan	GyeongIn National University of Education
HEINTZ Fredrik	Linköping University
HERSHKOVITZ Arnon	Tel Aviv University
HUANG Shu-Hsien	National Chin-Yi University of Technology
JEURING Johan	Utrecht University
KIM Mi Song	University of Western Ontario
KOHEN-VACS Dan	Holon Institute of Technology
KUO Bor-Chen	National Taichung University of Education
KURNIAWAN Oka	Singapore University of Technology and Design
KWOK Wai Ying	The Education Univeristy of Hong Kong
LE Nguyen-Thinh	Humboldt Universität zu Berlin
LI Robert K.Y.	City University of Hong Kong
LIEW Beng Keat	Republic Polytechnic
LIN Chang-Hsin	National University of Tainan
LU Bin	California State University Sacramento
MASON Jon	Charles Darwin University
MILRAD Marcelo	Linnaeus University
MOLLER Faron	Swansea University
OSMAN Kamisah	The National University of Malaysia
OTERO Nuno	University of Greenwich
ÖZÇINAR Hüseyin	Pamukkale University
RAISINGHANI Vijay	NMIMS
RAJENDRAN Ramkumar	IIT Bombay
SEOW Peter	National Institute of Education
SIN Kuen Fung	The Education Univeristy of Hong Kong
SIT Cindy	The Chinese University of Hong Kong
SONG Ki-Sang	Korea National University of Education
SPECHT Marcus	Delft University of Technology
TSAI Meng-Jung	National Taiwan Normal University
VAHRENHOLD Jan	Westfälische Wilhelms-Universität Münster
WADHWA Bimlesh	National University of Singapore

WU Longkai	National Institute of Education
WU Ying-Tien	National Central University, Taiwan
XU Li	University of Arizona
YANG Jie Chi	National Central University
YETER Ibrahim H.	Nanyang Technological University
YU Y.T.	City University of Hong Kong
ZEIGLER David	California State Unviersity, Sacramento
ZHANG Jinbao	Beijing Normal University

Table of Contents

FULL PAPER

SHORT PAPER

POSTER PAPER

Evaluating a Teacher Development Course in STEM with Artificial Intelligence Model Training: Problem-Solving Skills and Digital Creativity Development

Siu Cheung KONG *, Ye Aaron HU
The Education University of Hong Kong, Hong Kong
sckong@eduhk.hk , yhu@eduhk.hk

ABSTRACT

Many studies have highlighted the important role of STEM and Artificial Intelligence (AI) in K–12 education. However, not many studies have discussed the integration of STEM and AI in K–12 education. To fill this gap, this study investigates integrating STEM activities design with AI model training to promote simultaneous STEM and AI education. We collected data from a sample of 93 teachers before and after they attended a 6-hour teacher development course developed for the study. Among the 93 teachers, 85 completed the pre- and post-tests, which included concepts tests and the Technological Pedagogical Content Knowledge (TPACK) survey in four content-related areas. The results of a paired t-test indicated significant improvement with a medium effect size after the teachers attended the course. All 11 items of the TPACK survey yielded scores of between 3.62 to 4.32 out of 5. Moreover, 82.80% of the teachers demonstrated their digital creativity after attending the course, while 65.59% of the teachers demonstrated their understanding of AI concepts by applying them their creative ideas. One implication of the study is that promoting STEM and AI education together is feasible and can help teachers to do two things in one go.

KEYWORDS

Artificial Intelligence, Digital Creativity, Problem-Solving, STEM, Teacher Development

1. INTRODUCTION

‘STEM’ represents the integration of science, technology, engineering and mathematics (Yang, 2022). Integrating these subjects together makes STEM activities more meaningful (Council, 2012), increases students’ engagement and interest (Cotabish et al., 2013; Mohr-Schroeder et al., 2018) and improves students’ learning achievement (Barker & Ansorge, 2007; Nurlaely et al., 2017). The ubiquitous use of artificial intelligence (AI) in our daily life indicates the timeliness of considering educating all citizens about this subject (Touretzky et al., 2019). AI education in K–12 schools is currently very popular (Chiu, 2021). However, limited studies have explored the effectiveness of integrating STEM activities with AI model training in primary schools. Our study aimed to test the effectiveness of a teacher development course in fostering teachers’ ability and self-efficacy in teaching STEM integrated with AI.

2. LITERATURE REVIEW

2.1. STEM

According to Sullivan and Heffernan (2016), engaging in STEM-related activities can help students improve their problem-solving skills in four distinct ways: through casual reasoning, sequencing, conditional reasoning and engineering systems thinking. This study designed STEM activities focused on developing the participants’ Internet of Things (IoT) concepts and integrating AI with their problem-solving skills and digital creativity.

2.2. Internet of Things Concepts

Ashton (2009) first proposed the IoT concept, defining it as computers’ use of sensor technologies to monitor and comprehend the world without the assistance of humans. Al-Fuqaha et al. (2015) suggested that the functions of IoT are delivered by the following six elements: object identification, sensing, communication, computation, services and semantics. In this study, we adopted a simplified IoT concept as consisting of sensing, reasoning and reacting to design STEM activities in the primary school setting. We defined sensing as the process of detecting and transmitting data via sensors or a microcontroller with sensors, reasoning as using a processor and programming codes to process input through computation and predict system responses, and reacting as the final response of a system, in the form of offering services following communication among the computers, sensors and actuators.

2.3. AI

AI is an area of computer science that focuses on the creation of intelligent systems inspired by human intelligence and beyond (Sakulkueakulsuk et al., 2018). Kong and Zhang (2021, p. 12) defined ‘AI literacy’ as the ‘understanding of AI concepts and competencies in using AI concepts for evaluation and using AI concepts for understanding the real world’. In this study, we enabled primary students to experience AI data model training and use the model in a STEM artefact.

2.4. Integration of STEM with AI

The application of AI in STEM activities (AI-STEM), an emerging field, poses the challenge of integrating diverse AI techniques and complex educational elements to meet instructional and learning needs (Xu & Ouyang, 2022). In this study, we enabled primary school learners to experience AI model training and use it in a STEM artefact, thereby promoting AI literacy in STEM education.

2.5. Importance of Teacher Development

According to Wang and Cheng (2021), K–12 teachers face difficulties in attempting to teach activities that utilise AI, due to the teachers' lack of pedagogical experience with and knowledge of AI. In this study, teachers used Micro:bit, App Inventor and MakeCode programming environment to design STEM artefacts. The use of the AI model allows for a more human-friendly interaction with the STEM artefact, which helps teachers to understand the IoT and AI concepts and their relationship with daily life (Ghosh et al., 2018).

2.6. Technological Pedagogical Content Knowledge

Mishra and Koehler's (2006) Technological Pedagogical Content Knowledge (TPACK) framework assesses teachers' knowledge. A number of studies (e.g., Akyuz, 2018; Chai & Koh, 2017; Kong et al., 2020) have found that only some of the seven components of the TPACK framework are crucial for assessing teachers' performance. In this study, we used a modified TPACK framework with only the four components related to content knowledge: content knowledge' (CK), which refers to the IoT and AI concepts involved in STEM activities and the problem-solving skills involved in building the STEM system; 'technological content knowledge' (TCK), which refers to the understanding of the technological functions of each component – digital and non-digital – used for building the STEM system; 'pedagogical content knowledge' (PCK), which represents the blending of content and pedagogy into an understanding of how particular aspects of subject matter are organised, adapted and represented for instruction without using technology; and 'technological pedagogical content knowledge' (TPACK), which is the combination of CK, TCK and PCK in a learning context (Mishra & Koehler, 2006).

2.7. Current Study

This study investigated whether a teacher development course on teaching STEM with a focus on developing IoT concepts and integrating with AI can effectively establish teachers' skills and confidence in teaching STEM with AI. Therefore, we used a pre-/post-test design to assess IoT concepts and self-perceived problem-solving abilities before and after a related course. The course was developed by the research team and designed to foster problem-solving abilities and digital creativity through STEM activities. We proposed the following research questions:

a. Does the teacher development course significantly foster teachers' ability in teaching STEM with AI model training?

b. To what extent do teachers' have self-efficacy in teaching STEM with AI model training after completing the teacher development course?

c. Do teachers demonstrate digital creativity after completing the teacher development course?

3. METHOD

3.1. Participants

The sample consisted of 93 teachers from 45 primary schools in Hong Kong who will teach senior primary STEM activities in their schools. They were invited to complete a 6-hour professional development course. Among them, 85 teachers from 44 schools completed both pre- and post-tests of content knowledge as well as a TPACK survey, giving a response rate of 91.40%. Among these 85 teachers, 55 (64.71%) were male and 30 (35.29%) were female. The subjects' teaching experience varied from 1 to 30 years (*mean* = 11.54; SD = 7.84). The subjects they teach are shown in Table 1.

Table 1

Top Two Subjects Taught by Teachers in the Sample (N = 85)

Subject	No. of Teachers
Information Technology	54
Mathematics	51
General Studies	29
English Language	13
Chinese Language	8
Physical Education	5
Music	1
Other	1
Total	162

Note. Each teacher was asked to choose the one or two subjects that they teach the most.

3.2. Procedure

Before and after taking the teacher development course, the participants were given 10 minutes to complete content knowledge tests with the same set of questions. In addition, after attending the course, they were given 10 minutes to complete a TPACK survey on teaching STEM with a focus on developing IoT concepts and integrating the AI data model training (AIoT) into the lessons. They were also given another 10 minutes to write down their ideas regarding new AIoT applications, which should be different from the two examples used in the teacher development course. Each unit took 2.5 hours to complete and presented one example.

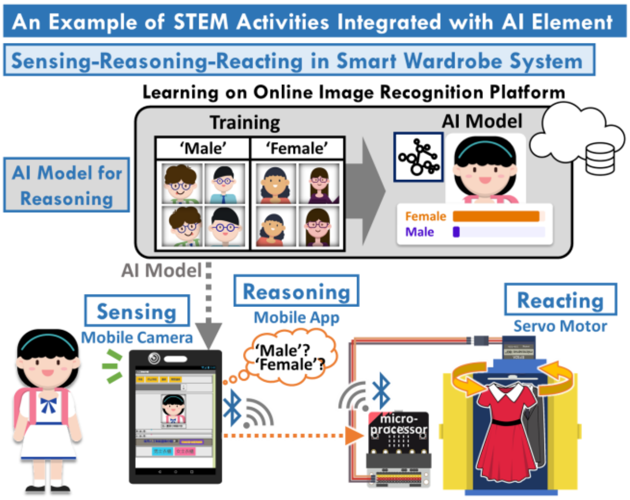
3.3. Teacher Development Course

The first unit introduced IoT concepts through the example of a remote-controlled greenhouse cooling system. In the processes of 'To Play' and 'To Inquire', the teachers gained an initial understanding of the IoT concept of sensing-reasoning-reacting when they play around with the STEM system and how the various components of the system express these concepts when they inquire about how each component works. In the system, a thermometer in microprocessor is responsible for sensing; an app on a mobile device is responsible for reasoning whether the temperature is high, say higher than 33 degrees Celsius; and the mobile app and a motor are responsible for reacting, specifically changing the background of the app to red and rotating the motor at a high speed when the temperature is high. Teachers dismantled the system into parts and were taught the function of each part with the support of worksheets through inquiry learning. The teachers then rebuilt the system using the ‘To STEM’ and ‘To Code’ processes. Through re-assembling the physical parts in the ‘To STEM’ process, the teachers developed their understanding of how the system works. They learnt that casual reasoning, sequencing, conditional reasoning and engineering systems thinking are needed in this re-assembling process to make the system work as designed. Thus, the first lesson imparted basic knowledge.

The second unit taught the concept of AI using the example of the development of a smart wardrobe, a system that provides a person with appropriate clothing by using face recognition to identify the person and rotating the wardrobe to the corresponding partition. Specifically, it presented male-oriented clothes if the system recognised a male face; and female-oriented clothes if it recognised a female face. In this unit, we worked with the teachers in AI model training to teach the AI to detect people’s faces and distinguish the category of users. We also taught the teachers how to extract the model and put it into the app. As in the previous session, the teachers played with and inquired about the system to understand how it works with the IoT concept of sensing-reasoning-reacting and with the AI data model. In the system, a camera on the mobile device is responsible for sensing, an AI data model embedded in the app is responsible for reasoning, and a servo motor is responsible for reacting according to the Bluetooth signal sent out by the mobile device and received by a microprocessor. The teachers then re-built the system to understand the relationships between its parts and practiced engineering systems thinking in the ‘To STEM’ and ‘To Code’ processes. Specifically, the teachers were required to solve various problems through causal reasoning when the system was not working as intended. In the ‘To Code’ process, the teachers applied sequencing to arrange coding blocks in the correct order and conditional reasoning to code for the system’s reactions to its sensing of different categories of images. After reflecting on the concepts and problem-solving skills in ‘To Reflect’, the teachers were guided ‘To Create’ by discussing and sharing their creative ideas for the design of other AIoT systems.

Figure 1

STEM System of the Smart Wardrobe, Including the AI Data Model Training



3.4. *Smart Wardrobe System*

The smart wardrobe system is an example of a STEM activity with AI element, as illustrated in Figure 1. In the system, a camera on a mobile device senses a user’s face according to an image the device captures. The mobile app then reasons out the category of user according to the pre-trained AI data model that uses the labels of ‘Male’ and ‘Female’. Through Bluetooth, the mobile app transmits data to a microprocessor according to the identified category. After receiving the data, the microprocessor instructs the servo motor to react and rotate in the appropriate direction and angle to show the appropriate category of clothes in the wardrobe.

For example, when the camera of the mobile device captures the characteristics of a female, the mobile app identifies the image as ‘Female’ and transmits a signal of ‘F’ to the microprocessor via Bluetooth. When the microprocessor receives this signal, it orders the servo motor to rotate to the angle of 180°, showing a dress to the user. The coding of the mobile application and the microprocessor is shown on Figure 2. Table 2 shows the four possible servo motor reactions according to the signal the microprocessor receives from the mobile app and the motor’s original position.

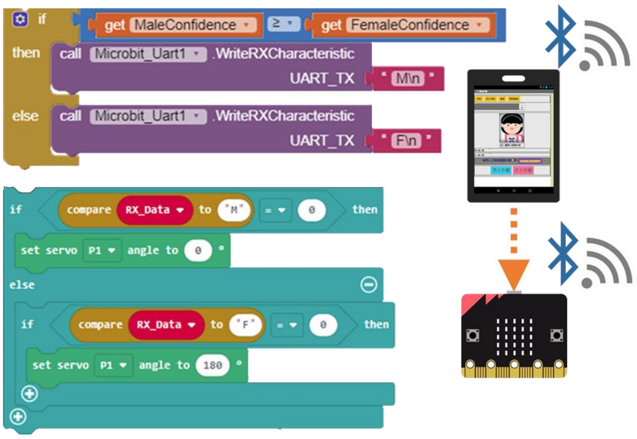
Table 2

Reaction of Servo Motor and Clothes Display According to Signal Received by Microprocessor

Signal Received	Original Angle of Servo Motor	Reaction of Servo Motor	Clothes Display
M	0°	Stay at 0°	Male
	180°	Rotate to 0°	
F	0°	Rotate to 180°	Female
	180°	Stay at 180°	

Figure 2

Coding Used in the Smart Wardrobe System



```
if get MaleConfidence ≥ get FemaleConfidence
then call Microbit_Uart1 .WriteRXCharacteristic
                                   UART_TX " M\n "
else call Microbit_Uart1 .WriteRXCharacteristic
                                   UART_TX " F\n "
```

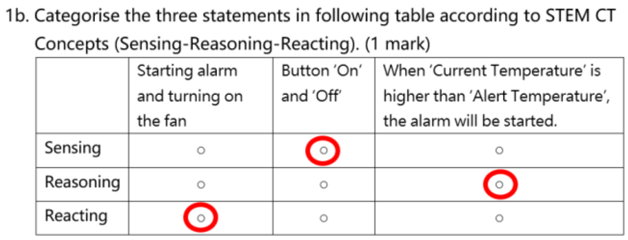
```
if compare RX_Data to "M" = 0 then
  set servo P1 angle to 0 °
else
  if compare RX_Data to "F" = 0 then
    set servo P1 angle to 180 °
```

3.5. *Instruments*

To assess the difference in teachers' ability to teach STEM after the course relative to their ability before the course, we prepared pre- and post-content tests. The test consists of nine items worth one point each, that assesses the teachers' problem-solving skills and their understanding of STEM concepts, namely two items regarding sequencing and coding, two items regarding causal reasoning, one item regarding conditional reasoning and coding, one item regarding engineering systems thinking and three items regarding sensing-reasoning-reacting. Figure 3 shows an example of an item assessing the teachers' understanding of the IoT concept of sensing-reasoning-reacting. Figure 4 shows an example of an item assessing the teachers' understanding of sequencing and the process of applying the AI model in the system.

Figure 3

Example of Test Item Assessing Understanding of the IoT Concepts of Sensing-Reasoning-Reacting

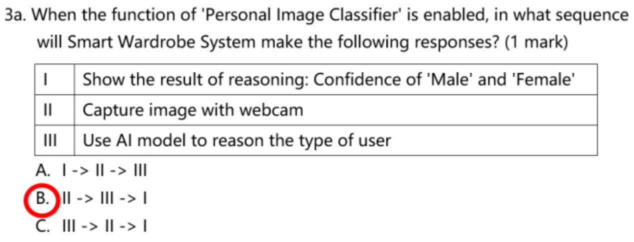


1b. Categorise the three statements in following table according to STEM CT Concepts (Sensing-Reasoning-Reacting). (1 mark)

	Starting alarm and turning on the fan	Button 'On' and 'Off'	When 'Current Temperature' is higher than 'Alert Temperature', the alarm will be started.
Sensing	○	[✓ ○]	○
Reasoning	○	○	[✓ ○]
Reacting	[✓ ○]	○	○

Figure 4

Example of Test Item Assessing Understanding of Sequencing and the Process of Applying AI Models



3a. When the function of 'Personal Image Classifier' is enabled, in what sequence will Smart Wardrobe System make the following responses? (1 mark)

I	Show the result of reasoning: Confidence of 'Male' and 'Female'
II	Capture image with webcam
III	Use AI model to reason the type of user

A. I -> II -> III
[✓ B.] II -> III -> I
C. III -> II -> I

To assess the teachers' self-efficacy in teaching STEM after the course, we developed a TPACK survey based on the model proposed by Mishra and Koehler (2006). The survey consists of 11 items related to content knowledge (CK, TCK, PCK, TPACK) relevant to implementing STEM activities with responses on a 5-point scale ranging from 1 (strongly disagree) to 5 (strongly agree).

After the course, the teachers were also required to design an IoT system with or without AI data model training. Two marks are awarded if the suggested designs include the sensing-reasoning-reacting concepts, one mark is given for designs that lacked detail and required elaboration and zero marks are given for designs with no proper suggestion of a new IoT or AIoT system or whose proposed system is identical to one of the systems discussed in the course.

3.6. *Analysis Procedure*

The test and survey data were processed and analysed using IMB SPSS 28 software. To investigate the effectiveness of the teacher development course, a paired t-test was conducted on the pre-test and post-test of the CK. To investigate the teachers' ability and confidence in STEM teaching, we calculated the mean and standard deviation (SD) of the TPACK survey.

4. RESULTS AND DISCUSSION

4.1. *Paired t-test on Pre-course and Post-course Content Knowledge Test*

The content knowledge test consists of nine items, each worth one point. The Cronbach's alpha for the test was α = .604. A paired sample t-test was conducted to determine the effect of the teacher development course on STEM teaching ability. The results indicate a significant difference between the total scores before (*mean* = 6.11; *SD* = 1.96) and after (*mean* = 7.59; *SD* = 1.31) the course (t (84) = 7.146, p < .001). The 95% confidence interval of the difference between the means ranges from 1.070 to 1.895, indicating a difference between the means of the samples. We, therefore, rejected the null hypothesis that there is no difference between the means and the associated conclusion that the training had no effect on the content test scores. Further, the Cohen's effect size (*d* = .775) suggests a medium level of practical significance (see Table 3).

Table 3

Paired t-test Results of the Pre-Course and Post-Course Scores on the Content Knowledge Test (N = 85)

Pre-test		Post-test			Effect Size
Mean	SD	Mean	SD	t-value	(Cohen's d)
6.11	1.96	7.59	1.31	7.146***	0.775

Note. *p<.05, **p<.01, ***p<.001.

4.2. *Results of the Post-course TPACK Survey*

The TPACK survey consists of 11 items, and the Cronbach's alpha of the survey is α = .950. The responses to all 11 items range between 3.62 and 4.32 out of 5 points (see Table 4). The results, presented in Table 4, show that after the course the teachers were confident in their abilities (concepts and problem-solving skills covered in the AI-integrated STEM activities) and using suitable pedagogy and technology tools to support their teaching.

Table 4

Results of the Post-course TPACK Survey (N = 85)

TPACK Item	Mean	SD
CK	4.03	0.62
I understand ‘sensing-reasoning-reacting’ in the operational process of IoT and related concepts, i.e., ‘sensing-reasoning-reacting’.	4.32	0.60
I have sufficient knowledge about STEM education in the IoT era.	3.85	0.79
I can use the computational thinking practices, such as sequencing, conditional reasoning, causal reasoning and engineering systems thinking, for problem-solving in the STEM activities.	3.92	0.73
PCK	3.87	0.72
I am capable of and willing to provide a complete STEM activity artefact for my students to play and to inquire.	3.94	0.75
I can identify and handle what learning difficulties students might have on technological innovation in STEM education.	3.80	0.78
TCK	3.79	0.75
I understand the functions that sensor, microprocessor, and actuator perform in the IoT systems.	3.81	0.76
I believe that the electronic parts of STEM teaching tools (e.g. Micro:bit, M5Stick), such as sensor, microprocessor, and actuator, can be used for fostering students’ digital creativity.	3.76	0.83
I can use STEM tools (e.g. Micro:bit, M5Stick) to organise STEM activities and foster students’ digital creativity.	3.73	0.81
TPACK	3.72	0.69
I can teach STEM lessons that appropriately combine the content of STEM, technological innovation and proper teaching approaches.	3.75	0.80
I can select and use technologies in my classroom that enhance what I teach, how I teach, and what students learn.	3.79	0.66
I can provide support and leadership in helping others to coordinate the use of STEM education, technological innovation, and teaching approaches at my school and/or district.	3.62	0.83

Note. All item responses are set on a 5-point scale ranging from 1 (strongly disagree) to 5 (strongly agree).

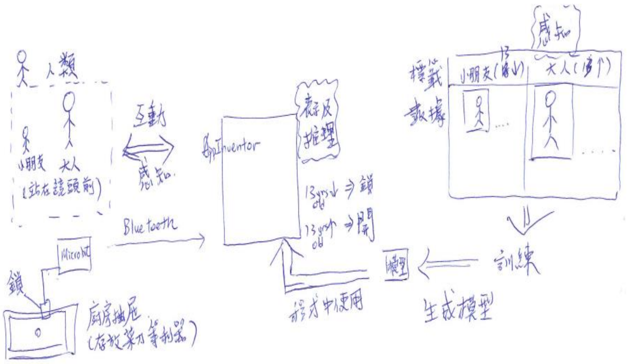
4.3. Evaluation of Digital Creativity of Teachers Based on their Proposed IoT/ AIoT STEM Systems

We received 87 creative ideas from 93 teachers. A full mark being two, 48 (55.17%) of the teachers earned a two, 29 (33.33%) earned a one and 10 (11.49%) earned a zero. A total of 77 (82.80%) teachers show digital creativity after attending the course, while 61 (65.59%) show their understanding of AI concepts by applying AI models in their proposed systems.

Figure 5 shows an example of a creative idea that earned a mark of two. It consists of an AIoT STEM system with two parts. In Part one, an AI data model is trained with two labels, namely ‘child’ and ‘adult’. The AI data model is then extracted and embedded in an app. If the STEM system detects an adult attempting to access utensils in the kitchen, it unlocks the drawer to grant access. However, if it detects a child attempting to access potentially dangerous utensils in the kitchen, it locks the drawer, thereby protecting the child from harm. This is possible because a camera in a mobile device captures the face of user and uses a pre-trained AI data model for reasoning and reacting. The STEM system also sends a message to adults when the app detects a child attempting to open the drawer.

Figure 5

Example of a STEM System That Protects Children from Accessing Dangerous Kitchen Utensils, Proposed by a Teacher in the Test of Digital Creativity



4.4. Conclusion, Limitations and Future Works

Our study provides evidence that a professional development course introducing teachers to IoT and AI concepts via STEM activities is feasible. We find that teachers can learn this content knowledge and feel competent in teaching STEM systems and introducing the IoT concepts of sensing-reasoning-reacting. We find that teachers feel confident to teach such IoT concepts and related problem-solving skills following such training. We also provide evidence that the introduction of AI data model training provides opportunities for teachers to be more creative in designing STEM systems.

This study has three limitations. First, the Cronbach’s alpha value of the content knowledge test is less than 0.7 and could benefit from further refinement, although it is within the acceptable range of 0.6 to 0.7 (Hair et al., 2010), particularly for newly designed instruments (Taber, 2018). Second, we did not administer a pre-course TPACK survey in this study, which should be done in the future studies to investigate the effectiveness of the training course in facilitating teachers' self-efficacy in teaching STEM integrating with AI model training. Third, we did not analyse the artefacts presented by the teachers in this study in detail. A more in-depth analysis may help develop effective guidance for teachers to be creative in proposing innovative ideas in designing STEM systems with IoT concepts. This, in turn, can help teachers to guide their students to develop digital creativity in their classes.

REFERENCES

Akyuz, D. (2018). Measuring technological pedagogical content knowledge (TPACK) through performance assessment. *Computers & Education*, *125*, 212–225.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376.

Ashton, K. (2009). That 'Internet of things' thing. *RFID Journal*, *22*(7), 97–114.

Barker, B. S., & Ansorge, J. (2007). Robotics as means to increase achievement scores in an informal learning environment. *Journal of Research on Technology in Education*, *39*(3), 229–243.

Chai, C. S., & Koh, J. H. L. (2017). Changing teachers' TPACK and design beliefs through the Scaffolded TPACK Lesson Design Model (STLDM). *Learning: Research and Practice*, *3*(2), 114–129.

Chiu, T. K. (2021). A holistic approach to the design of artificial intelligence (AI) education for K–12 schools. *TechTrends*, *65*(5), 796–807.

Cotabish, A., Dailey, D., Robinson, A., & Hughes, G. (2013). The effects of a STEM intervention on elementary students' science knowledge and skills. *School Science and Mathematics*, *113*(5), 215–226.

Council, N. R. (2012). *A framework for K–12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

Ghosh, A., Chakraborty, D., & Law, A. (2018). Artificial intelligence in Internet of things. *CAAI Transactions on Intelligence Technology*, *3*(4), 208–218.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2010). *Multivariate data analysis* (7th ed.). Prentice Hall.

Kong, S.-C., & Zhang, G. (2021, September). A conceptual framework for designing artificial intelligence literacy programmes for educated citizens. In S.C. Kong, Q. Wang, R. Huang, Y. Li, & T.C. Hsu (Eds.), *Conference Proceedings (English Track) of the 25th Global Chinese Conference on Computers in Education, GCCCE 2021* (pp. 11–15). Hong Kong: The Education University of Hong Kong.

Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education*, *151*, 103872.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*(6), 1017–1054.

Mohr-Schroeder, M., Bush, S., & Jackson, C. (2018). K12 STEM education: Why does it matter and where are we now? *Teachers College Record*.

Nurlaely, N., Permanasari, A., & Riandi, R. (2017). Student's STEM literacy in biotechnology learning at junior high school. *Journal of Physics: Conference Series*, *895*(1), 012155.

Sakulkueakulsuk, B., Witoon, S., Ngarmkajornwiwat, P., Pataranutaporn, P., Surareungchai, W., Pataranutaporn, P., & Subsoontorn, P. (2018). Kids making AI: Integrating machine learning, gamification, and social context in STEM education. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 1005–1010.

Taber, K. S. (2018). The use of Cronbach's alpha when developing and reporting research instruments in science education. *Research in Science Education*, *48*, 1273–1296.

Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K–12: What should every child know about AI? *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 9795–9799.

Wang, T., & Cheng, E. C. K. (2021). An investigation of barriers to Hong Kong K–12 schools incorporating artificial intelligence in education. *Computers and Education: Artificial Intelligence*, *2*, 100031.

Xu, W., & Ouyang, F. (2022). The application of AI technologies in STEM education: A systematic review from 2011 to 2021. *International Journal of STEM Education*, *9*(1), 1–20.

Yang, W. (2022). Artificial intelligence education for young children: Why, what, and how in curriculum design and implementation. *Computers and Education: Artificial Intelligence*, *3*, 100061.

Associating Learning Engagement with Changes in Computer Science Self-Efficacy in the Context of Game Design and Simulation Creation Activities in Rural Elementary Schools

Arif RACHMATULLAH[1*], Carol TATE[2], Andrea BEESLEY[3], Elise LEVIN-GURACAR[4]
[1,2,3]SRI International, United States of America, [4]TERC, United States of America
[1,2,3]arif.rachmatullah, carol.tate, andrea.beesley@sri.com, [4]elisesaadet@gmail.com

ABSTRACT

The current study examines the relationship between learning engagement in programming activities and changes in students' computer science (CS) self-efficacy. A total of 127 students from predominantly rural elementary schools in the United States (US) engaged in Scalable Game Design and Simulation Creation (SGD) activities for a week in each two consecutive semesters. They took a CS self-efficacy survey before the first and after the second sessions. Some students also completed an exit ticket after each day of activity to measure learning engagement and perceptions of the activities. We ran multilevel modeling and thematic analyses on the data. We found that during the students' first-semester activities, their learning engagement negatively correlated with changes in their CS self-efficacy. In contrast, in the second semester, we found that the greater the students' engagement with the activities, the more likely they were to experience increased CS self-efficacy. These findings were supported by teachers' and students' perceptions that, although students experienced frustration in the first semester, by the second semester they perceived the struggle to be more productive and therefore such an experience might be less damaging to their self-efficacy. This study emphasizes the need for more research into the complex relationship between learning engagement and self-efficacy beliefs, specifically the ways in which these two constructs interact over time with novice learners.

KEYWORDS

elementary, learning engagement, rural, self-efficacy

1. INTRODUCTION

Despite a growing need for computing knowledge in a wide range of fields, attracting and retaining a diverse population of students in advanced CS classes remains a challenge (Code.org, 2021). This is partly due to individuals lacking a sense of belonging within CS, coupled with or stimulated by beliefs of low capability or self-efficacy in CS (Nguyen & Lewis, 2020). The reasons for low self-efficacy can vary among individuals. Bandura indicated that experience plays a significant role in self-efficacy development (Bandura, 1977). Hence, students with early exposure to CS activities are more likely to develop higher CS self-efficacy. These higher levels of self-efficacy can increase students' likelihood of choosing, and feeling a sense of belonging in, a computationally intensive field or career. Although elementary school students are several years from choosing a college major or seeking a job in computing, many studies have suggested that the elementary years are a critical time for children to start understanding their interests (Shein, 2019; Wiebe et al., 2018), learning foundational CS concepts, and understanding the potential of CS practices when approaching various tasks (Vandenberg et al., 2021).

Countries around the world (e.g., United Kingdom, New Zealand, South Korea, Israel, China), have made comprehensive plans to provide CS learning opportunities to all students (e.g., Falkner et al., 2019). In the United States, CS experiences vary greatly across states and school districts due to the decentralized nature of the US education system (Peterson et al., 2021). This condition has led to the unequal distribution of CS offerings and qualified teachers (Code.org, 2021). Consequently, a gap in access to quality CS education experiences has emerged (Century et al., 2020). For example, CSforALL (2021) reported that CS classes are primarily available in elementary schools located in more developed areas, whereas schools, particularly elementary, located in rural areas rarely engage students with CS-related activities. Hence, students in rural elementary schools are likely to have less exposure to CS education and thus less opportunity to develop CS self-efficacy. At the same time, many CS curricular materials are distributed as open educational resources, which addresses issues of access and affordability. However, to accompany these resources, students need to have computing experiences in school. One approach to remedy this is to embed computing experiences into core classes such as mathematics, science, or English language arts.

Scalable Game Design and Simulation Creation (SGD) is a set of game and simulation design activities that are intended to foster and sustain student engagement in computer programming. SGD applies a project-first approach, in which students create an arcade-style game in the first lessons rather than studying principles before creating. Repenning et al. (2015) explained that such an approach "allows students to immediately engage in computer programming design experiences and to learn concepts as the need arises" (p. 8). Repenning and colleagues further posited that the project-first approach could increase student ownership, eventually increasing motivation to learn. SGD activities are conceptually framed with Csikszentmihalyi's (1997) flow theory and Vygotsky's (1978) zone of proximal development. These theories were combined into the zones of proximal flow theory (Basawapatna et al., 2013). The zones of proximal flow theory describes the need to balance students' skills with the challenges posed in the learning environment and provide appropriate scaffolding. Such measures help students acquire new skills and advance to more complex challenges without feeling frustrated or bored. Furthermore, students are presumed to experience greater engagement in such a learning environment.

Although conceptually SGD is promising in terms of engaging students in CS activities (Basawapatna et al., 2013; Repenning et al., 2015), the learning engagement level students exhibit during the activities is not yet well documented. In addition, the relationship between students' learning engagement in SGD and their CS self-efficacy is not well studied, especially with rural elementary school students. Thus, in the current study we examined this relationship in two-week-long SGD units spanning two semesters. This study contributes to the CS education literature by providing needed evidence on how CS self-efficacy develops among rural elementary students as they engage with extended and different types of CS-related activities. The current study was guided by the following research questions (RQs):

RQ1. To what extent does rural elementary school students' learning engagement in the SGD activities moderate changes in CS self-efficacy?

RQ2. How do teachers' perceptions of the SGD activities explain the relationship between rural elementary school students' learning engagement and their CS self-efficacy?

2. THEORETICAL FRAMEWORK

The current study is framed with self-efficacy theory. According to Bandura (1977), self-efficacy is defined as an individual's belief in their ability to perform a set of specific behaviors to bring about specific outcomes. Such outcomes can include, but are not limited to, academic performance or attitudes. Likewise, self-efficacy reflects an individual's confidence in their capacity to succeed in a particular situation. Thus, CS self-efficacy refers to students' confidence in their ability to succeed in CS-related tasks or courses.

Researchers have documented four sources of self-efficacy: mastery experiences, vicarious experiences, social or peer persuasion, and emotional and physiological states (Bandura, 1977; Bong & Skaalvik, 2003). The current study is focused on mastery and vicarious experiences. Although we acknowledge the power of the latter two sources of self-efficacy, mastery and vicarious experiences are more significantly linked to learning engagement (Linnenbrink & Pintrich, 2003). Bandura described mastery experiences as the experiences an individual has when successfully completing tasks (Bandura, 1977). Bandura further theorized that mastery experiences are the most impactful source of self-efficacy because they provide individuals with the most authentic evidence of whether they can succeed in a certain activity or field. In CS, much research has shown a significant positive correlation between K–12 students' self-efficacy and prior programming or CS-related activities (Hinckle et al., 2020; Rachmatullah & Wiebe, 2023). Regarding vicarious experiences, Bandura described these as the experiences of individuals who witness others similar to themselves succeed through sustained effort. Such conditions elevate students' beliefs in their capabilities to succeed in comparable activities. In the context of this study, the "others" would be classmates. We believe that these two sources of self-efficacy manifest in the students' learning activities as students actively seek for help and discuss the activities. Therefore, we hypothesized that learning engagement plays a role in students' development of CS self-efficacy.

3. METHODS

3.1. *Participants*

A total of 127 fifth-grade students (age *M* = 10.52, *SD* = 0.66) of six different teachers in three different schools located in the Midwestern region of the United States participated in the current study. The participating students were 45% male, 46% female, and 2% non-binary. The remaining students did not provide gender information. Most students identified their ethnicity as White (72%), followed by mixed (7%), and Black/African American, Latinx, Asian, and Native American (3%). The remaining students did not answer the ethnicity question. Only 11% spoke languages other than English at home. Sixty-four percent were familiar with computer programming and had had at least one coding experience. We also collected their computer or video game-playing frequency per week: zero = 9%, one to two days = 25%, three to four days = 24%, and five to seven days = 30%. The remaining students did not provide information about their gaming frequency.

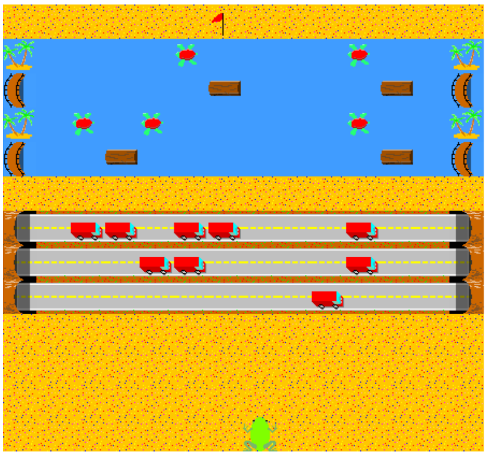


Figure 1. An example of Frogger's final display.

3.2. *Curricular Context*

Each semester, prior to implementing SGD activities, the teachers participated in a set of online asynchronous professional development (PD) sessions. The PD provided teachers with training on conceptual and practical components of SGD: game-design theory, computational thinking practices and patterns, CS big ideas, and step-by-step instruction for designing games and creating simulations with AgentCubes, an educational programming language for younger students to build 2D and 3D games and simulations. During fall 2021, the teachers took PD Course 1 for approximately six weeks. They learned the content and pedagogical practices for designing two games: Frogger and Journey. Frogger is based on the classic 1980's arcade game where the player controls a frog crossing a busy highway and then a river, dodging obstacles along the way. Figure 1 shows the final product of the Frogger activity students should have at the end of the activity. In Journey, students created a game in which the player is a traveler on a journey to reach a destination. The player travels on the ground amid walls with one or more chasers. The player needs to navigate the walls and avoid the chasers. After completing the PD courses and creating their own games for Frogger and Journey, the teachers implemented the lessons in their classrooms.

During the spring 2022 semester, the teachers took PD Course 2 on using computer programming to create two simulations: Contagion and Predator and Prey. Contagion focuses on designing and creating a simulation of the spread of an infectious disease. In the Predator and Prey activity, students design and create a simulation of small interactions in an ecosystem. Each set of game design and simulation creation activities took about five to six class periods.

3.3. Data Source and Collection

The data collection procedure is depicted in Figure 2. We collected data on students' CS self-efficacy, learning engagement, and perceptions of what they learned from the activities. In addition, we also collected data on teachers' perceptions of the activities as part of teacher exit ticket that they took after each day of activity.

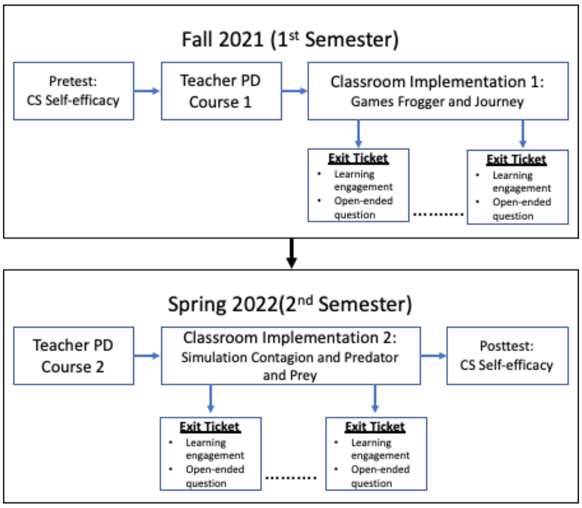


Figure 2. Data collection procedure.

CS self-efficacy. We used four 5-point Likert-type items measuring elementary school students' CS self-efficacy drawn from Kong et al. (2018) and Vandenberg et al. (2021). An example item is "I have confidence in my ability to do coding." Students took this instrument twice: before their teachers participated in the PD in fall 2021 (pretest) and after the last classroom activity in spring 2022 (posttest). We revalidated the instrument with our data and resulted in psychometrically sound items: The instrument was reliable with a reliability value of .822.

Learning engagement. We selected four 5-point Likert-type items from an instrument developed by Wang et al. (2014) to measure students' learning engagement. These items represent three dimensions of learning engagement: affective, behavioral, and cognitive. An example item is: "Today, in this lesson, I did not want to stop at the end of class." Students answered these questions several times as part of an exit ticket survey administered at the end of every lesson.

Student and teacher open-ended responses. As part of the exit ticket survey, students were also asked to answer the question, "What did you learn today in this lesson?". In addition, teachers also took an exit ticket survey with items asking them what challenges they faced and what went well at the end of every implementation day.

3.4. Data Analysis

Before performing further analyses, we generated aggregate scores for CS self-efficacy using the item-response theory method. This method standardizes the scores (called logit scores) on an interval scale that is more appropriate for statistical analyses than ordinal scale generated through calculating raw data (Boone et al., 2013). For learning engagement data, we generated an aggregate score for each student in each semester. Thus, each student would have one aggregate score in the fall 2021 and one in the spring 2022 semester. We used these aggregate scores in the subsequent analyses. Not all students took the exit tickets. In fact, only about 50% and 54% students submitted exit tickets in the fall 2021 and spring 2022 semesters, respectively. We acknowledge this incomplete data as one of the study's limitations.

We conducted multilevel modeling (MLM; Raudenbush & Bryk, 2002) to investigate the relationship between students' learning engagement and changes in their CS self-efficacy from pre- to post-intervention. Specifically, we separated the learning engagement scores into two sets, Fall 2021 and Spring 2022. This approach was intended to examine how learning engagement in each semester influenced students' CS self-efficacy. We also included other demographic factors (age, gender, ethnicity, multilingual status, familiarity with coding, previous coding experience, frequency of playing games, and CT assessment scores) in the model and set them as fixed variables. We applied a listwise deletion approach for missing learning engagement data. We were aware of the effect of this approach on the statistical power of the analysis.

We analyzed the qualitative data thematically to answer the second research question. First, we conducted an open-coding process to gather emerging qualitative codes in the data (Saldaña, 2021). Then, we applied axial coding to categorize those codes into more abstract categories (Saldaña, 2021). Finally, we used a selective coding process to generate themes to help interpret the quantitative findings. Three researchers met several times to generate the codes and themes and peer-debrief the themes (Patton, 2002). These meetings and debriefs were intended to mitigate individual researcher bias.

4. FINDINGS

4.1. RQ1. Relationship of Learning Engagement and CS Self-efficacy

Table 1 presents the MLM results. Students' learning engagement significantly moderated changes in students' CS self-efficacy, after accounting for demographic variables and CT scores. This was particularly true in spring 2022, the second semester in which students were exposed to SGD activities. This model accounted for

37.27% of between- and 47.69% of within-student variation. We then graphed these findings (see Figure 3).

Figure 3 illustrates that students who were highly engaged in the activities in fall 2021 experienced a greater decrease in their CS self-efficacy compared to those with low engagement. We observed an opposite pattern in spring 2022; after controlling for students' learning engagement in fall 2021, students' 2022 learning engagement had a positive and significant impact on the changes in CS self-efficacy. During the spring 2022 semester, highly engaged students increased in CS self-efficacy from pretest to posttest, while students with low learning engagement in spring 2022 decreased in their CS self-efficacy from pretest to posttest.

Table 1. Results of MLM (Notes: *SE* = standard error, LE = Learning Engagement)

Effect, Parameter	Estimate (*SE*)	*p*
CS Self-efficacy β_0		
Intercept, γ_{00}	8.37 (6.83)	.227
Age, γ_{01}	-0.58 (0.53)	.284
Gender, γ_{02}	0.97 (0.56)	.094
Ethnicity, γ_{03}	0.06 (0.11)	.606
Multilingual status, γ_{04}	-0.51 (1.04)	.628
Familiarity with coding, γ_{05}	-0.50 (0.72)	.493
Previous coding experience, γ_{06}	0.35 (0.68)	.607
Frequency of playing games, γ_{07}	0.56 (0.31)	.085
CT score, γ_{08}	-0.13 (0.27)	.634
LE in Fall 2021, γ_{09}	1.94 (0.99)	.060
LE in Spring 2022, γ_{10}	-2.89 (0.95)	.005
Intervention slope β_1		
Time, γ_{10}	-4.19 (2.25)	.074
LE in Fall 2021, γ_{11}	-1.34 (0.65)	.051
LE in Spring 2022, γ_{12}	2.32 (0.62)	.001
Random effects		
Variance components		
Between-student, τ_{00}	0.728 (SD = 0.85)	
Within-student fluctuation, σ^2	2.294 (SD = 1.51)	

4.2. RQ2. Students' and Teachers' Perceptions of the SGD in the Two Semesters

Based on students' and teachers' responses to the exit ticket surveys, we found that in the fall 2021 semester students felt that the activities were boring and hard. Some students expressed, "I learned I don't know anything", "[the activity is] boring ...", and "I did not make progress today I got stuck."

Even though they were struggling during the activities, some of the students found the activities to be fun. For instance, a student said, "Frogger is fun but hard." Many students expressed that they needed to persevere in this activity. Teachers also echoed these experiences that students put more effort into the fall's activities than in the spring's activities, and the teachers worried that many students needed a lot of help from others who had previous programming experience. For example, a teacher mentioned that "I had several students get onto the program with zero qualms and they put some serious effort into it." Teacher P02T01 said, "I love how students who are successful are willing to help others who are stuck." She expressed concern the help was too much: "But I do worry that they are doing too much for their classmates rather than offering assistance."

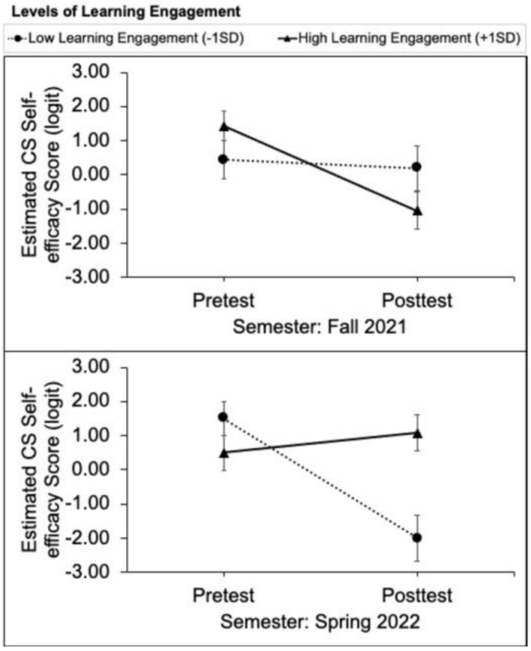


Figure 3. Interaction between learning engagement in two semesters and changes in CS self-efficacy. (Note: The Pretest and Posttest are the same time points in both semesters. Hence, they cannot be interpreted longitudinally, instead the interpretations should be done independently across semesters)

In the spring 2022 semester, the students indicated that the activities were easier than they anticipated, given the challenges of the fall 2021 activities. Students expressed, "I could remember what I learned in Frogger, and it helped me [in the spring 2022 activities]," "We did not have to put so much stuff on it this time," and "…if you get the work done [in the previous semester] it's easier." A student suggested that they needed practice or experience to understand the process: "I learned that it takes practice to be better and to understand what you are doing." Some students expressed that the activities were challenging yet fun. Consequently, they persevered and engaged with the activities until they obtained more clarity on the tasks. For example, a student said:

I learned that coding is fun, but it can be hard sometimes when something goes wrong. You try to do it first then if you cannot figure it out, you go ask the teacher…[then] you ask your other teacher. And you got to make sure you spell things correctly or it will mess you up.

Teachers noticed that once students figured out what they needed to do and successfully completed one task, the students could transfer their successful experience to another task. A teacher specifically said, “Once they created one agent, they were able to create all of the rest of them by themselves.” This sentiment was echoed by another teacher at another school:

> *Since we have not used the AgentCubes online program since beginning of December, I was surprised at how easily my students were able to create the agents in the simulation. They were building more than I was ready for them to build.*

However, some teachers also observed that their students had challenging moments with the activities, such as getting their agent to move. These difficult moments kept students engaged and they managed to persevere, resulting in completing the activities.

5. DISCUSSION AND IMPLICATIONS

Our findings indicated that in the first exposure to the SGD activities in the fall 2021 semester, students who were highly engaged in the activities experienced a decline in their CS self-efficacy. Contrarily, students who had high learning engagement in the second exposure to the SGD activities in the spring 2022 semester increased their CS self-efficacy. These findings suggest that in the first-time exposure to CS or SGD activities, the more students engaged in the activities the more they realized that they were not yet competent enough to work on the CS-related tasks. This aligns with previous studies that found both middle and high school students exposed to CS activities for the first several times do not significantly increase or even slightly decrease in their CS attitudes (Rachmatullah & Wiebe, 2023). In addition, according to Bandura (1977), mastery experiences are built incrementally as a source of self-efficacy. Hence, individuals need more time to develop stable mastery experiences to impact their self-efficacy positively. These mastery experiences can include students being familiar with the computer programming language. Therefore, by the second semester, when they had more familiarity with the computer programming language, students may have felt more competent in performing the tasks, even though the activities were of a different nature (simulation creation vs. game design). This might explain the increases in CS self-efficacy observed in the second semester.

Vicarious experiences might have also influenced students’ self-efficacy beliefs during the activities. The teachers indicated that a few students in the first semester activities helped other students finish the tasks, which might point out gaps in knowledge or experience among students. Such gaps might stimulate students who got help to compare themselves to similar students needing help. As a result, these students who needed help did not witness other students like themselves succeed in completing the task. However, in the second semester, as most students had gained more experience and confidence in their programming activities, they also observed that more students with the same level of experience as themselves could finish the tasks through sustained effort. Consequently, the activities in the second semester contributed to a positive change in CS self-efficacy for those highly engaged with the activities.

The findings on students who had low learning engagement levels were not surprising. Many studies have shown that students with low learning engagement tend to attain lower cognitive and affective learning outcomes (Marks, 2000), including self-efficacy. In the context of the current study, students who did not engage much in the SGD tasks and activities might not have had enough opportunities to test and evaluate their CS abilities in order to build their repertoire of CS experiences. Therefore, at the end of the activities, they did not strongly believe in their CS abilities compared to those with higher learning engagement.

Our study findings suggest implications for the design of evaluations of CS-related activities. Even though some CS activities could have a more positive impact on elementary school students’ CS self-efficacy (e.g., Phillips & Brooks, 2017), it might take more time for elementary school students who have limited prior CS experience, such as those in rural schools. As explained above, these students may need more time to gather evidence for building their CS self-efficacy. Thus, it is advisable that students with little prior CS experience have multiple learning opportunities rather than a one-time intervention. Furthermore, in an impact study, collecting data such as learning engagement may provide helpful information to interpret impact findings.

6. CONCLUSIONS, LIMITATIONS, AND FUTURE DIRECTIONS

We concluded that SGD activities could positively impact rural elementary school students’ CS self-efficacy. However, activities targeting these students should be extended over time rather than confined to a short period or single activity (e.g., Frogger only). In addition, the current study supports a contention that learning engagement and CS self-efficacy are significantly associated. Therefore, supporting students in CS-related activities to keep them engaged is imperative to maintain and improve their CS self-efficacy.

These findings should be interpreted carefully as they have some limitations. The first limitation is the variation in students taking the learning engagement survey. We found that some students did not take the survey after every day of implementation, which led us to create an aggregate semester score instead of analyzing engagement per activity (e.g., Frogger, Journey, and Contagion). Conducting such an analysis with complete daily exit ticket data would provide more detailed information about which activity contributed more to students’ changes in self-efficacy; thus, teachers and researchers would be able to better sequence the activities based (for example) upon the complexity of the activities. Second, we are aware of the impact of teacher characteristics (e.g., CT teaching self-efficacy, prior experience) on students’ learning engagement. However, due to the small sample size of both teachers and students, we could not run a three-level MLM to see the impact of teacher characteristics on changes in student CS self-efficacy. The small student sample size also prevented us from including the interaction of prior experience, learning engagement, and CS self-efficacy.

Lastly, we did not collect intermediate data between semesters; hence, we could not record how students' CS self-efficacy changed after one semester and longitudinally over time. Therefore, future studies could extend this study by collecting more data points.

7. ACKNOWLEDGMENTS

We thank our participating students and teachers. We want to also acknowledge the eMINTS research team at the University of Missouri for creating the program and delivering the PD to teachers. This work is supported by an Education Innovation and Research (EIR) grant from the U.S. Department of Education.

8. REFERENCES

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191-215.

Basawapatna, A. R., Repenning, A., Koh, K. H., & Nickerson, H. (2013, August). The zones of proximal flow: guiding students through a space of computational thinking skills and challenges. In *Proceedings of the ninth Annual International ACM Conference on International Computing Education Research* (pp. 67-74).

Bong, M., & Skaalvik, E. M. (2003). Academic self-concept and self-efficacy: How different are they really?. *Educational Psychology Review*, *15*(1), 1-40.

Boone, W. J., Staver, J. R., & Yale, M. S. (2013). *Rasch analysis in the human sciences*. Springer Science & Business Media.

Century, J., Ferris, K. A., & Zuo, H. (2020). Finding time for computer science in the elementary school day: a quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education*, *7*(1), 1-16.

Code.org. (2021). *Code.org 2021 Annual Report*. Retrieved https://code.org/files/code.org-annual-report-2021.pdf

CSforALL. (2021). *How accessible is computer science in rural communities?*. Retrieved from https://medium.com/csforall-stories/how-accessible-is-computer-sciencein-rural-communities-6b69e921a0a1.

Csikszentmihalyi, M. (1997). *Flow and the psychology of discovery and invention*. Harper Perennial, New York, 39.

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., ... & Quille, K. (2019, November). An international comparison of k-12 computer science education intended and enacted curricula. In *Proceedings of the 19th Koli calling international conference on computing education research* (pp. 1-10).

Hinckle, M., Rachmatullah, A., Mott, B., Boyer, K. E., Lester, J., & Wiebe, E. (2020, June). The relationship of gender, experiential, and psychological factors to achievement in computer science. In *Proceedings of the 2020 ACM conference on Innovation and Technology in Computer Science Education* (pp. 225-231).

Linnenbrink, E. A., & Pintrich, P. R. (2003). The role of self-efficacy beliefs in student engagement and learning in the classroom. *Reading & Writing Quarterly*, *19*(2), 119-137.

Marks, H. M. (2000). Student engagement in instructional activity: Patterns in the elementary, middle, and high school years. *American Educational Research Journal*, *37*(1), 153-184.

Nguyen, A., & Lewis, C. M. (2020, February). Competitive enrollment policies in computing departments negatively predict first-year students' sense of belonging, self-efficacy, and perception of department. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 685-691).

Patton, M. Q. (2002). *Qualitative research and evaluation method*s. Thousand Oaks, CA: Sage Publications.

Peterson, D., Goode, K., & Gehlhaus, D. (2021). *AI Education in China and the United States: A Comparative Assessment CSET Issue Brief*. Center for Security and Emerging Technology, Washington, D.C.

Phillips, R. S., & Brooks, B. P. (2017). *The hour of code: Impact on attitudes towards and self-efficacy with computer science*. Code.org, Seattle, WA, US.

Rachmatullah, A., & Wiebe, E. N. (2023). Exploring middle school students' interests in computationally intensive science careers: The CISCI instrument validation and intervention. *Science Education*, *107*(2), 333-367.

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods (Vol. 1).* Sage.

Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., ... & Repenning, N. (2015). Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *ACM Transactions on Computing Education (TOCE)*, *15*(2), 1-31.

Saldaña, J. (2021). *The coding manual for qualitative researchers*. Sage.

Shein, E. (2019). The CS teacher shortage. *Communications of the ACM*, *62*(10), 17-18.

Vandenberg, J., Rachmatullah, A., Lynch, C., Boyer, K. E., & Wiebe, E. (2021). Interaction effects of race and gender in elementary CS attitudes: A validation and cross-sectional study. *International Journal of Child-Computer Interaction*, *29*, 1-11.

Vygotsky, L. S., & Cole, M. (1978). *Mind in society: Development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wiebe, E., Unfried, A., & Faber, M. (2018). The relationship of STEM attitudes and career interest. *EURASIA Journal of Mathematics, Science and Technology Education*, *14*(10), 1-17.

Teachers' Understanding of Synergies Between Computational and Mathematical Thinking after a Summer Professional Development

Zuhal YILMAZ[1*], Karmen Williams[2], Fernando Alegre[3], Rose Kendrick[4], Juana Moreno[5]
[1,2,3,4,5] Louisiana State University, United States of America
zyilmaz@lsu.edu, kwil419@lsu.edu, falegre@lsu.edu, rkendr3@lsu.edu, moreno@lsu.edu

ABSTRACT

Emergent studies suggest that engaging in computer science education has the potential to support students' learning of key mathematical concepts due to the connection between mathematical and computational thinking (MT & CT). To create a rigorous learning environment that focuses on this connection, teachers must gain an in-depth understanding of the synergies between CT and MT, and learn the ways to integrate the common practices and aspects of both into their practice. Thus, we co-designed a five-week long summer professional development (PD) that cultivated various perspectives about the synergies between CT and MT. We explored the shifts in understanding of CT and its connection to MT with a cohort of ten teachers from multiple subject areas. The results suggest that the teachers displayed an overall improvement in the richness and depth of their descriptions of CT and MT, and provided a variety of examples of synergies between them. Most of the teachers recognized problem-solving, generalization and abstraction, and decomposition as synergies between CT and MT. They also explained more precisely how they used mathematical knowledge in computing activities after the PD. These results suggest that the PD may help teachers to integrate both types of thinking into their classroom practices. We also found that some aspects of MT and CT, such as modeling, did not surface in the data analysis. This finding will be helpful to chart the focus and design of future PDs.

KEYWORDS

computational thinking, mathematical thinking, in-service teacher, computer science, professional development

1. INTRODUCTION

There has been an increased interest in making computer science (CS) education a core subject in K-12 education (Menekse, 2015). Various studies emphasized the potential benefits of integration of CS education on students' mathematical learning (Alegre et al., 2022; Barcelos et al., 2018). However, teachers must first develop an understanding about this integration to create an effective learning environment for their students. Thus, we co-designed a professional development (PD) focusing on understanding what computational and mathematical thinking (CT and MT) include and highlighting the synergies between these two types of thinking.

CT is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be efficiently carried out by an information-processing agent." (Wing, 2006). Weintrop et al., (2016) also stated that CT is reformulating problems, thinking recursively, using decomposition and abstraction, choosing appropriate models, and using heuristic reasoning.

From a mathematics education perspective, the discourse on CT resonates with the fundamental theoretical frameworks of mathematics education like mathematical modeling (Lesh & Fennewald, 2013), problem solving (Schoenfeld & Sloane, 2016), generalization and abstraction (Tall, 1999), and quantitative reasoning (Duval, 1999). Thereby, these similarities inform and guide us to design a PD for teachers that focuses on the synergies between CT and MT.

2. CONNECTING MATHEMATICS AND COMPUTING

Using computing to aid students' mathematics and science learning has a long history. Early attempts to use programming "as a tool were based on unguided discovery" (Alegre et al., 2020, p. 992). This approach is shown to be ineffective for transfer (Mayer, 2004). However, in the past decade, this trend was reversed when the Bootstrap project (Schanzer et al. 2013) started to show evidence of transfer. Most recent studies focus on programming skills with a limited explicit connection to key mathematical concepts (Hickmott et al., 2018). Hickmott et al. (2018) reviewed 393 studies published up to the ten years after Wing (2006)'s article and found that one of the major gaps in the literature was the limited empirical studies that explicitly connected CT and MT. They suggested a lack of mathematics education expertise as the leading factor to numerous studies that "incidentally" make the connection.

Only a few studies (e.g. Barcelos et al., 2018; Brating & Kilhamn, 2021) explore learning of key mathematical ideas (e.g. algebraic thinking, functions, multiple representations) through engaging in computing. These studies suggested that engaging in computing tasks could enhance students' learning of key mathematical ideas.

2.1. *PD in CS: Connecting CT and MT*

CT is still a relatively new concept, many teachers lack the knowledge and resources necessary to successfully incorporate it into their curricula (Yadav et al., 2016; Wu et al., 2021). Providing effective PDs to support teachers' knowledge of CT and equip them with necessary resources are critical to improve students' learning experiences. Thus, we have reviewed the existing CS PD literature, particularly those focusing on supporting teachers' understanding of the synergies between CT and MT.

In our review, we only encountered a few PD studies (e.g. Hart et al., 2008; Wu et al., 2021) that focused on this connection. Hart et al. (2008) conducted a series of summer workshops to "provide teachers with innovative activities and ideas that link their secondary school mathematics curriculum with computer science." (p. 286). Wu et al. (2021) worked with 11 science and mathematics teachers in a co-designed PD focusing on enhancing teachers' confidence in integrating STEM-CS practices (e.g. modeling and simulation, and data practices) in their curricula. They found that teachers gained confidence and skills in designing STEM-CS curriculum.

Menekşe (2015) conducted a review on CS education PD in the US. She found that there is limited collaboration between researchers and practitioners in designing PD. Also, she found that the majority of computer science PD was shorter than a week and the support was not ongoing. She also found a few PDs focused on integration of CS in mathematics curriculum. To this end, we co-designed a 5- weeks long summer PD with mathematics and CS education researchers and practitioners. This PD focused on supporting teachers' understanding of CT and MT, and the synergies between both. We seek to answer the following research questions:

1- Which aspects of computational thinking were emphasized in teachers' description of CT before and after the co-designed PD?
2- Which aspects of mathematical thinking were emphasized in teachers' description of MT before and after the co-designed PD?
3- How did teachers' understanding of the synergies between CT and MT change after the co-designed PD intervention?

3. METHOD

Researchers interviewed ten teachers about their thinking and understanding of CT, MT and their synergies before and after the PD.

3.1. *Research Context and Participants*

Alina and Jessica were facilitators of the five-week long summer PD. Both facilitators taught the Introduction to Computational Thinking (ICT) course (Alegre et al., 2020) to 9th graders. Alina is a visual art teacher with a teaching experience of 12 years, and Jessica is a former mathematics teacher with 5 years of teaching experience. Before the PD, both facilitators were trained by an accomplished mathematics coach who has 21 years of experience and 4 years in CS education. Further, the facilitators and accomplished coach debriefed after each PD day and planned for the next day. Ten teachers from high-need schools engaged in PD for five hours per weekday. T4, T7, and T10 have a background in CS, T5 in robotics and T6 in mathematics. Teaching experience ranges from 0 to 17 years (average of 4.5 years). Three graduate students supported the facilitators by helping teachers with content knowledge and technical issues while they worked in breakout rooms.

The PD content focused on the following areas: problem solving, coding as an expression of ideas, decomposition, automation, generalization and abstraction, importance of order, and reification. The activities required use of mathematical concepts such as the coordinate system, functions, and algebraic expressions. For example, write a program based on this prompt: "Create a triangle that has a right angle at the left"; solve a word problem following these instructions: "do not calculate the solution in your head. Instead, just write an unevaluated expression in your program"; or replace repetitive parts of the code with a function in a loop. We also implemented diverse pedagogical strategies such as peer programming, code reviews, and working in small and whole groups.

3.2. *Data Sources and Analysis*

The primary data source of this study is the pre and post interviews of the participants. Each interview lasted 30-45 minutes. The lead researcher created the interview protocol based on essential aspects of CT (e.g. Weintrop et al., 2016, Wing, 2006) and MT (e.g. Schoenfeld & Sloane, 2016; Sternberg, 2012). Two researchers and an experienced ICT course teacher shared their feedback on the protocol. We piloted the protocol with a high school teacher and analyzed the pilot data to select questions which provided in-depth responses, and to improve the clarity of the questions. A few sample interview questions were: 1) How would you describe the connection between MT and CT? Can you give an example? 2) In what ways does summer PD support your understanding of the connection between MT and CT?

PD field notes were used as supportive data sources in the analysis. Thematic analysis of the interviews was used to characterize the different ways the teachers describe and exemplify the synergies between CT and MT. Content analysis of the field notes were used to identify the instances in which teachers connected mathematics and computation in PD activities. Practices (aspects) of CT and MT (e.g. abstraction, generalization, decomposition, problem solving) documented in the literature (e.g. Barcelos et al., 2018; Polya, 1945; Tall, 1999; Weintrop et al., 2016; Wing, 2006) guided the creation of codes. Two researchers independently coded the transcripts and the agreement rate was 84.6%. The researchers discussed the disagreements in the coding until reaching an agreement.

4. FINDINGS

The findings are reported in three separate subsections that focus on CT, MT, and the synergies between CT and MT, respectively. In each subsection, the changes in the teachers' understanding are documented in two forms: as categorizations of the aspects emphasized in the teachers' descriptions and as quoted examples.

4.1. *Changes in understanding of CT*

Analysis of the teachers' descriptions reveals changes in multiple aspects of CT as shown in Table 1.

Table 1. CT aspects

CT aspect	CT aspect-subcategories		Teacher										
			1	2	3	4	5	6	7	8	9	10	T
Problem Solving	Planning	Pre					X		X			X	3
		Post	X			X	X		X	X		X	6
	Precision	Pre										X	1
		Post				X					X		2
	Decomposition	Pre				X						X	2
		Post	X			X	X	X	X	X	X	X	8
Critical Thinking		Pre			X		X			X		X	4

	Post		X	X	X	X		X	X		X	7
Importance of Order	Pre											0
	Post				X			X	X	X	X	5
Using Algorithms	Pre							X			X	2
	Post							X			X	2
Functions	Pre			X				X	X			3
(e.g. Input-output)	Post	X		X			X			X	X	5
Language Aspects of Coding	Pre		X					X			X	3
	Post		X	X	X			X			X	5
Automation (Efficiency)	Pre											0
	Post					X	X				X	3

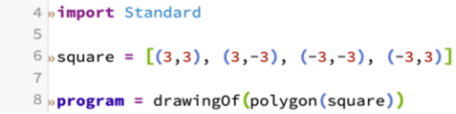
Analysis of the pre-interviews showed that only two teachers (T7, T10) could provide a meaningful description of CT. Both of them have a background in CS. Half of the teachers (T2, T3, T4, T5, T8) provided a vague description and very limited examples of CT, and three (T1, T6, T9) could not describe what CT means. After the PD, these three teachers could describe CT. For instance, T9 described it as: "[CT] *means taking a problem and working through that problem step by step to figure out how to get the desired output.*"

In the post, all the teachers provided a richer description of CT highlighting aspects such as problem solving, decomposition, coding, and order. Six teachers highlighted at least 4 important aspects of CT. However, details of their CT descriptions and examples still varied significantly.

In the pre-interview, 4 teachers indicated that CT includes problem solving. However, only 3 gave a limited explanation of why and how it includes problem-solving. For instance, T5 and T7 stated that in CT, as in a problems-solving process, they plan how to find the solutions. In the post, 8 teachers stated at least one problem-solving skill as they use CT. Only 4 out of them could explain the skill in-depth. For instance, T6 could not state any skills in the pre-interview. In the post, he stated: " [A] *skill of a problem solver involves breaking it down into small components and where you can plug them into a computer to help you automate the system to make solving that problem faster*."

The analysis also showed there is an increase in the number of teachers stating that CT includes coding (from 3 to 5) and critical thinking (from 4 to 7). The five teachers who mentioned coding in the post-interview also mentioned critical thinking. They seemed to perceive CT as a thinking type that requires skills beyond coding.

Even though not all the teachers mentioned abstraction and generalization in their pre or post interviews, they used the concepts during the summer PD. For example, in the 4th week of the PD, the teachers were asked to write a program to draw a square of any size using variables. T1 first drew a 6x6 square with the code shown in Figure 1.



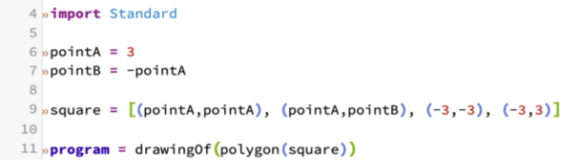
```
import Standard

square = [(3,3), (3,-3), (-3,-3), (-3,3)]

program = drawingOf(polygon(square))
```

Figure 1. Code to create 6x6 square

Then 2 teachers discussed how they could create a square of any size:

Jessica: Do you notice anything about those points on that list?
T4: They are all 3.
Jessica: Is there any way to create variables so you don't need to write 3 so many times?
T4: Set a variable and call it pointA = 3 and set another variable pointB = -point A

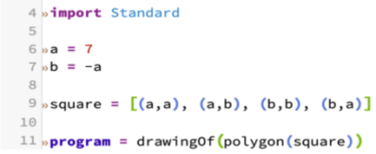
Then, the teachers started to change the code (Figure 2.)



```
import Standard

pointA = 3
pointB = -pointA

square = [(pointA,pointA), (pointA,pointB), (-3,-3), (-3,3)]

program = drawingOf(polygon(square))
```

Figure 2. Assigning variables

T1 realized that "seems longer than typing 3". Other teachers agreed. Then they found a solution calling the variable "pointA" as "a" and "pointB" as "b" (Figure 3).



```
import Standard

a = 7
b = -a

square = [(a,a), (a,b), (b,b), (b,a)]

program = drawingOf(polygon(square))
```

Figure 3. Generalized code to create any size square.

T4 stated the benefit of doing this is that "*you don't need to write all the points. This is a generalized solution to draw any square*".

4.2. *Changes in understanding of MT*

Analysis of the teachers' descriptions and examples reveals the following aspects of MT shown in Table 2.

Table 2. MT aspects

MT aspect		Teacher										
		1	2	3	4	5	6	7	8	9	10	T
Operations and calculations	Pre			X		X	X	X		X	X	6
	Post	X		X			X			X	X	5
Applying math to real life situations	Pre	X	X	X		X		X		X	X	7
	Post	X				X	X	X	X	X	X	7
Problem Solving	Pre			X			X	X				3
	Post		X			X	X	X	X	X	X	7
Process of producing an answer	Pre							X		X	X	3
	Post					X	X	X	X	X	X	6

Table 2 shows no significant change in the number of teachers for the first two categories. However, a conceptual progression in some of the teachers was observed. For instance, three (T6, T9, T10) of the six teachers who perceived MT as carrying out calculations, performing operations in the pre-interview mentioned this aspect in the context of problem-solving situations in the post-interview.

Conceptual progression was also observed in the second category. Although seven teachers stated MT requires applying mathematics to real life situations in the pre-interview, their examples (n = 5) for this aspect lack details. After the PD, the teachers gave more detailed examples of use of MT in real life situations. For instance, in the pre-interview, T1 said that we use MT in grocery shopping, and in the post, the same teacher stated::

MT might be used in grocery shopping, where you need to figure out what the cheapest price for something is. Just because one of them has a lower price on the tag doesn't mean it's the cheapest one, you're going to have to figure out how much per ounce it is, to see if it's actually cheaper.

While T6 and T8 could not give an example in the pre, they gave detailed examples of MT in real life in the post. T8:

You put coffee. You have to know how much coffee grinds that you're going to have to put in that coffee. And if you don't put enough you end up being really watery and not taste good. In a mathematical sense, there's a portion and that portion would be equivalent to some type of number.

Numbers changed significantly in the last two categories. The number of teachers indicating that MT includes problem solving increased from three to seven teachers from pre to post interview. In the post, teachers described the problem-solving process in more detail.

Only T10 stated MT encompasses proof and generalization when we asked what MT means in both interviews. Although other teachers did not state generalization in response to this question in the interviews, 6 teachers, including T10, stated generalization and abstraction is one of the synergies between CT and MT (see section below).

4.3. *Synergies between CT and MT*

Analysis of the teachers' responses revealed three main synergies between CT and MT as follows: 1) Mathematical concepts used in computation, 2) Engaging in problem solving 3) Practices used in both types of thinking.

4.3.1. *Mathematics Concepts Used in Computation*

Table 3 shows the distribution of the concepts that teachers stated in both interviews.

Table 3. Mathematics Concepts used in Computation

Categories		Teacher										
		1	2	3	4	5	6	7	8	9	10	T
Functions	Pre											0
	Post	X		X	X	X	X			X	X	7
Operations Calculations	Pre			X								1
	Post	X			X	X				X	X	5
Coordinates	Pre											0
	Post	X	X		X	X			X	X		6
Geometric Shapes	Pre											0
	Post	X			X	X						3

As seen in Table 3, before the PD, teachers saw no use of mathematical concepts in computation. Remarkably, in the post, most of them stated that they used various mathematical concepts such as operations, functions, and coordinates as they engaged in computation.

A unique feature of the PD was the use of computing keywords that prioritize connections to math over computer architecture. In this vein, the programming language used in the PD implements loops using a function called "distributed". T10 with CS background highlighted this connection as:

Applying an operation to a list of objects in your code, you have a function. That's called distributed, it takes what would normally be a for next loop, and puts it into and frames it in a way that it immediately invokes the distributive law of multiplication. So that is useful. And it reinforces ideas about how functions are composed in a mathematical expression, as well as being useful for coding too.

Similarly, T4 explained how he used math in automating repetition when asked to make a sun with 16 equidistance rays:

When I rotate my rays around my sun, I know how many angles are in a circle, 360, how many rays do I need to get, 16. Then 360 divided by 16 tells me what the angle difference between each ray is. And then, [I used] a distributed function [to create each ray], which is very similar to putting x outside of a parenthesis of two plus three, knowing that that x has to be distributed to 2x plus 3x.

4.3.2. *Engaging in Problem Solving in CT and MT*

Engaging in the problem solving steps (Polya, 1945) of "understand the problem, devise a plan, carry out the plan (solve) and look back (check and interpret)" was the most frequently stated synergy between CT and MT after the summer PD. Table 4 shows large changes in all the categories.

Table 4. Synergy of Engaging in Problem Solving

Categories		Teachers										
		1	2	3	4	5	6	7	8	9	10	T
Understand the Problem	Pre										X	1
	Post	X				X	X		X	X	X	6
Devise a Plan	Pre										X	1
	Post	X				X	X		X	X	X	6
Carry out the Plan	Pre										X	1
	Post	X	X			X	X		X	X		6
Check back	Pre										X	1
	Post	X	X		X					X	X	5
Persevere in Problem Solving	Pre											0
	Post		X		X			X				3
Generate Solutions in Multiple Ways	Pre											0
	Post					X	X					2

While six out of 10 teachers provided rich explanations for this synergy in their post-interviews, only one teacher with a CS background mentioned it in the pre-interview (See Table 4). However, in the post, these six teachers also provided examples for the problem-solving steps. For instance, T1 explained how they engaged in the first three steps of the problem-solving process together with decomposition strategies:

You need to use CT, sometimes to get a clearer understanding of a math problem. You need to figure out what the goal is [Understand the problem] *and how you're going to get there* [Devise a Plan]*, and then do calculations* [Carry out the Plan]*. It emphasized the importance of breaking things down step by step. That's what you have to do to figure out and to make mathematical decisions.*

Similarly, T10's stated in the post-interview:

You have some large problems, and you have to solve various pieces of it first [Carry out the Plan]*, and then come back to the larger problem with those results* [Check back].

As seen in Table 4, after the PD, three teachers (T2, T4, T7) indicated a critical practice used in both types of thinking: persevere in solving problems (NCTM, 2022). T2 stated:

When we did our code reviews with each other, because someone was struggling with the final image. What we would do is instead of giving them the answers, we would question them, ...so that they can solve the problem on their own. ... Like in solving a mathematics problem.

This quote of T2 highlighted the importance of scaffolding to support productive struggle and encourage perseverance while solving a problem that requires use of CT. T2 also stated that this process is similar in math problem solving.

Only 2 teachers stated that producing solutions in multiple ways is another synergy between CT and MT in the post-interview. For instance, T5 explained this synergy as:

Projects made you figure out a unique way with the limited knowledge that you have, because we have learned solid circles or how to draw circles or how to make any kind of oval-like shape. We will try to draw animals using only polygons and lines. It makes you think of unique ways to solve that problem with the limited information you have.

During the PD, all the teachers created, for instance, unique animal designs, sunny scenes, and pictures using polygons, lines and points. They acknowledged that there is more than one way to create the outcome just like in mathematics.

4.3.3. *Practices used in both CT and MT*

Analysis of the teachers' responses showed that the following practices are used both in CT and MT (Table 5).

Table 5. Common practices of CT and MT

		Teacher										
Categories		1	2	3	4	5	6	7	8	9	10	T
Generalization and abstraction	Pre										X	1
	Post	X	X		X			X		X	X	6
Automation (Efficiency)	Pre										X	1
	Post				X			X				2
Debugging	Pre										X	1
	Post		X		X			X			X	4
Decomposition	Pre											0
	Post	X	X	X	X	X			X		X	7
Importance of Order	Pre											0
	Post								X		X	2

In the pre-interview, only T10 indicated generalization and abstraction as one of the practices of CT and MT. In contrast, in the post-interview the majority of the teachers (n = 6) acknowledged generalization and abstraction as one of the common practices of CT and MT. For instance, T1 stated:

That's kind of a generalization. When you see something that is repeated in that code, you need to generalize it and kind of simplify it. So that goes with math too. You have to ...make things easier to understand for the outside viewer.

This quote showed how the "look for and express regularity in repeated reasoning" (NCTM, 2022) mathematical practice can also be used in CT and how it is connected to generalization.

In addition, all 6 teachers provided concrete examples when they used generalization and abstraction in the tasks that used CT and MT (See figure 3 as an example). During the PD, 9 out of 10 teachers explicitly noticed the regularities in the code and defined functions for the regularities.

Although the majority of the teachers thought generalization and abstraction as a common practice, two teachers stated efficiency and automation as a common practice in the post-interview. These teachers did not explain why they thought it is a common practice of CT and MT. T7 stated:

Trying to come up with a quick way of solving. So, problem solving when you're testing things,...to get the things out.

Another common practice of CT and MT stated in the post-interview was decomposition (n = 7) and debugging (n = 4). For instance, T2 said:

Learned how to think more in a mathematical sense, like using math to solve coding issues. I never would have thought that you could use math to figure out why your code is wrong. The second thing would be breaking apart code, like into pieces. In order to solve the problem, like taking it step by step until you figure out what exactly is wrong.

Since debugging is a skill based on concepts such as separation of concerns and decomposition, it is difficult to determine exactly which are the underlying concepts T2's comments are alluding to. Other aspects of troubleshooting, such as logical reasoning, were not mentioned.

The last common practice of CT and MT was the importance of order, and it was mentioned by two teachers in the post. However, during the summer PD, all the teachers observed the results of different orders, such as how the order in which the vertices of a geometric shape are joined affects the outcomes, or how order in code matters for creating layered objects. T7 explained this ordering practice in the post:

Because I think of PEMDAS, you have to use your order of operations. Same way with CT. I'm coding or creating an algorithm, I may need to put it in the right order, or it's not going to be right.

Four teachers (T1, T2, T4, T7) mentioned modeling as one of the synergies in the post-interview. Since they did not explain this synergy or give an example of modeling in which they used CT and MT, we did not classify this as one of the categories in Table 5.

5. RESULTS AND DISCUSSION

In a five-week PD focused on CT along with connections with MT, a significant shift in awareness of the centrality of problem solving in both types of thinking is observed from teachers of all backgrounds. Problem solving was mainly associated with the practices of decomposition and generalization/abstraction. However, decomposition seemed to be an unfamiliar concept to most teachers before the PD. The fact that the PD made this concept familiar to them is probably the reason that it was more explicitly stated than generalization during the post interviews.

The teachers demonstrated a progression in their understanding of CT and MT at varying levels. The dominant aspects emphasized in the post interview responses reflect the concepts stressed throughout the PD. The emphasis on decomposition, importance of order and generalization as aspects of CT, and calculations and applying mathematics to real-life situations as aspects of MT in teachers' responses were connected to how it is highlighted in the curriculum and by the PD facilitators. These results suggest a possible classification of CT concepts into a basic group (problem solving, decomposition and abstraction) and a more advanced group (precision, logical reasoning, automation and algorithms). A five-week PD seems to be suitable for learning the concepts in the former, but more time may be needed to internalize the concepts in the latter. Future work will investigate how the teachers' awareness of the concepts in the second group changes after a year of using them in the classroom.

The teachers' understanding of the synergies between CT and MT improved after the PD, in particular perceiving that both types of thinking types used in problem-solving. However, responses indicate some gaps in their understanding. Only a few teachers mentioned modeling as one of the synergies and yet, these teachers still had difficulty articulating the connections in detail. These results will support researchers in charting the focus and design of future PDs by considering which aspects of CT and MT will be explicitly explored with teachers during the PD. Also, there is a need for conducting a follow up future study on how the teachers make these connections in their classroom practices after the PD.

6. ACKNOWLEDGMENTS

This research project is partly funded by NSF award CNS-1923573 and the U.S. Dep. Of Education award U411CI90287.

7. REFERENCES

Alegre, F., Underwoood, J., Moreno, J., & Alegre, M. (2020, February). Introduction to CT: a new high school curriculum using codeworld. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 992-998).

Alegre, F., Yilmaz, Z., Moreno, J & Kendrick, R. (2022, November). Culturally relevant computing tasks: Evidence of synergies between students' mathematical and computational thinking. *44th Annual Conference of the North American Chapter of the International Group for the Psychology of Mathematics Education*, Nashville, TN.

Barcelos, T. S., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *J. Univers. Comput. Sci.*, *24*(7), 815-845.

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning*, *23*(2), 170-185.

Duval, R. (1999). Representation, vision and visualization: cognitive functions in mathematical thinking. Basic Issues for Learning.

Hart, M., Early, J. P., & Brylow, D. (2008, March). A novel approach to K-12 CS education: Linking mathematics and computer science. In *Proceedings of the 39th SIGCSE technical symposium on Computer Science Education* (pp. 286-290).

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education*, *4*(1), 48–69.

Lesh, R., & Fennewald, T. (2013). Introduction to part I modeling: What is it? Why do it? In *Modeling students' mathematical modeling competencies* (pp. 5-10). Springer, Dordrecht.

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American psychologist*, *59*(1), 14.

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education*, *25*(4), 325–350.

Polya, G. (1945). *How to Solve It*. Princeton University Press.

Schanzer, E., Fisler, K., & Krishnamurthi, S. (2013). Bootstrap: Going beyond programming in after-school computer science. In *SPLASH education symposium*.

Schoenfeld, A. H., & Sloane, A. H. (2016). *Mathematical thinking and problem solving*. Routledge.

Tall, D. (Ed.). (1991). *Advanced mathematical thinking* (Vol. 11). Springer Science & Business Media.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining CT for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wu, S., Peel, A., Bain, C., Horn, M.S., Wilensky, U. (2021). Different paths, same direction: Do teachers learn CT in stem practices through professional development. Proceedings of the *5th APSCE International Computational Thinking and STEM in Education Conference*, (pp. 52-57).

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254.

Abstraction and Problem Solving in a CS Curriculum for 4th Grade

Mor Friebroon-Yesharim*, Ronit Ben-Bassat Levy, Michal Armoni
Department of Science Teaching, Weizmann Institute of Science, Rehovot, Israel
mor.friebroon@weizmann.ac.il*, ronit.ben-bassat-levy@weizmann.ac.il, michal.armoni@weizmann.ac.il

ABSTRACT

The importance of teaching computational thinking (CT) as part of K-12 education has led, among other factors, to the inclusion of computer science (CS) as a K-12 subject in many countries. Consequently, many CS curricula for different age levels have been or are being developed. On the long way between the policymakers and the learners, every curriculum has several manifestations. For example, differences may exist between the intended curriculum and the one implemented in class. In a large research project, we are currently investigating the evolution of a 4th-grade curriculum in CS, starting with the vision of the policymakers, who set the teaching goals, and proceeding through the formal curriculum, then to the corresponding teacher training and classroom implementation, until the attained curriculum, as reflected in the students' learning outcomes. Here, we focus on the manner in which the formal curriculum reflected the process of CS problem solving, including its inherent nature as a process that involves transitions between multiple layers of abstraction. This major aspect of CS is perceived by many as especially important for developing CT. Our findings indicate that the treatment of problem solving in this curriculum was characterized by lower levels of abstraction. Specifically, the presentation of problems was often characterized by lower-level descriptions, and the solution process often neglected the level of an algorithm and hence, also the transitions in which it was involved.

KEYWORDS

Introduction to computer science, problem solving, abstraction, elementary school, curricular analysis

INTRODUCTION

In recent years, K-12 computer science (CS) education has been constantly gaining momentum, even at young age levels, namely, in elementary schools and even earlier (Oda, Noborimoto, & Horita, 2021). Consequently, CS curricula have been or are currently being developed in many countries (Caspersen et al., 2022). A major factor governing the design of such curricula is the broad worldwide acknowledgment of the importance of teaching computational thinking (CT), and of the potential effect of teaching CS on the development of CT (Wing, 2006). In a large research project, we examine the evolution of a 4th-grade (ages 9-10) CS curriculum, within the framework of curricular analysis, in which a curriculum is viewed as having several manifestations, representing it in different phases of this evolution (Valverde et al., 2002). For example, a curriculum may be manifested as an intended curriculum, representing the vision of the policymakers, or as an attained curriculum, representing its learning outcomes. Our research project analyzes this curriculum as manifested in several phases of its evolution: the policymakers' vision, the elaborate formal curriculum, the teacher training, the implementation by teachers, and the students' learning outcomes. The policymakers' vision included several teaching goals, among which is the development of CT, regarding which they emphasized problem solving and abstraction. Embedded in the framework of curricular analysis, the research project follows the educational goals of the curriculum throughout the phases of its evolution. Here, we focus on the phase of the formal curriculum and the above-mentioned goal, namely, problem solving and abstraction. Besides being an educational goal of this curriculum, problem solving is widely acknowledged as a central process carried out by CS scientists and professionals, and hence it is also a major component of CT (Wing, 2006), the learning of which can contribute to general problem-solving skills (Lee & Junoh, 2019). Similarly, abstraction is widely recognized as a major aspect of CT (Wing, 2008). In the context of this paper, we used an integrated perspective of CS problem solving and abstraction. That is, we examined problem solving in CS as a process that is built on and driven by algorithmic abstraction; thus, it inherently involves transitions between levels of abstraction. Hence, abstraction plays a major role in teaching problem solving effectively, and problem solving is a highly effective context for teaching abstraction. Obviously, problem solving as well as abstraction have additional facets, which should also be addressed when aiming to achieve the two educational goals of abstraction and problem solving. These facets are addressed in other parts of our research project and will be reported elsewhere. Here, we will focus only on the integrated perspective of algorithmic abstraction.

Using content analysis, we examined the treatment of problem solving in the 90-page formal curriculum document for 4th grade, specifically the employment and the reflection of the different abstraction levels. Since formal curricula lay the foundation for the actual teaching process, the way they address educational goals is crucial for their success. This paper contributes to the field of CT and CS education by offering and demonstrating a way to investigate this with regard to the important CT educational goals of problem solving and abstraction.

This paper is organized as follows: Section 1 presents relevant background and related work. Section 2 presents the research question, and Sections 3 and 4 present the methodology and findings, respectively. Finally, Section 5 discusses the results and presents the conclusions.

1. BACKGROUND

1.1. Problem solving in CS and CT

As noted above, problem solving is considered by many as the essence of CS. The exact interpretation of this term is not always clear and CS laymen may interpret it in a way that is consistent with misconceptions of the nature of CS (Taub, Armoni, & Ben-Ari, 2012); however, there is no doubt that finding efficient and correct (or optimal) solutions for algorithmic problems lies at the heart of the discipline, and the rich set of ideas, concepts, skills, and thinking patterns that are inherent to algorithmic problem solving play an essential and necessary role in the expertise of computer scientists and professionals. At the same time, problem solving is an important skill in other sciences and professions as well as in daily life. In fact, teaching the art of problem solving is an educational challenge for educators of different disciplines, addressed by very active research areas, for example, in mathematics education (Schoenfeld, 2016) and science education (Kohl & Finkelstein, 2008). Some aspects of problem solving are necessarily discipline dependent. Nevertheless, there is a substantial common core. Problem solving requires the use of strategies and heuristics, meta-cognitive abilities for controlling the solution process, the ability to look beyond immediate contexts and resources, as well as creativity and flexibility. The amorphic, inconcrete nature of the components on this list is what makes it such an educational challenge. The call for teaching CT offers a way to cope with this challenge. Since CT encompasses all the thinking patterns, ideas, and skills employed daily by computer scientists and professionals, including those aspects required for problem solving, learning CT by experiencing CS may foster the development of these aspects, thus also fostering the acquirement of the common core abilities. Indeed, several educators have demonstrated the effectiveness of teaching problem solving in CS as a means of developing students' general problem-solving abilities, even at early ages (e.g., Lee & Junoh, 2019; Kazanci, 2017).

1.2. Problem solving and abstraction

Abstraction is a fundamental CS idea, recognized by many throughout the history of CS as the essence of the discipline (Armoni, 2013). It has many manifestations in different CS contexts (e.g., modularization, generalization, modeling, decomposition, and representation). Hence, it is also a major component of CT (Wing, 2008). As a deep and abstract idea, it is difficult to capture CS abstraction by means of a compact definition. Consequently, teaching abstraction in the context of CS is challenging, and students of different age levels have difficulties in understanding and employing abstraction (e.g., Haberman, 2004).

Abstraction is closely related to problem solving; it is used for creating an appropriate model for thinking about a given problem and for devising appropriate techniques for solving it (Aho & Ullman, 1972). Algorithmic problem solving employs algorithmic abstraction, where details of the problem at hand or aspects of its solution are either ignored or considered, depending on context and needs. Hence, the process of solving an algorithmic problem involves back and forth transitions between levels of abstraction. Many scholars have considered the ability to work at different layers of abstraction and to move freely between them as the expertise of competent computer scientists (e.g., Knuth, 2003; Wing, 2006; Dijkstra, 1975).

When studying the perceptions of undergraduate students regarding the concept of an algorithm, Perrenet, Groote, & Kaasenbrood (2005) constructed a hierarchical category system (hereafter referred to as the PGK hierarchy) in which each category reflects a perception of this concept at a certain level of abstraction. From another perspective, this hierarchy describes the process of algorithmic problem solving, where each level reflects an abstraction level employed during the algorithmic problem-solving process. The PGK hierarchy consists of 4 levels, defined as follows, from high to low: The highest level is the *Problem level*. Every process of solving algorithmic problems starts at this level, where the solver gains insights into the problem at hand and understands its nature and characteristics. For example, the solver may look for similarities to other, possibly more familiar problems, or experience with possible decompositions of the problem into smaller or simpler problems. At the level below, the *Algorithm level*, the solver designs an algorithm that solves the problem at hand. Moving to the level below, the *Program level*, the solver translates the solution, namely, the algorithm, into a programming language (such as Scratch), resulting in a concrete executable computer program. At the lowest *Execution level*, the program is executed, thus enacting the solution. The PGK hierarchy is simple and general enough to also apply it for relatively simple problems and solutions, which may be more suitable for younger students. Hence, the hierarchy concurs with Bruner's spiral teaching (1960), which is especially suitable for learning fundamental ideas.

Armoni (2013) developed a framework, based on the PGK hierarchy, for teaching abstraction to novices through algorithmic problem solving. Statter and Armoni (2020) integrated this framework into a middle-school introductory CS course and studied its effect when used by several teachers. They found that the teachers could employ the framework and that it was highly effective regarding all relevant aspects, such as using all levels of abstraction and specifically the Algorithm level, moving freely between levels of abstraction and identifying the appropriate level to work at during different phases of the solution process, and explaining solutions at the Algorithm level instead of detailing program components.

In an earlier part of our large research project (Friebroon-Yesharim & Armoni, 2022) we extended the PGK hierarchy to also include the Problem-programming level (as the second highest level) in which the problem was described using programming terms. The need for the new level emerged from the data, since the problem descriptions were included in the curriculum document. The extended hierarchy was denoted as PGK*.

1.4. The new Israeli curriculum

The new Israeli CS curriculum for elementary schools was made public by the Ministry of Education in 2016. Its implementation started in the 2017/8 school year, in about 300 schools. It was planned for two school years, starting in 4th grade, 60 hours per year. It utilizes Scratch for the 4th grade and robotics for the 5th grade. In the 4th grade, the curriculum covers topics such as algorithmics, input and output, logical expressions, variables, conditional execution, and iterative execution. Since Scratch is used, the curriculum also covers event-driven programming concepts, such as agent communication and concurrency. The curriculum was published as an elaborate document intended mostly for the teachers, and included worked examples and unsolved challenges that can be used by teachers. Despite its length, it is not a textbook, but rather, a resource for introducing teachers to the curriculum and helping them to use it.

2. RESEARCH QUESTION

Our focus on the integration of abstraction and problem solving yielded the following research question:

How does the formal 4th-grade CS curriculum reflect the abstraction-driven process of algorithmic problem solving?

The research was approved by the IRB of the Weizmann Institute of Science and by the Ministry of Education.

3. METHODOLOGY

The data for the study consist of the first version of the 4th-grade CS curriculum, as published by the Ministry of Education, which contains 90 pages. However, we were only interested in those parts that deal with problem solving and demonstrate it. Hence, the data for analysis included all worked examples that included the process of solving a given problem. This curriculum included 35 worked examples, of which 24 dealt with problem solving. The others did not refer to an algorithmic problem at all. For example, some exemplified the translation of a given algorithm (without referring to its purpose or the task it was designed to achieve) into Scratch, whereas others presented scripts including a newly learned instruction. In addition, an appendix included 15 examples, where each example contained a link to a project taken from the Scratch website, as well as a short text with didactic comments and observations concerning the project; however, since the examples referred to external ready-made projects, they were not presented as problem-solving activities; therefore, we did not include them in our data for analysis.

To address the research question, we employed a qualitative approach, using tools for document analysis (Azungah, 2018). The analysis was based on the PGK* hierarchy, thus reflecting our combined perspective of problem solving and abstraction. We used the hierarchy as a 5-category system in a deductive content analysis (Azungah, 2018) of all parts of the text that demonstrated problem-solving processes. A basic text segment could be a picture, a sentence, or where needed, segments of sentences that are fully contained at a specific level of abstraction, and any expansion of them would violate this condition. Each basic test segment was coded by the abstraction level it reflected. Using this coding, we could also learn about the transitions between levels of abstraction that took place within parts that referred to problem solving.

The analysis was performed by the first author. Then, the second author independently analyzed about 25% of the data text, after which an iterative process of review began, involving the third author, who is an expert in the field. In each cycle, disagreements (including new ones, revealed during the review) were discussed and resolved, until a need to update the coding guidelines was recognized and they were revised accordingly. This process ended when the guidelines were stable, and all disagreements were successfully resolved, namely, 100%-agreement was achieved. At the point, the data were analyzed again, using the stable guidelines. Coding was performed using Atlas.ti software.

4. RESULTS

We present our findings concerning the problem-solving processes included in the curriculum document. To depict the findings, we used transition graphs (for example, see Figure 1), in which the Y-axis includes the five PGK* levels of abstraction and the X-axis represents process progression over time. A problem-solving process is represented by a broken line in which each segment denotes an advance to a new phase in the problem-solving process, whereas a phase and the abstraction level at which it was performed are represented by a point on a process line. Thus, a transition between abstraction levels is reflected by a non-horizontal segment. We decided to organize these graphic representations by chapters, namely, a graph for each of the 10 chapters included in the formal curriculum, where a graph depicts all problem-solving processes included in the corresponding chapter (using different colors). This way, a learning progress, characterized by a better use of abstraction levels over time, is easier to recognize. Since both Chapters 1 and 8 did not include worked examples of problem solving, this analysis yielded 8 graphs. Owing to space limitations, we will present only half of them.

Figure 1 presents the graph obtained for Chapter 4. This chapter presents the concept of a variable. It included four examples (Prob1 to Prob4, respectively). For example, Prob1 started with an algorithmic problem whose description did not use any programming-related terms. This was the simple problem of multiplying a number by 5. However, the solution process skipped the Algorithm level and the process moved directly to the Programming level, where it also ended. As depicted in Figure 1, Prob1 was the only example in this chapter that started with a high-level description of a problem. The other three worked examples were presented as Scratch-related problems (for example, Prob4 was described as changing the size of a sprite), namely, their coded process started at Level 4. In these three examples, the complete processes were characterized by a low level of abstraction, since the problems were described in terms of programming and were solved only at the Programming level. The worked example of Prob3 has a unique and interesting characteristic. In this rather lengthy example, the problem was first solved using variables. Then, after executing the solution, the process climbed back to the Programming level to present a solution without variables. This was the only example in the formal curriculum that included more than one solution for the same problem.

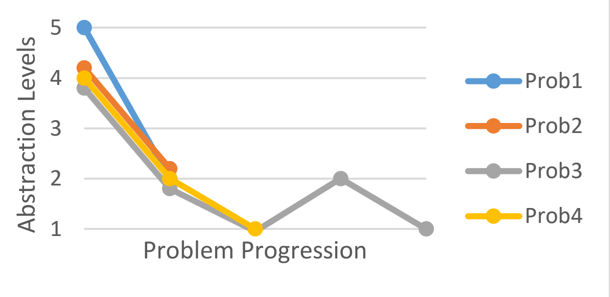


Figure 1. The problem-solving processes in Chapter 4 by phases and transitions between levels of abstraction

Similarly, Figure 2 presents the graph corresponding to Chapter 5 (which introduced conditional execution). This chapter included three worked examples. All three started with a problem description that referred to the Scratch world. Both the first and the second examples skipped the Algorithm phase, whereas the third example passed through this level. Only in the second example did the process reach the execution phase, whereas the processes described in the other two examples ended at the Programming level.

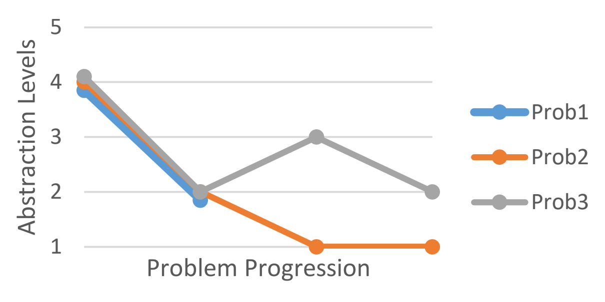


Figure 2. The problem-solving processes in Chapter 5 by phases and transitions between levels of abstraction

Chapter 6 (Figure 3) discussed the complete process of algorithmic problem solving. In line with this, it included one example of a decreasing process that started with a non-programming-related problem, which was first solved algorithmically and only then was the solution translated into programming.

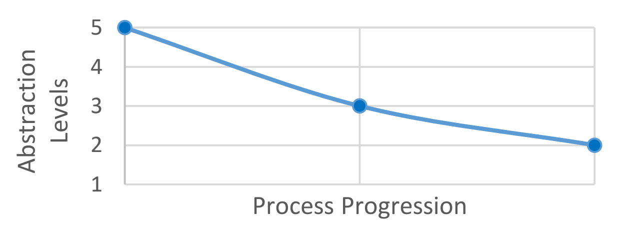


Figure 3. The problem-solving process in Chapter 6 by phases and transitions between levels of abstraction

Finally, the last chapter, Chapter 10 (see Figure 4), discussed iterative execution. It included three examples, all of which were described without involving programming-related terms. Two of them demonstrated relatively complex problem-solving processes that involved moving back and forth between levels of abstraction, including the Algorithm level. In contrast, the solution of the first was fully contained at the Programming level.

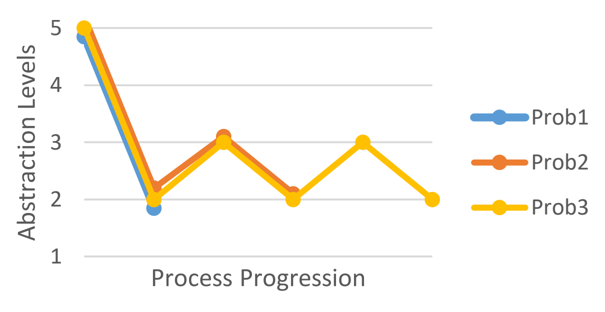


Figure 4. The problem-solving processes in Chapter 10 by phases and transitions between levels of abstraction

Figure 5 depicts another angle of our analysis. It combines all the 24 worked examples depicted in all the chapter-related graphs (of which we presented four), looking only at their starting abstraction level, which refers to the problem description. As can be seen from Figure 5, more than half of them were described in programming-related terms.

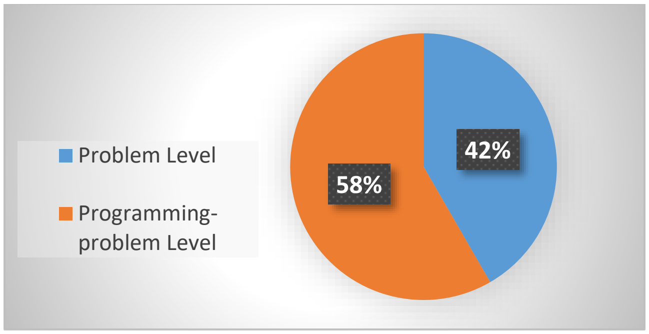


Figure 5. Problems' descriptions by abstraction level

In contrast, Figure 6 examines the starting point of the solution process. Namely, examining each process at the part that comes immediately after the problem description. Evidently, most solutions skipped the Algorithm level and went straight to programming.

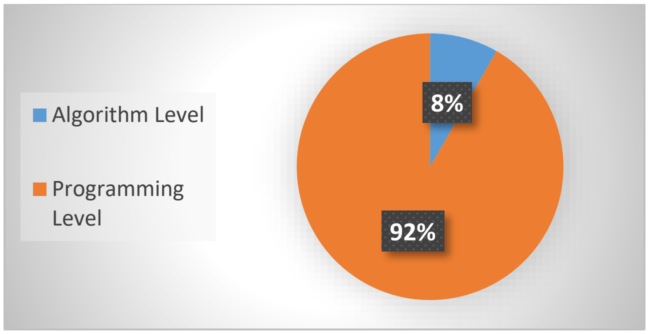


Figure 6. Problems' solutions by initial abstraction level

5. DISCUSSION AND CONCLUSIONS

The study reported here examined the way that the 4th-grade CS curriculum demonstrated the use of abstraction in algorithmic problem solving. Since both abstraction and algorithmic problem solving lie at the heart of CS and are often considered as educational goals of CS education (as is the case for the curriculum investigated in this study), examining their integrated treatment is an important part of assessing the quality of CS curricula. Furthermore, since both abstraction and problem solving are also considered major CT components whose learning can foster the development of general problem solving and abstraction skills, assessing their integrated treatment is highly important also from the perspective of CT. Therefore, our method of assessment significantly contributes to computing education research in general, and to CS/CT curricular design, in particular. As shown below, interesting insights can be gained using the information obtained by this method.

In order to address problem solving, CS/CT curricula should encourage the use of worked examples and assignments that deal with problem solving. Conforming with this principle is beneficial in several aspects. Clearly, the larger the relative part of such examples and assignments is, the more emphasized is the aspect of problem solving and possibly, the influence of such a curriculum in acquiring problem-solving skills is larger. In addition, emphasizing problem solving may foster an authentic image of CS as a discipline in which the cognitively challenging practice of problem solving plays a major role, potentially offering a remedy for prevalent misconceptions regarding the nature of CS. Furthermore, problem-based learning is known as a didactic strategy that promotes meaningful learning. In the case of the curriculum discussed here, of the 35 worked examples, 24 demonstrated algorithmic problem solving. There were also 4 assignments, each presenting a problem-solving challenge. Apparently, the presence of problem solving in this curriculum is noticeable, but it has room for improvement.

Another relevant factor is the extent to which abstraction is employed in problem solving. As a fundamental idea, internalizing abstraction calls for explicit teaching and for taking advantage of contexts that lend themselves to the use of abstraction. Therefore, missing the opportunity to integrate the teaching of abstraction into problem-solving teaching events may lead to a lower impact regarding the learning of abstraction. It may also promote or strengthen an inaccurate image of CS, for example, by recognizing CS with lower abstraction levels of algorithmic problem solving, namely, programming. In the curriculum discussed here, most of the solved problems were described using programming terms; that is, their description actually mixed two levels, obscuring the more abstract level of a problem by elements of the relatively lower level of programming. This may also encourage students to continue directly into programming, while skipping the algorithm phase. Indeed, most of the solutions began at the Program level and the Algorithm level was utilized (not necessarily as the initial level) in only 6 of the 24 problem-solving processes. Interestingly, this curriculum dedicated an entire chapter (Chapter 6) to the process of algorithmic problem solving, and indeed this chapter included an example in which the description of the given problem did not involve programming terms, and in which the solution began at the Algorithm level (Figure 3). However, Chapter 6 included only this one example, and apparently the characteristics of this example were not very prevalent outside this chapter. Similar to programming-related problem descriptions, neglecting to emphasize the Algorithm level may also lead to limited abstraction skills as well as to an inaccurate image of CS, in which the role of programming is larger than it should be.

Moving between levels of abstraction was exemplified in all problem-solving processes. However, some of these processes were very short, involving as few as one segment, from the Problem-programming level to the Program level (e.g., Prob1 in Figure 1). Moreover, most processes did not involve increasing segments, potentially leading to the interpretation of problem solving as a one-way process, and to identifying problem solving with the narrower concept of top-down design. Nevertheless, there are also relatively complex processes (see Prob2 and Prob3 in Figure 4), which nicely demonstrate multiple transitions, back and forth. Adding more processes of this nature to the curriculum, especially in the latter chapters, may improve the treatment of transitions between levels.

The analysis we presented here also highlights the challenges of developing new curricula and some of the pitfalls in which misconceptions may be enforced, perspectives may be narrowed, and educational goals may fail, somewhere along the long way from the vision of policymakers to the young students. As part of our research project, investigating the integration of abstraction and problem solving (as well as other aspects) continues to the other curricular manifestations, by examining it in teachers' training, in classrooms, and finally, in students' learning outcomes. We hope to report on the results shortly.

6. REFERENCES

Aho, A. V. & Ullman, J. D. (1972). *The Theory of Parsing, Translation, and Compiling*. Prentice-Hall, Inc.

Armoni, M. (2013). On teaching abstraction in CS to novices. *Journal of Computers in Mathematics and Science Teaching*, *32*(3), 265-284.

Azungah, T. (2018). Qualitative research: deductive and inductive approaches to data analysis. *Qualitative Research Journal*, *18*(4), 383-400.

Bruner, J. S. (1960). *The Process of Education*. Harvard University Press.

Caspersen, M. E., Diethelm, I., Gal-Ezer, J., McGettrick, A., Nardelli, E., Passey, D., ... & Webb, M. (2022). *Informatics Reference Framework for School*. ACM Press.

Dijkstra, E. W. (1975). About robustness and the like. EWD 452. *The Archive of Dijkstra's Manuscripts*. (1975). http://www.cs.utexas.edu/users/EWD/.

Friebroon-Yesharim, M., & Armoni, M. (2022). The Tale of an Intended CS Curriculum for 4th Graders, The Case of Abstraction. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2* (pp. 623-623).

Haberman, B. (2004). High-school students' attitudes regarding procedural abstraction. *Education and Information Technologies 9*(2), 131–145.

Kazanci, Z. (2017). Teaching computer algorithm as a medium of improving problem solving skills in elementary school students. *International Journal of Social Science and Humanity*, *7*(3), 179.

Knuth, D. E. (2003, June). Bottom-up education. In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 2-2).

Kohl, P. B., & Finkelstein, N. D. (2008). Patterns of multiple representation use by experts and novices during physics problem solving. *Physical Review Special Topics-Physics Education Research*, *4*(1), 010111:1-13.

Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, *47*(6), 709-716.

Oda, M., Noborimoto, Y., & Horita, T. (2021). International trends in K-12 computer science curricula through comparative analysis: Implications for the primary curricula. *International Journal of Computer Science Education in Schools*, *4*(4), 24-58.

Perrenet, J., Groote, J. F., & Kaasenbrood, E. (2005). Exploring students' understanding of the concept of algorithm: levels of abstraction. *ACM SIGCSE Bulletin*, *37*(3), 64-68.

Schoenfeld, A. H. (2016). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics (Reprint). *Journal of Education*, *196*(2), 1-38.

Statter, D., & Armoni, M. (2020). Teaching abstraction in computer science to 7th grade students. *ACM Transactions on Computing Education (TOCE)*, *20*(1), 8:1-37.

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, *12*(2), 8:1-29.

Valverde, G. A., Bianchi, L. J., Wolfe, R. G., Schmidt, W. H., & Houang, R. T. (2002). *According to the Book: Using TIMSS to Investigate the Translation of Policy into Practice through the World of Textbooks*. Springer Science & Business Media.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725.

A Review of Learning Progressions in K-12 Computational Thinking Education

Yutian Ma
The University of Hong Kong, Hong Kong
Yutian95@connect.hku.hk

ABSTRACT

Computational thinking (CT) integration in K-12 education has gained rapid attention in recent years. To facilitate the integration, determining the age-appropriate learning progressions (LP) for K-12 students is vital. This study systematically reviewed 25 CT-related LPs articles to gain an in-depth understanding of the existing CT LPs, and how they were developed. The results show that LPs studies in CT education are in the preliminary stage: most of the LPs are not fully developed or haven't been validated; the CT elements were addressed unevenly in the LPs studies; the targeted grain size and grade range vary among the studies. Suggestions for the development and validation of LPs in CT education are provided for future work.

KEYWORDS

Computational Thinking, Learning Progression, K-12

1. INTRODUCTION

Popularized by Wing (2006), computational thinking (CT), has been viewed as a fundamental skill that every child should learn in the 21st century. Encouraged by this claim, researchers, educators, and policymakers have recently promoted the integration of CT into basic education (Tang et al., 2020). However, the approaches in different countries vary significantly regarding educational objectives, educational levels, and position in the school curricula (Bocconi et al., 2016). And there is no widely accepted comprehensive standard on what, how, and when to teach, learn and assess CT at the K-12 level. The common challenge of promoting CT into K-12 education faced by all countries is to develop age-appropriate learning sequences paired with teaching materials and to prepare teachers with relevant knowledge and skills (Falkner & Vivian, 2015).

For some well-established subjects such as math and science, learning progression (LP) studies have emerged over the past years to address the need of representing what students should know and be able to do at different levels more precisely (Duncan & Rivet, 2018). LPs normally refer to the descriptions of the increasing levels of complexity of the disciplinary knowledge and practices for students from primary schools to develop and refine skills over time (Battista, 2011). The LPs aim to work as hypothetical models to inform the design of the curriculum, teaching materials and assessment, hence, promoting teaching and learning (Jin et al., 2019).

The main difficulty in creating age-appropriate learning progressions of CT lies in defining CT and identifying what it entails (Grover & Pea, 2013; Bocconi et al., 2016). The different conceptualizations of CT and diverse emphasis on CT elements may lead to different learning progressions when designing curriculum, instruction, and assessment for CT. This study aims to identify what kinds of LPs of CT have been developed, how they have been developed as well as the appropriateness and effectiveness by reviewing and examining the current studies on CT LPs at the K-12 level.

2. THEORETICAL FRAMEWORK

2.1. Definitions and Elements of CT

The most widely accepted CT definition was proposed by Wing (2011), which stated that "*CT is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.*" (p.1). Based on this general definition, CT has been broken into sets of elements and dimensions from very different perspectives by researchers and experts (Tang et al., 2020; Zhang & Nouri, 2019). The debate lies in whether the information-processing agent could be a machine or human; whether CT skills can be learned outside programming etc. Those who hold a broader view may propose dimensions mainly on general thinking habits such as abstraction, pattern recognition, generalization, decomposition, and algorithmic thinking (Selby & Woollard, 2013). Others may also include the CS and programming concepts such as modularizing, iterative, recursive and parallel thinking (Grover & Pea, 2013).

Table 1. Classification of CT Elements.

Mental Processes: **Mental strategies useful to solve problems** E.g., Algorithmic Thinking, Logical Thinking, Problem Decomposition and Modularization, Abstraction, Pattern Recognition, Generalization, Evaluation,
Methods: **Operational Approaches widely used by computer scientists** E.g., Automation, Data Representation, Parallelization, Simulation, Programming
Practices: **Typical practices used in the implementation of computing machinery-based solutions** E.g., Experimenting and Iterating, Testing and Debugging, Reusing and Remixing
Transversal skills: **General skills that can enhance thinking like a computer scientist** E.g., Create and Design, Communicate and Collaborate, Reflect, Be tolerant for ambiguity, Be persistent when dealing with complex problems.

Lodi and Martini (2021) analyzed several CT definitions and classified the elements of CT into four categories: mental processes, methods, practices and transversal skills. Table 1 presents the explanations of the categories and corresponding example elements.

2.2. Definitions of Learning Progression

CT Learning progression studies have just emerged in recent years, while LPs in math and science education have been widely explored for their critical potential to align and develop curriculum, instruction, and assessment (Jin et al., 2019). Learning progression is defined as "the descriptions of the successively more sophisticated ways of thinking about a topic that can follow one another as children learn about and investigate a topic over a broad span of time" (National Research Council, 2007, p.214).

Duncan and Hmelo-Silver (2009) summarized four key characteristics of LP. First, LPs are structured around fundamental concepts and practices of a discipline. Second, LPs usually encompass a lower and an upper anchor. Third, layered intermediate levels of achievement, referring to the coherent and reasonable networks of knowledge and skills that reveal students' levels of understanding and competency, are provided to link the two anchors. Usually, the construction of the sequences is based on analysis of domain research and empirical studies on student learning and LPs. Lastly, LPs are influenced by instruction and curriculum. In mathematics education, a similar term learning trajectory (LT) is widely used to refer to the learning paths.

In summary, the conceptualizations and definitions of CT held by the experts and researchers decide the sub-elements involved in CT LPs and the learning contexts of CT. LPs could be developed in different ways and forms, reflecting different characteristics. Also, LPs need to be validated thus to support the establishment of curriculum, design of instruction and assessment effectively. Based on these premises, the following research questions (RQs) are proposed to guide the review and gain an in-depth understanding of CT LPs. In the process of developing CT LPs in K-12 education:

RQ1: What are the definitions, learning context and sub-elements of CT in the LPs studies?

RQ2: What are the conceptualizations of LP?

RQ3: What are the methods used to develop and justify the applicability of the CT LPs?

RQ4: What are the characteristics (age appropriateness, grade span and size) of the existing CT LPs?

3. METHOD

To answer the research questions, an integrative review (IR) is conducted to is adopted since it doesn't limit the type of the selected literature to empirical studies (Toronto, 2020). It is suitable for providing a holistic understanding of an emerging topic by following a systematic search and synthesis of the appraised literature (Toronto, 2020).

3.1. Search Strategies

The preferred reporting of items for systematic review and meta-analysis (PRISMA) protocol is used to guide the process of selecting relevant studies (Moher et al., 2009). Six databases were searched: Scopus, Web of Science, SpringerLink Electronic Resources, Ebscohost, ProQuest Research Library, and Learn Techlib. The search terms with Boolean operators used in this review were as follows: ("computational thinking") AND ("learning progression" OR "learning path*" OR "learning trajector*"). Since the term CT was popularized by Wing in 2006, the publication date of the articles searched is set as from 2006 to April 2021.

3.2. Study Selection

For the selection of relevant articles, six specific inclusion criteria are generated based on the research questions. The study should a) address LPs in the context of CT education; b) report the LPs at the K-12 level; c) report detailed LPs or the development process of LPs; 4) be a theoretical or empirical study; 5) not be a short paper or a poster; 6) written in English.

The Process of the study selection is presented in the PRISMA flow diagram as shown in Figure 1.

Identification: Records identified through database searching (n=1640) (Scopus=441, LearnTechLib =718, Ebscohost=158, WoS=32, ProQuest=55)
Screening: Records after duplicates removed = Records for Screening (n=1199)
Eligibility: Full-text articles assessed for eligibility (n=140) + Snowball sampling added (n=3)
Incuded: Studies included in this review (n=25)

Figure 1. PRISMA Flow Diagram of Article Selection

4. RESULTS

25 articles were selected based on the above criteria, including journal articles (n=10), conference proceedings (n=13), book chapters (n=1), and reports (n=1). 14 of them are empirical studies, and 11 of them are non-empirical studies.

4.1. CT Definitions, Learning Contexts and Sub-elements

Of the 25 studies, 24% of them (n=6) didn't provide an explicit definition or explanation of CT. 76% of the studies (n=19) explicitly define or explain CT provide an explicit definition of CT or explanation of CT.

The most cited definition is Wing's (2006, 2011) interpretation of CT (n=8), Brennan and Resnick's (2012) framework (n=4), and the CSTA and ISTE's (2011) operational definition (n=3).

Given that different authors hold different views about whether CT could be acquired outside programming, the LPs were developed under different learning contexts. Most of the studies (n=16, 64%) develop LPs in programming contexts. A small group of studies (n=6, 24%) combine both unplugged (non-technology) and programming contexts to develop LPs. Only two (8%) studies develop LPs solely in unplugged contexts.

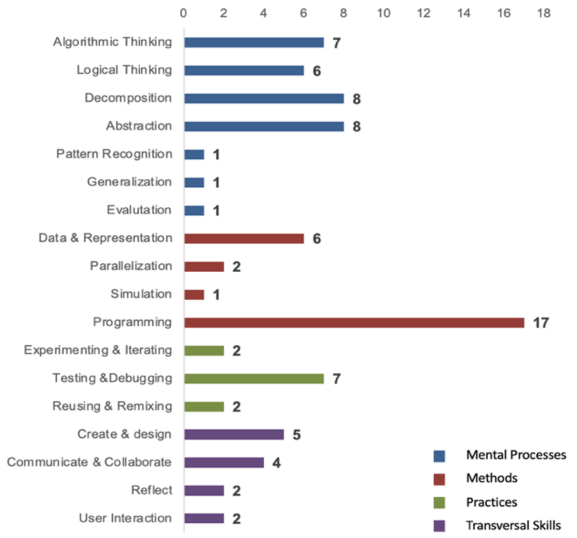


Figure 2. CT Elements in LP Studies

As mentioned in section 2.1, Lodi and Martini (2021) classified CT elements into four categories: mental processes, methods, practices and transversal skills. The most investigated category is methods (n=19), followed by mental processes (n=14). Practices (n=7) and transversal skills (n=6) were less discussed. Among all the selected studies, 13 of them focused on a single category, and 12 studies included CT elements from more than one category. However, these studies didn't pay much attention to exploring the connections among elements from different categories. Only a few studies (n=3) pointed out the connections between algorithmic thinking and programming (Ko & Delgado, 2013; K–12 Computer Science Framework, 2016; Dwyerz et al., 2014). Although CT has been decomposed into different dimensions for educational practices, the development of CT requires the combination of mental strategies, operational approaches, implementation practices as well as transversal skills.

Figure 2 demonstrates the distribution of the CT elements that appeared in the LP studies. The covered CT elements are very uneven in these studies. The most investigate CT element is programming (68%, n=17), including concepts such as sequence, conditional, loop, variable, operator, event etc. Then follows by abstraction, decomposition, algorithmic thinking, debugging, logical thinking, data and representation This indicates the high emphasis on programming in the development of CT, which is consistent with the narrow distinction between CT and programming in empirical studies (Ezeamuzie & Leung, 2021).

4.2. Conceptualization of LP

LP studies adopted different terminology to refer to the learning sequences. Around half of them (n=14, 56%) adopted "learning progressions"; 24% of them (n=6) used "learning trajectories"; 12% of them (n=3) adopted "path/pathway"; 1 of them mainly used "sequences" while citing CT learning progressions and learning trajectories studies, 1 curriculum framework didn't use either of the terms.

Among the 25 articles, 22 of them provided a definition or explanation of these terms, indicating two main conceptualizations of LP. One dominant conceptualization adopted by 44% of the studies (n=11) regards LP as the scope and sequence of different levels of ideas or skills that should be taught and assessed. For example, the description could be "A learning progression is a sequence of subskills that need to be mastered..."(Dwyerz et al., 2014, p.2), "...knowledge that is ordered as hierarchic constructs..." (Niemel et al., 2017, p.3). Another 36% of studies (n=9) may borrow the definition widely accepted in math and science LP research which emphasize the role of students' ways of thinking in the development of LP. Examples are "...descriptions of the successively more sophisticated ways of thinking about a topic..." (Rich et al., 2019, p.745).

These two conceptualizations suggest different approaches to developing LP (Ducan & Rivet, 2018). The first conceptualization implies a logical analysis approach, determining the scope and sequence based on the logical analysis of normative knowledge in the domain and the conventional wisdom of practice (Daro et al., 2011). The latter conceptualization indicates an evidence-based approach, by which LPs are grounded in actual empirical research on how students' understanding of core ideas grows (Ducan & Rivet, 2018).

4.3. Methods for Developing and Justifying CT LPs

Around two-thirds of the studies (n=17, 68%) adopt an evidence-based approach, and the rest one-third of the studies (n=8, 32%) utilize a logical analysis approach. Compared with the conceptualizations of LP discussed in section 4.3., the inconsistency between the conceptualizations of LPs and the operational processes of these studies could be identified.

4.3.1. Evidence-based Approach

Studies seeking empirical evidence on students' learning process (n=17) can be categorized into theoretical LPs (n=7) and empirical research of student learning (n=10).

The theoretical LPs refer to the LPs that are developed based on empirical literature without further validation by empirical data. There are also two different ways to propose the theoretical LPs. Zhang and Nouri (2019) and Sullivan and Heffernan (2016) reviewed the empirical research and proposed the LPs based on the empirical evidence about what CT skills can be obtained by students. Another five studies conducted by Rich's team (Rich et al., 2019, Rich et al., 2020, Rich et al., 2018a) and the K-12 CS framework (2016) not only seeking empirical evidence but also theoretical support to decide the appropriate sequence of the LPs due to the lack of related empirical research.

The focus of the empirical research of student learning diverges into three dimensions:1) develop empirical LPs based on assessing students' programming artefacts (Alves et al., 2020; Seiter & Foreman, 2013; Moreno-León et al., 2017) such as Scratch and App Inventor projects to characterize levels of sophistication; 2) observing short or long teaching experiments to develop the LPs (Ber, 2019; Kalas et al., 2018; Israel & Lash, 2020, Zhang et al., 2020); 3) implement the LP-based instruction and curriculum in real practices (Freina et al., 2018; Djurdjevic-Pahl et al., 2017; Dwyerz et al., 2014).

4.3.2. Logical Analysis Approach

Different from evidence-based LPs, LPs developed through logical analysis (n=8) are all theoretical LPs without further validation. Half of the studies in this group (Angeli et al.,2016; Iyer, 2019; Rose et al., 2020; Høholt et al., 2021) presented the LPs directly based on the researchers' experience and understanding of the topic without providing a comprehensive analysis method. The other studies adopted different methods, but all derived from analysis of the discipline structure and the conventional wisdom of the practice. For example, Ko and Delgado (2013) developed a hypothetical LP of algorithmic thinking in K-12 mainly based on the programming textbooks to identify the core concepts, and relevant curriculum framework to decide the age appropriateness and the analogy of learning CT as learning a foreign language. Allsopp and Misfeldt (2019) took two steps to develop the LP of programming: 1) use a concept specification map to overview critical programming concepts about each other from the most general one to the most specified one; 2) identify and decide progression levels for when different concepts are taught by identifying sub-maps for different levels through at least 10 iterations. Instead of relying on the existing curriculum framework or content, Niemel et al. (2017) rely on teachers' experience by collecting teachers' learning data from a CT MOOC course and analyzing the ideas and proposals in their essays to form a learning trajectory.

4.4. Characteristics of the Existing CT LPs

4.4.1. Age Appropriateness

Among the LPs studies, 40% of them didn't connect the different levels of LPs with exact age or grade though they pointed out the overall grade span of LPs such as K-12 or K-8. The reason could be that most of the empirical studies report that the participants had little/no programming experience before entering the study, no matter which grade they were in (Rich et al., 2017). Similar evidence on some learning concepts can be found among different age groups, making it difficult to tie the learning concepts with age or grade. K-12 CS framework (2016) also relied on the wisdom of related math and science research and practices to determine the grade level with LPs.

4.4.2. Grade Span and Grain Size

The grade span of the LPs varies among the studies. 76% of the LPs (n=19) include no less than two grade bands or key stages. 20% of the LPs (n=5) targeted one grade band or one grade.

The grain size of LPs refers to the grade range of each level of the LPs (Ducan & Rivet, 2018). One that includes levels within a grade can be viewed as a finer-grain size. Therefore, we mapped the studies into three categories based on the grain size: coarse (n=15), medium (n=4) and fine grain size (n=5) represent the covered range of each level of a progression from grade band, grade level to within a grade. LPs with a coarse grain size are in majority, which has the potential to inform curriculum standards and the design of large-scale assessments. The LPs with a finer grain size are better suited for instructions and curriculum development. However, it is difficult to ensure coherence among the LPs with different granularity due to the diversity of the covered progress variables.

5. DISCUSSION

This section discusses important findings and the three important implications for future research in this field: 1) leveraging the power of both logical analysis and empirical research to enrich the development of CT LPs; 2) collecting both qualitative and quantitative data to validate the existing LPs; 3) viewing CT as a whole competency and making explicit the connections among the sub-elements of CT when developing LPs.

5.1. Combining Two Approaches to Develop CT LPs

The results show two dominant conceptualizations of LP suggesting two different ways to develop LPs in CT education: evidence-based approach and logical analysis approach. No matter which conceptualizations they held, they sought empirical evidence on how students' understanding, and skill developed to generate the LPs. CT in K-12 education is an emerging topic, the research and didactic materials of which are largely based on theoretical reasoning and experience (Alves et al., 2020). What students learn may not match what to teach in the curricula (Zhang & Nouri, 2019). Informed by constructivism theory believing that learners construct new knowledge based on pre-existing knowledge, Dwyerz et al. (2014) also identified the need to understand how students learn and what they already know before entering the curriculum.

However, several studies that conceptualized LP as students' successively more sophisticated ways of thinking way of thinking of a topic didn't ground on students' empirical evidence. (Rose et al., 2020; Ko & Delgado, 2013). The possible reason for this could be the lack of empirical work on students' thinking and understanding in k-12 CT education (Ko & Delgado, 2013).

Both two approaches are important to inform curriculum, instruction and assessment. The logical analysis of the domain developed by experts points out the important knowledge that students should learn. However, the judgement about the age appropriateness of the content may be questionable. Current research on CT/CS learning in K-12 is not sufficient and is primarily composed of small, short-term studies, conducted with students from diverse populations at different times in different contexts (Hsu et al., 2018). Also, the studies vary widely in quality, methods, and underlying theoretical models, applied to different subjects (Kalelioğlu, 2018). Considering the problematic aspects and the strengths of the two roads to LPs, we proposed that developing hypothetical LPs should start with a dialogue between these two essential but incomplete resource standards and research on CT/CS learning in K-12.

5.2. The Validity Issues of LPs in CT

LPs are conjecture models of learning over time when they first developed no matter based on logical analysis or empirical evidence. They are hypothetical models by nature that need to be empirically validated (Duncan & Rivet, 2018). Thus, the threats to the validity of the LPs come from two aspects: the development phase and the validation phase. In the development phase, theoretical LPs could be influenced by the limitations of the reviewed studies in terms of sample size, the duration of the interventions etc. (Sullivan & Heffernan, 2016; Rich et al., 2018b). For LPs that emerged from empirical studies, researchers reported the LPs are not fully developed (Israel & Lash, 2020; Seiter & Foreman, 2013), given the limited number of collected students artefacts (Moreno-León et al., 2017) and the lack of consistency of the assessment adopted by different teachers (Zhang et al., 2020). These LPs should be tested and refined in multiple settings across students with different backgrounds to provide a practical pathway to inform curriculum development, classroom teaching, and assessment of CT/CS education.

Most of the LPs in the reviewed studies are hypothetical LPs that haven't been validated yet. The difficulties in monitoring students' learning systematically lie in the lack of reliable assessment methods to capture students' level of thinking and content knowledge (Djurdjevic-Pahl et al., 2017; Freina et al.;2018; Rich et al., 2020).

LP studies in science provide insights to validate LPs by using both qualitative and quantitative methods to evaluate the meaning of, the order of and the distinctions of the LP levels (Jin et al., 2019). Researchers normally use LP-based assessment to validate LPs. Cognitive interviews about the assessment were conducted to examine whether the LPs capture students' thinking (Jin et al., 2019). Wright maps and Rasch models are often used to validate the order and the distinctions of the LP levels.

5.3. Viewing CT as a Whole Competency

The reviewed studies show an uneven distribution of the investigated CT elements. The most popular CT element is programming which has been viewed as an effective way to cultivate CT. The high emphasis on programming in CT studies may narrow the distinctions between them and overlook the problem-solving aspect of CT.

Also, LP studies tend to investigate the CT elements individually instead of pointing out the connections among elements from different categories. However, only developing the single elements of CT may lose sight of the big picture of CT as a problem-solving competency. Therefore, the LPs studies should consider the dependency among the different elements to provide a coherent understanding of students' developmental progressions of CT. Curriculums developed based on the individual LPs may lead to a risk that students see multiple topics as a disjoint set of independent academic ideas and practical skills without knowing the big picture of CT (Bell, 2018).

6. REFERENCES

Allsopp, B. B., & Misfeldt, M. (2019). Visual representations supporting the implementation of a K12 programming curriculum in open and democratic educational institutions. *Proceedings of the European Conference on E-Learning, ECEL*, *2019-Novem*, 24–31.

Alves, N. D. C., von Wangenheim, C. G., Hauck, J. C. R., Borgatto, A. F., & Andrade, D. F. (2020). An item response theory analysis of the sequencing of algorithms & programming concepts. In *Proceedings of the International Conference on Computational Thinking Education*.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47-57.

Battista, M. T. (2011). Conceptualizations and issues related to learning progressions, learning trajectories, and levels of sophistication. *The Mathematics Enthusiast*, *8*(3), 507-570.

Bell, T. (2018). Computer Science in K-12 Education: The Big Picture. *Olympiads in Informatics*, *12*(7), 3-11.

Bers, M. U. (2019). Coding as another language: a pedagogical approach for teaching computer science in early childhood. *JOURNAL OF COMPUTERS IN EDUCATION*, *6*(4), 499–528.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing Computational Thinking in Compulsory Education - Implications for policy and practice. In *Joint Research Centre (JRC)*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

CSTA, & ISTE (2011). Operational Definition of Computational Thinking for K-12 Education. Retrieved from http://www.iste.org/docs/pdfs/Operational-Definition-of-Computational-Thinking.pdf

Daro, P., Mosher, F. A., & Corcoran, T. B. (2011). Learning trajectories in mathematics: A foundation for standards, curriculum, assessment, and instruction.

Djurdjevic-Pahl, A., Pahl, C., Fronza, I., & El Ioini, N. (2017). A Pathway into Computational Thinking in Primary Schools. In T.-T. Wu, R. Gennari, Y.-M. Huang, H. Xie, & Y. Cao (Eds.), *Emerging Technologies for Education* (pp. 165–175)

Duncan, R. G., & Hmelo-Silver, C. E. (2009). LPs: Aligning curriculum, instruction, and assessment. *Journal of Research in Science Teaching*, *46*(6), 606–609.

Duncan, R. G., & Rivet, A. E. (2018). Learning progressions. In *International Handbook of the Learning Sciences* (pp.422–432).

Dwyerz, H., Hilly, C., Carpenterz, S., Harlow, D., & Franklin, D. (2014). Identifying elementary students' pre-instructional ability to develop algorithms and step-by-step instructions. *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 511–516.

Ezeamuzie, N. O., & Leung, J. S. (2022). Computational thinking through an empirical lens: A systematic review of literature. Journal of Educational Computing Research, 60(2), 481-511.

Falkner, K., & Vivian, R. (2015). A review of Computer Science resources for learning and teaching with K-12 computing curricula: an Australian case study. *Computer Science Education*, *25*(4), 390–429.

Freina, L., Bottino, R., & Ferlino, L. (2018). A Learning Path in Support of Computational Thinking in the Last Years of Primary School. In H. Gentile, M and Allegra, M and Sobke (Ed.), *GAMES AND LEARNING ALLIANCE, GALA 2018* (pp. 16–27).

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43.

Høholt, M., Graungaard, D., Bouvin, N. O., Petersen, M. G., & Eriksson, E. (2021). Towards a model of progression in computational empowerment in education. *International Journal of Child-Computer Interaction*, *29*, 100302.

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: mathematics plus computational thinking. *INTERACTIVE LEARNING ENVIRONMENTS*, *28*(3, SI), 362–382.

Iyer, S. (2019). Teaching-Learning of Computational Thinking in K-12 Schools in India. In S.-C. Kong & H. Abelson (Eds.), *Computational Thinking Education* (pp. 363–382).

Jin, H., Mikeska, J. N., Hokayem, H., & Mavronikolas, E. (2019). Toward coherence in curriculum, instruction, and assessment: A review of learning progression literature. *Science Education*, *103*(5), 1206–1234.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Kalelioğlu, F. (2018). Characteristics of studies conducted on computational thinking: A content analysis. In *Computational thinking in the STEM disciplines* (pp. 11-29).

Ko, P., & Delgado, C. (2013). A Proposal for a Hypothetical K-12 Learning Progression Set for Algorithmic Thinking. *American Educational Research Association, San Francisco, California*

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883-908.

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & PRISMA Group*. (2009). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Annals of internal medicine*, *151*(4), 264-269.

Moreno-León, J., Robles, G., & Román-González, M. (2017). Towards Data-Driven Learning Paths to Develop Computational Thinking with Scratch. *IEEE Transactions on Emerging Topics in Computing*, *8*(1), 193–205.

National Research Council. (2007). *Taking science to school: Learning and teaching science in grades K-8*. National Academies Press.

Niemel, P., Partanen, T., Harsu, M., Leppänen, L., & Ihantola, P. (2017). Computational thinking as an emergent learning trajectory of mathematics. *ACM International Conference Proceeding Series*, 70–79.

Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, D. (2018a). Decomposition: A K-8 computational thinking learning trajectory. *ICER 2018 - Proceedings of the 2018 ACM Conference on International Computing Education Research*, 124–132.

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2018b). K-8 learning trajectories derived from research literature. *ACM Inroads*, *9*(1), 46–55.

Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, D. (2019). A k-8 debugging learning trajectory derived from research literature. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 745–751.

Rich, K. M., Franklin, D., Strickland, C., Isaacs, A., & Eatinger, D. (2020). A Learning Trajectory for Variables Based in Computational Thinking Literature: Using Levels of Thinking to Develop Instruction. *Computer Science Education*, *00*(00), 1–22.

Rose, S., Habgood, M. P. J., & Jay, T. (2020). Designing a Programming Game to Improve Children's Procedural Abstraction Skills in Scratch. *Journal of Educational Computing Research*, *58*(7), 1372–1411.

Seiter, L., & Foreman, B. (2013). Modelling the learning progressions of computational thinking of primary grade students. *ICER 2013 - Proceedings of the 2013 ACM Conference on International Computing Education Research*, 59–66.

Selby, C., & Woollard, J. (2013). Computational Thinking : The Developing Definition. *ITiCSE Conference 2013*, 5–8.

Sullivan, F. R., & Heffernan, J. (2016). Robotic Construction Kits as Computational Manipulatives for Learning in the STEM Disciplines. *Journal of Research on Technology in Education*, *48*(2), 105–128.

Tang, K. Y., Chou, T. L., & Tsai, C. C. (2020). A Content Analysis of Computational Thinking Research: An International Publication Trends and Research Typology. Asia-Pacific Education Researcher, 29(1), 9–19.

Toronto, C. E. (2020). Overview of the Integrative Review. *A Step-by-Step Guide to Conducting an Integrative Review*, 1-9.

Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Wing, J. (2011). *Research notebook: Computational thinking-what and why?* The Link Magazine

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers and Education*, *141*(September 2018), 103607.

Zhang, L. C., Nouri, J., & Rolandsson, L. (2020). Progression of Computational Thinking Skills in Swedish Compulsory Schools with Block-based Programming. *ACE 2020 - Proceedings of the 22nd Australasian Computing Education Conference, Held in Conjunction with Australasian Computer Science Week*, 66–75.

Use Community Problem-Solving to Engage All Students in Computational and Statistical Thinking

Li XU
University of Arizona, USA
lxu@arizona.edu

ABSTRACT

Guzdial (2022) suggested that one of the reasons computer science courses fail societally marginalized students is because “the curriculum remains disconnected from students’ lives and is not designed to serve them.” How to make learning real to students? There are examples and ways to engage students in their learning including but not limited to hands-on projects, group works, problem solving, and real-world connections. In this paper, I will reflect my experiences in designing and developing a course that utilizes community problem-solving to encourage all students to identify and solve real-world problems that happen in their community. The course framework integrates Course-based Undergraduate Research Experience (CURE), which offers all students opportunities to engage in research as a part of the structured course. CURE provides an effective approach to engage students into more active learning (Kuh, 2008), and support students to accomplish learning outcomes and persistence (Freeman et al., 2014) in STEM majors. The integration of CURE in the course design can facilitate teaching Computational Thinking (CT) and Statistical Thinking (ST). The paper presentation intends to provide specific topics to add to curricula by the structured and organized course development based on an integrated framework to engage students in learning using meaningful community problem solving. Additionally, I will describe the teaching strategies that are employed to facilitate teaching CT and ST, as well as the challenges and constraints to develop the course framework and implement the course development.

KEYWORDS

Course-based Undergraduate Research Experience, Computational Thinking, Statistical Thinking, Community Problem-Solving

1. INTRODUCTION

In today’s rapidly updated technological landscape, Computational Thinking (CT) and Statistical Thinking (ST) are in high demand across many industries including tech and software development, finance, and healthcare. Nolan and Temple Lang (2010) noted that “Computational literacy and programming are as fundamental to statistical practice as mathematics.” Horton and Hardin (2021) believe that “the leading thinkers of the next decade will be those who seamlessly knit together tools from both statistics and computing and that how we think about statistics will be informed by complementary computational thinking.” CT helps students to develop problem solving skills and creativity that can be applied in many professions and their daily lives. ST helps students to develop competency to make data-driven decisions and evaluate arguments critically. Educators agree that CT and ST are critical cognitive dispositions students need to adopt. Both CT and ST have been recognized as critically important learning outcomes in computing curricula offered in higher education institutions.

CT and ST are different skill sets: CT focuses on supporting students to develop proficiency in programming, while ST centers on statistical analysis and machine learning. Despite the increasing need to develop CT and ST skills, historically students view CT and ST difficult and unpleasant to learn. Educators view teaching CT and ST as a challenging task to approach due to the multidisciplinary nature of CT and ST as well as limited resources and training to teach the subjects effectively. Woodard and Lee (2021) reported on a study that computing is difficult and not intuitive. Ben-Zvi and Garfield (2004) also summarized that “many statistical ideas and rules are viewed complex, difficult, and/or counterintuitive.” Students are typically intimidated by the subjects in CT and ST. Especially, socially marginalized students who don’t have equal access to technology and resources face additional challenges due to their lack of prior exposure to CT and ST. Moreover, Guzdial (2022) suggested that one of the reasons computer science courses fail societally marginalized students is because “the curriculum remains disconnected from students’ lives and is not designed to serve them.”

Then, how to make learning real to students? And how to engage all students in learning CT and ST and prevent them from shying away from the subjects? Nowadays there are a large amount of community data sources available for learning and teaching data-driven CT and ST. Processing community data involves the use of computational and statistical methods and techniques to analyze, manipulate, and draw insights from data that pertains to people who live in the communities. Utilizing inclusive community problem solving is a powerful strategy to engage all students in learning CT and ST using public community data. The community data can engage students in working on problems that are important to their community. The process involves identifying community problems, gathering data, developing a solution, refining, and evaluating the solution. The problem-solving results are meaningful and could make a positive impact on their local community. The community problem-solving process can be implemented as a CURE project in a course development. CURE, which stands for Course-based Undergraduate Research Experience, involves conducting a research process to address questions or solve problems experienced by everyone in a class. Researchers reported that a CURE project in a classroom can engage students into more active learning (Kuh, 2008), support students to accomplish learning outcomes and persistence (Freeman et al., 2014), and improve graduation rates and retention in STEM majors. CURE provides an effective approach to offer research experience at scale (Mogk and Goodwin, 2012; NAC, 2015; Wei and Woodin, 2011) before students approach the senior capstone course. According to previous studies, evidence shows that CURE is beneficial to students by hands-on research experiences that support students to develop knowledge and competency skills in STEM fields.

In my institution, faculty integrated both CT and ST in our Applied computing curriculum. We recognized the role of computing to teach ST as noted by Nolan and Temple Lang (2010). In the program, we offer two introductory courses: one is for CT and the other is for ST. We also believe it is necessary to support students to strengthen CT and ST skills explicitly in the program curriculum after the introductory courses. One undergraduate course I designed follows the two introductory courses on CT and ST intends to synthesize student learning in CT and ST in data-driven problem solving. In this paper I will present the course design that aims to use community problem-solving to engage students by integrating CURE into an online course to explicitly teach both CT and ST in computing. By using community problem solving to learn the thinking skills, students can see the real-world implications of their learning results and develop connections between their learning and responsibilities within their community. The paper presentation centers around the below research questions:

R1: What topics to teach in CT and ST while applying CURE to solve problems using community data?
R2: How to use community problem solving to teach CT and ST effectively?
R3: What are the challenges and constraints involved in what and how to teach CT and ST?

The integration of CURE with CT and ST centering around community problem solving enables students to "develop a mind set with a strong focus on data – the collection of data and, through analyzing it appropriately, using this to bring about beneficial insights and changes," which is required by graduates who study data science (Danyluk and Leidig, 2021). In this paper, Section 2 addresses the first research question R1, describing the integrated course framework on CT, ST and CURE as well as student learning outcomes. Section 3 addresses R2, listing teaching strategies to teach CT and ST so that the course content makes sense to students and engages students into learning by doing. Section 4 addresses R3, summarizing the challenges and constraints regarding what and how to employ community problem solving to teach CT and ST within a course development. And finally, Section 5 concludes the study and plans the future works.

2. TEACHING FRAMEWORK

When developing the course content, the instructor is mindful to explicitly teach and formally train students in both Computing Thinking (CT) and Statistical Thinking (ST), which are related but distinct skills that are important for students to learn in today's data-driven and technology focused world. The course framework emphasizes the computing and statistical literacies students need to know, experiences students need to develop their characters, practical knowledge they need to learn so that they can develop competencies to solve problems driven by CT and ST. In particular, the course content is designed based on reflections on the questions raised by Nolan and Temple Lange (2010) originally:

1. When they graduate, what ought our students be able to do computationally, and are we preparing them adequately in this regard?
2. Do we provide students the essential skills needed to engage in statistical problem solving and keep abreast of new technologies as they evolve?
3. Do our students build the confidence needed to overcome computational challenges to, for example, reliably design and run a synthetic experiment or carry out a comprehensive data analysis?
4. Overall, are we doing a good job preparing students who are ready to engage in and succeed in statistical inquiry?

CT is a problem-solving approach that involves breaking down complex problems into smaller, manageable pieces and using algorithms to solve the small pieces. Wing (2011) describes CT as "the thought process involved in formulating problems and their solutions so that solutions are represented in a form that can be effectively carried out by an information-processing agent." When designing the course to support students to develop CT skills, the instructor asked: What do students need to work with data that is not well formed and ready to analyze using a statistical analysis or machine learning method? That is, teaching CT in this course focuses on guiding students to collect and process data, and use computational methods to solve problems. The course design includes data processing using simulations and collecting community data that describe social and economic factors and quality of life in community. The CT topics that spread in the course framework include abstraction, decomposition, pattern recognition, data structures, formulating problems as an optimization problem, and problem-solving using simulations and algorithms. Through learning the CT topics, students will also develop the disposition to think in an Object-Oriented fashion, automate problem solutions using scripts and Python libraries, and implement efficient problem solutions with the power of computing.

In contrast, ST involves collecting and analyzing data, making inferences and predications, and using statistical and machine learning models to understand relationships between data variables. It provides a way of understanding a complex world by describing it in relatively simple terms that nonetheless capture essential aspects of its structure or function, and that also provides us some idea of how uncertain we are about that knowledge (Poldrack, 2018, p. 15). The course design distinguishes ST from CT and other thinking processes and covers ST topics focusing on using data and statistical methods to make informed decisions. The course topics include descriptive and inferential statistics, probability, regression analysis, experimental data analysis, data visualization, and machine learning. Students learn ST as an interactive process using a software tool Jupyter Notebook where they run Python code. The ST process students conduct is combined with their CT process using Python programming and writing presentations. Each step of the ST process is determined by the information gained from previous steps of the thinking processes. The concepts students learn with data and skills they develop can help them build a foundation for their future success in data-intensive professions. The course design intends to get a balance between how much students should get into why/how algorithms do and how students use Python library functions like black boxes to solve problems in CT and ST.

CURE is integrated in the course design intentionally to weave CT and ST within a meaningful real-world project development using community problem solving. The community data students use is downloaded from the web site developed by the researchers at the University of Wisconsin Population Health Institution (2023). Students are also encouraged to integrate community data from other data sources if they see necessary to approach their research questions. The CURE element in the course design gets students working not just community data, but also important community problems. The CURE element in the course design breaks into pieces in the course framework to facilitate students to iteratively develop data-driven questions and solve them. Students gain experiences in problem solving, critical thinking, data analysis, and effective communication, which are valuable skills that are transferable to a wide range of careers. By the end of the project development, students will be able to

- Define a community challenge as specific research questions
- Design a feasible research plan
- Collect, analyze, and transform data to information
- Discuss research results in a socioeconomic context
- Generate hypothesis statements based on explorative study on data
- Synthesize related evidence in a research project presentation

Additionally, the instructor intends to use CURE to encourage dispositions in students. The dispositions include appreciation of the power of CT and ST, programming, machine learning, and research rigor. The instructor intends to foster a joy of data among students with a foundation in CT and ST. The research project development requires students to build team works and collaboratively work on various activities. Students gradually develop a sense of ownership on their project development and come to realize the difference they can make by their creative work. Students will become more aware of undergraduate research and its potential benefits. This will help them to build confidence and raise awareness about the importance of data-driven research and practice. The learning aims to inspire students with a strong sense of professional disposition while working with data in a team.

3. TEACHING STRATEGIES

The course development emphasizes data-driven critical thinking and problem solving in CT and ST during the teaching/learning process. John Dewey (1916) rooted critical thinking in the students' engagement with a problem. According to the preference matrix method (Paxton, 2006), if an individual can "make sense" of and "get involved" in the course learning environment, the individual prefers the environment and then it is likely that the person will spend time within the environment. The community problem-solving utilized in the course design provides students opportunities that make sense and engage them involved in active learning, which leads students to learn CT and ST productively. To use community data to teach effectively, the instructor specifically utilizes four teaching strategies focusing on the two key dimensions including "make sense" and "involvement".

Strategy 1: Solve Real-World Problems

The CURE project in the course design requires students to solve real-world problems using community data. The problems provide students authentic, complex, and open-end problems to solve, which challenge and motivate students to apply CT and ST in new and innovative ways. While developing their research project, students need to undergo the empirical enquiry cycle including problem elicitation, data wrangling, data analysis, formulation of research findings, and presentation of findings and conclusions/recommendation. Students will assess relevant data to address questions of their interests and communicate their findings using clear and concise visualizations and arguments to help make their research discoveries accessible and understandable.

The community problem solving also provides a problem-based learning (PBL) to facilitate student learning on the various topics in CT and ST as well as the CURE project development. Instead of following step-by-step instructions, students take more active roles to drive the learning process by working with data to solve the various community problems and questions. They are not only learners but also contributors to develop the course content on CT and ST. The objectives to learn CT, ST and CURE consist of the acquisition of the ability to apply the computation and statistics concepts and techniques creatively in a variety of contexts and situations. When working on the real-world problems, students give and receive immediate feedback on their work via discussions with collaborators and peers by doing online discussions and meetings.

The instructor sets clear goals and expectations to make sure students understand what they are expected to achieve and what they will be learning while approach the real-world problems. The weekly learning modules have clear, staged objectives to facilitate project development. The learning module on the CURE project has relevant and accessible resources including a data set students can initialize data exploration, Python tutorial scripts to do preliminary data analysis, team work guidelines as well as instructions to access literature works on community data process. Additionally, the instructor uses various communication venues including feedbacks, online discussions, and synchronous meetings to celebrate successes and encourage students to learn from their failures during the iterative problem solving process.

Strategy 2: Scaffold Community Problem Solving

The course design utilizes a process of gradually build students' understanding, confidence, and skills in a supportive way to approach the community problem solving in the CURE project development. To scaffold the project development, the instructor first introduces the community data, and encourages students to explore the community data and think about the problems and questions they are interested at tackling. To approach the collaborative learning in the CURE project, students need to document regular progress reports and share them with others when doing weekly online discussions. The reflection reports are expected to present how the problem is formulated, what data is identified to be collected in the problem context, how data variables are brought into structure that makes analysis possible, how the structured data is analyzed, how a solution is designed and implemented, how the research results are evaluated, and how to communicate the research discoveries.

The course content is developed using a progression model composed of three steps: *use*, *modify* and *create*. The instructor intends to use the model as a pattern of engagement to support student learning and maintain a level of challenge while avoiding too much learning anxiety. The online lectures, which allow students to participate synchronously and are also recorded for students to review later, combine presentation and demo code writing. The weekly lectures highlight what questions to ask along how problems are solved. When students approach the assessment assignments in the various learning modules, they need to compare the assignment questions with the questions in the lectures, modify the given code, trace execution steps, and empirically explore and evaluate problem solutions.

Note that the course design intentionally introduces the means and tools in CT and ST so that students can be equipped with tools when they approach problems designed in the course assignments and the CURE project. In general, the set of assignment problems are relatively well structured and designed to promote learning in purposeful and engaging activities. The CURE project is to support students to synthesize their learning on CT and ST and transfer the skills to solve open-end problems.

Strategy 3: Conduct Inclusive Learning

Learning and teaching happen in classrooms. The course design supports to conduct inclusive learning by creating an environment that is respectful and accessible to all students. The data used in the CURE project includes public-health data that describe social, economic, environmental conditions and health outcomes in communities, which ensures every student, regardless of their background, can find the data meaningful to their life and is exposed to a wide range of perspectives and experiences. Especially, the data reflect perspectives and experiences of underrepresented populations, which can help students promote equity and equal access to opportunities for learning and growth.

The course design provides support and accommodations for all students. To accommodate needs from diverse students, the course development utilizes multiple communication channels. First, the online course site organizes course materials including the course syllabus as well as learning materials for the designed course topics. Second, during each week, the weekly synchronous meetings are offered for the class to meet and discuss concepts, techniques, and project development. The meetings focus on active learning to allow students to make input on the learning and assessment activities. Third, the online discussion forums are used to encourage and support students to reach out to the instructor and their peers outside the online weekly meeting time. Lastly, students are expected to work closely with the instructor as well as their group partners and classmates using emails, GitHub, Zoom, Microsoft Team, and other social platforms chosen by study/project groups.

Inclusive learning is also reflected in the CURE project that students work together to resolve problems. Team project management is viewed as an important skill to teach and engage students. As indicated by a literature survey on Software Engineering (SE), "learn by doing" is the most widely used approach to providing practical experiences in SE education (Marques, Quispe, & Ochoa, 2014). While developing the CURE project, students work collaboratively to come up with their research project proposal and the final project report. They are encouraged to play one or more roles with different responsibilities, but they work towards the common goals to accomplish problem solving. At the beginning of the project development, the instructor suggests each team to establish processes for resolving conflicts and making decisions. The instructor also encourages reflections and discussions among team members about challenges and successes of their teamwork, and how to promote inclusivity in project development.

Strategy 4: Write to Learn

Writing to learn is an important strategy used to enhance learning in CT and ST. Bean and Weimer (2011, p.24) argue that writing provides one of the best ways to help learn active, dialogic thinking skills. Hazzan (2008) also suggests conducting reflections and states that reflection "increases one's awareness of the objects with which one thinks and may therefore systematically and consciously lead one to think …."

The course design supports students to engage in informal writing including weekly journaling, summaries, explanations, writing the problem-solving process, reflections and self-assessment, to communicate with peers and instructors. For example, during each week, students will reflect what they have learned, how they learned, and how to improve their data analysis and visualizations as well as the learning process. The set of scaffolding online-discussion questions unfold what and how students are expected to develop their projects. Through the weekly input from students, both students and the instructor can identify areas where students need further clarifications. Therefore, the instructor can monitor student learning closely and coach the learning process based on the persistent student input via the informal and reflective writing. During the research project development, students will also document their project development progress and collaborate learning in their project log. For the weekly reflective writing, online instructions on how to write a reflective post and how to comment peers' posts are provided. For a project log entry students need to journal and document the research process, a guideline is also provided to facilitate students to approach the writing task.

Students also engage with writing through the writing and programming assignments. For each assignment, students would approach a writing task where they are guided to approach a set of problems/questions that can be approached by analyzing and visualizing data. Students would generate visualization plots and present them with explanations and evaluations. Writing in the assignments intends to support students to "think out loud" as they code, make mistakes, and fix them, which is viewed as the best way to instill good habits to write code on problem solving and data analysis (Nolan and Temple Lang, 2015). Each assignment rubric contains a criterion regarding writing presentation, which is used to support students to write to learn and get feedback to improve their writing.

Lastly but not the least, students have opportunities to practice rigorous writing extensively through composing their project proposal and project report. The project development includes specific writing instructions on how to write the project proposal and develop the project report. The instructor intends to use the instructions to guide students through the writing process of a quantitative research, supporting students to develop awareness of disciplinary genre conventions, such as organization, design, style, mechanics, and citation format.

4. DISCUSSIONS

The course development with CURE was started in Spring 2022. The instructor has been actively collecting data to verity the effectiveness of course framework and teaching strategies. The instructor believes that teaching both computing and statistical analysis skills together ensures students formally trained in their education. Using community problem solving to teach CT and ST based on an integrated course framework with CURE can provide students with an engaging, hands-on, meaningful learning experience, which can impact their understanding of data and their ability to analyze data. The interdisciplinary, active learning can support students to apply CT and ST to real-world problems, conduct data analysis, and develop solutions to complex community challenges. The community problem-solving experience can provide a strong foundation for students who can later continually learn the evolving CT and ST skills. Additionally, students can become more involved in their community and take an active role in addressing important local, regional, and global problems.

The course design advocates teaching both CT and ST explicitly via a single course development. The course framework covers and weaves five of the six proposed divisions of data science including "data exploration and preparation; data representation and transformation; computing with data; data modeling; data visualization and presentation" proposed by David Donoho (2018) in "50 Years of Data Science." Even though the sixed division, science about data science, is not covered in the course design, the activities students need to do in the five divisions demands a great deal of efforts that are usually greater than a regular course students take in Applied Computing.

Specifically, the integrated learning presents several challenges. First, the course framework needs to balance depth and breadth of CT and ST topics. It is important to provide students with a rather comprehensive understanding of key concepts and tools in CT and ST. The course design, however, proves that it is challenging to employ sufficient in-depth details on CT and ST topics in a single undergraduate course. Additionally, CT and ST evolve rapidly. The course design demands continual efforts to keep up with the latest technology developments.

CT and ST are interdisciplinary skills that involve a combination of mathematics, computer science, and other disciplines. Both include complex and abstract concepts, which make them difficult for students to grasp, especially if students lack sufficient prior experience or exposure to the skills. Teaching CT and ST together is even more challenging when offering the course to the group of students who have different learning styles and backgrounds, which is quite typical in our Applied Computing program. Based on the instructor's experiences of offering the course in the previous two terms, the course was perceived as a difficult course for students who are less confident in their computational and statistical skills in general.

The integration of CURE makes the course even more challenging with the rest of CT and ST topics, especially when students need to synthesize their learning in the community problem solving. In general, our students have variant levels of CT and ST skills, which impact their collaboration to solve problems in CURE and impose more challenges to manage CURE project development effectively. To approach community problem solving, automating computational procedures is a foundational computing action. When students have limited programming and data analysis skills, the real-world problem solving becomes hard barriers for them to approach learning CT and ST. When students struggle with real-world problem solving, they may become frustrated, which can result their lack of understanding on the community data and reduce their motivation to learn.

Lastly, the integration of CT, ST, and CURE requires a significant investment of time and resources, including the time and resources needed to prepare students to access high quality community data, scaffold instruction for effective and productive community problem solving, and sustain the CURE project to ensure it is relevant and meaningful. In addition, assessing student learning in CT, ST, and CURE using tools such as Python, Jupyter Notebook, and GitHub proves challenging, and needs a delicate combination of developing hands-on projects, resolving questions in written assignments, as well as surveys and interviews on the use of special software tools and libraries. Overall, teaching CT and ST effectively needs a framework that is efficient and principled in how we formulate, decompose, analyze, and answer data-driven questions.

5. CONCLUSIONS AND FUTURE WORKS

There are several practical outcomes resulted by this study. First, the paper presents a course framework that utilizes community problem solving to weave CT and ST concepts and techniques as well as the CURE project development into a coherent course design. Second, four teaching strategies are laid out to demonstrate how to teach CT and ST and support students to have a meaningful CURE. Third, the paper presents the challenges and constraints to teach both CT and ST explicitly using one course development. The future work of the study is to analyze student learning and attitude data and investigate how the course framework and teaching strategies support students to accomplish the learning outcomes. Additionally, to strengthen the CURE driven by community problem solving in the course implementation, we are looking for opportunities to collaborate with local organizations, business, and government agencies, i.e., getting communities involved into the community problem solving process. The community connection can help student to gain more understanding on the data and problems they are solving and see both CT and ST are essential skills for everyone to succeed in today's world.

6. REFERENCES

Bean, J. C, & Weimer M. (2011). *Engaging ideas: The professor's guide to integrating writing, critical thinking, and active learning in the classroom,* 2nd Edition. Jossey-Bass.

Ben-Zvi, D., & Garfield, J. (2004). Statistical literacy, Reasoning, and Thinking: Goals, Definitions, and Challenges. *The Challenges of Developing Statistical Literacy, Reasoning and Thinking*, 3-16, Kluwer Academic Publishers.

Committee for Convocation on Integrating Discovery-Based Research into the Undergraduate Curriculum, Division on Earth and Life Studies, Division of Behavioral and Social Sciences and Education, and National Academies of Sciences, Engineering, and Medicine (2015). *Integrating Discovery-Based Research into the Undergraduate Curriculum: Report of a Convocation*. Washington, D.C.: National Academies Press.

Danyluk, A., & Leidig, P. (2021). Computing Competencies for Undergraduate Data Science Curricula. *ACM Data Science Task Force*. Downloaded from https://dstf.acm.org/

Dewey, J. (1916). *Democracy and Education*. New York: Macmillan.

Donoho, D. (2018). 50 Years of Data Science. QUBES Educational Resources.

Guzdial, M. (2022). Toward Justice in Computer Science through Community, Criticality, and Citizenship. *Communications of the ACM*, 65(5), 42-44.

Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., & Wenderoth, M.P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proc. Natl. Acad. Sci. 111*, 8410-8415.

Hazzan, O. (2008). Reflections on Teaching Abstraction and Other Soft Ideas. *SIGCSE Bull, 40*(2), 40-43.

Horton, N.J., & Hardin, J.S. (2021). Integrating Computing in the Statistics and Data Science Curriculum: Creative Structures, Novel Skills and Habits, and Ways to Teach Computational Thinking. *Journal of Statistics and Data Science Education,* 29(S1), S1-S3.

Kuh, G. (2008). *High-Impact Educational Practices: What They Are, Who Has Access to Them, and Why They Matter*. Washington, DC: Assn of Amer Colleges.

Marques, M.R., Quispe, A., & Ochoa, S.F. (2014). A systematic mapping study on practical approaches to teaching software engineering. In *2014 IEEE Frontiers in Education Conference,* 1-8.

Mogk, D.W., & Goodwin, C. (2012). Learning in the field: Synthesis of research on thinking and learning in geosciences. *Geological society of America special papers, 486(0)*, 131-163.

Nolan, D.A, & Temple Lang, D. (2010). Computing in the Statistics Curriculum. *The American Statistician*, 64, 97-107.

Nolan, D.A, & Temple Lang, D. (2015). *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*. Boca Raton, FL:CRC Press.

Paxton, J. (2006). The preference matrix as a course design tool. In *Proceedings of the 6th baltic sea conference on computing education research: Koli calling 2006*, 124–127. New York, NY, USA: ACM.

Poldrack, R. A. (2018). *Statistical Thinking for the 21st Century*. Stanford University.

Population Health Institution (2023), *County Health Rankings Model*, from https://www.countyhealthrankings.org/explore-health-rankings/county-health-rankings-model

Wei, C.A., & Woodin, T. (2011). Undergraduate Research Experiences in Biology: Alternatives to the Apprenticeship Model. *CBE-Life Sci. Educ. 10*, 123-131.

Wing, J. (2011). Computational thinking: What and Why. Retrieved December 10, 2019, from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

A Study on Learners' Different Learning Disposition About Cooperative Learning Across Different Languages for Instruction and Learning in the General Education Course on Speculative Reason

Tzu-Keng FU[1*], Jung-Yin FANG[2]
[1]College of General Studies, Yuan Ze University, Taiwan (R.O.C.)/Taoyuan City
[2]Graduate Institute of Learning and Instruction, National Central University, Taiwan (R.O.C.)/Taoyuan City
futzukeng@saturn.yzu.edu.tw, jungyin.f@gmail.com

ABSTRACT

This study discusses general education courses about speculative reason at YY university in Taiwan. We use the Kano's two-dimensional quality model to classify the quality attribute of nine perspectives of these courses about speculative reason, and moreover use the Importance Performance Analysis to study instructors' performance, course materials, and learners' cooperative learning in these courses. We aim to show that domestic students' performance regarding cooperative learning are different across different languages for instruction and learning.

KEYWORDS

Cooperative learning, Kano's two-dimensional quality model, importance-performance analysis, general education course on speculative reason

1. INTRODUCTION

The feasibility of using instructional materials for teaching philosophy for children (P4C) in general education courses about courses about logical thinking and speculative reason (SR courses, thereafter) at the university level has been studied, where a new-style SR course arrangement was implemented at the YY university (Fu, 2021). SR courses lies at the intersections of STEM course and humanities. We use as an example two general education classes in the SR course *Language and Thought* in this study: an English-taught SR course 1091L&TE and a Chinese-taught SR course 1082L&TC. This SE course pertains to the life science section of the general education curriculum at the YY university which aims to conduct the speculative reason on the scientific study of cognitive aspect of human language. The same new-style SR course arrangement was also implemented in these courses, lectures and whole-class discussions and interactions in the first half and group final projects, small group discussions and presentations in the second half. We also used instructional materials for teaching P4C to stimulate whole-class interactions in the first half of the course and we guided the groups to work on their final group projects in the second half *in the classroom*.

We wanted to know whether these instructional materials were also beneficial for students in terms of opening their minds and encouraging active interaction with their classmates in the classroom. We also wanted to know what their perceptions were toward collaborative learning in general education SR courses. A Kano's Two-Dimensional Quality Analysis (Kano analysis) and an Importance-Performance Analysis (IPA) were conducted to understand how the quality attributes of cooperative learning were perceived by these learners after they finish the courses. (Fu, 2021)

Collaborative learning can play an important role in SR courses. However, learners have not been taught *how* to learn cooperatively in SR courses. For example, in courses about philosophy, learners not only need to be able to debate and argue about philosophical theories, but they also need to be able to defend and justify their own philosophical positions. Hence, learners need to interact with others to *learn* how to debate, defend, and justify various positions. This kind of interaction is a part of collaborative learning. Nevertheless, based on our past observations of teaching SR courses, learners rarely know how to interact with others, let alone learn together with them. Therefore, the first step in implementing collaborative learning in SR courses is to guide learners to interact with others. In our past teaching, we misunderstood that cooperative learning could be achieved only if some sort of group tasks had been arranged in a course.[1] Therefore, in this study, we use a new-style SR course arrangement to help learners understand how to engage in cooperative learning in class step by step.

In our new-style SR courses, we assigned the *final group project as a learning task*. This task was divided into several smaller ones and each one was assigned as a course unit in the second half of the semester. We paid special attention to the following three points before implementing this new arrangement: first, to strengthen the group discussion activities in class every week; second, to strengthen the whole-class discussion activities in class every week; and third, to lead and guide the final group projects *in class*. Instructional materials for teaching P4C were the medium for implementation. Most of the course materials for stimulating students to learn using cooperative learning methods were the instructional materials for teaching P4C course in this new-style SR course arrangement.

We realized that learners need instructors' guidance before inviting them to participate in collaborative learning. Moreover, we found that before engaging in cooperative learning, groupmates as a whole need to be aware of their common goal: a final group project in our case.

---

[1] We followed other instructors and required learners to finish their term group reports in a semester and we evaluated their performance.

Cooperative learning activities, such as get-to-know-you activities with groupmates, can be organized to encourage group members to respond and interact with each other. Therefore, to enable learners to have sufficient time to engage in group work cooperatively and work on their final group project, the arrangement of discussions *in the classroom* may be important, i.e., these discussions are an independent unit of a course.

We arranged the course as follows:

A. Assigning *Discussion Plan* of instructional materials for teaching P4C each week after giving a theoretical and informative introduction to each unit.

B. Assigning an *Essay Writing Session* for the midterm exam. Students use this session to deepen their understanding of relevant materials, which also serves as a springboard for them to consider issues for their final group project.

C. Pushing them to consider issues for their final group projects in classroom.

2. DATA ANALYSIS TOOL

2.1. Analysis of the quality attributes of SR courses – the application of Kano's model

In the past, producers focused on the one-dimensional quality, that is, consumers will only be satisfied when a certain quality factor is sufficient, otherwise they are not. However, when judging the pros and cons of quality attributes, there is a shift from the producer-oriented to the consumer-oriented perspective, as this notion of one-dimensional quality is not enough to grasp the true attributes of quality properly.

Kano, Seraku, N., Takahashi, and Tsuji provided the so-called Kano's model in (1984). This model divides quality attributes into the following five categories: (Figure 1)

(1) Attractive Quality: If this quality attribute element is sufficient, the consumer will be satisfied. If it is not sufficient, the consumer may accept it but will not be satisfied.

(2) One-dimensional Quality: If this quality attribute element is sufficient, consumers will be satisfied; if it is not sufficient, consumers will not be satisfied.

(3) Must-be Quality: If this quality attribute element is sufficient, consumers will take it for granted, so satisfaction will not increase because of it being sufficient; if it is not sufficient, consumers will not be satisfied.

(4) Indifferent Quality: Regardless of whether the quality attribute element is sufficient or not, it has no effect on consumer satisfaction.

(5) Reverse Quality: This quality attribute element is sufficient to make consumers dissatisfied; if it is not sufficient, customers will be satisfied.

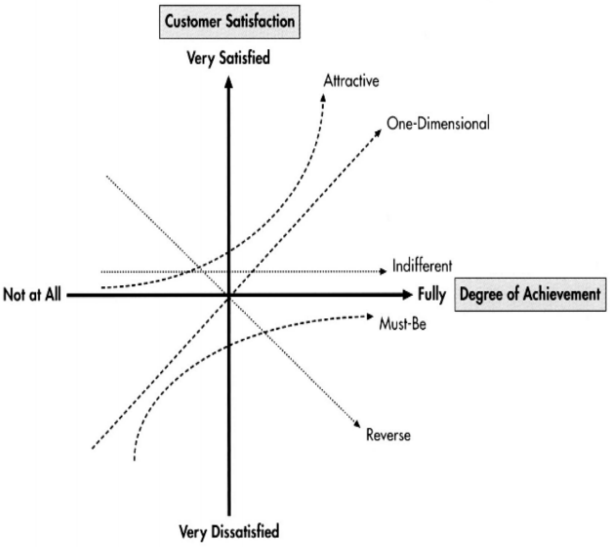


Figure 1. Kano's two-dimensional model

2.2. Quality analysis of each unit of the course – IPA

The IPA model was proposed by Martilla and James (1977). It was initially used to examine the service performance of the automotive industry and analyze product or service attributes. In the late 1970s, IPA began to be used in service quality research in various fields, but also in improving existing planning and decision-making. At present, the concept of service is also applied to the higher education system as the basis for teaching improvement and instructors' professional growth. In the IPA method, the degree of importance is represented on the *x*-axis and the degree of satisfaction is represented on the *y*-axis. Taking the average of the scores obtained from the degrees of importance and satisfaction as the cut-off point to obtain the two quadrant IPA two-dimensional matrix, we then mark the values obtained by the importance and satisfaction of each attribute in the four quadrants.

According to Matzler, Bailom, Hinterhuber, Renzl and Richler (2004), the four quadrants of the IPA pattern matrix are expressed as follows (Figure 2):

The quadrant (I) indicates that the degrees of importance and performance (satisfaction) are high, and the attributes falling within this quadrant should be kept (keep up the good work). The quadrant (II) indicates that the degree of importance is low, but the degree of performance is high, and the attributes falling within this quadrant are oversupplied (possible overkill). The quadrant (III) indicates that both importance and performance are low, and the attributes within this quadrant have a low priority. The quadrant (IV) indicates that the degree of importance is high but the degree of performance is low, and the attributes falling within this quadrant are the focus of supplier improvement (concentrate here).

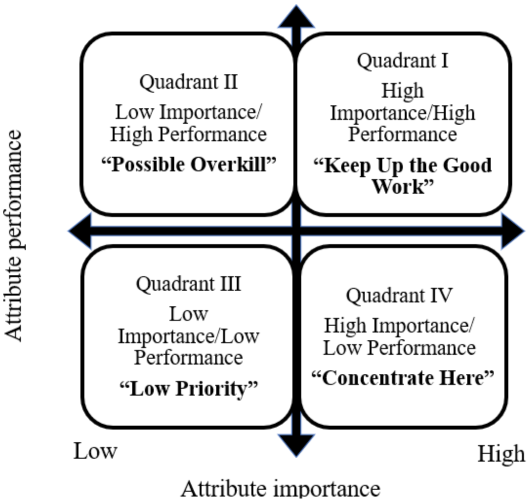


Figure 2. IPA two-dimensional matrix

The Kano's model uses questionnaire survey methods to understand the consumers' perception of whether a certain quality attribute is sufficient or not. After finding the perception of sufficient and inadequate situation of a certain quality attribute, the two-dimensional quality characteristics of each quality attribute can be categorized using the evaluation binary table of quality attribute elements (Table 1). The special feature of the questionnaire used in this model is that for each quality attribute, a set of positive and negative pairs of question items (as shown in Table 2) must be designed. The selected options of elements are cross-checked and then classified into their quality attributes.

Table 1. Evaluation Binary Table of Quality Attribute Elements.

Insufficient Sufficient	Like	Must-be	Neutral	Live with	Dislike
Like	Q	A	A	A	O
Must-be	R	I	I	I	M
Neutral	R	I	I	I	M
Live with	R	I	I	I	M
Dislike	R	R	R	R	Q

Resource: "How to make product deployment projects more successful by integrating Kano's model of customer satisfaction into quality function deployment," by Matzler, & Hinterhuber, 1998, Technovation, 18(1), 32.

Table 2. An example of a set of positive and negative pairs of question

	How do you feel if the following is available?	How do you feel if the following is not available?
The appropriate schedule	1 2 3 4 5	1 2 3 4 5
The appropriate difficulty	1 2 3 4 5	1 2 3 4 5

1: Dislike, 2: Live-with, 3: Neutral, 4: Must-be, 5: Like.

3. THIS STUDY AND DATA ANALYSIS

3.1. *Subjects of This Study*

The subjects of this study are students in two SR courses in Spring semester 2020 and Fall semester 2020, namely an English-taught SR course 1091L&TE and a Chinese-taught SR course 1082L&TC. The course material is *Philosophical Inquiry: Instructional Manual to Accompany Harry Stottlemeier's Discovery* (Lipman, Sharp, Oscanyan, 1984).

3.2. *Research Instrument*

This study uses the Kano-IPA Questionnaire, which is used on SR courses in (Fu, 2021). The questionnaire comprises three parts, namely the Kano's model, IPA, and personal background variables. The first part uses Kano's model as the main questionnaire method, which comprises the available course elements and unavailable course elements. This questionnaire uses a self-edited scale to examine nine aspects of SR courses, namely schedule (item 01), difficulty (item 02), the adequacy of course materials (item 03), the contents of course materials interest me (item 04), instructors' explanation and guidance (item 05), instructors' responses to questions (item 06), peers' mutual assistance (item 07), the whole-class interaction (item 08), and the arrangement of presentation (item 09).

This questionnaire used the Likert five-point scoring model in the first part where 1 means "dislike", 2 means "live with", 3 means "neutral", 4 means "must be", and 5 means "Like". The subjects select the appropriate response to each question based on their personal perceptions about the course. Using the same questions from the first part of the questionnaire, another Likert five-point scoring model was used in the second part where 1 means "not at all important", 2 means "slightly important", 3 means "normal", 4 means "fairly important", and 5 means "very important" (Table 1).

Reliability refers to the stability and reliability of the scale. In this study, Cronbach's α coefficient is used for reliability analysis, and the scores of the questions are tested for internal consistency.

For the Kano's model part, the internal consistency of this questionnaire is $\alpha = 0.923$ when the elements of teaching are *sufficient* and $\alpha = 0.945$ when the elements of teaching are *insufficient*, indicating that the questionnaire has good internal consistency.

For the IPA part, the internal consistency of this questionnaire on *importance* is $\alpha = 0.919$ and *performance* is $\alpha = 0.939$, indicating that the questionnaire has good internal consistency.

3.3. *Statistical Analysis*

The information from the samples in the SR courses was collected through a questionnaire. A total of 64 questionnaires were collected, there were no incomplete questionnaires, leaving a total 64 valid questionnaires.

The quality attributes are classified into five items by the evaluation binary table of quality attribute elements (Table 2): attractive quality attribute (A), must-be quality attribute

(M), one-dimensional quality attribute (O), indifferent quality attribute (I), and reverse quality attribute (R).

Since the subjects' views on the quality attribute of an element are not the same, this study adopts a counting method to count the subjects' answers on each quality attribute of an element to classify them.

In this study, **two SR courses with different languages for instruction and learning** are used as independent variables, and the quality attributes of three perspectives of cooperative learning used in these courses are dependent variables. The independent sample T-test is used to analyze the differences between two SR courses with different languages for instruction and learning. The chi-squared test is used to analyze the frequency counts in item 07, item 08, and item 09 of two courses.

4. DATA ANALYSIS

4.1. *Kano's Model Analysis*

In this section, the Kano's model analysis is provided with respect to nine aspects (Table 3). There is not much difference between three perspective of cooperative learning (item 07, item 08, item 09) when we classify the quality attributes of learners' perceptions on cooperative learning.

Table 3. Kano's Model Analysis for SR courses

	item.4	item.5	item.6	item.7	item.8	item.9
1091L&TE	O	O	O	O	I	O
Percentage	39.4%	48.5%	51.5%	33.3%	39.4%	42.4%
1082L&TC	I	O	O	O	I	I = O
Percentage	41.9%	45.2%5	61.3%	32.3%	41.9%	29.0%

4.2. *IPA*

The IPA-Distribution diagram for the 1082L&TC (Figure 3) shows that course materials interest me (item 04), instructors' explanation and guidance (item 05), and instructors' responses to questions (item 06) belong to the quadrant I (high importance/high performance; keep up the good work). Peers' mutual assistance (item 07), whole-class interactions (item 08), and the arrangement of presentation (item 09) belong to the quadrant III (low importance/low performance; low priority), where the value of item 09 lower is than the others.

IPA-Distribution diagram for the 1091L&TE (Figure 4) shows that course materials interest me (item 04) belongs to the quadrant III, instructors' explanation and guidance (item 05), and instructors' responses to questions (item 06) belong to the quadrant I. Peers' mutual assistance (item 07) falls on the *x*-axis, slightly lean to quadrant II, whole-class interactions (item 08) belongs to quadrant I, and the arrangement of presentation (item 09) belong to the quadrant II (possibly overkill).

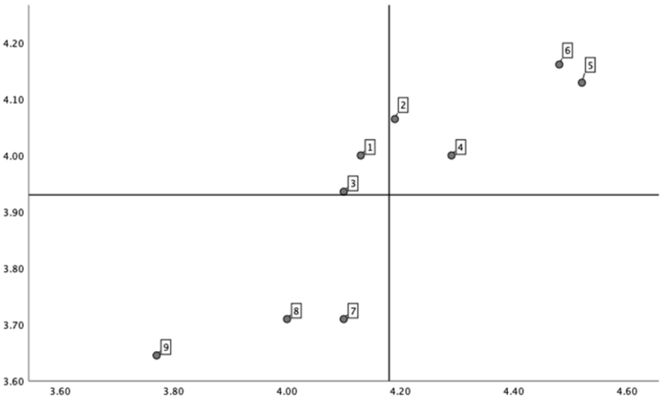


Figure 3. IPA-Distribution Diagram for the 1082L&TC

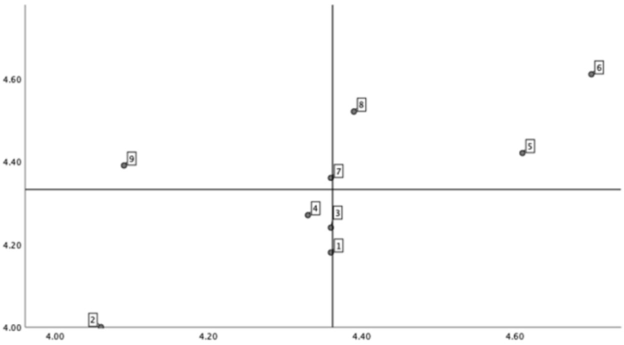


Figure 4. IPA-Distribution Diagram for the 1091L&TE

4.3. *The Independent Sample T-Test for IPA*

The results of independent sample T-test are as follows: There is no significant difference between IA7, IA8, and IA9 across 1091L&TE and 1082L&TC, where t=-1.140(p>.05), t=-1.690(p>.05), and t=-1.114(p>.05), respectively. It shows that there is no difference in the results for three perspectives of cooperative learning. There is significant difference between IB7, IB8, and IB9 across 1091L&TE and 1082L&TC, where t=-2.614(p<.05), t=-3.495(p<.01), t=-3.141(p<.01), respectively. It shows that there is a difference in the results for three perspectives of cooperative learning in these courses. Moreover, we check the mean of IB7, IB8, and IB9: for the mean of the IB7, the value is 3.71 in 1082L&TC but 4.36 in 1091L&TE; for the mean of the IB8, the value is 3.71 in 108L&TC but 4.52 in 109L&TE; for the mean of the IB9, the value is 3.65 in 108L&TC and but 4.39 in 109L&TE. It shows that the value of the IPA-performance of 1091L&TE is better than that of 1082L&TC.

Table 4 The Independent Sample T-Test

	Variable	N	Mean	T-value	P-value
IA7	1082L&TC	31	4.10	-1.140	.259
	1091L&TE	33	4.36		
IA8	1082L&TC	31	4.00	-1.690	.096
	1091L&TE	33	4.39		
IA9	1082L&TC	31	3.77	-1.114	.270
	1091L&TE	33	4.09		
IB7	1082L&TC	31	3.71	-2.614	.011
	1091L&TE	33	4.36		

IB8	1082L&TC	31	3.71	-3.495	.001
	1091L&TE	33	4.52		
IB9	1082L&TC	31	3.65	-3.141	.003
	1091L&TE	33	4.39		

IA7: Importance of Peers’ Mutual Assistance

IA8: Importance of Whole-Class Interaction

IA9: Importance of Arrangement of Presentation

IB7: Performance of Peers’ Mutual Assistance

IB8: Performance of Whole-Class Interaction

IB9: Performance of Arrangement of Presentation

4.4. The chi-squared test for the three perspectives of cooperative learning

As shown in previous sections, three perspectives of cooperative learning (item 07, item 08, and item 09) of the course 1091L&TE are classified as O or M by Kano’s model analysis, which can be taken as the necessary condition (W = O + M) of the course; three perspectives of cooperative learning of the course 1082L&TC are classified as I or W. In this study we use the chi-squared test for the three perspectives across 1091L&TE and 1082L&TC. Crosstabs (table 5) shows that there does not exist statistically significant difference between the frequency counts in item 07, item 08, and item 09, where $\chi^2 = .947$，p<.05.

Table 5. Crosstab between 1091L&TE and 1082L&TC

	1091L&TE		1082L&TC		χ^2	p
	frequency	%	frequency	%		
I	31	49.2	32	50.8	.947	.332
W	47	57.3	35	42.7		

5. FINDINGS

According to (Fu, 2021), the learners’ perceptions of cooperative learning will be changed after the implementation of the new-style SR course arrangement. However, at the end of these new-style SR courses, the learners in the English-taught SR course 1091L&TE have a better perception of the performance of the cooperative learning than the learners in the Chinese-taught SR course 1082L&TC, where all learners are native-Mandarin speaker.

6. DISCUSSION AND CONCLUSION

The scores of student evaluation of teaching in both courses were satisfactory: 1082L&TC scored 4.63 and 1091L&TE scored 4.65. The language for instruction and learning in 1082L&TC and 1091L&TE were Mandarin and English, respectively hereby we can tease out their differences across languages in this study as follows:

According to the IPA, we found that course materials which can arouse learners’ interest (item 04) belongs to “keep up the good work” (high importance/high performance) even though the learners of the Chinese-taught SR course (1082L&TC) take it as quality attribute I in our Kano’s model analysis.

In the English-taught SR course (1091L&TE), item 04 belongs to “low priority” (low importance/low performance) even though the learners take it as quality attribute O in our Kano’s model analysis.

Collaborative learning is the main learning method in this new-style SR course arrangement in academic year 2020-2021. The learners of 1082L&TC regard three perspective of cooperative learning (item 07, item 08, and item 09) as quality attribute I consistently. All these items belong to the quadrant III which means that learners taken them as low importance and low performance perspectives in this course.

However, the learner of 1091L&TE regard item 07 as quality attribute O, item 08 as quality attribute I, and item 09 as quality attribute O = I, and item 08 belong to the quadrant I, item 09 belongs to the quadrant II (high importance/low performance; concentrate here), and item 07 seems to be on the *x*-axis, slightly lean to the quadrant II, hence can be read as “keep up the good work” or “concentrate here”.

Table 6. A Comparation between 1091L&TE & 1082L&TC

		No.4	No.7	No.8	No.9
1091L&TE	Quality Attribute	O	O	I	O
	Quadrant	Q1	*x*-axis	Q1	Q2
1082L&TC	Quality Attribute	I	O	I	O=I
	Quadrant	Q1	Q3	Q3	Q3

The result of independent sample T-test shows that there is no difference across 1091L&TE and 1082L&TC for three perspectives of cooperative learning with respect to the IPA-importance (*x*-axis) but there is an obvious difference across 1091L&TE and 1082L&TC for three perspectives of cooperative learning with respect to the IPA-performance (*y*-axis) even though two groups of students are native-Mandarin speaker.

In other words, the learners in 1091L&TE have a different learning disposition about cooperative learning from the learners in 1082L&TC with respect to the IPA-performance of the course. The same results of instructors’ performance were found in Kano’s model analysis and IPA distribute diagrams: quadrant *I* and quality attribute **O** while the same new-style SR course arrangement and courses materials were implemented.

7. REFERENCES

Fu, T.-K. (2021). Applying New-Style Course Arrangement for Courses about Speculative Reason to Influence University Students’ Perceptions on Cooperative Learning: Using Curriculum Materials about Philosophy for Children to Promote the Whole-Class Interactions and Peer’s Mutual Assistance.

Journal of General Education: Concept & Practice, 9(2), 1-38. (In Mandarin)

Lipman, M., Sharp, A. M., & Oscanyan, F. S. (1984). *Harry Stottlemeier's discovery: Instructional manual: Philosophical enquiry*. Rowman & Littlefield.

Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Attractive quality and must-be quality. *Journal of Japanese Society for Quality Control*, 14, 39-48.

Martilla, J. A., & James, J. C. (1977). Importance-performance analysis. *Journal of Marketing*, 41(1), 77-79.

Matzler, K., & Hinterhuber, H. H. (1998). How to make product deployment projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. *Technovation*, 18(1), 25-38.

Matzler, K., Bailom, F., Hinterhuber, H. H., Renzl, B., & Pichler. J. (2004). The asymmetric relationship between attribute-level performance and overall customer satisfaction: A reconsideration of the importance-performance analysis. *Industrial Marketing Management*, 33(4), 271-277.

Evaluation of an Instructional Design for Developing Computational Thinking Skills Using Four-bar Linkage Bionic Robots

Shaun-Wen CHEN*, Ju-Ling SHIH
Graduate Institute of Network Learning Technology, National Central University, Taiwan
*seanchen54017@gmail.com, juling@cl.ncu.edu.tw

ABSTRACT

In this study, we designed a learning activity to develop computational thinking skills and learn linkage mechanisms by assembling a bionic robot, and examined students' learning performance on the instructional design. In today's society, robots can be found everywhere serving human beings, which shows that robots play an important role in society and help people to solve repetitive tasks. Therefore, understanding some mechanical principles will definitely be helpful. In order to understand them effectively, it is necessary to develop computational thinking skills. In this study, we designed a Quadruped bionic robot assembly kit using a 3D printed skeleton combined with Lego and DC motors. The purpose of this study is not only to attempt to build a bionic robot, but also to construct an instructional design for first-time learners of STEM and robotics that allows them to learn basic mechanical structures and develop CT skills at a minimal cost. The results showed significant growth in student learning outcomes, and most of the students developed CT skills while assembling their own bionic robots.

KEYWORDS

Bionic Robot, STEM, Linkage Mechanism, Instructional Design, Computational Thinking

1. INTRODUCTION

STEM is an important factor that can influence technological and economic development (Xie, Fang, & Sauman, 2015), STEM fields are an important topic for everyone to learn in modern society. The ability to effectively learn topics in STEM fields requires the skills of computational thinking (CT). According to the current trend of technological development, robotics is one of the major aspects of development (Li et al., 2011). In this study, the linkage mechanism in mechanical structures is considered to be important for current junior high school students because linkage mechanisms are often encountered in life, and learning about linkage mechanisms enables learners to better understand spatial concepts. Therefore, this study combines the linkage mechanism with the bionic robot to teach the learners about linkage and to develop CT skills through the process of assembling the bionic robot.

In this program, students will venture into the STEM field to learn about bionic robots, linkage mechanisms and also develop computational thinking skills. Students will also assemble their first bionic robot. In this activity, students will be given some components to assemble their designed robot and will be guided to find out how to make their bionic robot move faster and more stable on various terrains. In this paper, we use mainly qualitative analysis, supplemented by quantitative analysis, to investigate the following research questions.

1. How did the learners perform in CT skills during the process of building the bionic robots?
2. How well did the learners understand and absorb the knowledge of linkage mechanism?
3. How the learners feel and feedback?

2. LITERATUR REVIEW

2.1. Computational Thinking

Nowadays, computational thinking affects almost every discipline of study, including the sciences and humanities (Bundy, 2007). Computational thinking is a skill that is widely used in all areas of today's society. Computational thinking does not mean thinking like a computer, but rather having humans and computers work together to solve problems (Wing, 2006). Computational thinking is an important problem-solving skill and a thinking skill that is applicable to many disciplines (Bundy, 2007). In order to learn computational thinking skills, education is important to improve and strengthen intellectual skills so that CT can be applied in various disciplines (Wing, 2011). CT was categorized by Selby and Woollard (2013) into five major themes: Abstraction, Decomposition, Algorithm, Evaluation, and Generalization.

1. **Abstraction** refers to the development of rules that can solve similar problems to simplify information and finally create relationships between problems, (Council, 2010) and display only the information needed (Peel & Friedrichsen, 2017).

2. **Decomposition** is the process of determining the relationship between substantive elements and elements after categorizing the potential elements of the problem. Different strategies are used for decomposition, such as type, nature, and variables (Rich, Egan, & Ellsworth, 2019).

3. **Algorithm** is the formulation of rules that can solve similar problems step by step, which eventually results in a series of steps to solve a problem in order to solve similar problems later on (Peel & Friedrichsen, 2017).

4. **Evaluation** is the process of checking whether the developed algorithms and solutions are feasible. Various attributes need to be evaluated, including accuracy, efficient speed, effective use of resources, and ease of use.

5. **Generalization** is the process of repurposing and reapplying pieces to similar or unique problems, building models, rules, principles, or observational patterns to test predictions (Selby & Woollard, 2013).

CT is a fundamental skill that enables one to navigate today's complex technology (Peel & Friedrichsen, 2017). In this study, a one-day camp was conducted through the above five themes. During this one-day camp, students learn to use computational thinking skills through the process of assembling bionic robots and develop the ability to use CT skills effectively in future problems in various disciplines.

2.2. STEM-Based Instructional Design

STEM stands for four core areas: science, technology, engineering, and mathematics (Xie, Fang, & Sauman, 2015). In recent years, STEM education has become one of the core foundations of educational significance. It is a necessary course of study in order to enable students to deal with real-life and work-related problems more effectively in the future (Julià & Antolí, 2019). Instructional design for STEM education has become an important part of the curriculum. The design of a curriculum that engages students and effectively strengthens their STEM competencies has become an important issue for teachers. Khalil and Elkhider (2016) identified five pedagogical principles for teaching and learning.

1. Learners are committed to engaging in real-life problem solving.
2. Activating existing knowledge as a basis for new knowledge.
3. Learners are presented with new knowledge.
4. Learners apply new knowledge.
5. The learner integrates the new knowledge into the learner's world.

There are many approaches to instructional design, and this study is based on a project-based learning approach to design an activity that allows students to learn about linkage and develop CT skills through the process of assembling a bionic robot. Project-based learning is an instructional design that effectively helps learners learn from the process of creating a project (Guo et al., 2020). The project-based learning approach focuses on the following points, (1) knowledge is constructed, (2) prior knowledge must be taught first, (3) the whole is slowly constructed from the parts, and (4) efforts are required to engage in purposeful activities to build useful knowledge structures (Gómez-del Río & Rodríguez, 2022). The activity in this study was designed based on the combination of the above-mentioned STEM-based instructional design and project-based learning, and the activity involved students in knowledge and skills such as STEM, computational thinking skills, and linkage mechanisms.

2.3. Bionic Robot with Linkage Mechanism

Legged robots are more adaptable to changes in terrain and have been one of the main focuses of research by some companies in recent years. More and more people are developing various types of legged robots for various situations. For example, the Massachusetts Institute of Technology (MIT) developed a bionic cheetah that can walk on rough terrain (Singh & Kotecha, 2020). The design of bionic legs with a linkage mechanism has become a major focus for legged robots, and the best references are humans and other mammals that have already developed a unique way of walking (Li et al., 2011).

There has been little research on the use of bionic robots in instructional design. Therefore, this study developed a bionic robot package to inspire students to learn relevant knowledge and acquire CT skills at the same time. Students are guided through the process of assembling, improving, and enhancing their self-designed bionic robots.

3. Activity DESIGN

3.1. Learning Objectives

Students develop CT skills as they learn about building bionic robots. Learning objectives include:

1. Understand what is a bionic robot;
2. How to assemble a M-shaped bionic robot;
3. How to assemble a Cross-shaped bionic robot;
4. Know what a Dead Point is and how to eliminate it;
5. Understand crank-rocker mechanism and double rocker mechanism;

As the learners assembled their bionic robots, the students filled out their assembly process on a record sheet and the record sheet had scaffoldings to help the students with the assembly. The CT skills used by the learner can be analyzed based on the following aspects of the activity:

1. Understand the classification of the four-bar linkage mechanism and be able to indicate what kind of classification the linkage mechanism is. (Abstraction)
2. Understand each part of a four-bar linkage and be able to state the function of each part. To select the parts needed for the assembly of a bionic robot. (Decomposition)
3. Find a ratio that allows the bionic robot to walk smoothly and over obstacles when assembling the bionic robot. (Algorithm)
4. Evaluate the performance of the bionic robot after assembling it and suggest areas for improvement. (Evaluation)
5. After a few assemblies, the assembly process can be formalized to help make the next assembly more efficient. (Generalization)

At the end of the activity, the teacher collects the record sheets, which also serves as an assessment of the student's CT performance. The results of the pre-test and post-test provide an indication of how well the students learned from the activity.

3.2. Activities

After testing the students' prior knowledge of linkage mechanism and bionic robot, the teacher introduced the learners to the types of robots and the basic knowledge of linkage mechanism. Then, students were led to assemble the basic M-shaped bionic robot and cross-shaped bionic robot. Then, students were given 150 minutes to investigate, assemble, and improve their own bionic robots. Finally, students put their bionic robots on an obstacle course to compete for speed. The topics, content, and objectives of the course are shown in *Table 1*.

Table 1. Topics, Objectives, and Contents of the course

Topics	Content	Objectives
Course	**Robots**: Legged robots, wheeled robots, robotic arms, drones, bionic beasts, bionic robots.	Know what robots are and be able to identify the types of robots.
	Basic Linkage Mechaniksm: Basic terms, functions, examples.	Learn the basics of the linkage mechanism and understand its application.
Bionic Robot Assembly	**M-shaped Bionic Robot Assembly**: Double-rocker mechanism, examples, assembly method.	Learn the assembly method of M-shaped and its classification in the linkage mechanism.
	Cross-shaped Bionic Robot Assembly: Crank-rocker mechanism, example, assembly method.	Learn the assembly method of cross-shaped and its classification in the linkage mechanism.
Hands-on Practice & Competition	Modify the original M-shaped or cross-shaped or even invent their own bionic robot with linkage and try to pass the obstacle course in the shortest time.	Each learner can assemble their own linkage bionic robot within time.

The three topics are described in details as below.

3.2.1. Course: Robots

In this session, the teacher introduces the definition of robot and introduces various types of robots to the students, and let the students watch videos of these robots in action. The types of robots include: legged robots, wheeled robots, robotic arms, drones, bionic beasts, and bionic robots (shown in *Figure 1*).

Figure 1. Course teaching about robots

3.2.2. Course: Basic Linkage Mechanism

This section introduces the basic terms, definitions and applications of the basic linkage mechanism. Examples of uses for linkage mechanisms such as lifts, step-on trash cans, engines, water pumpers, etc. (shown in *Figure 2*).

Figure 2. Course teaching about linkage

3.2.3. Bionic Robot Assembly: M-shaped

In this study, a learning material was developed using SolidWorks to draw out the structures and a 3D printer to print it. Excluding failed prints, a set of materials for a bionic robot took about 7 hours to print. Nevertheless, the long start-up preparation can produce high cost-value teaching project materials.

The printed skeleton is combined with LEGOs and DC motor to form a bionic robot assembly kit (shown in *Figure 3*). And this kit can assemble the M-shaped and cross-shaped two bionic robots (Chen, Shih, and Chen, 2022).

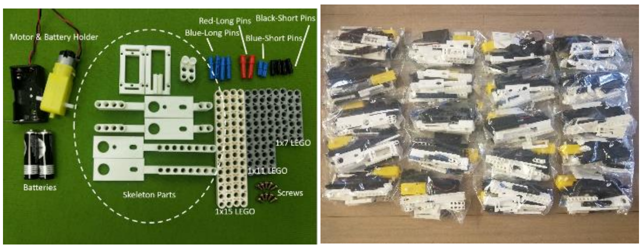


Figure 3. Four-bar linkage bionic robot kit

Before starting the assembly, the teacher walks the students through what an M-shaped bionic robot (shown in *Figure 4*) is and its classification in the linkage mechanism as a double-rocker mechanism. Next, to demonstrate basic assembly and layering concepts, the teacher guides the learners step-by-step through the assembly of a basic M-shaped bionic robot so that the students can fully understand how to assemble a bionic robot and understand the possibilities of improving it.

Figure 4. M-shaped Bionic Robot

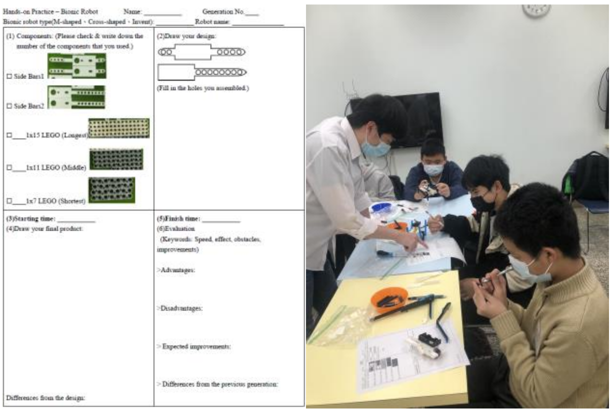
3.2.4. Bionic Robot Assembly: Cross-shaped

After learning how to assemble the M-shaped bionic robot, the teacher led the learners to understand what a cross-shaped bionic robot is (shown in *Figure 5*) and its classification in the linkage as a crank-rocker mechanism. Like earlier, the basic assembly and layering concepts are demonstrated to the learners.

Figure 5. Cross-shaped Bionic Robotic

3.2.5. Hands-on Practice & Competition

In this session, learners can choose to assemble and improve the two bionic robots they have just learned, or develop their own linkage bionic robot. During the process, learners will be asked to record their progress on a record sheet (shown in *Figure 6*). The record sheet requires the learner to fill in information about the type of bionic robot they plan to build, the parts they choose, the drawings of the design, the drawings of the finished product, the strengths and weaknesses, the improvements that can be made, and the differences from the previous generations. Learners can create the next generation of bionic robots based on their own record sheet and continue to refine it. During the hands-on practice session, the teacher will ask each learner if they need any help and give direction when the learner encounters a bottleneck.


Hands-on Practice – Bionic Robot Name: ________ Generation No.___
Bionic robot type(M-shaped、Cross-shaped、Invent): ________ Robot name: ________

(1) Components: (Please check & write down the number of the components that you used.)
☐ Side Bars1
☐ Side Bars2
☐___1x15 LEGO (Longest)
☐___1x11 LEGO (Middle)
☐___1x7 LEGO (Shortest)

(2)Draw your design:
(Fill in the holes you assembled.)

(3)Starting time: ________
(4)Draw your final product:
Differences from the design:

(5)Finish time: ________
(6)Evaluation
(Keywords: Speed, effect, obstacles, improvements)
>Advantages:
>Disadvantages:
>Expected improvements:
>Differences from the previous generation:

Figure 6. Record Sheet

3.3. *Reflection & Post-test*

Students presented their final bionic robots to the class and demonstrated their design concepts, as well as an analysis of their strengths and weaknesses. There was also a post-test to review the students' understanding of the linkage mechanism and the bionic robot.

This is the entire process of the activity, which lasted 4 hours in total. There is one class in the morning and one in the afternoon, and both classes have exactly the same content. Through the course, learners understand what a linkage mechanism is and learn how to develop a simple bionic robot on their own, and most importantly, develop CT skills in the process.

4. RESEARCH DESIGN

4.1. *Research framework*

The study was conducted as a one-day summer camp for 32 junior high school students aged 12 to 14, 23 males and 9 females. The event was held at National Central University in Taiwan. Due to the large number of students, the students were divided equally into two groups of 16 students in the morning session and 17 students in the afternoon session, both with the same content. The event started with a simple quiz with 5 questions about the curriculum to determine the prior knowledge level of the students. Then, depending on the level of the students, the entire course was conducted for 4 hours. After the students had mastered the basics, they were led by the teacher to learn how to assemble the bionic robots, which took 60 minutes. After experiencing the bionic robot assembly process and learning from the robot models, students spent approximately 150 minutes building their own bionic robots. At the same time, students will be able to place their creations on an obstacle course for a timed competition. Finally, at the end of the competition, the teacher will guide the students to review the key points learned from the activity and reflect on their work. Students' feedback was obtained by distributing a survey and analyzing their learning through post-testing (shown in *Figure 7*).

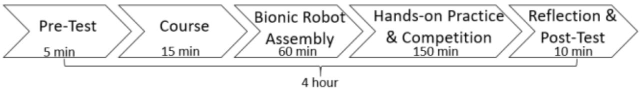


Figure 7. Research Process

4.2. *Research Tools*

4.2.1. *CT Performance*

This study analyzes the record sheets written by the learners in the hands-on sessions. The record sheet was designed based on the five CT themes categorized by Selby and Woollard (2013). The learners' performance on CT was analyzed based on their completion of the record sheet (*Table 2*).

Table 2. Record sheet and the CT skills it represents

CT Skills	Record Sheet Content	Examples
Abstraction	Able to draw the expected type of design.	The learner wrote down he/she is going to assemble the M-shaped, and the design drawing shows the obvious M-shaped structure.
Decomposition	Can select the required components.	Learners select the components they need according to their own design.
Algorithm	Assemble the same bionic robot according to the learner's own design.	The learner's finished product is the same as the design.
Evaluation	Evaluate finished products and propose possible areas of improvement.	Learners evaluate their finished product in the evaluation section.
Generalization	Complete the entire record sheet.	Learners complete each column in the record sheet.

4.2.2. *Knowledge Acquisition of Linkage Mechanism*

The learners' learning effectiveness was analyzed through the difference between the pre-test and post-test. Both the pre-test and post-test had 5 single-choice questions, each with 20 points, for a total of 100 points. The questions were the same, but the order of the options was different.

5. RESULTS & DISCUSSION

5.1. *CT Performances*

Learners' record sheet completion was analyzed using the five CT themes. Every learner completed at least one record sheet, 15 completed two, and 1 completed three.

Each learner submitted at least one record sheet. 26 of the 32 learners were able to abstract the bionic robot in their minds and represent it through drawing during the first assembly. Fifteen learners were better at using CT skills to assemble the bionic robot and make a second robot, and 12 of them used abstraction skills in the second build. The majority of the learners performed well in abstraction skills (Abstraction) (*Table **3***).

Table 3. Learners' performance on Abstraction in CT

Learners	Abstraction		
	1st	2nd	3rd
32	26 (32)	12 (15)	1 (1)

27 out of 32 learners were able to break down the bionic robot as a whole in their minds and pre-select each part one by one in the first assembly. 12 out of 15 learners in the second assembly were able to perform the decomposition skill. This shows that most of the learners performed well in the decomposition skills (Decomposition) (*Table **4***).

Table 4. Learners' performance on Decomposition in CT

Learners	Decomposition		
	1st	2nd	3rd
32	27 (32)	12 (15)	1 (1)

25 of the 32 learners were able to assemble their bionic robots based on their designs and to investigate and improve their finished products to find a set of algorithms for the bionic robots they designed to cross the obstacles smoothly. 11 of the 15 learners in the second assembly achieved the skill of decomposition. It can be seen that most of the learners performed quite well in algorithm skills. (Algorithm) (*Table 5*).

Table 5. Learners' performance on Algorithm in CT

Learners	Algorithm		
	1st	2nd	3rd
32	25 (32)	11 (15)	1 (1)

26 out of 32 learners were able to evaluate their results and give reasonable solutions after making the finished product. In the second assembly 7 out of 15 learners did the skill of evaluation. Most of the learners performed well in the evaluation skill. (Evaluation) (*Table **6***).

Table 6. Learners' performance on Evaluation in CT

Learners	Evaluation		
	1st	2nd	3rd
32	26 (32)	7 (15)	0 (1)

Only 18 of the 32 learners were able to use their computational thinking flexibly to generalize the process of their design of the bionic robot so that their next design could be better and more effective. Only 4 of the 15 learners in the second assembly were able do the generalization skill. It can be seen that most of the learners were not familiar with the generalization skills (Generalization) (*Table **7***).

Table 7. Learners' performance on Generalization in CT

Learners	Generalization		
	1st	2nd	3rd
32	18 (32)	4 (15)	0 (1)

Although most junior high school students can apply the abstraction, decomposition, algorithms, and evaluation of computational thinking skillfully, it is somewhat difficult for them to generalize the whole system process. Although the data indicated that the students did not perform well in the generalization skills, the learners were able to answer the questions during the activity and said that they just wanted to focus on assembling the bionic robot so they did not record the process on the log sheet.

5.2. *Knowledge Acquisition of Linkage Mechanism*

32 pre-tests and 32 post-tests were collected from a total of 32 people. Since one person's pre-test was not submitted, the student's post-test score was not counted.

According to the pre-test and post-test returned by the learners, the average score of the pre-test was 39.38, while the average score of the post-test was 75.00, showing a significant improvement (p=.000) (*Table 8*) . This shows that the course is effective for the learners and that the learners are learning most of what the teachers want to teach.

Table 8. Paired Samples T-Test for Pre and Post-test

T-test	N	Mean	SD	t	p
Pre-test	32	39.38	17.949	-8.320^{***}	.000
Post-test	32	75.00	22.143		

****p<.001*

5.3. *Overall Satisfactions*

A total of 32 valid satisfaction questionnaires were collected from a total of 32 students. The satisfaction questionnaire was divided into five topics for learners to self-examine, including course content richness and difficulty, learning attitude, learning motivation, and overall satisfaction (as shown in *Table 9*).

Table 9. Overall Satisfactions

Themes	Means
Course Difficulty	4.17
Course Richness	4.22
Learning Attitude	4.20
Learning Motivation	4.25
Overall Satisfaction	4.27

Students were mostly satisfied with the difficulty and richness of the course content, with an overall average of 4.14 and 4.18. Most students found the introduction of robots to be easy to understand, while students were more confused by the introduction of M-shaped bionic robots, but still found the M-shaped and cross-shaped bionic robots to be informative, while in the course on basic linkage, students found it less informative. Students' learning attitude (*M* = 4.17) and motivation (*M* = 4.21) were mostly high in the overall activity. Most of the students thought that the linkage mechanism was important and interesting and worthwhile for them to learn. Because the overall activity increased students' interest in the linkage mechanism and the bionic robot (*M* = 4.21), students were willing to ask questions when they encountered difficulties and gained a great sense of accomplishment when they completed their products. Students generally felt that the course topics and the teaching materials were clear and understandable. Although the students' expectations of the overall activity may have been affected by not being able to bring the completed bionic robot home, most students were satisfied with the overall activity (*M* = 4.23).

Based on the above data, the overall activity enhanced students' understanding of linkage mechanisms and robotics and developed the learners' skills in using CT. The learners were engaged while assembling the bionic robots, and although they were less willing to write a record sheet while assembling, they were still motivated to try to use their CT skills to help them investigate the bionic robots.

6. Acknowledgements

This study is supported in part by the National Science and Technology Council (previously known as Ministry of Science and Technology) of Taiwan, under MOST 108-2511-H-008 -016 -MY4.

7. REFERENCES

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, 1*(2), 67-69. http://www.inf.ed.ac.uk/publications/online/1245.pdf

Chen, S. W., Shih, J.-L., & Chen, Y. M. (2022). Four-bar Linkage Quadruped Biorobotic Instructions: Gamified Design and Development. *The 30th International Conference on Computers in Education*.

Chen, S. W., & Shih, J.-L.. (2022, June). A STEM-based Learning Activity Instructional Design of Quadruped Bionic Robots. *In CTE-STEM 2022 conference*. https://doi.org/10.34641/ctestem.2022.461

Council, N. R. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking*. The National Academies Press. doi:10.17226/12840

Gómez-del Río, T., & Rodríguez, J. (2022). Design and assessment of a project-based learning in a laboratory for integrating knowledge and improving engineering design skills. *Education for Chemical Engineers*. https://doi.org/10.1016/j.ece.2022.04.002

Guo, P., Saab, N., Post, L. S., & Admiraal, W. (2020). A review of project-based learning in higher education: Student outcomes and measures. *International Journal of Educational Research, 102*, 101586. https://doi.org/10.1016/j.ijer.2020.101586

Julià, C., & Antolí, J. Ò. (2019). Impact of implementing a long-term STEM-based active learning course on students’ motivation. *International Journal of Technology and Design Education 29*, 303–327. doi:10.1007/s10798-018-9441-8

Khalil, M. K., & Elkhider, I. A. (2016). Applying learning theories and instructional design models for effective instruction. *Advances in physiology education, 40*(2), 147-156. doi:10.1152/advan.00138.2015

Li, Y., Li, B., Ruan, J., & Rong, X. (2011, September). *Research of mammal bionic quadruped robots: A review.* In 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM), 166-171. doi: 10.1109/RAMECH.2011.6070476.

Peel, A. & Friedrichsen, P. (2017). Algorithms, Abstractions, and Iterations: Teaching Computational Thinking Using Protein Synthesis Translation. *The American Biology Teacher, 80*(1), 21–28. doi:10.1525/abt.2018.80.1.21

Rich, P. J., Egan, G., & Ellsworth, J. (2019, July). *A framework for decomposition in computational thinking.* In Proceedings of the 2019 ACM conference on innovation and technology in computer science education, 416-421. doi:10.1145/3304221.3319793

Sawyer, R. K. (2014). *The future of learning: Grounding educational innovation in the learning sciences.* The Cambridge handbook of the learning sciences, 726-746.

Selby, C. & Woollard, J. (2013). *Computational thinking: the developing definition*. Conference: Special Interest Group on Computer Science Education (SIGCSE), 2014. URL http://eprints.soton.ac.uk/id/eprint/356481

Singh, R., & Kotecha, R.. (2020, April). Quadruped Robot Gait Planning for Enhanced Locomotion and Stability. Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST). doi:10.2139/ssrn.3572191

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011). Research notebook: Computational thinking—What and why. *The link magazine, 6*, 20-23. https://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf

Xie, Y., Fang, M., & Shauman, K. (2015). STEM Education. *Annu Rev Sociol, 41*, 331-357. doi:10.1146/annurev-soc-071312-145659

An Exploratory Study on the Relationship between Computational Thinking and Motivation for Gaming among University Students in Japan

Masanori FUKUI[1*], Masakatsu KURODA[2], Yuji SASAKI[3], Tsukasa HIRASHIMA[4]
[1]Center for University Education, Tokushima University, Japan.
[2]Faculty of Literature, Kobe Women's University, Japan.
[3]Graduate School of Media and Governance, Keio University, Japan.
[4]Graduate School of Advanced Science and Engineering, Hiroshima University, Japan.
f-masanori@tokushima-u.ac.jp, m-kuroda@suma.kobe-wu.ac.jp, y.sasaki@keio.jp, tsukasa@lel.hiroshima-u.ac.jp

ABSTRACT

This study explored the relationship between computational thinking and motives for using games to obtain knowledge for the development of computational thinking through games. A survey with 312 university students revealed that computational thinking, creativity, algorithmic thinking, and critical thinking were related to numerous motives for game use. Using this data, we then conducted a canonical correlation analysis and examined game subjects. Findings demonstrated that achievement, friendship, cooperativity, and problem-solving, recognition, study and creativity, and critical thinking were related to each other.

KEYWORDS

education using a game, game education, computational thinking, empirical study

1. INTRODUCTION

1.1. *Purpose of This Research*

We explored the relationship between computational thinking and motives for using digital games, including video and television games, to acquire knowledge for fostering computational thinking using games in the future.

1.2. *Background of This Research*

As the advanced information society progresses, developing 21st-century skills is essential for success in the future. Such skills refer to the literacy skills required in the digital age of the 21st century and beyond. These are defined as skills comprising 10 items across four categories: creative and innovative thinking; critical thinking, problem-solving, and decision-making; learning and metacognition (knowledge about cognitive processes) as a way of learning; information literacy and literacy in information and communication technology as a way of working; communication and collaboration as tools for work (teamwork); citizenship in local and global communities; life and career planning; and personal and social responsibility (cultural awareness and responsivity) as tools for social life (Binkley *et al.*, 2011). These skills encompass the attitudes and competencies required to cope with rapidly changing times. Accordingly, it will be imperative in education to cultivate them in various ways.

Additionally, computational thinking is deemed a necessary 21st-century skill (Haseski *et al.*, 2018). Computational thinking is defined by Wing (2006) as a process of organizing and representing problems in a computer-solvable way. The Programme for International Student Assessment (PISA) stated that they would test computational thinking in mathematics education tests in the future (Organisation for Economic Co-operation and Development, n.d.). Therefore, the importance of computational thinking is expected to increase further.

To foster computational thinking, various methods have been practiced such as programming, unplugged computing, and so on. Activities to foster computational thinking using game development have been implemented (*e.g.*, Brennan, 2022). Scratch, for example, incorporates the "game" element of moving a cat, which is commonly used in the introductory programming education stage. Harvard University (n.d.) offers a practical method for developing computational thinking on their web page, "Computational Thinking with Scratch."

Because the use of game materials has long been studied for their effectiveness in increasing learner motivation and interest (*e.g.*, Malone, 1980), it is essential to continue research on the development of computational thinking using game materials. It is also imperative to incorporate this not only into programming education but also into unplugged computing and regular classes to enhance computational thinking.

1.3. *Identification of Problems*

The development of computational thinking through programming activities and game subjects has a specific level of usefulness. Therefore, programming should be enhanced using game materials, whereas a methodology for fostering computational thinking using game elements should be developed.

However, differences in readiness among the students, such as differences in awareness of programming, interest in games, and previous game-playing experience, may result in varying development of computational thinking. Undoubtedly, the appropriate subject matter may differ depending on differences in readiness. Because programming has already been implemented in several countries, we focused on differences in attitudes and approaches toward games.

One of the readiness factors was the learner's motivation for playing games. There are various motives; for example, Sherry *et al.* (2006) surveyed students in elementary school through college in the U.S. to understand game use and satisfaction. They found that students played games as a challenge, competition, distraction, excitement, fantasy, or social interaction. In a game user study, Iguchi (2013) surveyed game use and satisfaction among Japanese university students and found that the seven factors were fantasy, recognition, preference, achievement, friendship, study, and diversion. In this study, we used Iguchi's game use and satisfaction scale because the survey subjects were Japanese university students.

If the relationship between game playing and computational thinking is clarified, classes will foster computational thinking. Hence, understanding the relationship between gameplay and computational thinking can enhance future computational thinking development. However, such a study is currently non-existent.

Therefore, this study exploratively examines the relationship between the motive for using games and computational thinking and seeks ways to enhance the cultivation of computational thinking through game development and classroom practices that incorporate game elements.

2. METHODOLOGY

2.1. Survey Targets and Survey Procedure

A survey was conducted in November–December 2022 among second-year university students majoring in game development in Japan, taught by one of the authors. We surveyed 312 individuals (180 men and 132 women, an average age of 19.63, SD = 0.70), and 271 responded to the survey request (effective response rate was 86.86%). The survey duration was approximately 10 min.

As an ethical consideration in conducting the survey, we did not seek any personally identifiable information such as name, initials, e-mail address, or student ID number. Before the survey, the content was explained to the respondents, and they were instructed to only respond if they agreed with its content. Their responses were considered as their consent.

2.2. Measurement Scales

We selected five factors (creativity, algorithmic thinking, cooperativity, critical thinking, and problem-solving skills) and 29 items for the Japanese version of the Computational Thinking Scale (Bando & Motozawa, 2021), originally developed by Korkmaz *et al.* (2017). Additionally, seven factors (fantasy, recognition, preference, achievement, friendship, study, and diversion) and 29 items of the Game Uses and Gratifications Scale (Iguchi, 2013) were prepared to measure gaming motivation.

In the aforementioned cases, we measured computational thinking and motivation for gaming using a 5-point Likert scale from 5 ("Strongly agree") to 1 ("Strongly disagree"). Both scales have been used in surveys of university students and are considered valid for this study.

2.3. Analysis of Procedure

First, descriptive statistics for the "computational thinking" and "game uses and gratifications" scales were calculated. Next, the correlation coefficients between the factors of computational thinking and the motivation for gaming were calculated. Furthermore, to comprehensively examine the relationship between game-use motivation and computational thinking, we conducted a canonical correlation analysis using the seven variables of game-use motives as the first group and the five variables of computational thinking as the second group.

3. RESULTS

3.1. Descriptive Statistics

Tables 1 and 2 show the results of the descriptive statistics of the "computational thinking" scale and the "game uses and gratifications" scale. The results found that the mean scores for all items except for problem-solving were above the medium score of 3.00. The mean values of cooperativity and creativity for computational thinking and fantasy, achievement, and friendship as motives for gaming tended to be relatively higher than those of the other factors. Cronbach's alpha coefficients were calculated for each factor and ranged from .75 to .86, indicating that all items were safe to use. Based on the suggestion that averaging several questions be regarded as an interval scale (Carifio & Perla, 2008), this study considered the computational thinking and game use and satisfaction scales as interval scales, respectively evaluating them as follows:

Table 1. Descriptive Statistics of the Computational Thinking Scale

	Mean	***SD***
Creativity	3.54	0.59
Algorithmic thinking	3.06	0.83
Cooperativity	3.67	0.92
Critical thinking	3.17	0.77
Problem-solving	2.99	0.72

(*n* = 271)

Table 2. Descriptive Statistics of the Game Uses and Gratification Scale

	Mean	***SD***
Fantasy	4.23	0.72
Recognition	3.24	1.03
Preference	3.57	0.95
Achievement	4.12	0.86
Friendship	4.29	0.88
Study	3.70	1.06
Diversion	3.42	1.12

(*n* = 271)

3.2. Correlation between Computational Thinking and Motives for Gaming

To determine the correlation coefficient, Spearman's correlation coefficient was calculated. Table 3 shows an association between computational thinking and motives for gaming as: algorithmic thinking and study; cooperativity and achievement; friendship and critical thinking; friendship and study. In contrast, problem-solving was not related to motives for gaming. More precisely, creativity, algorithmic thinking, and critical thinking were related to different motives for game use, while friendship and study were related to many factors of computational thinking. Moreover, fantasy, recognition, and diversion were not related to computational thinking.

3.3. Canonical Correlation Analysis

In the previous section, we examined the relationship between computational thinking and motives for using games through correlation analysis. However, since game use motives are complex, such as wanting to win to show off to others or immersing oneself in fantasy alone and not being disturbed by anyone, various patterns can be assumed. Additionally, computational thinking is complex and cannot be explained by a single factor. Therefore, we analyzed the motives for game use and computational thinking by assuming that each was composed of a complex set of factors. The two canonical correlation coefficients calculated, λ_1 and λ_2, were significant at the 1% level (table 4). The canonical loadings at λ_1 and λ_2 are shown in Table 5.

Table 4. Canonical Correlation Coefficients

	λ	*df*	χ^2	*p*
λ_1	.367	35	86.954	.000
λ_2	.312	24	48.790	.002
λ_3	.244	15	21.766	.114
λ_4	.113	8	5.598	.692
λ_5	.091	3	2.184	.535

Table 5. Canonical Loadings of Each Factor

First group	λ_1	λ_2
Fantasy	-.218	-.146
Recognition	-.118	**-.572**
Preference	-.139	-.511
Achievement	**-.503**	-.288
Friendship	**-.854**	-.365
Study	-.003	**-.856**
Diversion	-.009	-.035
Second group	λ_1	λ_2
Creativity	-.231	**-.829**
Algorithmic thinking	.102	-.486
Cooperativity	**-.956**	-.104
Critical thinking	-.251	**-.732**
Problem-solving	**-.440**	.276

For the first canonical variable λ_1, in the first group, the coefficients were larger in the order of friendship and achievement with negative values. In the second group, the coefficients were larger in the order of cooperativity and problem-solving with negative values.

For the second canonical variable λ_2, in the first group, the coefficients were larger in the order of study and recognition with negative values. In the second group, the coefficients were larger in the order of creativity and critical thinking with negative values.

4. DISCUSSION

The results of the first canonical variable showed that users with a high awareness of playing games with friends and a high awareness of playing games because they are happy to accomplish complex tasks tended to have a high awareness of solving problems in cooperation with their peers and of being flexible when solving problems. This implies that the motives for game playing are reflected in actual problem-solving. For users whose motive for using games is to play cooperatively with friends, it is assumed that playing with one who aims to solve problems cooperatively will allow them to solve problems without considerable barriers.

The results of the second canonical variable showed that users who played games, because they wanted to learn and to be recognized by others, tended to be more able to make plans to solve complex problems. Considering these users, it is assumed that using materials that require planning to solve complex problems in actual problem-solving will allow them to solve problems without considerable barriers. In addition, it may be adequate to focus on motives for using games that enhance elements of computational thinking and use teaching materials that can foster these motives for using games.

Regarding the relationship between the motive for using games and game genres, Iguchi (2015) found that game genres associated with achievement were fighting games, adventure, RPG (Role-Playing Game), and action, whereas game genres associated with friendship were MMORPG (Massively Multiplayer Online Role-Playing Game), FPS (First Person Shooter), shooting, strategy simulation, fighting games, adventure, RPG, and action. Furthermore, the game genres associated with the study were MMORPG, FPS, shooting, strategy simulation, love simulation, fighting games, adventure, RPG, and action, whereas game genres associated with recognition were FPS, shooting, fighting games, adventure, and action. By incorporating these game elements, it is possible to develop a subject matter that enhances the motivation to use a particular

*Table 3***.** Correlation between "Computational Thinking" and "Game Uses and Gratifications" Scale

	Creativity	Algorithmic thinking	Cooperativity	Critical thinking	Problem solving	Fantasy	Recognition	Preference	Achievement	Friendship	Study	Diversion
Creativity	1.00											
Algorithmic thinking	0.34**	1.00										
Cooperativity	0.27**	0.01	1.00									
Critical thinking	0.49**	0.51**	0.22**	1.00								
Problem solving	0.18**	0.09	0.30**	0.14*	1.00							
Fantasy	0.11	0.10	0.08	0.03	-0.01	1.00						
Recognition	0.12	0.13*	0.06	0.11	-0.13*	0.27**	1.00					
Preference	0.16**	0.08	0.06	0.06	-0.10	0.41**	0.28**	1.00				
Achievement	0.15*	0.04	0.22**	0.13*	0.08	0.31**	0.33**	0.30**	1.00			
Friendship	0.16**	0.06	0.30**	0.18**	0.08	0.24**	0.37**	0.25**	0.32**	1.00		
Study	0.29**	0.18**	0.03	0.24**	0.00	0.35**	0.18**	0.40**	0.30**	0.25**	1.00	
Diversion	-0.03	0.13*	0.02	0.01	-0.06	0.18**	0.23**	0.10	0.15*	0.19**	0.04	1.00

*$p < .05$, **$p < .01$ (n = 271)

game and, consequently, computational thinking. However, there is an overlap in the game genres relevant to the motives for using each game.

4.1. Implications for classroom activities

When students develop games using Scratch or other similar tools in school, it is assumed that game genres such as simple RPGs, action, and adventure games are more effective for enhancing cooperativity and problem-solving skills. To foster creativity and critical thinking, creating shooting, action, and adventure games are assumed to be more effective than other genres. Furthermore, the use of adventure and action games may be effective for developing computational thinking skills, making them useful in the classroom. However, since learner preferences vary, providing multiple options and including the elements mentioned above is necessary to make them effective.

5. CONCLUSIONS

5.1. Summary of this research

We studied the relationship between computational thinking and motives for using games to enhance future game-based computational thinking. Based on a survey of university students, we proposed suggestions for practices oriented toward developing computational thinking using game materials.

The study results provide insights for developing computational thinking using game materials. Moreover, this type of research focusing on the relationship between the motives for gaming and computational thinking has never been conducted before and is considered novel and original.

Based on the findings, it is necessary to conduct research on the development of computational thinking in the context of gamification and practical models by conducting surveys, examining methods, and practically assessing their usefulness. Moreover, these findings may enhance digital game-based learning that fosters computational thinking.

5.2. Limitations of this research

The data collected in this study through a self-report survey are subjective, and further analysis using objective measures should be conducted. Additionally, given the wide range of game genres and the prevalence of multiple genres in many games, it is necessary to examine the relationship between these multiple game genres and computational thinking.

Future research should focus on the development of computational thinking to expand on these issues.

6. ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI (Grant Number 21K13644) and the Inamori Foundation.

7. REFERENCES

Bando, T., & Motozawa, A. (2021). The Relationship between Computational Thinking and Grit among University Students, *Journal of the Japan Society of Technology Education*, 63(1), 23–29. [in Japanese]

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2011). Defining Twenty-First Century Skills. 10.1007/978-94-007-2324-5_2.

Brennan, K. (2022). A case for why: School, society, self. In S.-C. Kong & H. Abelson (Eds.), *Computational thinking education in K–12: Artificial intelligence literacy and physical computing* (pp. 15–28). MIT Press. 10.7551/mitpress/13375.003.0004

Carifio, J., & Perla, R. (2008). Resolving the 50-Year Debate Around Using and Misusing Likert Scales, *Medical Education*, 42, 1150–1152.

Harvard University. (n.d.). *Computational Thinking with Scratch*. Retrieved January 9, 2023, from http://scratched.gse.harvard.edu/ct/index.html.

Haseski, H., Ilic, U., & Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends, *International Education Studies*, 11, 29–42. 10.5539/ies.v11n4p29.

Iguchi, T. (2013). Video Game Uses and Gratifications of University Students in Japan: Towards Establishing Game User Research, *Journal of Information and Communication Research*, 31(2), 67–76. 10.11430/jsicr.31.2_67 [in Japanese]

Iguchi, T. (2015). An Investigation into Video Game Use by University Students with a Particular Focus on Game Genre and Motivation, *Journal of Information and Communication Research*, 33(2), 41–51. 10.11430/jsicr.33.2_41 [in Japanese]

Korkmaz, Ö., Cakir, R., & Özden, M. Y. (2017). A Validity and Reliability Study of the Computational Thinking Scales (CTS). *Computers in Human Behavior*, 72, 558–569.

Malone, T. W. (1980). *What makes things fun to learning? A study of intrinsically motivating computer games*, technical report, Xerox Palo Alto Research Center, Palo Alto, CA.

Organisation for Economic Co-operation and Development (n.d.). *PISA 2022 Mathematics Framework*. Retrieved January 9, 2023, from https://pisa2022-maths.oecd.org/ca/index.html.

Sherry, J. L., Greenberg, B. S., Lucas, K., & Lachlan, K. A. (2006). Video Game Uses and Gratifications as Predictors of Use and Game Preference, *International Journal of Sports Marketing and Sponsorship*, 8, 213–224.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35. 10.1145/1118178.1118215

AI with supporting technologies entrusts diversity and inclusion in research issues for underrepresented students in educational applications

Iza GUSPIAN[1], Stephen J. H. YANG[1*]
[1]Department of Computer Science and Information Engineering, National Central University,
No 300 Zhongda Rd, Zhongli District, Taoyuan City 32001, Taiwan
izaguspian@gmail.com, Stephen.Yang.Ac@gmail.com

ABSTRACT

This study aims to see that diversity and inclusion can be considered in educational AI applications where the research issues of underrepresented students are there. It is known that the role of AI is very helpful in student learning, and the delivery of learning material by the teachers. Hence, we try to cover this gap which other researchers ignore through bibliometric analysis in education. The results of the analysis show that in 2019-2022 the trend tends to increase dramatically on these issues involving the latest technology. We believe that engaging AI on social, health/medical, and cultural/literacy issues will eliminate the discrimination that places equity on underrepresented students according to the findings. Meanwhile, machine learning, deep learning, chatbot, robotics, and NLP are even more rife when juxtaposed with the Covid 19 happens at the same time. The role of AI really contributes to prediction, assessment/evaluation, adaptive learning systems, and intelligent tutoring system in education which in turn makes it easier for underrepresented students to gain access, and teachers who carry out learning activities in it.

KEYWORDS

Artificial intelligence, diversity and inclusion, underrepresented students, education

1. INTRODUCTION

Diversity and inclusion in education that place a common view between educators, students and academics is something that is fought for by the existence of artificial intelligence with its supporting technologies. The same opportunity to gain full achievement in school with the potential of students to develop well (Hague, 2022). Presenting our analysis by looking at research issues, knowing how researchers mitigate bias by using AI techniques and supporting tools to inform instructional decisions, modelling, and predict learning outcomes for underrepresented students in educational applications, with surveys obtained from referenced literature confirm a comprehensive understanding of addressing various problems and providing solutions, (Dieterle et al., 2022; & Truby, 2019).

In identifying students, model predictions, tracking and designing and implementing lessons, exams, and individual feedback are employed by AI in education (Zhang & Aslan, 2022). The AI systems, algorithms utilized can improve student practice and make it more personal in learning, in particular for underrepresented students. An instance of an intelligent tutoring system (ITS) is technology that can help student learning be better and patterned. (Dieterle et al., 2022; & Decuypere et al., 2021) reveal that reforming the necessary conditions with digital tools and AI-enhanced learning platforms in every part, as well as continuous improvement of collaborative, and informal learning to increase the engagement of digital experiences.

This phenomenon compels us to do a lot to realize underrepresented students that they get equal rights through artificial intelligence, indeed, with the existence of researchers who have offered their studies have contributed in taking part in this case. Our study investigates the research issues of diversity and inclusion for underrepresented students in AI educational applications which help pedagogue and researchers find the suitable formula. However researchers have identified analyzes of diversity, equity, and inclusion for students from children at school to higher education in artificial intelligence context, sometimes neglecting to look at issues specially to underrepresented students in an education (e.g. Jora et al., 2022; Chiu, 2019; Perez et al., 2022; Roche et al., 2022; Bracarense et al., 2022; & Bagunaid et al., 2022). As is known, the role of AI has a major influence, especially in the context of the research on underrepresented students, which important to get success of AI education through diversity and inclusion (Xia et al., 2022; Delaine et al., 2016; & Ibe et al., 2018), while our research focuses on findings, analyzing bibliometrics research issues in education for underrepresented students who have the potential to generate fresh ideas and new strategies.

The following research questions are trying to be proposed:

1. What are the most research trends of diversity and inclusion in artificial intelligence for underrepresented students from 2008-2022?
2. How does AI with supporting technologies embed diversity and inclusion in research issues for underrepresented students?
3. What is the role of artificial intelligence in diversity and inclusion of underrepresented students in educational applications?

2. METHODOLOGY

We derived the Scopus, Web of Science (WoS), and IEEE Xplore databases for analysis of research papers on diversity and inclusion in research issues for underrepresented students in educational applications through AI and its supporting technologies. Bibliometric analysis is included in the methodology of this paper from 2008-2022, knowing the development movement from year by year in which AI systems/tools are present in providing the same opportunity to students who are underrepresented in the education field. Query string through (TITLE-ABS-KEY (diversity) OR TITLE-ABS-KEY (minority) OR TITLE-ABS-KEY (underrepresented)) AND (TITLE-ABS-KEY (inclusion) OR TITLE-ABS-KEY (inclusive)) AND (TITLE-ABS-KEY (artificial AND intelligence)). In searching 179 articles linked to diversity and inclusion in research issues for underrepresented students, while in filtering only 60 articles focus on education, hence we eliminate unrelated such as non-English, book series, book, and excluded education areas in analyzing the full text of 47 articles including education context.

3. RESULTS

The bibliometric analysis in Figure 1 started in 2008 (1 document) and shows progress from 2011 to 2016, although only one each study in 2011-2013, 2014 (2 documents), 2015 and 2016 each at 3 documents, zero count in 2009, 2010 and 2017, so that there are fluctuations between these year intervals. Growth occurred from 2018 to 2022, each study in 2018 (4 documents), 2019 (2 documents), 2020 (6 documents), 2021 (8 documents) and 15 studies in 2022.

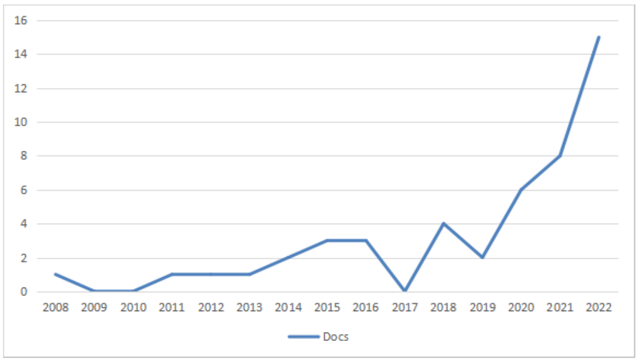


Figure 1. Analysis by year

Research trends in Figure 2 with keyword analysis on diversity and inclusion in artificial intelligence for underrepresented students in 2008-2016 are more towards serious games, digital games for mental therapy. Whereas in 2017-2018 the learning system began to emerge with the AI trend by using robotic teaching connected to social interaction. In 2019 -2022 there are machine learning, deep learning, chatbot, NLP, and robotics which continue to discuss diversity and inclusion in underrepresented students by involving neural network technology in interaction and showing compelling trends.

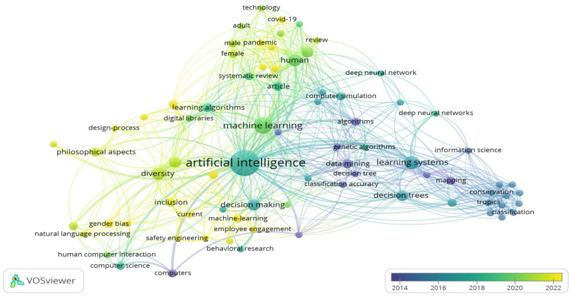


Figure 2. Research trend

Table 1 illustrates that machine learning was mostly used by researchers as many as 10 in social and health/medical issues, followed by deep learning, chatbot, robotic, and NLP. We have research issues many in the social, health/medical, cultural/literacy areas, while others are not specifically told in.

Table 1. AI technologies

Issues	AI technologies	Count
Social	Machine learning	7
	Deep learning	4
	Chatbot	3
	Robotic	2
Health/Medical	Machine learning	3
	Deep learning	2
	Chatbot	3
	Robotic	3
Cultural/Literacy	Chatbot	1
	Robotic	1
	NLP	4

The role of AI in underrepresented students greatly contributes, in Table 2 known that the predictions are most often found at 12 along with biased predictions in the algorithms that are present, while it provides the assessment/evaluation at 11, and lastly adaptive learning systems and intelligent tutoring systems in 5 each.

Table 2. Role of AI

Role of AI	Count
Prediction	12
Assessment/Evaluation	11
Adaptive Learning System	5
Intelligent Tutoring System	5

4. DISCUSSIONS

The following results identify it in our analysis based on the following research questions:

4.1. What are the most research trends of diversity and inclusion in artificial intelligence for underrepresented students from 2008-2022?

Research trends from 2008-2022 to address the diversity in artificial intelligence for underrepresented students around fit pattern for diversity and inclusive design. We analyzed and found that from 2008-2016 serious games involved artificial intelligence by discussing mental health conditions or medical (Bernardini et al., 2014). Meanwhile, 2017-2018 the learning of AI machine learning on social interaction, and robotic teaching for underrepresented students (Liu et al., 2017; & Zamin et al., 2018). Lastly, there will be a significant increase in 2019-2022 by involving a neural network technology which research on human behavior and the habits themselves, so this can accelerate behavior research with human computer interaction to eliminate bias in predicting whether in the context of gender, race, and ethnicity through NLP (Shi et al., 2020; & Ahmed et al., 2022). The trend of the distribution by subject area dominates from the computer science in connecting to artificial intelligence, how it predicts bias and improves the performance of underrepresented students.

4.2. How does AI with supporting technologies embed diversity and inclusion in research issues for underrepresented students?

Many researchers apply machine learning and deep learning and it is becoming a trend that neural networks for learning, correcting errors, and providing automatic feedback using machines or internet-based have great potential to contribute to education. Whereas, we realize that direct learning is still needed because the teacher's touch and face-to-face interactions add to the social impact even though the impact of Covid 19 is often found in health/medical issues. ML, DL, chatbots, robotics, and NLP are indeed available AI technologies to facilitate teaching and learning activities, in this case we found several research issues that can motivate underrepresented students in terms of their social interaction and eliminate the discrimination in cultural/literacy (Alahmadi et al., 2020; & Morrison, 2021) or as a health context in terms of gender mostly female, race, sex, and ethnicity. While during the Covid-19 pandemic in health crisis (Mateos et al., 2022), utilizing AI technologies was widely used as a therapy (Patrascoiu et al., 2022), it is undeniable that underrepresented students by the availability of technology can access via laptops or cellphones. In short, AI with assistive technology can play a key role in promoting diversity and inclusion in research issues for underrepresented students by enhancing accessibility, identifying biases, promoting diversity in data collection, and providing personalized support (Cheong et al., 2021; Henne et al., 2021; & Patrascoiu et al., 2022).

4.3. What is the role of artificial intelligence in diversity and inclusion of underrepresented students in educational applications?

It is important to design algorithms in line with technologies to understand populations in socially marginalized or other terms, presenting predictions and assessment/evaluation on the role of AI needed to encourage accurate and purposeful representation in AI systems, here we found gender bias, and race mostly (Ahmed et al., 2022; Lillywhite, B., & Wolbring, G; & Hamdi et al., 2022). Meanwhile, adaptive learning has been verified as challenging and important in education (Yang, et al., 2013), hence adaptive learning is needed for students to support them in executing their skills and knowledge, where the goal is quality learning and maximum quantity results in absorbing information related to learning, as well as the learner’s individual needs. Well-designed ITS specifically pedagogical is known to be able to complement and replace other learning as a supporting tools and an authoring tools that is considered for student with disabilities which are known to be effective, (Dermeval et al., 2017; & Ahuja et al., 2022).

5. CONCLUSIONS

Our results with a bibliometric analysis from 2008-2022 show that the contribution of AI in education to diversity and inclusion for underrepresented students by looking at trends, issues, and the role of AI has a positive impact on students using AI technologies as a reference for educators. Obviously, the role of AI in prediction and assessment/evaluation is still performing, but with the same enthusiasm, AI technologies are created that strengthen underrepresented students such as machine learning, deep learning, chatbots, robotics, and NLP, which are three research issues namely social, health/medical, cultural/literacy related to gender, ethnicity, disabilities, race, and language, which requires equity in the education system. Meanwhile, trends up to 2019 provide an accurate measure for predicting and evaluating human behavior through neural networks to help underrepresented students, teachers, and maximize time efficiency.

6. REFERENCES

Ahmed, Md. A., Chatterjee, M., Dadure, P., & Pakray, P. (2022). The role of biased data in computerized gender discrimination. *IEEE/ACM 3rd International Workshop on Gender Equality, Diversity and Inclusion in Software Engineering (GEICSE*), Pittsburgh, PA, USA, 2022, 6-11, doi: 10.1145/3524501.3527599.

Alahmadi, D., Babour, A., Saeedi, K., & Visvizi, A. (2020). Ensuring inclusion and diversity in research and research output: A case for a language-sensitive NLP crowdsourcing platform. *Applied Sciences*, 10(*18*), 6216. MDPI AG. https://doi.org/10.3390/app10186216

Ahuja, N. J., Dutt, S., Choudhary, S. lohmor, & Kumar, M. (2022). Intelligent tutoring system in education for disabled learners using human–computer interaction and augmented reality. *International Journal of Human–Computer Interaction, 1–13*. Informa UK Limited. https://doi.org/10.1080/10447318.2022.2124359.

Bagunaid, W., Chilamkurti, N., & Veeraraghavan, P. (2022). AISAR: Artificial intelligence-based student assessment and recommendation system for e-learning in big data. *Sustainability*, 14, 10551. https://doi.org/10.3390/su141710551.

Bernardini, S., Porayska-Pomsta, K., & Smith, T. J. (2014). ECHOES: An intelligent serious game for fostering social communication in children with autism. In *Information Sciences,* 264, 41-60. Elsevier BV. https://doi.org/10.1016/j.ins.2013.10.027.

Bracarense, N., Bawack, R. E., Wamba S. F., & Carillo, K. D. A. (2022). Artificial intelligence and sustainability: A bibliometric analysis and future research directions. *Pacific Asia Journal of the Association for Information Systems*. doi: 10.17705/1pais.14209.

Cheong, M., Leins, K., & Coghlan, S. (2021). Computer science communities: Who is speaking, and who is listening to the women? using an ethics of care to promote diverse voices. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.* https://doi.org/10.1145/3442188.3445874.

Chiu, T. K. F. (2022). Designing inclusive and diverse artificial intelligence (AI) curriculum for K-12 education. *Digital Communication and Learning*. https://doi.org/10.1007/978-981-16-8329-9_3.

Dieterle, E., Dede, C., & Walker, M. (2022). The cyclical ethical efects of using artifcial intelligence in education. *AI & Society*. https://doi.org/10.1007/s00146-022-01497-w.

Decuypere, M., Grimaldi E., & Landri P. (2021). Introduction: Critical studies of digital education platforms. Crit Stud Educ 62(*1*), 1-16. https://doi.org/10.1080/17508487.2020.1866050.

Delaine, D. A., Williams, D. N., Sigamoney, R., & Tull, R. G. (2016). Global diversity and inclusion in engineering education: Developing platforms toward global alignment. *International Journal of Engineering Pedagogy*, 6(*1*).

Dermeval, D., Paiva, R., Bittencourt, I. I., Vassileva, J., & Borges, D. (2017). Authoring tools for designing intelligent tutoring systems: A systematic review of the literature. *International Journal of Artificial Intelligence in Education,* 28(*3*), 336-384. Springer Science and Business Media LLC. https://doi.org/10.1007/s40593-017-0157-9

Hague, S. (2022). Diversity and inclusion in schools. *Pearson*. https://www.pearson.com/content/dam/one-dot-com/one-dot-com/uk/documents/educator/schools/issues/inclusion/diversity-and-inclusion-in-schools-report.pdf.

Hamdi, Y., Mulder, N., & Abdelhak, S. (2022). Women in systems science and gender parity: Why and how to democratize the "technology, innovation, and society nexus." *OMICS: A Journal of Integrative Biology,* 26(*6*), 329-338. Mary Ann Liebert Inc. https://doi.org/10.1089/omi.2022.0055.

Henne, K., Shelby, R., & Harb, J. (2021). The datafication of #MeToo: Whiteness, racial capitalism, and anti-violence technologies. *Big Data & Society,* 8*(2), p. 205395172110558*. SAGE Publications. https://doi.org/10.1177/20539517211055898.

Ibe, N. A., Howsmon, R., Penney, L., Granor, N., DeLyser, L. A., & Wang, K. (2018). Reflections of a diversity, equity, and inclusion working group based on data from a national CS education program. *Proceedings of the 49th ACM technical Symposium on computer science education*, 711–716.

Jora, R. B., Sodhi, K. K., Mittal, P., & Saxena, P. (2022). Role of artificial intelligence (AI) in meeting diversity, equality and inclusion (DEI) goals. *8th International Conference on Advanced Computing and Communication Systems (ICACCS).* IEEE. doi: 10.1109/ICACCS54159.2022.9785266.

Lillywhite, B., & Wolbring, G. (2022). Emergency and disaster management, preparedness, and planning (EDMPP) and the 'social': A scoping review. *Sustainability,* 14(*20)*, 13519. MDPI AG. https://doi.org/10.3390/su142013519.

Liu, R., Salisbury, J. P., Vahabzadeh, A., & Sahin, N. T. (2017). Feasibility of an autism-focused augmented reality smartglasses system for social communication and behavioral coaching. *Frontiers in Pediatrics*, 5. Frontiers Media SA. https://doi.org/10.3389/fped.2017.00145.

Mateos-Sanchez, M., Melo, A. C., Blanco, L. S., & García, A. M. F. (2022). Chatbot, as educational and inclusive tool for people with intellectual disabilities. *Sustainability*, 14(*3*), 1520. MDPI AG. https://doi.org/10.3390/su14031520.

Morrison, J. D. (2021). Beyond Lead Generation. In People-Centered Approaches Toward the Internationalization of Higher Education (pp. 193–217). IGI Global. https://doi.org/10.4018/978-1-7998-3796-1.ch010.

Perez, A. B., Perez, D. B., Molina, D.C., Cruz, D., Montenegro, R., Vazquez, L., & Xochicale, M. (2022). Piloting diversity and inclusion workshops in artificial intelligence and robotics for children. arXiv:2203.03204v2.

Patrascoiu, L. A., Folostina, R., Patzelt, D., Blaj, M. P., & Poptean, B. (2022). E-tools for personalizing learning during the pandemic: Case study of an innovative solution for remote teaching. *Frontiers in Psychology,* 13. Frontiers Media SA. https://doi.org/10.3389/fpsyg.2022.751316.

Roche, C., Wall, P. J., & Lewis, D. (2022). Ethics and diversity in artificial intelligence policies, strategies and initiatives. *AI & Society.* https://doi.org/10.1007/s43681-022-00218-9.

Shi, S., Wei, S., Shi, Z., Du, Y., Fan, W., Fan, J., Conyers, Y., & Xu, F. (2020). Algorithm bias detection and mitigation in lenovo face recognition engine. *Natural Language Processing and Chinese Computing, 442–453.* Springer International Publishing. https://doi.org/10.1007/978-3-030-60457-8_36.

Truby, J. (2020). Governing artificial intelligence to benefit the UN sustainable development goals. *Sustainable Development.* Wiley. doi: 10.1002/sd.2048.

Xia, Q., Chiu, T. K. F., Lee, M., Sanusi, I. T., Dai, Y., & Chai, C. S. (2022). A self-determination theory (SDT) design approach for inclusive and diverse artificial intelligence (AI) education. *Computers & Education 189.* https://doi.org/10.1016/j.compedu.2022.104582.

Yang, T. C., Hwang, G. J., & Yang, S. J. H. (2013). Development of an adaptive learning system with multiple perspectives based on students' learning styles and cognitive styles. *Educational Technology & Society*, 16(*4*), 185–200.

Zamin, N., I. Arshad, N., Rafiey, N., & S. Hashim, A. (2018). Robotic teaching aid for disabled Children: A sustainable solution for industrial revolution 4.0. *International Journal of Engineering & Technology,* 7*(2*), 200-203. Science Publishing Corporation. https://doi.org/10.14419/ijet.v7i2.28.12912.

Zhang, K., & Aslan, A. B. (2021). AI technologies for education: recent research & future directions. *Computer & Education: Artificial Intelligence*. 2. 100025. https://doi.org/10.1016/j.caeai.2021.100025.

Designing a Culturally Responsive, Equitable and Inclusive K-12 Curriculum on Computational Thinking and Problem-Solving skills

Elavarasi MANOGARAN[1], Shoaib DAR[2]
[1,2]Pi Jam Foundation, India
elavarasi.manogaran@thepijam.org, shoaib.dar@thepijam.org

ABSTRACT

Computational Thinking and problem-solving skills are the 21st century thinking skills that need to be democratized for all K-12 students. But teaching and learning of these concepts is generally associated with stereotypical views of ‘complexity’ and ‘who can learn them’ that might hinder confidence levels in students and increase entry barriers especially for the underserved and minority. This paper explores culturally responsive design strategies that help include all students irrespective of their gender, race, socio-economic class or ethnicity, in learning and practicing computational thinking and problem-solving skills. We propose ten design strategies that includes using semantic waves in a culturally responsive way, including students’ tribe in the curriculum, building on familiar local contexts, telling a tale, acknowledging the digital divide, dual approach to problem solving, looking at technology from a critical lens, practicing responsible digital citizenship, digital wellbeing and spreading waves of kindness and compassion. These strategies will enable students to be efficient computational thinkers and problem solvers while also closing the stereotypes and intersectional gender gap associated with computer science and STEM education.

KEYWORDS

computational thinking, problem solving, equity in STEM & computing education, culturally responsive curriculum design, semantic waves

1. INTRODUCTION

Historically marginalized and underserved groups continue to feel left out or under-confident about pursuing computing and STEM education. There has been significant research over the last decade to make computing education equitable, culturally relevant (Morales-Chicas, 2019) and responsive (Ron Eglash, 2013), especially in the United States, by including students’ culture, community connections and lived experiences. Computing education in India has reinforced the focus on computational thinking and 21st century skills to build a stronger workforce (Ministry of Human Resource Development, 2020), yet the inclusion of these thinking skills and problem-solving using technology is still in its nascent stages in the K-12 education space. The lack of access to technology tools and quality education resources were the major bottlenecks for attending to the needs of marginalized students in computing education. But in certain contexts, like Jammu & Kashmir (J&K), India, access to tools and resources is not the only constrain. These students are shy and sensitive and carry wounds from their past history and culture, yet were determined to solve their communities’ biggest challenges like access to school, unprecedented weather etc, through technology. We created a culturally responsive curriculum for J&K, through a video course called ‘Let’s Code’ with the special intention of broadening participation among young girls and under-served students in computing education.

2. CULTURALLY RESPONSIVE COMPUTING CURRICULUM

In this paper, our goal is to provide culturally responsive design strategies for curriculum creators and educators in creating equitable and inclusive computing education resources and encourage participation among under-represented students and minority girls. Our video-based curriculum comprises 24 lessons with an average video length of 6 minutes. There are three levels to the course design as shown in Figure 1, which includes computer science principles, problem solving & computational thinking skills and programming. The course also includes advance concepts such as AI, bigdata and machine learning besides introducing students to the foundations of computer science.

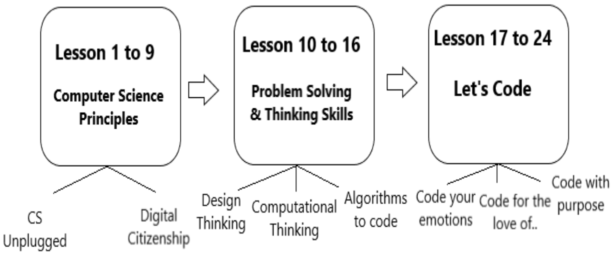


Figure 1. Course Design & Structure of Let’s Code

3. DESIGN STRATEGIES

3.1 Include Students’ Tribe in the Curriculum

When students see themselves and their environments in what they learn, they are most likely to find learning more meaningful and purposeful. We showed students role models from their own community who had excelled in a myriad of STEAM fields like literature, military and law, who had also pioneered computing and technology. Kalpana Chawla, the first Indian born women and robotic arm operator to go to space, Ayesha Aziz, the youngest female pilot, who got her flying license at 16 years, Mawya Sudan, the first female fighter pilot from Kashmir in the Indian air force, Reshma Saujani, a lawyer and founder of a non-profit who teaches girls how to code, are some of the many role models. We also introduced a fictional, humanoid android co-instructor to help with the course and named her A.I.S.H.A (Artificial Intelligence She/Her Humanoid Android) as shown in Figure 2.

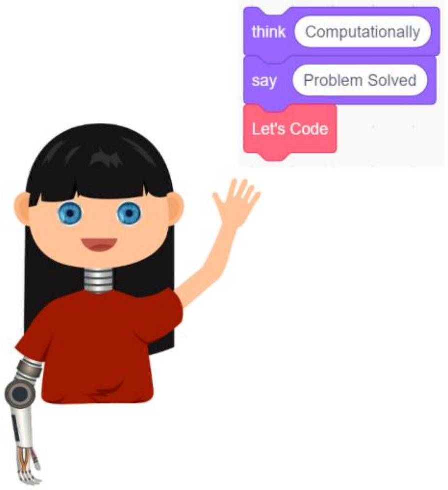


Figure 2. Co-instructor AISHA (Artificial Intelligence She/Her Humanoid Android), illustrated by Nitya Tiwari

AISHA being an AI herself not only taught but also learnt alongside students. Having characters such as AISHA, whose name reflected the identity of Kashmiri students, helped immensely to let our students feel that they too belong to computing and STEM.

3.2 Build on Familiarity in the Local Context, in addition to Low Floor and High Ceiling

It is imperative that our 21st century learners are introduced to topics like Artificial Intelligence, Big data and Machine Learning as they get introduced to computer Science. However low floor it is, concepts like machine learning might be intimidating to students for the first time. Using examples from their local context and observable surface culture, helps students connect readily with any concept. We heard from students about what stood out in the local Kashmiri context and apples were one of the local favourites. Kashmir's biggest economy is the apple industry with about 70% of India's total production of apples coming from Kashmir. We asked students to compare a robotic arm that could be trained to pick apples and analyse its efficiency with a human harvest. They not only learnt how a machine could be trained but were prompted to think critically about the effect of human biases as machines were trained to harvest apples. This way, students got a lot of room for thinking, and building on their familiarity helped them learn new and challenging concepts effortlessly.

3.3 Use Semantic Waves in a culturally responsive way to teach complex Computational Thinking Skills

Semantic waves was introduced by Karl Maton as part of the Legitimation Code Theory and was re-introduced in CS by Paul Curson, Queen Mary University of London (Curzon, 2019) using unplugged activities. Descending and ascending a semantic wave is especially challenging for computer science concepts with high semantic complexity, like algorithms or abstraction. To make this process seamless, we used culturally relevant examples during both the unpacking and repacking phase of the wave.

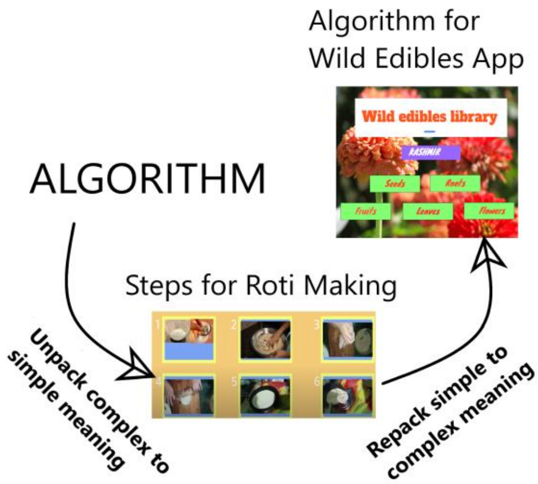


Figure 3. Semantic waves used to teach the concept of 'Algorithms' in a culturally responsive way

Students learnt about the concept of algorithms by unpacking concrete examples (from the local context) like making a roti and then move up to re-pack complex meanings by creating an algorithm as shown in Figure 3 to build a wild edibles app specific to Kashmir.

3.4 Tell a Tale and Give Life to it

Story-telling is an ancient, alluring and culturally universal form of teaching that students use to connect and reflect what they learn with their everyday lives. Stories have the power to reclaim the joy of learning computer science as students learn better by making content personally relevant. Above all, they feel included in the highly stereotyped field of computer science. Here is an excerpt from the computer story of our video lesson 'Let's cook a computer story', which personifies computer parts as real-life characters as shown in the Figure 4, "*Cache is the super-woman who tosses the ingredients to the CPU in lightning speed — but don't judge her by her looks! GPU is the girl who's got talent. She could do plating, presentation and what not. She's got a keen eye for imagery. And Drive, as you can see is a boy with special abilities. He lost his hand but uses a hand of a magic wand, to read, write or sketch almost anything. He could remember so much more than RAM and he is the one his friends rely on, to keep everything safe*"

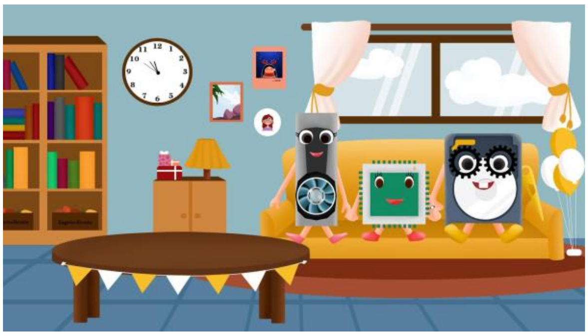


Figure 4. Meet GPU, Cache and Drive (from left to right) at their friend Sensor's birthday party (Illustrated by Nitya Tiwari, Edited by Vishesh Banerjee)

3.5 *Make students look at technology from a Critical Lens*

While students learnt to write their own algorithms, we also wanted to encourage them to think critically about the algorithms they encounter in their day-to-day lives. Students were introduced to the concept of algorithmic bias or human bias as they learnt how computers could be trained to pick apples based on Kashmiri farmers' apple picking preferences. In our example, Northern Kashmir is known for the quantity of apple produce, whereas southern Kashmir is known for its quality. Some farmers (might prefer picking apples a bit earlier than they are completely ripe and store them in cold storages, while others prefer to wait until the right time to harvest (until completely ripe). An important question to encourage students to ask is, "Are we accommodating all farmers' opinions into our training data or is the training data biased?" Students also think about other ill effects of algorithms such as using biased training data to train a machine to evaluate students for school or college admissions or to identify thieves and criminals. A well-deserved student may lose the opportunity of education or someone legitimate may be convicted for a crime they did not do. Students were introduced to big data and poor privacy and were also encouraged to ponder over how they can use this data to make their community better.

3.6 *Teach them to be kind & responsible Digital Citizens*

Digital citizenship is beyond digital literacy and digital safety. We teach students about cyber-safe behaviours such as creating strong passwords, not sharing their private information online and being mindful of phishing and other cyber-attacks while also identifying and verifying legitimate information online. But we didn't stop by teaching about the don'ts of cyber-safety and cyber-bullying. We also taught them how to be responsible and kind online. Students learnt not just the ill-effects of cyber-stalking and sexting, but also learnt how to respond and act constructively in such situations.

3.7 *Encourage Problem Solving from both a Design and Computational perspective*

A true digital native does not stop with exploring and living responsibly in physical and virtual spaces. One also exercises his civic responsibility by solving powerful problems with technology. In computing curriculum, problem solving is an area often envisioned to be deserving complexity and hence lesson plans focus on the 'challenge' or 'logic' factor, instead of assessing how 'meaningful' or relevant it is to students. Research suggests that students, especially girls, feel more confident about a problem, if it makes them feel a sense of purpose with their own community. We opted a *dual-approach to problem solving* as shown in Figure 5, using both *design and computational thinking* techniques. Students follow the design thinking approach to create projects as simple as a 'paper boat that doesn't sink' to creating something as complex as a 'wild edibles library to help improve malnutrition in India'. The key is to empathize, contemplate and question if the problem is meaningful to the student before dwelling deep.

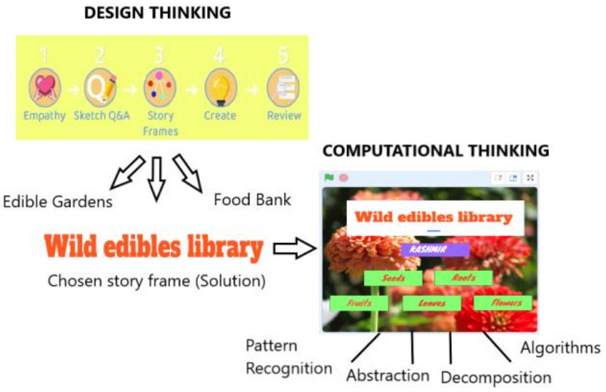


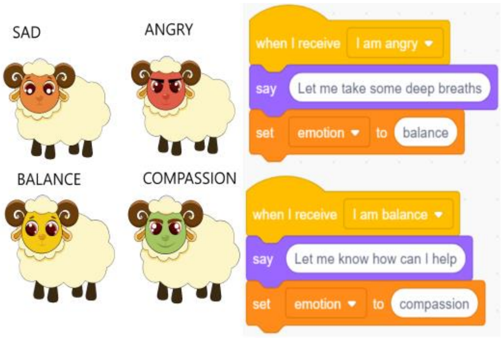
Figure 5. Dual approach to problem solving using computational and design thinking skills

3.8 *Create awareness about Digital Wellbeing*

While it was important to teach students about the digital skills they need be developing and how to be kind and responsible cyber-citizens, it was crucial to also teach them to be mindful of their technology usage. We taught them to stretch, breathe and take digital breaks to not let technology take control over their wellbeing. We encouraged practices such as setting screen-free times, taking regular digital breaks and even introduced some stretching poses like chair pose, forward bend fairy, turtle and lion's breath.

3.9 *Support students to acknowledge the Digital Divide*

Access to computers and internet is still a far cry in many parts of India. Just 4% of the rural households have access to computers as compared to 23% in urban areas (GOHAIN, 2020). Little over 15% of rural households have access to internet services (Kundu, 2020). Only 28% woman have access to the internet when compared to 72% of men (Kala, 2019). Due to this gendered digital divide, it has been increasingly difficult for woman to gain access to technology. By including these statistics, we make students aware of the digital divide and make them acknowledge that lack of access to a computer or internet doesn't have to mean lack of digital skills. We provided our students with low-cost computing tools and free block-based coding apps such as Code Mitra (Code Mitra, 2022) and encouraged collaborative sharing of these tools and resources such that even the most marginalized and underserved doesn't have to feel left out.



```
when I receive  I am angry ▾
say  Let me take some deep breaths
set  emotion ▾  to  balance

when I receive  I am balance ▾
say  Let me know how can I help
set  emotion ▾  to  compassion
```

Figure 6. Left: Emotions as 'Variables' (Illustrated by Nitya Tiwari) Right: Block-based code for emotions as variables

3.10Spread waves of Kindness and Compassion

Computer Science is often perceived as a field that does not find any relevance with being empathetic or building morals. Research has found that CS majors are less likely to see their roles in solving global problems, fostering justice, improving other people's lives or learning about different cultures or religion. There is a pressing need than ever, for a value-based, inclusive and equitable CS education, besides a skilled workforce. We teach students how to navigate their emotions as they code for emotions as 'variables' as shown in Figure 6, catch their thoughts using 'conditions' and contemplate about self-regulation as they code. Integrating socio-emotional learning to teach core programming concepts can guide students to be a good human besides being a good coder.

Table 1. Some of the Culturally Responsive Examples used to teach a CT Concept, Skill or Practice

CT Concept, Skill or Practice	**Culturally Responsive examples**
Computer Parts & Functions	CPU as the master chef who adds & subtracts flavors to his Rajma
Internet	Data packets and routers explained using hay distribution in sheep farms
Machine Learning & Bias	Apple harvesting in Kashmir using Robotic arms
Algorithms	Students learn algorithms by learning roti-making
Sequencing, Precision & Correctness	Explained through Rotimatic – a fully automatic roti making robot
Loops; Self & Social awareness	Coding for infinite loops of anger and kindness
Conditionals; self-awareness	Coding to catch their thoughts using conditionals and tag them as positive or negative
Variables; Self-awareness	Coding for emotions as variables and learning to navigate them

4. IMPLEMENTATION AND FINDINGS

More than 6000 students from Jammu & Kashmir across grade levels VI to XII took the video-based computing course on the Diksha platform. Our course was created with support from UNICEF India, Samagra Shiksha, JKKN and Diksha. Out of all students who took the course, 52% were reported to be girls and 48% boys. For greater than 94% of learners, this was the first-time they were exposed to block-based computing and coding. The curriculum has empowered thousands of minority girls to think, create and dream big with technology and prepare them to be strong, creative, critical and compassionate problem solvers. Our analysis shows that 83% of girl students were open to pursuing a career related to STEM, post completion of course compared to 32% at the beginning and 87% of students were able to articulate the link between Computer Science and phenomenon in everyday life. The course was featured in the UNICEF stall at the G20 education summit 2023 and UN TES (United Nations Transforming Education Summit, Collection of best practices, 2022) as one of the best practices in digital learning transformation. Table 1 shows some of the teacher testimonials after taking the Let's Code course.

Table 1. Teacher Testimonials from Let's Code course

Through contextualized curriculum, I hope to see my Kashmiri girls emerge stronger and higher in the tech field (Surya Suraf, Client partner, Tech Mahindra)
CS is for everyone, it builds problem solving and creative skills in the process (Monica Sharma, Govt School, Udhampur)
I had a fear of coding. But I realized that it's just another language. Just like we use a language to talk to each other, code is a language we use to communicate with the computer and CS is beyond coding (Sonika Bhandari, Govt. School, Jammu)

5. CONCLUSION

This study described our approach to encourage participation among under-represented students and minority girls through designing a culturally responsive, equitable and inclusive computational thinking course curriculum. The course progression and design offered students the possibility to engage with the content seamlessly without any prior exposure to computer science or computational thinking. Inclusion of culturally responsive examples proved to be an effective design strategy to reduce entry barriers for learners especially girl students.

6. REFERENCES

Code Mitra. (2022, July 15). Retrieved from Google Playstore: https://play.google.com/store/apps/details?id=org.thepijam.codemitra&gl=US

Curzon, P., Waite, J., Maton, K., & Donohue, J. (2020, October). Using semantic waves to analyse the effectiveness of unplugged computing activities. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education* (pp. 1-10).

GOHAIN, M. P. (2020, July 21). 23% of urban population has access to computers, only 4% of .. *Times of India*.

Kala, R. R. (2019, September 17). High gender disparity among internet users in India. *Financial Express*.

Kundu, P. (2020, May 05). DIGITAL DIVIDE, Indian education can't go online – only 8% of homes with young members have computer with net link. *Scroll.in*.

Ministry of Human Resource Development. (2020). National Education Policy. *Governtment of India*.

Morales-Chicas, J. C. (2019). Computing with relevance and purpose: A review of culturally relevant education in computing. *International Journal of Multi-cultural Education*, 125-155.

Ron Eglash, J. E. (2013). Towards culturally responsive computing education. *Communications of the ACM*, 33-36.

United Nations Transforming Education Summit, Collection of best practices. (2022, November). Retrieved from https://knowledgehub.sdg4education2030.org/computer-science-students-india

The Learning Effectiveness of the Computational Thinking Instructional Tool named AI2 Robot City and Its Sorting Extended Version

Ting-Chia HSU[1*], Wei-Ni WEN[1], Mu-Sheng CHEN[1], Tai-Ping HSU[1]
[1]Department of Technology Application and Human Resource Development, National Taiwan Normal University, Taiwan
ckhsu@ntnu.edu.tw, roru0945@gmail.com, mushengchen946@gmail.com, taipinshe@gmail.com

ABSTRACT

This study applied the computational thinking (CT) instructional tool named *AI2 Robot City* for learning the application of artificial intelligence (AI) on image recognition, and used its extended version, named the smart-warehouse unit, for practicing the bubble sort algorithm in a CT course. The participants in this quasi-experimental research were 50 university freshmen. This study compared the learning achievement and test anxiety of the students learning online with those of the students learning offline. The results showed that the learning achievement of the online students was significantly higher than that of the offline students. The test anxiety of the online students was significantly lower than that of the offline students. It is inferred that the use of the offline board game needs more self-regulated learning time to carry out interaction.

KEYWORDS

online learning, artificial intelligence in education, bubble sort, computational thinking

1. INTRODUCTION

Along with the widespread emphasis on computational thinking (CT), the development of artificial intelligence (AI) is also booming. Moreover, Artificial Intelligence in Education (AIED) is becoming one of the hottest topics in every domain of education (Hwang et al., 2020). The demand for AI courses in universities around the world is growing (Goel & Joyner, 2017). Therefore, integrating AI application into CT courses has drawn a great deal of attention. When AI application in non-CS courses needs to be developed, past research has also pointed out that visualization of programming and programming lessons with physical examples have considerable benefits for novice student learning (Scherer et al., 2020). Battistella and von Wangenheim (2016) also found that using games to teach programming can enhance learning for beginners by visualizing programming effects.

For instance, some AI STEM curriculums have been designed for non-CS (i.e., non-Computer Science major) undergraduates (Hsu et al., 2023; Lin et al., 2021). Since AI is an important branch of CS, its related knowledge and skills are closely related to computer programming based on CT processes. Higher education nowadays not only emphasizes CT for students in every domain, but has also begun to explore the learning of AI application. The study of AI is part of the curriculum in the field of CS, but non-CS students do not need to study the relevant content as deeply as those in the CS department. Considering the possibility of the development of AI application in various fields and daily lives, it is necessary to teach non-CS students some basic knowledge and skills for the purpose of understanding the application of AI. In order to design relatively general courses to help non-CS students learn, it is more appropriate to choose topics which involve the main application trends of AI and the experience of the students. At present, AI is widely used in various fields, including marketing, finance, agriculture, medical care, and transportation. In these application fields, transportation should be the subject that students have the most daily contact with. Therefore, a previous study pointed out that the emergence of smart technologies is prompting major changes in the logistics and transportation industries, such as Smart Warehousing (Chung, 2021). The application is related to the Internet of Things technology and AI application, such as self-driving cars and smart warehousing, involving image recognition. Therefore, we developed an extension to the CT instructional tool, AI2 Robot City, named smart warehousing, which is used to learn image recognition application integrated with bubble sorting for the scenario of smart warehousing.

In conventional instruction, these courses are taught by teachers' narration and demonstration (Malhotra et al., 2021; Nelke & Winokur, 2020). The current study not only attempted to investigate the test anxiety of non-CS students when learning AI application, but also the learning achievement of students' actively learning AI application online and offline was hypothesized differently. In particular, the COVID-19 pandemic of the past two years has made online learning necessary, so as to consider how to teach online, especially hands-on practice When students conduct courses remotely, the originally planned physical teaching materials like the CT board game will need to be transformed in the curriculum design to prevent them from being difficult to access and operate.

Some studies have pointed out that hands-on courses were particularly affected during the pandemic, because it is hard to give students opportunities to perform practical exercises (Oliveira et al., 2021). When learning practical knowledge and skills, if students cannot personally experience the operation process, it will be difficult to obtain corresponding learning experience to support their learning of the relevant content. Even though distance learning has a great impact on practical courses, if the relevant teaching aids can be designed so that students can operate at home, it may be able to make up for the aforementioned impact of distance learning on practical courses. Chung et al. (2019) pointed out that mobile learning with appropriate learning content can provide learning support for students, improve their learning performance, and reduce their cognitive load. Yang et al. (2019) also pointed out that undergraduates choose to continue learning through mobile devices if the three self-regulation needs of students (i.e., autonomy, competence, and relevance) can be met. Therefore, when students are studying alone at home, in addition to the teaching materials provided by the teacher, if the teaching materials and teaching aids designed on mobile devices (such as mobile phones and tablets) can be combined, it may be more convenient for students to operate. In addition, well-designed online learning can allow students to control their learning progress more freely, help them learn independently, and improve their learning effectiveness.

To focus on CT, several studies have pointed out that the results of assisted simulation programming through learning tools such as robots and AI application can enhance students' learning of CT (Kopcha et al., 2021; Merkouris & Chorianopoulos, 2019). The reason may be related to the way of cognition described in the theory of embodied cognition, that is, the brain is not the only cognitive resource which the students use to solve problems, but is achieved through the body and its perception guidance in the world (Wilson & Golonka, 2013). The operation process of physical instructional tools can also provide learners with shared physical performance and embodied cognition. Therefore, if it is possible to design a physical instructional tool for students who need to apply AI, it may be effective to enhance students' learning achievements and motivations related to AI application in the CT course.

2. METHOD

2.1. *Instructional Tools*

The students who learned in the classroom (i.e., offline) were mainly taught by the teacher with the instructional tool named AI2 Robot City and its extended sorting application shown as Figure 1. The students who learned online used the mobile learning videos. At the same time, the instructional tools were sent to the students' homes. Figure 1 shows the instructional content which all the students used in the study, regardless of whether learning in online or offline mode.

Figure 1. Screenshots of Instructional Tools

2.2. *Participants*

There were 28 freshmen participating in the offline learning of AI2Robot City and its extended version of bubble sorting, which was the control group in the study. There were 22 freshmen participating in the online learning of AI2Robot City and its extended version of bubble sorting, which was the experimental group in the study.

2.3. *Measuring Tools*

The test anxiety scale in the motivational strategies learning questionnaire (Pintrich et al., 1991) was employed to measure the anxiety of the participants while learning AI application offline and online in the CT course. There were five items with a 7-point scale, ranging from strongly disagree to strongly agree on a scale of 1 to 7.

The pre-test and post-test of learning achievement was reviewed by experts with more than 20 years of teaching experience. The content of the test includes 20 multiple-choice questions, four filling questions, and four quizzes, totaling 100 points.

2.4. *Experimental Process*

Figure 2 shows the procedure of the treatments. In addition to the pre-test, there were nine periods spent in the instructional experiment, with three periods per week. Each period took 50 minutes. Before following the experimental procedure shown in Figure 2, the students in both groups had spent three periods of class time learning the basic syntax of MIT App Inventor and the basic concept of sorting. The statistical method used in this study was analysis of covariance (ANCOVA).

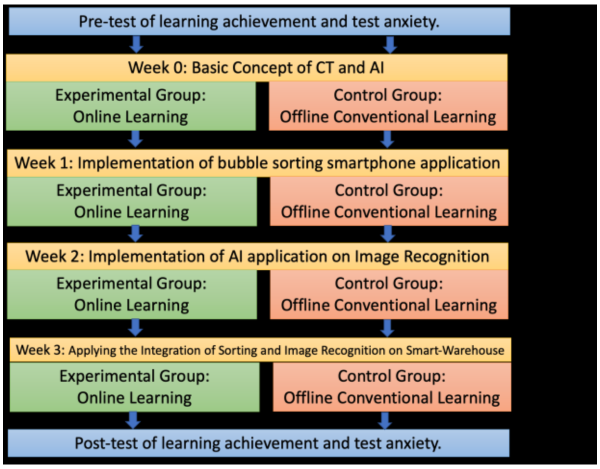


Figure 2. Experimental Procedure

2.5. *Research Questions*

This study aimed to answer two research questions, as follows: 1) Was the learning achievement of the online learning better than the learning achievement of the offline learning in the use of AI2 Robot City and its extended version of bubble sorting? 2) Was the test anxiety of the online learning students better than the test anxiety of the offline learning students when using AI2 Robot City and its extended version of bubble sorting?

3. RESULTS

3.1. *Learning Achievements*

The results of the ANCOVA on learning achievement showed that the experimental group (Adjusted mean =

74.84) scored significantly higher than the control group (Adjusted mean = 67.60), shown as Table 1.

Table 1. ANCOVA of Learning Achievements

Group	*N*	Mean	*SD*	Adjusted Mean	*SE*	*F*
Experimental	22	74.82	9.41	74.84	2.12	6.525^{*}
Control	28	67.64	11.50	67.60	1.88	

$^{*}p < .05$

3.2. *Test Anxiety*

The results of the ANCOVA on test anxiety showed that the experimental group (Adjusted mean = 4.03) scored significantly lower than the control group (Adjusted mean = 4.41) on the 7-point scale, shown as Table 2.

Table 2. ANCOVA of Test Anxiety

Group	*N*	Mean	*SD*	Adjusted Mean	*SE*	*F*
Experimental	22	4.00	0.71	4.03	0.13	4.671^{*}
Control	28	4.43	0.87	4.41	0.12	

$^{*}p < .05$

4. CONCLUSIONS

The results of the study show that the learning achievement of students learning with the developed instructional tool in both online and offline modes significantly improved. Furthermore, the students who studied online had better learning achievement after nine periods of treatment. It is inferred that this result may be due to the fact that the students' prior ability was sufficient to support their individual online learning. At the same time, online learning and the supplemental instructional tool allowed students to conduct self-regulated learning. The students could watch teaching videos and use the instructional content and tools at their own learning pace. In addition to submitting the completed mobile phone programs any time, many students also recorded the learning process by screen video, which built their learning confidence. In terms of test anxiety, it may be due to the fact that students' operation time was limited by the classroom arrangements, and their learning pace was confronted with peer pressure in the classroom, while online learning students could check the teaching materials and operations at any time according to their own abilities and speed. Therefore, while retaining the opportunity for learning at their own pace remotely without peer pressure, they could minimize their anxiety. Finally, students are not used to active inquiry learning in traditional physical courses, and tend to get used to being guided, while online learning students have no physical environment to rely on, and so have a greater opportunity to actively explore when they are provided with enough physical instructional tools and online learning materials.

5. REFERENCES

Battistella, P., & von Wangenheim, C. G. (2016). Games for teaching computing in higher education–a systematic review. *IEEE Technology and Engineering Education, 9*(1), 8-30. https://doi.org/10.1111/bjet.12314

Chung, C.-j., Hwang, G.-j., & Lai, C.-l. (2019). A review of experimental mobile learning research in 2010–2016 based on the activity theory framework. *Computers & Education, 129*, 1-13. https://doi.org/10.1016/j.compedu.2018.10.010

Chung, S.-H. (2021). Applications of smart technologies in logistics and transport: A review. Transportation *Research Part E: Logistics and Transportation Review, 153*, 102455. https://doi.org/10.1016/j.tre.2021.102455

Goel, A. K., & Joyner, D. A. (2017). Using AI to teach AI: lessons from an online AI class. *AI Magazine, 38*(2), 48-59. https://doi.org/10.1609/aimag.v38i2.2732

Hsu, T.-C., Chang, C., & Lin, Y.-W. (2023). Effects of voice assistant creation using different learning approaches on performance of computational thinking. *Computers & Education, 192*: 104657. https://doi.org/10.1016/j.compedu.2022.104657

Hwang, G.-J., Xie, H., Wah, B. W., & Gašević, D. (2020). Vision, challenges, roles and research issues of Artificial Intelligence in Education. *Computers and Education: Artificial Intelligence, 1*, 100001. https://doi.org/10.1016/j.caeai.2020.100001

Kopcha, T. J., Ocak, C., & Qian, Y. (2021). Analyzing children's computational thinking through embodied interaction with technology: a multimodal perspective. *Educational Technology Research and Development, 69*(4), 1987-2012. https://doi.org/10.1007/s11423-020-09832-y

Lin, C.-H., Yu, C.-C., Shih, P.-K., & Wu, L. Y. (2021). STEM based Artificial Intelligence Learning in General Education for Non-Engineering Undergraduate Students. *Educational technology & society, 24*(3), 224-237. https://www.jstor.org/stable/10.2307/27032867

Malhotra, S., Kumar, A., & Dutta, R. (2021). Effect of integrating IoT courses at the freshman level on learning attitude and behaviour in the classroom. *Education and Information Technologies, 26*(3), 2607-2621. https://doi.org/10.1007/s10639-020-10376-0

Merkouris, A., & Chorianopoulos, K. (2019). Programming embodied interactions with a remotely controlled educational robot. *ACM Transactions on Computing Education (TOCE), 19*(4), 1-19. https://doi.org/10.1145/3336126

Nelke, S. A., & Winokur, M. (2020). Introducing IoT subjects to an existing curriculum. *IEEE Design & Test, 37*(6), 24-30. https://doi.org/10.1109/MDAT.2020.3005358

Oliveira, G., Grenha Teixeira, J., Torres, A., & Morais, C. (2021). An exploratory study on the emergency remote education experience of higher education students and teachers during the COVID-19 pandemic. *British Journal of Educational Technology, 52*(4), 1357-1376. https://doi.org/10.1111/bjet.13112

Scherer, R., Siddiq, F., & Viveros, B. S. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior, 109*, 106349. http://doi.org/10.1016/j.chb.2020.106

Wilson, A. D., & Golonka, S. (2013). Embodied cognition is not what you think it is. *Frontiers in Psychology, 4*, 58. https://doi.org/10.3389/fpsyg.2013.00058

Yang, S., Zhou, S., & Cheng, X. (2019). Why do college students continue to use mobile learning? Learning involvement and self-determination theory. *British Journal of Educational Technology, 50*(2), 626-637. https://doi.org/10.1111/bjet.12634

ENABLING STEAM POSSIBILITY & EMPOWERING CLIL PRACTICE: A STEAM-INTEGRATED & CLIL-ORIENTED CURRICULUM DESIGN FOR PRIMARY STUDENTS

Kuay-Keng YANG[1], Yi-Hung LIAO[2], Sheng-Yi WU[3]*
[1]Department of Science Communication, National Pingtung University, Taiwan
[2]Mount Dawu College, National Pingtung University, Taiwan
[3]International Master Program in STEM Education, National Pingtung University, Taiwan
kuaykeng@gmail.com, yliaog@mail.nptu.edu.tw, digschool@gmail.com *

ABSTRACT

Despite the rise of STEAM education globally, there are still relatively deficiencies in the current teaching aids and teaching materials in STEAM education. Using a CLIL-for-STEAM pedagogical framework, this study explores students' inquiry competencies by examines the learning contents of magical power of electricity and traffic lights, matching the curriculum guidelines of Taiwanese 4th-grade students, to develop STEAM-integrated teaching and learning materials in a bilingual format to correspond to the bilingual nation policy. The goal of this course is to encourage students to use their prior knowledge and develop their problem-solving skills through interdisciplinary problem situations in a bilingual environment.

KEYWORDS

STEM education, interdisciplinary, teaching aids, teaching materials, CLIL (Content and Language Integrated Learning)

1. INTRODUCTION

The concept of STEM (Science, Technology, Engineering, and Mathematics) was developed in 1990, and STEM education has also gained attention in education since then. The integration of art into design has been proposed by some scholars as a way to cultivate students' creativity and interest in STEM learning (Sochacka, Guyotte, & Walther, 2016) as a way of teaching STEAM (Science, Technology, Engineering, Arts, and Mathematics). Additionally, some scholars have pointed out in recent years that students should not learn in only one field, but should learn from multiple fields based on the needs of their students (Saorín, Melian-Diaz, Bonnet, Carrera, Meier, & De La Torre-Cantero, 2017).

STEAM-related studies (Kanadli, 2019) have found that teachers' professional training, equipment, and environment are crucial in promoting STEAM education. In this context, STEAM teaching aids and teaching materials play a key role. However, the STEAM teaching aids and materials available at present only require assembling materials according to the instructional steps, claiming to deliver STEAM-oriented learning outcomes despite the fact that the content often does not align with the learners' existing knowledge and skills. Additionally, there are no relevant STEAM teaching materials available to support the bilingual education policy at the moment. As a result of this research study, bilingual STEAM teaching materials and teaching aids will be developed based on students' knowledge at a certain stage, so that students can integrate different subjects and solve life problems utilizing the knowledge they have acquired.

2. BILINGUAL STEAM PROJECT-BASED LEARNING EDUCATION CURRICULUM DESIGN INTRODUCTION

For STEAM education curriculum design, using problem-, project-, or design-based tasks to provide students with the opportunity to solve real-world situations is a practical teaching method (National Research Council, 2014). The study utilized STEAM project-based learning to allow students to work on a real-world project, guiding and encouraging them to integrate and apply their knowledge and skills across five different subjects (Science, Technology, Engineering, Arts, and Mathematics) by proposing an idea, designing a prototype, and testing the prototype.

The design of a bilingual STEAM project-based learning education curriculum will be introduced to a student in three steps (as shown in figure 1). Table 1 describes the scenario problems and how students’ STEAM knowledge and skills will be demonstrated through scoping and defining problems, followed by solving the identified problems.

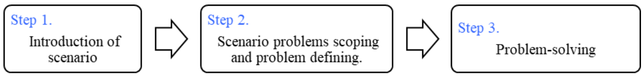
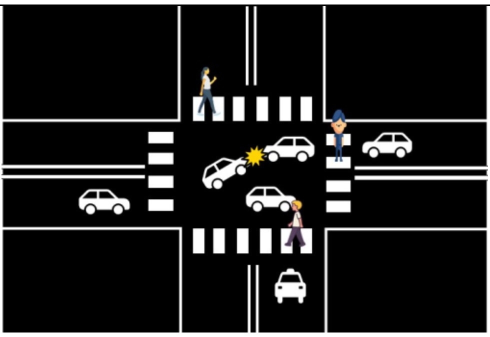


Figure 1. the three-step of Bilingual STEAM project-based learning education

Table 1. Traffic light STEAM project-based learning Curriculum design

Step 1. Introduction of scenario
Car accidents happen every day at the intersection



Mother: Watch out, Dino. There are so many cars on the road today.
Dino: What is the loud sound? What happened?
Mother: Oh. No! There is a car accident. The two cars crashed into each other at the intersection.
Dino: That is sad. Are they going to be okay?
Mother: Let's hope for the best! We all need to be careful on the road.

Step 2. Scoping and defining the scenario problem.

Scoping scenario problems [See-Think-Wonder]

- **See**: What do you observe?
- **Think**: How can you explain what is happening?
- **Wonder**: What questions do you have?

Defining scenario problem in subjects:

Science: *How to set up traffic lights with a simple circuit? Can simple circuits be connected in series or parallel to individually light up the red, green, and yellow bulbs?*

Technology: *How can we use technology to set an automatic traffic light?*

Engineering: *How can we build a stable traffic light stand with recycled cardboard boxes*

Arts: *What are the color configuration order of traffic lights and the visual effect of colors*?

Mathematics: *How do you calculate the number of seconds for the red, green, and yellow lights according to the traffic flow?*

Step 3. Problem-solving

- **Science (Problem & Task →Brainstorm & Inquiry →Analyze & Summarize)**

Problem and Task: How to set up traffic lights with a simple circuit? Can simple circuits be connected in series or parallel to individually light up the red, green, and yellow bulbs?

Brainstorm & Inquiry:

Brainstorm: *Connect the red, green, and yellow LEDs with independent switches (as shown in figure 1)*

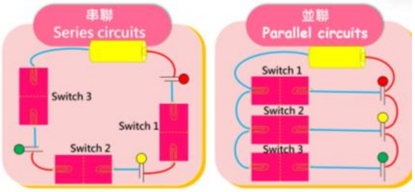


Figure 1. series and parallel circuits

Inquiry: Specified the science research questions:

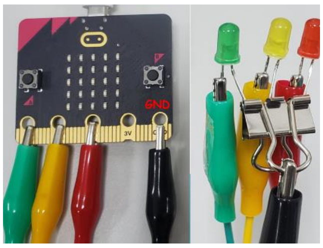
What happens when turning on each switch in a series or a parallel circuit?

Analyze and Summarize: *Which connection mode (serial or parallel) will meet the changing needs of traffic lights?*

- **Technology (Problem & Task → Explore & Experiment →Discover & Explain)**

Problem & Task: *How can we use technology to set up an automatic traffic light?*

Explore & Experiment: *"Loop" will be used to program an automatic traffic light that changes from green to red to yellow.*



d

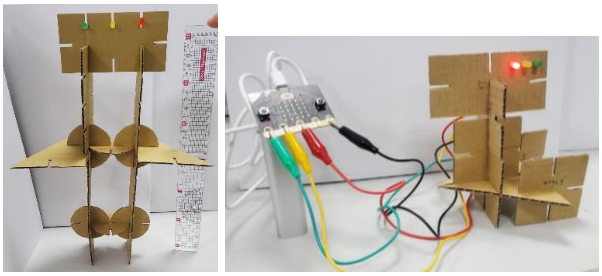
Discover & Explain: *Verify the results of programming data*

- **Engineering (Problem & Task→Imagine & Plan→Create & Improve)**

Problem & Task: How can we build a stable traffic light stand with recycled cardboard boxes.

Imagine & Plan: How can we develop ideas and plan the design for the product in accordance with the requirements?

Create & Improve: What can we do to prototype, test, and redesign?



- **Art (Problem & Task → Research & Design → Perform & Share)**

Problem & Task: What are the traffic lights and the visual effect of colors?

Research & Design:

What are the differences between left-hand and right-hand traffic on traffic light order?

What is the visual effect of red, green, and yellow colors on a human being?

Perform & Share:

Appreciate the beauty of color light and create free creation of color light overlapping

- **Mathematics (Problem & Task → Problem solving)**

Problem & Task:

It is a 100-meter intersection, and Dino wants to ensure that all the cars waiting for the red light can pass in one green light. Assuming that each car needs a parking space of 5 meters, what is the minimum time for a green light?

Information:

Time required to pass the intersection for the first car is ***five seconds***

Time required to pass the intersection for the second car is ***three seconds***

Time required to pass the intersection for the third and over car is ***one seconds***

Problem solving: *[Teachers guides students to solve problems through graphic representation]*

3. SIDE BY SIDE FOR STEAM AND CLIL

As an educational objective that emerged from the European multilingual policy in the mid-1990s, Content and Language Integrated Learning (CLIL) is a pedagogical approach that aimed at enabling citizens to use three European languages functionally (L1/first Language + 2 objective/target languages) (Council of Europe, 2007). While the primary aim of CLIL was to encourage citizens to become equipped with an additional language (Mehisto et al. 2008; Marsh, et al. 2012; Coyle et al., 2010) to promote economic advances and global competitiveness, CLIL, especially in Asian counties, has more recently evolved into a pedagogical methodology connected to the teaching and learning of English (Wu & Lin, 2019; Tsou & Kao, 2018; Abduh & Rosmaladewi, 2018; Lin, 2022) which encompasses not only the acquisition of four linguistic skills but also the impacting of socio-cultural parameters. Today, science and language learning are ubiquitously important aspects of education, which is why the project was conceived. By integrating STEAM education into content and language integrated learning (CLIL), the project also stimulates the development of a community of practice between science-based subject experts, language education professionals, experienced in-service practitioners and pre-service teachers that promotes the development of students' scientific and language competences.

Mediating between the dual pedagogical objectives on second/foreign language acquisition and curricular subject content, the framework of CLIL featured with the 4Cs (Content, Communication, Cognition, and Culture/Community) (Mehisto et al., 2008; Coyle et al., 2010) is helpful in understanding how CLIL can be conceptualized from a broader perspective as well as to be applied in this STEM Cardboard Challenge of CLIL-based project. The 4Cs framework places an integrated teaching and learning emphasis where content is considered intricately linked to and inseparable from communication, culture, and cognition. Regarding the" CONTENT," lessons are scaffolded to build upon the student's prior knowledge of different subject matter, so the student can build an understanding and comprehension of the subject content and even apply it with an integrated learning mode. For instance, in the unit of "Traffic Lights," students were guided to review, integrate and apply the knowledge of "setting up a switch,” "the power of electricity,” "the series and parallel circuit,” "the functional meaning of color light," and even "the land area calculation" to design a traffic light at the intersections to solve the traffic problems. Secondly, CLIL-based classes emphasize" COMMNICATION" in which students are at the center and teachers give them the opportunity to talk and work in pairs or groups, especially during the experiment sessions. As a result, students are learning new content while simultaneously improving their acquisition and use of the target language. The teacher's role would be to serve as a good subject knowledge expert and language resource to provide support whenever necessary to facilitate meaningful and fluid communication. To design a functional cardboard tower, students were provided with ample meaningful and scaffolding writing and speaking exercise to use the target vocabulary and sentence patterns to discuss the planning, communicate with the peers and even describe their products at the final showcase project. Moreover, a CLIL lesson will be effective if the teacher embraces real-world experiences to the classroom and assists students in developing their" COGNITION" to think for themselves as critical thinkers observing and discovering the world as well as problem solvers addressing and improving their livelihoods. It is about helping students develop their cognition, construct their own understanding, and challenge themselves to create for innovation regardless of their ability or linguistic proficiency. Echoing Bloom's taxonomy of the learning pyramid, this STEAM project started with eliciting students lower-order thinking (LOTs, such as remembering and observing what they have learned before, understanding and associated with the real-world problem of traffic accidents, experimenting and applying the proposed solutions of traffic light designing) and followed up with developing their higher-order thinking (HOTs: distinguishing and analyzing the properties of lights, electricity, cardboard structures, and programming, evaluating the alternatives for effectiveness and efficiency, and finally collaborating and creating as a team for the best solutions to the firstly identified real-world problem). Lastly, the world is facing more and more unprecedented challenges as we go forward. It is essential for every 21st-century learner to demonstrate tolerance and understanding in order to celebrate and realize our multicultural and multilingual world. This project, characterized as a CLIL-oriented and STEAM-based classroom, was essentially a meaning-making learning "CULTURE" and "COMMUNITY" for subject knowledge and language acquisition. It is also considered an incubator for cultivating intercultural awareness and understanding, such as appreciating the local designs with colors, distinguishing the different directions of traffic lights around the world, connecting the basic color theory to personal feelings, and even the universal design in architecture and engineering.

4. Conclusion

Through examining the learning contents of the magic of electricity power and traffic lights, this study explores students' inquiry competencies using a CLIL-for-STEAM pedagogical framework, to develop STEAM-integrated bilingual materials that correspond to the curriculum guidelines of Taiwanese 4th-grade students. A real-world scenario is designed according to students' prior knowledge, and the students will be able to learn science, technology, engineering, arts, and mathematics through the problem-inquiry-analyze method. In this study, the curriculum development design is presented, and further empirical research will be conducted to assess the effectiveness of textbooks and teaching aids.

REFERENCES

Council of Europe. (2007). *From linguistic diversity to plurilingual education: Guide for the development of language education policies in Europe*. Retrieved from Council of Europe Language Policy Division: https://rm.coe.int/16806a892c

Coyle, D., Hood, P., & Marsh, D. (2010). *CLIL: Content and language integrated learning.* Cambridge, UK: Cambridge University Press.

Kanadli, S. (2019). A Meta-Summary of Qualitative Findings about STEM Education. *International Journal of Instruction, 12*(1), 959-976.

Lin, T. B. (2022). The foundation of intercultural communication: The practice and challenges of bilingual rducation in Taiwan. *Journal of Curriculum Studies, 17*(1), pp. 1-13.

Mehisto, P., Marsh, D., & Frigols, M. J. (2008). *Uncovering CLIL: Content and language integrated learning in bilingual and multilingual education.* Oxford, UK: Macmillan.

Marsh, D., Mehisto, P., Wolff, D., & Frigols Martín, M. J. (2012). *European framework for CLIL teacher education*. Retrieved from https://ebuah.uah.es/xmlui/bitstream/handle/10017/14881/CLIL-Framework_Marsh.pdf?sequence=1&isAllowed=y

Sochacka, N. W., Guyotte, K. W., & Walther, J. (2016). Learning together: a collaborative autoethnographic exploration of STEAM (STEM + the arts) education. *Journal of Engineering Education, 105*(1), 15–42.

Saorín, J. L., Melian-Diaz, D., Bonnet, A., Carrera, C. C., Meier, C., & De La Torre-Cantero, J. (2017). Makerspace teaching-learning environment to enhance creative competence in engineering students. *Thinking Skills and Creativity, 23*, 188-198.

Tsou, W-L., & Kao, S-M. (2018). *Exploring CLIL: A Resource Book. Bookman Books*, Taipei City

Wu, Y., & Lin, A. M. (2019). Translanguaging and trans-semiotising in a CLIL biology class in Hong Kong: Whole-body sense-making in the flow of knowledge co-making. *Classroom Discourse, 10*(3-4), 252–273.

Is Computational Thinking Self-Efficacy aligned with Computational Thinking Comprehension?

Jongpil CHEON[1*], Tianxiao YANG[2], Seonkwan HAN[3]
[1,2]Texas Tech University, U.S.A.
[3]Gyeongin National University of Education, South Korea
jongpil.cheon@ttu.edu, tianxyan@ttu.edu, han@gin.ac.kr

ABSTRACT

This study identified the alignment between students' CT self-efficacy level and CT comprehension level. The descriptive statistics and content analysis of 315 undergraduate students' survey responses found that there was considerable discrepancy between the beliefs and comprehension. The underperformance of abstraction and evaluation comparing other CT components was detected as well. Based on the results, implications of CT assessments and instructional strategies to enhance CT thinking skills are discussed.

KEYWORDS

Computational thinking, CT self-efficacy, CT comprehension, Discrepancy

1. INTRODUCTION

Computational Thinking (CT) became an important competency for problem-solving skills (Shute et al., 2017; Wings, 2006) in both K-12 and higher education especially in the STEM disciplines. Numerous studies have been conducted to construct CT frameworks, measure CT competencies, and propose pedagogical methods. The CT assessment is one of the primary concerns in CT education since the objectives and assessments should be aligned together to measure the effectiveness of the CT activities.

Learners' self-efficacy of CT is a common method to measure their CT skills because the confidence influences their thinking skills, and cognitive competency affects their beliefs. However, few studies have determined if the self-efficacy levels are aligned with their understanding or performance of CT skills. Higher level of CT self-efficacy would lead higher level of CT understanding or skills, but beliefs and skills may not be consistent in different components of CT. Thus, this study aimed to investigate if there is a discrepancy between the CT self-efficacy and CT comprehension. The potential misalignment was investigated with descriptive, correlation and predictive statistics. The findings provided insights of CT assessments and CT education for both researchers and educators. The research questions were:

- To what extent do students perceive their CT self-efficacy?
- To what extent do students comprehend CT?
- Are the CT self-efficacy and CT comprehension correlated together?
- Does CT self-efficacy predict CT comprehension level?

2. CT COMPONENTS

The problem solving with computing power became an integral part of our daily lives and work environments. Since CT was introduced as fundamental problem-solving skills, the goal of CT education is to introduce CT as a thinking skill with various learning activities, such as block-based programming, unplugged activities, or robotics (Kwon, Cheon, & Moon, 2021).

There is little agreement about what CT encompasses, and sub-categories of CT were named as CT components, concepts, or steps. Wing (2006) initially posed five CT processes (i.e., problem reformulation, recursion, problem decomposition, abstraction, and systematic testing), and the National Research Council (2010) defined the CT model by identifying new five CT processes (i.e., hypothesis testing, data management, parallelism, abstraction, and debugging). Brennan and Resnick (2012) proposed a CT framework with three key dimensions (i.e., concepts, practice, and perspectives). Anderson (2016) developed one of the most famous CT models including a) decomposition, b) pattern recognition, c) abstraction, d) algorithm, and e) evaluation as the five CT processes. The five CT components were used to measure the CT self-efficacy and comprehension in this study.

3. CT ASSESSMENT

A systematic review of CT assessment (Tang et al., 2020) reported that CT processes were assessed with artifacts (e.g., Program codes or portfolio) or multiple-choice questions. Most of the assessments were related to fundamental CT concepts, such as loop, conditionals, variables, etc. On the other hand, CT attitudes and motivation were collected with surveys or interviews, and CT self-efficacy was usually measured through a survey (Bower et al., 2017). In addition, student's programming activities were recorded with a screen cast software, and video files were analyzed.

Although programming is a common approach to teach and assess CT skills, CT should be thinking skills which can be applied to other disciplines. The computing power is crucial for CT, but the specific thinking steps are playing an important role in solving structured and ill-structured problems. Thus, learners should be able to define CT with the CT components, and it would be significant to measure their comprehension of CT with their own definition since their programming code or test score may not reflect their actual comprehension of the thinking skills. Therefore, this study collected CT definition from an open-ended question to determine learner's comprehension levels and compare it to their confidence levels.

4. METHOD

4.1. Data Collection

The participants were 315 undergraduate student who enrolled a course "Computing and Information Technology" in a large public university in the United States. The course covered CS basic knowledge and CT skills with programming activities with Scratch.

They were asked to take a survey at the end of the course, the survey included five 5-point Likert scale items to indicate their CT self-efficacy levels in terms of five CT components: (a) Decomposition, (b) Pattern Recognition, (c) Abstraction, (d) Algorithm, and (e) Evaluation. The last item was a short-essay question to state what computational thinking is.

4.2. Data Analysis

First, the CT self-efficacy data was analyzed with descriptive statistics. Second, the researchers used the directive qualitative content analysis technique to create and develop a coding framework measuring students' CT comprehension. It is called "directive" because it allowed the researchers to start with an existing conceptual framework and develop new categories under the framework (Hsieh & Shannon, 2005). In this study, the researchers used the "CT Definition" with its five components (Anderson, 2016) which is same as the categories of CT self-efficacy. A double-coding protocol was conducted, and the interrater reliability was 0.81. Next, the frequency of each CT components for each participant was calculated, and the data was used for a comparison, correlation and regression.

5. RESULTS

5.1. To what extent do students perceive their CT self-efficacy?

The mean scores of self-efficacy levels on all five CT components were higher than 3.83 out of 5 as shown in Table 1. The participants were positively confident on all areas in CT.

Table 1. CT Self-efficacy Scores

	Mean	Percentile
Decomposition	4.02 (.827)	80.4%
Pattern Recognition	3.94 (.876)	78.8%
Abstraction	3.85 (.880)	77.0%
Algorithm	3.98 (.806)	79.6%
Evaluation	3.83 (.917)	76.6%

5.2. To what extent do students comprehend CT?

The results (See Table 2) show that when students were required to describe the CT, they often would mention setting up a step-by-step solution (i.e., Algorithm, 36.8%, 116 codes) and analyzing a problem by dividing it into smaller parts (i.e., Decomposition, 30.8%, 97 codes). Comparatively, they rarely mentioned that identifying the common features among different objects (i.e., Pattern Recognition, 20.0%, 63 codes). Moreover, only a limited number of responses mentioned detecting errors (i.e., Evaluation, 11.4%, 36 codes) and distinguishing the important information from unimportant information (i.e., Abstraction, 9.5%, 30 codes).

Table 2. CT Self-definition

	Frequency	Percentile
Decomposition	97/315	30.8%
Pattern Recognition	63/315	20.0%
Abstraction	30/315	9.5%
Algorithm	116/315	36.8%
Evaluation	36/315	11.4%

5.3. Are the CT self-efficacy and CT comprehension correlated together?

A spearman correlation was conducted repeatedly to examine the relationships between the CT self-efficacy items and their corresponding CT comprehension performances. The results in Table 3 shows that the Spearman r values in terms of the correlations between CT self-efficacy and CT comprehension are 0.093 (Decomposition), 0.085 (Pattern Recognition), 0.018 (Abstraction), 0.055 (Algorithm), and -0.094 (Evaluation). All the correlations are insignificant (p > 0.05). It means that none of students' self-efficacy in decomposition, pattern recognition, abstraction, algorithm, and evaluation was significantly correlated to their exact CT performance regarding the corresponding CT component.

Table 3. Spearman r between CT self-efficacy (CTSE) items and their corresponding CT comprehension (CTC)

Association (CTSE ←→ CTC)	Spearman's rho	Sig.
Decomposition	0.093	0.099
Pattern Recognition	0.085	0.133
Abstraction	0.018	0.750
Algorithm	0.055	0.327
Evaluation	-0.094	0.096

5.4. Does CT self-efficacy predict CT comprehension level?

A binominal logistic regression was conducted repeatedly to ascertain if each CTSE item can predict its corresponding CTC performance (i.e., the occurrence of corresponding CT comprehension code). The Hosmer and Lemeshow test was used to examine if each predictive binominal logistic regression model adequately fitted the data (i.e., Goodness of Fit). Table 4 shows that all the Hosmer and Lemshow test results are insignificant (p > 0.05). It means each of the five predictive models adequately described the data set.

Table 4. Hosmer and Lemeshow Test for Model Fit (CTSE: CT Self-efficacy; CTC: CT Comprehension)

Predictive Model (CTSE → CTC)	Chi-Square	df	Sig.
Decomposition	2.413	2	0.299
Pattern Recognition	5.250	2	0.072
Abstraction	0.548	2	0.760
Algorithm	4.488	2	0.106
Evaluation	0.679	2	0.712

However, the specific logistic regression statistics shown in Table 5 indicates that all the predictive relationships were insignificant ($p > 0.05$). It means that students' self-efficacy in their decomposition, pattern recognition, abstraction, algorithm, or evaluation did not predict their corresponding higher CTC performance.

Table 5. Binominal Logistic Regression Predicting Each CT Comprehension Code Based on Each Corresponding Survey Item (CTSE: CT Self-efficacy; CTC: CT Comprehension)

Predictive Model (CTSE → CTC)	B	S.E.	Wald	df	Sig.
Decomposition	0.268	0.159	2.826	1	0.093
Pattern Recognition	0.192	0.170	1.266	1	0.261
Abstraction	0.118	0.227	0.270	1	0.603
Algorithm	0.178	0.150	1.411	1	0.235
Evaluation	-0.275	0.180	2.329	1	0.127

In sum, the insignificant predictive effects identified that students' CT self-efficacy could be unreliable proof of their academic success regarding their CT comprehension.

6. DISCUSSION

The findings show that there is discrepancy between the learners' beliefs toward computational thinking and their ability to define computational thinking. More specifically, the self-efficacy levels of all five CT components were higher than 75 % while the comprehension levels of all CT components were lower than 40 %. Also, the self-efficacy levels were neither correlated with nor predictable to the CT comprehension levels. The results could reveal the following implications for researcher and educators.

First, researchers should not rely on only self-efficacy data as an outcome since higher self-efficacy beliefs do not guarantee higher learning outcomes (Pajares, 1996). More diverse or multimodal data, such as program artifacts, observations, refection journals, portfolio, interview or quizzes, should be collected to measure accurate learners' CT comprehension.

Second, the underperformance of CT comprehension levels in the findings yields to rethink about learning contents and instructional strategies in CT education. In most of CT courses, programming or coding activities are the main topics, and instructor may assume that the CT components or process could be automatically possessed as learners create various products with a block-based programming language. However, it would be better to provide not only hands-on practices but also other types of activities to recapture coding procedure to visualize the CT steps and enhance thinking skills. The applications of CT steps in real-life examples should be integrated into other subject areas.

Third, the comprehension levels of "Abstraction" (9.5%) and "Evaluation" (11.4%) were comparatively lower than other CT components. These weaknesses are also found in previous studies in that abstraction is often ignored by students (Ezeamuzie, Leung, & Ting, 2022), and evaluation (i.e., debugging) is treated as an automatic programming behavior (Huang et. al., 2022; Vourletsis, Politis, & Karasavvidis, 2021) although detecting and fixing errors is significant skills for CT. The two thinking skills should be explicitly taught with appropriate instructional strategies. For example, unplugged activities to visualize abstraction procedure examples could be helpful, and introducing different debugging methods with samples would enhance CT comprehension level.

On other hand, the study has some limitations. Since the self-efficacy level for each CT component was measured by only one survey item, the reliability could be questioned. The CT comprehension levels were measured with only responses to one open-ended question. Other data could be included for more accurate assessment of CT comprehension.

7. REFERENCES

Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning & Teaching, 15*, 226-234.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (Vol. 1, p. 25).

Bower, M., Wood, L. N., Lai, J. W. M., Howe, C., Lister, R., Mason, R., et al. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education, 42*(3), 53–72.

Ezeamuzie, N. O., Leung, J. S., & Ting, F. S. (2022). Unleashing the potential of abstraction from cloud of computational thinking: A systematic review of literature. *Journal of Educational Computing Research, 60*(4), 877-905.

Hsieh, H.-F., & Shannon, S. (2005). Three approaches to qualitative content analysis. Qualitative Health Research, 15, 1277-1288. doi:10.1177/1049732305276687

Huang, M., Cheon, J., Park, E., & Fan, X. (2022). Investigating challenges of debugging tasks in an undergraduate computational thinking course. *Proceeding of the Annual Convention of the Association for Educational Communications and Technology, 2022*(1), 154-164.

Kwon, K., Cheon, J., & Moon, H. (2021). Levels of problem-solving competency identified through Bebras computing challenge. *Education and Information Technologies, 26*(5), 5477-5498.

National Research Council. (2010). *Committee for the Workshops on Computational Thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, D.C: National Academies Press.

Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of Educational Research, 66*(4), 543 – 578.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 1-22.

Vourletsis, I., Politis, P., & Karasavvidis, I. (2021). The effect of a computational thinking instructional intervention on students' debugging proficiency level and strategy use. In *Research on E-Learning and ICT in Education* (pp. 15-34). Springer, Cham.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*, 33-35.

Cultivating Students’ Computational Thinking and Digital Literacy in University Courses: A Hong Kong Example

Ka Wai Cecilia CHUN[1*], Ping Fu FUNG[2*], Yat Chiu LAW[3*], Kwok Wing SUM[4*]
[1,2,3,4]The Chinese University of Hong Kong, Hong Kong
kawaichun@cuhk.edu.hk, pffung@cse.cuhk.edu.hk, yclaw@cse.cuhk.edu.hk, kw04@cuhk.edu.hk

ABSTRACT

This paper introduces two core information technology (IT) courses which were newly developed and offered to all first-year undergraduate students in a university in Hong Kong starting from 2022-23. The two courses aim to equip students with computational thinking (CT) and digital literacy (DL) skills, both of which are identified as the 21st century skills essential in the digital age. An evaluation study was conducted with the first cohort of students taking the courses in the first year of full implementation. This paper presents the features of the courses, walk through the design and methodology of the evaluation study, and summarizes the major findings. The evaluation study shows that after taking the courses, students had significant improvements in CT and DL self-efficacy. Implications of the findings are discussed and directions for future research are suggested.

KEYWORDS

computational thinking, digital literacy, university education, evaluation study

1. INTRODUCTION

In the digital age, the competence and confidence in handling, evaluating and using digital data are becoming more and more important. University education plays an essential role of equipping students with the necessary related skills and attitudes so that they would not be drowned in the sea of data but be able to make use of data for living, learning and working in their future. In view of the pressing need of students, two new core IT courses, namely Digital Literacy and Computational Thinking, were designed and offered to all entrants to the undergraduate programmes in a university in Hong Kong. Two sets of courseware were developed, and an evaluation study was conducted to evaluate the effectiveness and to inform the curriculum review of the courses. The following section will brief on the conceptual underpinnings of the study and the main features of the two courses.

2. BACKGROUND

2.1. Computational Thinking

The term “computational thinking” was coined by Papert (1980) and reintroduced and elaborated by Wing (2006), who defined computational thinking as “solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science”. To put it simply, CT is to think like a computer scientist in face of problems (Wing, 2014). An important difference between CT and computer science is that CT is not only programming, but also a universal skill which, as argued by researchers, can be applied to the learning of and problem-solving in other disciplines. CT has attracted increasing attention since Wing (2006) used the term in her seminal paper, which is reflected by the fact that many countries have mandated computing education in K-12 (Kong et al., 2019), and large-scale international educational assessments have incorporated CT as one of their test domains (IEA, 2023; OECD, 2019). According to a review study by Tang et al. (2020), about 70% of CT assessment research was conducted in K-12 settings, whereas only about 15% was conducted in colleges, indicating a lack of evidence of how effectively CT education is implemented at the university level.

2.2. Digital Literacy

The emergence of the term “digital literacy” dates back to mid-1990s, when it first appeared in the literature and extended the meaning of literacy from print reading and writing to comprehending and using digital information on the Internet. As defined by Gilster (1997), digital literacy is the “ability to understand and use information in multiple formats from a wide variety of sources when it is presented via computers”. Two decades later, the European Commission provided a more refined definition in Digital Competence Framework (DigComp), which delineates the concept of DL into five competence areas, i.e., information and data literacy, communication and collaboration, digital content creation, safety, and problem solving (Carretero et al., 2017). While DL overlaps with CT in the area of computer programming, it is claimed that some aspects of CT (e.g., algorithmic thinking) can be developed in the absence of digital technology (Let’s Talk Science, 2018). In contrast to the myth that “digital natives” who are born in the digital age have a self-developed capacity to use digital technology, scholars pointed out that DL could be taught and IT education is important in this regard (Fraillon, 2019; Ng, 2012).

2.3. Features of Course Design and Implementation

To cover aspects of both CT and DL while satisfying the needs of different major programmes, two courses, version P and R respectively, were developed and implemented. They share the same contents except that Course P focuses on using Python while course R focuses on the statistical tool R. Major programmes are advised to choose a version that complements the major study of their students. For example, Statistics Programme picked version P. It is expected that students would gain the following intended learning outcomes (ILOs) from the courses, with each ILO annotated as DL and/or CT related: (1) Use spreadsheet to organize and process data (CT, DL); (2) describe the importance of information security and data privacy (DL); (3) apply methods and tools to obtain and use data properly

(CT, DL); (4) solve problems in a computational thinking style (CT); (5) process and analyse textual (Course P) / quantitative (Course R) data using software packages (CT, DL); (6) interpret and present data accurately to suit different application scenarios (DL).

The two university-level courses had been piloted twice in 2021-22, along with a thorough evaluation study. The findings were presented to different faculties internally to help them understand the importance and the effect of the new courses on students' CT and DL. After some fine-tunings, full implementation of the courses has started in 2022-23, serving 3500 Year 1 undergraduates annually. Both courses were enhanced, and the research instruments were validated during the pilot run.

The courses cover topics of digital literacy, digital data security and privacy, data acquisition, data preparation, data processing, misuse of statistics, introductory data science, modeling (such as decision tree, regression, k-means and classification), data visualization, CT in Python/R, practical applications and real-life examples, such as MNIST handwritten digit recognition, Iris dataset, email spam filtering. The courses also emphasize hands-on experience with computational tools to master the concepts and knowledge firmly. For example, students would practise Python/R coding with variables and objects, selection and repetition constructs, defining and calling functions, using provided packages and APIs including NLTK, tidyverse, matplotlib, ggplot2, etc. On top of solid lectures, there are weekly lab classes and take-home assignments as well as a capstone project. During the lab, students are required to Bring Your Own Device (BYOD) so that the classes can be conducted in interactive classrooms rather than computer rooms. Each student is also assigned a personal Virtual Machine (VM) with 8 CPU cores and 16GB RAM for intensive learning tasks. Irrespective of their device capabilities and system settings, a homogeneous VM computing and learning experience can be delivered over Remote Desktop to the VM. Licensed and standard software are provisioned in the VM to free students from setup issues and hassles. To embrace cloud computing, Jupyter Notebooks and open notebook computing platforms are introduced so that students can apply in their project and their further study.

3. RESEARCH QUESTIONS

The aim of this study is to evaluate the effectiveness of the two new core IT courses when they are fully implemented. Specifically, it investigates whether and to what extent the intended learning outcomes of the courses have been achieved, and provides evidence for the review and enhancement of curriculum design and implementation. This paper will address the following three research questions (RQ):

RQ1: What is the effect of taking the courses on students' CT and DL self-efficacy (course-specific)?

RQ2: What is the effect of taking the courses on students' DL self-efficacy (general)?

RQ3: What is the effect of taking the courses on students' attitudes towards ICT for learning?

4. METHODOLOGY

4.1. Data and Participants

The data is drawn from the evaluation study of the two IT courses in the First Term of 2022-23. Students enrolled in the two courses were invited to participate in pre-course and post-course questionnaire surveys, which were conducted online in September 2022 and from December 2022 to January 2023 respectively. The response rates of the two surveys were 63.9% (N=1044) and 37.8% (N=618) respectively.

The sample comprises 490 students who participated in both pre-course and post-course surveys. A vast majority of them were in Year 1 (99.2%) and the remaining were in Year 2 to Year 4 (0.8%). There were slightly more females (50.4%) than males (49.6%). The students were studying in Faculties of Engineering (35.5%), Business Administration (31.8%), Arts (24.1%), or interdisciplinary or double degree programmes (8.6%). About 59% of students did not have any learning experiences of programming or coding, while some students had learnt programming or coding for less than one year (23.0%), 1-3 years (11.7%) and over 3 years (6.1%).

Apart from students' background information, the surveys collected students' perceptions on three major constructs before and after they took the courses. The first construct, DL and CT self-efficacy (course-specific), is derived from the intended learning outcomes of the courses. The second and third constructs are DL self-efficacy (general) and attitudes towards ICT for learning respectively. Operationalization of these three constructs is described in the following sub-sections.

4.2. Operationalization of Major Constructs

Self-efficacy is a self-belief in one's capacity which has been shown to have positive correlation with academic performance (Honicke & Broadbent, 2016). In this study, two types of self-efficacy are measured. CT and DL self-efficacy (course-specific) is measured by using a 14-item scale adapted from the ILOs of the two courses. Example items include "Use basic data models to describe digital data" and "Solve problems in a computational thinking style". The participants rated their confidence in doing 14 tasks on a 7-point scale ranging from "Very unconfident" (1) to "Very confident" (7). As the two courses share the same ILOs except for the programming language (Python or R) that they cover, all sample students were asked to respond to the same items except for one item which was tailored for students taking each of the two courses (Item 7P/7R: "Use Python/R functions to retrieve digital data). The internal consistency (Cronbach's alpha) of the scale is high for the pre-course and post-course surveys (0.96 and 0.97 respectively).

DL self-efficacy (general) scale is adapted from the Self-Efficacy Scale for Digital Competences in Schools by Mannila et al. (2018). It is included in this study to provide converging evidence on the effectiveness of the courses. Mannila et al. (2018) developed this 27-item self-efficacy scale based on European Union's framework DigComp 2.0 (Vuorikari et al., 2016). In our surveys, 16 items were selected which are related to the contents covered by the two courses. Compared with CT and DL self-efficacy (course-specific), DL self-efficacy (general) is a more generic but not course-specific measure of students' learning outcomes. Example items include "Adapt my searches based on knowledge about how search engines produce results" and "Plan and design a solution to a problem in the form of step-by-step instructions". The participants were asked to rate their confidence in doing 16 tasks on a 7-point scale ranging from "Very unconfident" (1) to "Very confident" (7). The internal consistency (Cronbach's alpha) of the scale is high for the pre-course and post-course surveys (0.96 and 0.97 respectively).

Attitudes towards ICT for learning are measured by using a 5-item scale adapted from the questionnaire developed by Ng (2012). Example items include "I like using ICT for learning" and "I learn better with ICT". Participants were asked to rate how much they agreed with 5 statements on a 5-point scale ranging from "Strongly disagree" (1) to "Strongly agree" (5). The internal consistency (Cronbach's alpha) of the scale is high for the pre-course and post-course surveys (0.91 and 0.94 respectively).

5. RESULTS AND DISCUSSION

To examine the effectiveness of the two courses in promoting the three major outcomes, students' ratings in the pre- and post-course surveys are compared. A series of paired samples t-tests were conducted for CT and DL self-efficacy (course-specific) (Table 1), DL self-efficacy (general) (Table 2), and attitudes towards ICT for learning (Table 3). The items in the tables are sorted in descending order of the pre-post differences in the respective construct. In each pair of pre-post comparison, the larger value is highlighted in bold.

For CT and DL self-efficacy (course-specific), the averages of pre- and post-course ratings across the 14 items are 3.95 and 4.68 respectively. Table 1 shows significant increases in rating in all 14 items, indicating that students had significant improvements in their confidence in solving problems in a computational thinking style and handling digital data, and that the ILOs of the two courses were achieved. The largest increase appears in Item 12 (Use Virtual Machine (VM) for remote computing). It is the item where students had the lowest pre-course rating, which may be due to students' lack of experience of using VM before taking the courses. On the other hand, Item 2 (Understand the importance of information security and data privacy) shows the smallest increase. It is the item where students had the highest pre-course rating. A possible reason is that information security and data privacy is emphasized in IT education at primary and secondary levels in Hong Kong (Education Bureau, 2022) and students might have already learnt the related topics before entering the university.

For DL self-efficacy (general), the averages of pre- and post-course ratings across the 16 items are 4.82 and 5.11 respectively. Table 2 shows significant increases in rating in 13 items, indicating that students had an overall significant improvement in their confidence in understanding and using digital data. Two items (Item 16 and Item 2) do not show any significant post-course change, whereas one item (Item 7) shows a significant decline. The largest increase appears in Item 13 (Protect digital equipment from undesired access online), where students had the lowest pre-course rating. It is interesting to note that even though students understood well the importance of information security as mentioned in the above sub-section, many of them were not confident enough to practice it by protecting their digital devices until they took the courses. On the other hand, Item 7 (Recognize hate speech in discussing online) shows a significant but small decline. While recognizing hate speech is not one of the foci of the courses, further analysis is needed to investigate the reason for this finding and whether it is related to any student background factors.

Table 1. Paired Samples *t*-tests for CT & DL Self-efficacy (Course-specific)

Item	Pre-course		Post-course		
	Mean	SD	Mean	SD	*t*
12	3.34	1.73	**4.48**	1.65	-13.02***
11	3.71	1.59	**4.71**	1.46	-12.55***
7R	3.52	1.68	**4.37**	1.74	-3.44***
1	4.18	1.53	**5.01**	1.34	-11.24***
10	3.96	1.65	**4.78**	1.48	-10.53***
13	3.35	1.65	**4.15**	1.64	-8.96***
5	3.88	1.61	**4.64**	1.44	-10.11***
8	3.70	1.79	**4.40**	1.63	-8.60***
9	4.29	1.54	**4.93**	1.42	-8.49***
6	4.18	1.50	**4.80**	1.37	-8.48***
7P	3.55	1.75	**4.12**	1.58	-6.73***
4	4.34	1.59	**4.90**	1.42	-7.53***
3	4.57	1.43	**5.07**	1.25	-6.63***
2	4.77	1.41	**5.14**	1.26	-5.02***

*** $p < 0.001$

Table 2. Paired Samples *t*-tests for DL Self-efficacy (General)

Item	Pre-course		Post-course		
	Mean	SD	Mean	SD	*t*
13	4.12	1.55	**4.82**	1.47	-8.83***
10	4.20	1.45	**4.86**	1.35	-8.80***
12	4.22	1.59	**4.82**	1.44	-7.74***
9	4.56	1.28	**4.96**	1.32	-5.76***
1	4.75	1.47	**5.14**	1.36	-4.97***
11	4.52	1.40	**4.90**	1.35	-5.39***
5	4.82	1.47	**5.16**	1.39	-4.52***
4	5.03	1.38	**5.29**	1.33	-3.77***
3	4.80	1.34	**5.05**	1.33	-3.61***
6	5.00	1.39	**5.23**	1.32	-3.21**
14	4.83	1.44	**5.03**	1.41	-2.64**
15	4.98	1.39	**5.18**	1.35	-2.76**
8	5.29	1.41	**5.44**	1.28	-2.25*
16	5.16	1.34	**5.28**	1.31	-1.68
2	**5.30**	1.38	5.23	1.36	0.87
7	**5.51**	1.26	5.36	1.35	2.17*

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$

For attitudes towards ICT for learning, the averages of pre- and post-course ratings across the 5 items are 3.83 and 3.81 respectively. Table 3 shows no significant post-course changes except for one item, i.e., Item 3 (ICT makes learning more interesting) where there is a significant but small decline. In contrast to the overall significant improvements in CT & DL self-efficacy (course-specific) and DL self-efficacy (general), the overall insignificant post-course change in students' attitudes may be because attitudes normally take a longer time to change when compared with knowledge growth.

Table 3. Paired Samples *t*-tests for Attitudes towards ICT for Learning

Item	Pre-course		Post-course		
	Mean	SD	Mean	SD	*t*
4	3.69	0.88	**3.73**	0.92	-0.90
5	3.83	0.84	3.83	0.86	-0.05
2	3.86	0.83	3.86	0.86	0.00
1	**3.89**	0.83	3.82	0.92	1.50
3	**3.91**	0.83	3.82	0.91	2.02*

* $p < 0.05$

6. CONCLUSION

The design and provision of CT and DL courses for all first-year undergraduate students is a challenging step to take in university education, given the diverse educational background, learning needs and interests and programming experiences of students. Yet, the promising improvements made by students as shown in the present study proves that this bold step is worth taking. For one thing, this study has substantiated the important role of formal IT education in promoting CT and DL of students at university level. The integration of hands-on lab sessions and technologies such as VM and Jupyter Notebooks into the courses appears to be the key to success of the courses. For another, the study has addressed the call for more CT assessments at higher educational level and more reliability and validity evidence of CT assessments (Tang et al., 2020). The valid and reliable instruments used in this study are invaluable tools which can be applied to other intervention studies for assessing their effectiveness in enhancing CT and DL self-efficacy and attitudes. Finally, the absence of post-course improvement in students' ICT-related attitudes warrants our attention. To address this issue, more daily life examples are included in lectures, and bonus lab tasks are provided for able students to achieve extended ILOs in the Second Term of 2022-23. It remains to be seen whether these improvements in design may bring about changes to students' attitudes.

7. ACKNOWLEDGEMENT

We thank the anonymous reviewers for their constructive comments. This research is supported by Teaching Development and Language Enhancement Grants funded by the University Grants Committee (2022-25; 2019-22).

8. REFERENCES

Carretero, S.; Vuorikari, R. and Punie, Y. (2017). *DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use*. EUR 28558 EN.

Education Bureau. (2022). *Information Literacy for Hong Kong Students" Learning Framework (Updated Version) (Draft).*

Fraillon, J. (2019, August). Digital literacy: Myths and realities. In K. Cockle (Ed.), *Research Conference 2019: Preparing students for life in the 21st century: Identifying, developing and assessing what matters: Proceedings and Program* (pp.68-72). Melbourne: Australian Council for Educational Research.

Gilster, P. (1997). *Digital literacy*. New York: Wiley Computer Publications.

Honicke, T., & Broadbent, J. (2016). The influence of academic self-efficacy on academic performance: A systematic review. *Educational research review, 17*, 63-84.

International Association for the Evaluation of Educational Achievement [IEA]. (2023). *ICILS 2018: International Computer and Information Literacy Study 2018.*

Kong, S. C., Abelson, H., & Lai, M. (2019). Introduction to computational thinking education. In S. C. Kong & H. Abelson (Eds.), *Computational thinking education* (pp. 1-10). Singapore: Springer.

Let's Talk Science. (2018). *Computational Thinking Framework.*

Mannila, L., Nordén, L.-Å., & Pears, A. (2018, August). Digital competence, teacher self-efficacy and training needs. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 78-85).

Ng, W. (2012). Can we teach digital natives digital literacy? *Computers & Education, 59*(3), 1065-1078.

Organisation for Economic Co-operation and Development [OECD]. (2019). *Computer Science and PISA 2021.*

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798.

Vuorikari, R., Punie, Y., Carretero Gomez, S., Van den Brande, G. (2016). *DigComp 2.0: The Digital Competence Framework for Citizens. Update Phase 1: The Conceptual Reference Model*. Luxembourg Publication Office of the European Union. EUR 27948 EN.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2014). *Computational thinking benefits society. 40th anniversary blog of social issues in computing*, 2014, 26.

SCRATCH PROGRAMMING IN SCIENCE: THE IMPACT ON COMPUTATIONAL THINKING AND STUDENT ACHIEVEMENT

Noraini LAPAWI[1], Samri CHONGO[2*], Hazrati HUSNIN[3]
[1,2]Science Department, Institute of Teacher Education Penang Campus, Penang, Malaysia
[3] Centre of Teaching and Learning Innovation, National University of Malaysia, Selangor, Malaysia
noraini.lapawi@ipgm.edu.my, samrichongo@gmail.com, hazrati@ukm.edu.my

ABSTRACT

Students' low achievement in science is due to their lack of skills to solve complex problems. Therefore, computational thinking skills (CT) are skills that need to be applied to help students solve problems. The purpose of this study was to identify the impact of the CT Science Module on CT achievement and skills. This study was conducted using a quasi-experimental design involving 61 Form One students in Sabah, Malaysia. Two teaching approaches were the treatment group using modules integrated with Scratch programming and the control group using a conventional approach. Students' CT skills were measured by using the Bebras Task and their achievement was assessed using objective and subjective questions on the topic of Matter. Both instruments had high validity and reliability and were analyzed by ANCOVA. The results showed that CT skills were not significant, while achievement was significant. The integration of CT skills through Scratch programming was crucial to empower STEM education to help students solve problems and further improve achievement.

.

KEYWORDS

computational thinking, STEM education, problem-solving, matter

1. INTRODUCTION

The technological development in the Fourth Industrial Revolution has had a great impact on all aspects of human life such as the economy, social, and education systems globally. Currently, students are equipped with 21st-century skills in the education system so that they can survive and compete globally, especially in solving problems and being able to think creatively and critically. Accordingly, Basic Computer Science (BCS) and Computer Science (CS) were introduced in Malaysia in 2017 through the KSSM revision as elective subjects (MOE, 2016). Their objectives are to enable students to solve complex problems through computational thinking (PC) using computer-based solutions (MOE, 2016). Computer science is linked to engineering thinking because it can build a system that interacts with the real world (Wing 2006). CT skills are 21st-century skills for future generations that must be developed (Seneviratne 2017). However, there is little research on how to integrate CT skills with other subjects in the curriculum, especially Science as a way to solve more complex problems.

CT skills are different from other approaches to solving problems. CT skills involve solving problems, designing systems, and understanding human behavior with the basic concepts of computer science (Wing 2006). Therefore, the steps to solve the problem are based on computer science (Yadav, Hong, & Stephenson 2016). In addition, CT is more systematic by using abstraction and decomposition to solve more complex problems (Wing 2006). Yadav, Hong, and Stephenson 2016 agree with Wing's (2006) view that abstraction and decomposition are CT skills in solving problems. The clear difference is CT skills adapt computer science concepts to solve problems compared to other approaches. The CT skills are effective in solving problems (Faber et al. 2017; Yadav, Hong, & Stephenson 2016). The importance of CT can be seen in everyday life which is heavily influenced by algorithms without realizing its importance in learning that can help solve problems (Seneviratne 2017). However, very few integrate CT skills through programming in pedagogy (Parimalah et al. 2019). Therefore CT skills need to be applied to students in the education system so that they can solve problems in life and be able to compete in the digital economy era.

2. LITERATURE REVIEW

Computational thinking uses heuristic thinking to find solutions to problems (Wing 2006); how problem-solving is made to be easier to solve. CT is a cognitive or thinking process that involves thinking logically to solve problems and artifacts, procedures, and understanding the system better (Csizmadia et al. 2015). Problem-solving involves working towards a goal (Robertson 2001; Hesse et al. 2015) and to achieve the goal, the ideas obtained do not have a clear or routine solution (Hesse et al. 2015). In the process of solving problems, technology plays an important role (Siti Hendon 2016), especially computing technology that involves programming.

Computing is one of the widely used technologies that combine IT, Computer Science, digital literacy, and problem-solving using CT skills. (Webb et al. 2017). CT are problem-solving activity (Selby & Woollard 2010). Programming is a process that involves identifying, analyzing, and understanding a problem, evaluating feasible solutions, generating algorithms, as well as implementing solutions in the form of coding (Dagiene & Stupuriene 2016; Webb et al. 2017). The presence of computing technology is crucial to develop CT through programming to solve problems. *Scratch* is one form of programming that contributes to problem-solving skills

(Kalelioğlu & Gülbahar 2014). According to Nikiforos, Kontomaris dan Chorianopoulos (2013), American MIT scientist, Resnick has proposed Scratch programming as a method to solve problems because it makes a person think. Programming refers to the broader activity of analyzing problems, designing solutions, and implementing them, while coding is the implementation stage of problem-solving in a specific programming language (Bocconi et al. 2016).

Science subjects are often considered difficult, especially in the topic of Matter. This is because the matter is made up of particles that are abstract and difficult for students to understand and often cause misconceptions among students as well as among teachers (Nike Kusuma et al. 2016; We, L.Y., 2004; Keong, T.C. 2008). Cokelez & Dumon (2005) also shows that students always have misconceptions about the concept of atoms and molecules. The abstract situation makes it difficult for students to use their imagination for elements that cannot be seen with the naked eye and are difficult to describe (Mohamad Bilal Ali & Norida Md Dalhar, 2009). Misconceptions about atoms and molecules do not only involve diagrams but also explain the concept of atoms and molecules (Cokelez & Dumon 2005). The misconceptions cause problems and disrupt the learning of students to understand the basic concepts of matter (Salmiza & Haslinda 2015; Özmen & Ayas 2003; Adbo & Taber 2009). Therefore, the integration of CT skills in science needs to be implemented to overcome the misconceptions.

The process of applying CT skills can be done using a computer for programming through plugged-in Scratch and Python (Tsarava et al. 2017). CT should be integrated into all subjects in the T&L, especially in a plugged-in way because it can help students be more creative, critical, innovative, and able to solve problems. Scratch is a useful visual tool for teaching and studying programming. It allows novice programmers to understand concepts such as logical structures, variables, event-driven processing, and debugging (Yukselturk & Altiok 2016). Programming through Scratch is very useful because of the visual interface, pedagogy, and CT factors that provide an interesting and easy-to-use learning environment. Programming through Scratch is very useful because of the visual interface, pedagogy, and CT factors that provide an interesting and easy-to-use learning environment (Saltan & Kara 2016; Yukselturk & Altiok 2016). Previous studies show that CT skills through Scratch help solve problems and understand a concept among students (Kalelioğlu & Gülbahar 2014; Moreno-León, Robles,& González 2015; Su et al. 2014)

CT skills include algorithmic thinking, decomposition, generalization, identifying patterns, abstraction, and evaluation (Csizmadia et al. 2015; Selby & Woollard 2013). According to Grover & Pea (2013), the five CT skills mentioned have similarities in the study (Csizmadia et al. 2015; Selby & Woollard 2010). Meanwhile, Angeli et al. (2016) stated CT skills include algorithm, decomposition, generalization and identifying patterns, abstraction, and debugging. Nowadays, CT concepts or skills are often an issue and are still inconsistent. However, based on renowned researchers, CT concepts or skills that are commonly applied in the world of education can be divided into five, namely decomposition, abstraction, algorithms, pattern recognition, and evaluation. All five skills are applied in the Sc-CT Module. These five skills can be summarized in Table 1.

Table 1. CT skills and their description

CT skills	Description
Algorithm	Involves using an orderly sequence of steps in the process of solving a problem or completing a task (Faber et al. 2017).
Abstraction	Eliminates irrelevant aspects of the problem, the individual is not distracted and can directly pay attention to the important aspects of the problem (Faber et al., 2017). In the process of choosing important steps, students need to analyze data to draw conclusions and develop general principles (Yadav, Hong, and Stephenson 2016).
Decomposition	A way of thinking about artifacts in the context of looking at their components or smaller parts. The parts can then be understood, solved, developed, and evaluated separately (Csizmadia et al. 2015).
Evaluation	Explain how a process is more effective and efficient. The effectiveness and efficiency of solving a problem are evaluated. Evaluation skills in CT are telling someone to find the most effective or efficient solution to the problem (Faber et al 2017).
Pattern recognition	The approach for problem-solving can be used or applied to other and similar problems. The elements of the problem can be used to solve problems in other situations and can be improved based on the situation (Faber et al 2017).

Based on the literature, it can be concluded that CT skills can help students solve problems. Thus, that prompts researchers to study the effectiveness of the Sc-CT Module on CT skills and Science achievement. The objectives and hypotheses of this study are as follows;

Objectives:

a) Evaluate the effectiveness of the Sc-CT Module on the achievement of the topic of Matter
b) Evaluate the effectiveness of the Sc-CT Module on computational thinking

Hypotheses :

H_{01}: There was no significant difference in mean post-achievement test scores between the treatment group and the control group.

H_{02}: There was no significant difference in the mean scores of the computational thinking skills test between the treatment and control groups.

3. METHODOLOGY

3.1 Research design

This study involved a quasi-experimental design using a pre-test and post-test design. The experiment aimed to test the effectiveness of the CT Module on the CT skills of the students who were undergoing the intervention in the study. The CT module is a teaching approach that integrates CT in the teaching of Science involving two treatment groups and one control group. The first treatment group is to see the effect of the teaching approach on those who use the CT Module that integrates CT skills through Scratch programming. Whereas the control group is a group that follows a conventional teaching approach. A general overview of the research design is shown in Table 2.

Table 2. Quasi-experimental study design

Groups	Pre-test	Intervention	Post-test
Treatment	U_1	X_1	U_2
Control	U_1	X_2	U_2

Note

U_1 : Pre-test
U_2 : Post-test
X_1: Sc-CT Module
X_2 : Conventional

3.2 Sample

The sampling method used in the study was purposive sampling. This study involved 65 students, 31 from the control group and 34 students from the treatment group from two public schools.

3.3 Teaching Procedures

The teaching and learning process for the control and experimental groups used the same standard, which is according to the Standard Document of Curriculum and Assessment (DSKP) for Form 1 science issued by the Ministry of Education Malaysia (MOE). The teaching approach for the experimental group was to integrate CT skills using Scratch programming and the control group through the conventional method.

3.3.1 Experimental group teaching procedure (treatment)

The intervention in this study used the Sc-CT Module integrated into Science subjects. This study was conducted in school according to the normal learning schedule which was conducted twice a week. Each week took 120 minutes in the T&L process over five weeks. However, the project produced would be completed outside of learning time. The teaching approach for the experimental group produced two projects using Scratch 3.0. Before the intervention, pre-tests and briefings were given to teachers and students separately, while post-tests were conducted as soon as the intervention was completed. Students solve problems using programming collaboratively in small groups (3-4 pupils).

3.3.1 Teaching procedures of the control group

The teaching and learning method of the control group still used a problem-solving approach that follows the MOE model and used existing technology such as computers in the teaching of the topic of Matter. Other activities such as experiments were also carried out and the teaching period was four weeks.

3.4 Instruments

3.4.1 Computational Thinking Skills Test (UKPK)

There are 15 objective questions in the Computational Thinking Skills Test (UKPK) that allocates one mark for each correct answer. The UKPK questions are constructed from the Beabras tasks which consist of three levels according to the degree of difficulty, namely easy, medium and difficult. CT skills, difficulty levels, and percentages are shown in Table 3 and the total allocated time to answer this question was 45 minutes.

Table 3. CT skills and difficulty level of each question

Question No	CT Skills/Concepts	Difficulty Level	Percentage (%)
1	*Algorithm*	Easy	26.7%o
2	*Abstraction*)	Easy	
3	*Algorithm*	Easy	
4	*Decomposition*	Easy	
5	*Abstraction*	Medium	46.6%
6	*Evaluation*	Medium	
7	*Pattern Recognition*	Medium	
8	*Decomposition*	Medium	
9	*Algorithm*	Medium	
10	*Abstraction*	Medium	
11	*Pattern Recognition*	Medium	
12	*Decomposition*	Hard	26.7%
13	*Evaluation*	Hard	
14	*Pattern Recognition*	Hard	
15	*Evaluation*	Hard	

3.4.2 Achievement Test for the Topic of Matter (UPPJ)

Student Achievement Test for the Topic of Matter (UPPJ) aimed to evaluate the effectiveness of the Sc-CT Module on student knowledge in the topic of Matter. Two sets of questions are used which are the pre and post-test. Both sets of questions have the same number of questions, the form of the question, the level of difficulty, and the scope. The content scope in the pre and post-test questions ensures that the questions are included in all T&Ls conducted. The pre and post-test questions consist of three types, namely objective, structural, and essay questions. There are 15 objective questions, three structural questions, and two essay questions in both the pre and post-test.

3.5 Data Analysis

Data were analyzed using ANCOVA to test the effectiveness of the Sc-CT Module on the CT skills and Science achievement.

4. FINDINGS AND DISCUSSION

*4.1 **Comparison of Science tests between groups***

Table 4 shows that the ANCOVA test for post-UPPJ was significant, F(65) = 132.725, $p < 0.001$, with effect size, *partial* η^2 = 0.682. According to Cohen (1988), the suggested guidelines (0.01 = small, 0.06 = medium, 0.138 = large), based on the results, *partial* η^2 = 0.682, the effect size was large. Therefore, hypothesis H_{01} was rejected. The Sc-CT Module showed to improve the achievement of the topic of Matter compared to the conventional approach.

The approach of integrating CT skills through the programming of Inquiry-Based Learning (IBL) and Project-Based Learning (PBL) approaches encourages students to think critically, creatively, and able to solve problems (Mannila et al. 2014). PBL and IBL approach based on scaffolding with guidance from friends and teachers in the Zone of Proximal Development (ZPD) allowed students to solve a variety of more complex problems (Basawapatna et al. 2013). PBL also provides opportunities for students to work together in groups to carry out hands-on activities to stimulate students to develop abstract concepts (Bicer et al. 2015) which causes many misconceptions (Garnett & Treagust 1992a, 1992b). Projects produced through Scratch allow students to better understand abstract concepts in Science, particularly at the microscopic level (Kamisah & Lee 2013).

Table 4. Results of the ANCOVA test for the post-UPPJ by group

Source	Sum of squares	df	Mean square	F	Sig.	Partial eta squared
Groups	896.476	1	896.47	132.72	**0.000**	0.682

Significance = 0.05

The findings of the study are also supported by previous studies that plugged-in activities (Scratch) can help solve problems among students in understanding a concept (Kalelioğlu & Gülbahar 2014; Moreno-León et al. 2015; Su et al. 2014) and increase achievement (Basu et al. 2017; Rodriguez et al. 2016). Samri et al. (2021) also proved that the integration of CT skills through Scratch programming can improve achievement in Chemistry.

4.2 Comparison of CT skills between groups

The ANCOVA test in Table 5 showed no significant difference, F(65) = 0.896, p = 0.347, with a small effect size (*partial* η^2 = 0.014). Therefore, H_{02} failed to be rejected. However, based on the mean post-UKPK score obtained, it was found that the treatment group outperformed the control group.

Table 5. Results of the ANCOVA test for post-UKPK by groups

Source	Sum of squares	df	Mean square	F	Sig.	Partial eta squared
Groups	0.338	1	0.338	0.896	**0.347**	0.014

Significance = 0.05

Findings from past studies show that the process of applying computational thinking skills requires a longer time to get significant results, especially when using Scratch. This finding can be proven through the results of a study conducted by Kalelioğlu dan Gülbahar (2014) which was only carried out for one hour a week for five weeks and was not significant. However, Korkmaz and Oluk (2016) showed that for a relatively long period (six weeks), the findings of the study were significant. In most cases, it shows that the activities carried out to apply CT skills require more time (Atmatzidou & Demetriadis 2016). This finding clearly shows that CT skills need a longer time to show significant findings. However, the descriptive statistical comparison of the mean score in Figure 1 shows that the treatment group is better than the control group.

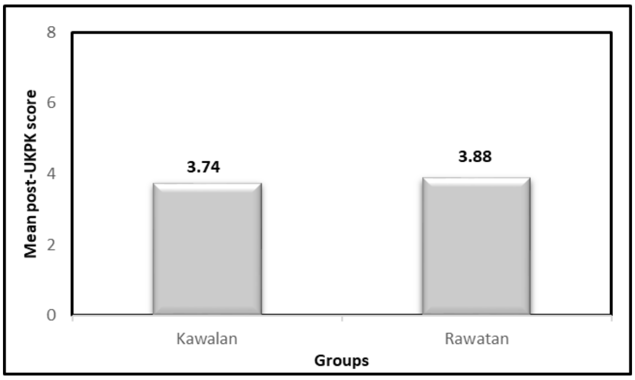


Figure 1. Comparison of Mean Scores of Control and Treatment Groups

In the five applied CT skills, the components of decomposition, algorithms, and evaluation showed better achievement using the Sc-CT Module based on the activities carried out as shown in Figure 2. The algorithm skills applied in the Sc-CT Module through the PBL approach are shown in Figure 3, while the decomposition skills are in Figure 4. Algorithm skills are key to understanding computer science (Duncan & Bell 2015), including planning, programming, and exploring complex computational problems and design techniques (Duncan & Bell 2015). In computer science, most people use algorithm skills to solve problems (Reichert, Couto Barone & Kist, 2020) and algorithmic visualization is more effective for students (Hundhausen et al., 2002). In general, problem-solving in CT often uses decomposition and abstraction (Wing, 2006; Yadav, Hong & Stephenson, 2016).

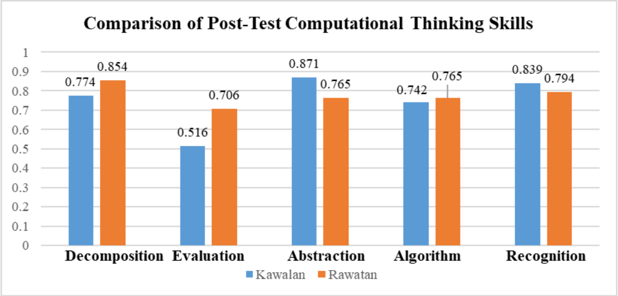
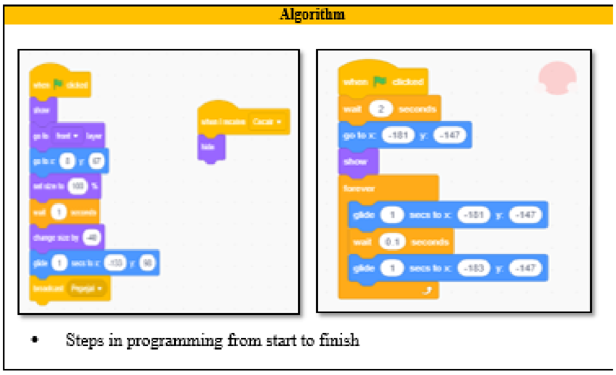


Figure 2. Comparison of CT Skills in Post-Test



```
when clicked
wait 2 seconds
go to x: -181 y: -147
show
forever
glide 1 secs to x: -181 y: -147
wait 0.1 seconds
glide 1 secs to x: -183 y: -147
```

Figure 3. Algorithm Skills applied through the Project Production (PBL)

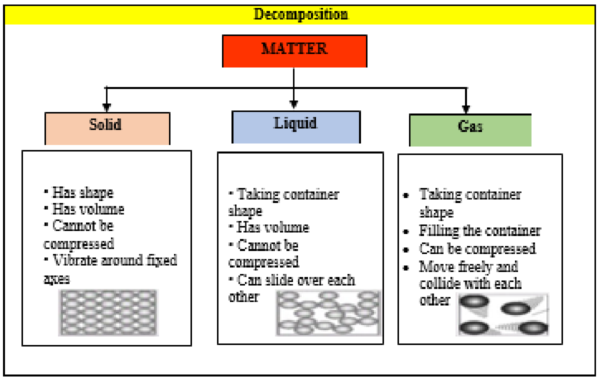


Figure 4. Decomposition skills applied by students

In this study context, the algorithm, decomposition, and evaluation skills show a better improvement after the CT application because the activities carried out in the T&L integrate these skills. Similarly to achievement in Science, abstract concepts can be understood by students through visualization with Scratch programming.

5. CONCLUSION

This study provides an opportunity for other researchers to integrate CT skills plugged in, particularly through Scratch programming. In the teaching process, teachers can integrate CT skills for all subjects, especially Sciences. The findings show that the integration of skills can significantly improve Science achievement. Even though it is not significant for CT skills, the mean score shows an increase for students who are involved with the integration. In integrating CT skills, it is necessary to focus on algorithm skills because it can improve the overall CT better than other skills. Therefore, policymakers in the education system in Malaysia need to provide a suitable module for all subjects to integrate CT skills.

CT skills are very much needed by today's generation and need to be integrated into the education system to help students systematically solve problems. Through the integration of CT skills in Science, 21st-century skills can also be instilled. However, to help educators with CT skills, especially those involving programming, it is necessary to collaborate with other institutions such as universities and polytechnics so that programming skills can be better understood quickly.

6. REFERENCES

Adbo, K. & Taber, K.S. 2009. Learners' Mental Models of the Particle Nature of Matter: A study of 16-year-old Swedish science students. *International Journal of Science Education* 31(6): 757–786.

Angeli, C., Voogt, J., Fluck, A. & Webb, M. 2016. A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. . . *Educational Technology & Society* 19(3): 47–57.

Atmatzidou, S. & Demetriadis, S. 2016. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems* 75: 661–670.

Basawapatna, A., Repenning, A., Koh, K.H. & Nickerson, H. 2013. The Zones of Proximal Flow: Guiding Students Through a Space of Computational Thinking Skills and Challenges 67–74..

Basu, S., Biswas, G. & Kinnebrew, J.S. 2017. Learner modelling for adaptive scaffolding in a computational thinking-based science learning environment. *Research Gate*

Bicer, A., Navruz, B., Capraro, R.M., Capraro, M.M., Oner, T. & Boedeker, P. 2015. STEM schools vs. Non-STEM schools: Comparing students' mathematics growth rate on high-stakes test performance. *International Journal on New Trends in Education and Their Implications* 6(1): 138–150.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A. & Engelhardt, K. 2016. Developing Computational Thinking in Compulsory Education: Implications for Policy and Practice. Report No. . Luxembourg

Cokelez, A. & Dumon, A. 2005. Atom and molecule: Upper secondary school French students' representations in long-term memory. *Chemistry Education Research and Practice* 6(3): 119–135.

Csizmadia, A., Curzon, P., Humphreys, S., Ng, T., Selby, C. & Woollard, J. 2015. *Computational thinking: A guide for teachers*. United Kingdom: Computing At School.

Dagiene, V. & Stupuriene, G. 2016. Informatics concepts and computational thinking in K-12 education: A Lithuanian perspective. *Journal of Information Processing* 24(4): 732–739.

Faber, H.H., Wierdsma, M.D.M., Doornbos, R.P. & Van Der Ven, J.S. 2017. Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network* 12: 13–24.

Garnett, P.J. & Treagust, D.F. 1992a. Conceptual difficulties experienced by senior high school students of electrochemistry : Electric circuits and oxidation- reduction equations Conceptual Difficulties Experienced by Senior High School Students of Electrochemistry : Electric Circuits and ox. *Journal of Research in Science Teaching* 29(2): 121–142.

Garnett, P.J. & Treagust, D.F. 1992b. Conceptual Difficulties Experienced by Senior High School Students of Electrochemistry : Electrochemical (

Galvanic) and Electrolytic Cells. *Research in Science Teaching* 29(10): 1079–1099.

Grover, S. & Pea, R. 2013. Computational thinking in K–12: A review of the state of the field. *Journals.Sagepub.Com* 42(36): 38–43.

Hesse, F., Care, E., Buder, J., Sassenberg, K. & Griffin, P. 2015. A Framework for Teachable Collaborative Problem Solving Skills. In P. Griffin & E. Care (eds.). *Assessment and Teaching of 21st Century Skills, Educational Assessment in an Information Age*. pp. 37–57. Mebourne, Australia.

Kalelioğlu, F. & Gülbahar, Y. 2014. The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education* 13(1): 33–50.

Kamisah, O. & Lee, T.T. 2013. Impact of interactive multimedia module with pedagogical agents on students' understanding and motivation in the learning of electrochemistry. *International Journal of Science and Mathematics Education* 12(2): 395–421.

Korkmaz, Ö. & Oluk, A. 2016. Comparing students' Scratch skills with their computational thinking skills in terms of different variables. *I.J. Modern Education and Computer Science* 11: 1–7.

Mannila, L., Dagiene, V., Demo, B. & Settle, A. 2014. Computational thinking in K-9 education. *Dl.Acm.Org*

Ministry of Education Malaysia (MOE). 2016. *Basics Computer Science: Standard curriculum and assessment document*. Putrajaya.

Moreno-León, J., Robles, G. & González, M.R. 2015. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED-Revista de Educación a Distancia.Número* 46: 1–23.

Nike Kusuma, W., Prayitno & Fauziatul, F. 2016. Guru Kimia Pada Topik Struktur Atom Menggunakan Instrumen Diagnostik Two-Tier. *Jurnal Pembelajaran Kimia* 01(2): 38–41.

Nikiforos, S., Kontomaris, C. & Chorianopoulos, K. 2013. MIT Scratch: A powerful tool for improving teaching of programming. In *Conferrence on Informatics in Education 2013*.

Özmen, H. & Ayas, A. 2003. Students' Difficulties In Understanding Of The Conservation Of Matter In Open And Closed-System Chemical Reactions. Report No.

Robertson, S.I. 2001. *Problem-Solving*. First. United Kingdom: Psychology Press.

Rodriguez, B., Rader, C. & Camp, T. 2016. Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment. In *ITiCSE '16, July 09-13*. pp. 95–100. Arequipa, Per: ACM.

Salmiza, S. & Haslinda, A. 2015. Kajian mengenai kefahaman pelajar tingkatan empat terhadap konsep asas jirim dan zarah. *Asia Pacific Journal of Educators and Education* 30: 89–104.

Saltan, F. & Kara, M. 2016. ICT Teachers' Acceptance of "Scratch" as Algorithm Visualization Software. *Higher Education Studies* 6(4): 146–155.

Samri, C., Kamisah, O. & Nazrul, A.N. 2021. Impact of the Plugged-in and Unplugged Chemistry Computational Thinking Modules on Achievement in Chemistry 17(4).

Selby, C.C. & Woollard, J. 2010. Computational thinking: The developing definition. In *Conference'10*. City, State, Country.: ACM

Seneviratne, O. 2017. Making Computer Science Attractive to High School Girls with Computational Thinking Approaches: A Case Study Oshani. In P.J. Rich & C.B. Hodges (eds.). *Emerging Computational and Policy on Research, Practice, Thinking.Educational Communications and Technology: Issues and Innovations*. pp. 21–32. Cham, Switzerland: Springer International Publishing AG.

Siti Hendon, S.A. 2016. Transforming Science Teaching Environment for the 21st Century Primary School Pupils. *Malaysian Online Journal of Educational Technology* 4(4): 68–76.

Su, A.Y.S., Yang, S.J.H., Hwang, W., Huang, C.S.J. & Tern, M.-Y. 2014. Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy. *British Journal of Educational Technology* 45(4): 647–665.

Tsarava, K., Moeller, K., Butz, M., Trautwein, U., Pinkwart, N. & Ninaus, M. 2017. Training computational thinking: Game-based unplugged and plugged-in activities in primary school. *Research Gate* 687–695.

Webb, M., Davis, N., Bell, T., Katz, Y.J., Reynolds, N., Chambers, D.P. & Sysło, M.M. 2017. Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies* 22(2): 445–468.

Wing, J.M. 2006. Computational Thinking. *Communications of The ACM* 49(3): 33–35.

Yadav, A., Hong, H. & Stephenson, C. 2016. Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends* 60: 565–568.

Yukselturk, E. & Altiok, S. 2016. An investigation of the effects of programming with Scratch on the preservice IT teachers ' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*.

SELLAM - an Interactive Educational Platform for Children to Learn Programming Concepts

Lakshitha Wijesinghe[1], Vinura Perera[2*], Erandi Herath[3*], Prabodha Abeynayake[4*]
[1,2,3,4] Department of Computer Engineering, University of Sri Jayewardenepura, Sri Lanka
lakshithaprabath@sjp.ac.lk, vinuraperera2@gmail.com, erandikghs@gmail.com,
pyabeynayake@gmail.com

ABSTRACT

Learning computer programming is becoming increasingly important for children in today's digital world. There are many platforms available for kids to learn computer programming concepts from a young age. However, the challenge lies in finding a self-learning tool that suits the preferences of students. Many existing platforms have limitations and are not always effective in imparting knowledge to children. According to our survey, the results indicated that existing platforms have limitations in catering to the preferences of students. Moreover, there were challenges in using these platforms in school classrooms, which made it difficult for teachers to monitor and assist students effectively. As a solution, we propose a new, game-based, self-learning, interactive platform called "SELLAM" for children to learn computer programming concepts. The platform also includes self-learning materials for students to guide them through concepts. First, we incorporated the students' feedback acquired through the survey to design the framework according to their preferences. Second, we proposed three games with unique features to address the aforementioned limitations. Finally, we evaluated our platform in comparison to other platforms based on student feedback. "SELLAM" received positive responses for its gradually increasing complexity, simple control buttons, engaging game characters, graphics, and challenging tasks that require logical thinking.

KEYWORDS

Game-based Learning, Computational Thinking, Computer Programming

1. INTRODUCTION

The demand for the design of computer applications or programs is growing. In the US alone, the predicted percentage change in employment from 2021 to 2031 is 25%. This is much larger than the average expected change (5%) for all other occupations (U.S. Bureau of Labor Statistics, 2022). The recent Information and Communications Technology (ICT) skill gap analysis report by the ICT Industry Skills Council (ICTISC), Sri Lanka also highlights the importance of continuous growth of employment in the sector (ICT Industry Skills Council, 2021). To minimize this gap, it is important to motivate school children to pursue a career in Information Technology. One way to address this is to introduce Computational Thinking (CT) concepts directly to the school curriculum. However, finding skilled educators to deliver such content remains a challenge. Particularly, for developing countries such as Sri Lanka, this is even more challenging. To mitigate this effect, it is important to introduce self-learning tools and automate the learning process as far as possible. This paper explores the possibility of helping school children in Sri Lanka to discover and cultivate a passion for computer programming.

Coding games are an effective tool to teach Computational Thinking (CT) to students (Zhang, Wong, & Chan, 2022). CT is identified as a fundamental skill that every student should possess, despite not choosing to work in computer science in the future (Lodi & Martini, 2021). In the literature, there are two types of environments named open task environments and goal-oriented environments to practice CT and coding practices. For instance, several European countries use open-task environment-based tools such as ScratchJr to drive their initiatives for computer science (Bers, 2018).

There are many platforms in the market to teach programming concepts to children through games. We have studied over twenty-five platforms to recognize their limitations through user feedback and reviews. Among the available platforms, most of the apps are not programming language oriented. They mainly focus on teaching basic programming concepts to children. From game-based platforms, only a few have multiple games. Also, almost all the games include most common programming concepts such as loops, conditional statements, and functions. All the platforms we have studied support the common mobile Operating Systems such as Android and iOS while some of them additionally support Windows and MacOS.

We found several limitations in the available platforms through a background survey. Some games lack enough examples to learn the concepts. Further, some have monotonous levels that do not allow children to have a challenge. Furthermore, some of the games had poorly designed interfaces, which adversely affected the learning experience. Some platforms had a huge amount of content that a child was unable to bear. Most of the games were developed without considering the children's ages. Hence, some of the games didn't have suitable difficulty levels relevant to each age. Finally, players didn't contribute much to solving the problems which violate the purposes of those gaming platforms.

After identifying all the limitations in the currently available learning platforms, we have designed an interactive platform called "SELLAM" to address those limitations.

Table 1. Aimed programming concepts and CT components in the games of "SELLAM"

Games	Programming concepts	CT components
Bunny Hops	Logical thinking, Loops, Conditional statements, Problem-solving	Decomposition, Abstraction, Algorithm Design, Pattern Recognition
Arrange Beads	Pattern identifying, Nested loops and nested Conditional statements, Basic mathematical logics for coding, Variables	Decomposition, Pattern Recognition, Abstraction, Algorithm Design
Escape World	Conditional statements, Logical thinking, Debugging and simple Algorithms, Functions	Algorithm Design, Abstraction

Figure 1. Interface of the game "Bunny Hops"

2. METHODOLOGY

2.1. *School Survey*

First, we conducted a survey on school children from Sri Lanka. The new curriculum in the Sri Lankan education system offers Information and Communication Technology from grade 6-11 (Educational Publications Department.,2018) where it is a core subject from grade 6-9. For grade 10 and 11, the subject is optional. The curriculum of Information and Communication Technology offers lessons on basic programming concepts from grade 7 onward and Pascal programming in grade 11.

Our team conducted a survey on students to identify the practical issues in delivering the curriculum. The questionnaire is attached to appendix 1. We collected data from 219 participants from grade 6 to 11.

2.2. *Game Design*

For the game designs, we used Unity platform, and the programming in Unity is done with C#. We used Adobe Photoshop for graphics design. Our mobile application consists of three main games named, "Bunny Hops", "Arrange Beads" and "Escape World". All three games focus on teaching computer programming concepts and CT components to children with different strategies (Table 1).

The first game is "Bunny Hops" (Figure 1) which is a source-to-destination based game where the user has to move a bunny using the basic coding principles to reach the final destination while avoiding obstacles. The user can drag and drop control components to create the instruction sequence which moves the game character. This game helps the child to understand the idea of abstraction, where the child has to ignore unnecessary characteristics of surroundings in the game and find the most essential component, which is the "carrot" object, to reach the destination and win the game. And also, it teaches the concept of algorithm design where the child must follow a step-by-step procedure to reach the destination by dragging and dropping the items and identifying the blocks which they have to reach by properly analyzing the path. The instructions of this game consist of elementary movements such as moving in fundamental four directions, jump, and advanced concepts such as if-else conditions, loops, etc. The difficulty of the game increases and more advanced programming concepts are introduced while the child goes through the levels. So, this helps the child to learn the CT component of decomposition since each level is focusing on smaller problems which focus on certain concepts by increasing the complexity of the concepts when the levels are increasing. The last level of the game achieves the CT component of pattern identifying where the child has to use previous knowledge about all the concepts they have learned and use the knowledge about previous patterns to solve the problem.

The second game is "Arrange Beads" (Figure 2) which is built on the concept of identifying patterns. The game is based on a given pattern in each level where the user has to identify the pattern and come up with a logical way to simulate the same pattern. The uniqueness of this game is that the coding environment given to the user is organized into a suitable structure to provide a friendly environment for users where they could experiment with different inputs, simulate multiple times, experiment themselves, and reach a logical solution by trial and error. The complexity of the patterns and the complexity of the solution increases with each level. The game is organized into different levels with increasing difficulty in each level to give the users the experience in the decomposition CT component. The complex problem of identifying a core pattern and coming up with a logic to rebuild it is broken down into manageable pieces through levels. This game is mainly based on pattern recognition CT components. All the levels of the game provide the space for users to look for the patterns in the game and use their prior knowledge in solving similar problems for the current problem. The game is designed in such a way that each past problem is incorporated into the next levels to provide pattern recognition skills to the users. The game also has several control buttons and details that are not directly associated with the level to give the users the ability to identify the details that are relevant to solving the current problem and ignore the details that aren't relevant to the issue at hand. For each problem, the users are expected to come up with

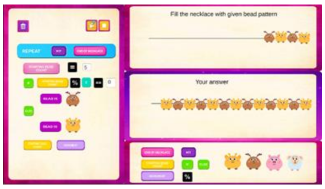


Figure 2. Interface of the game “Arrange Beads”

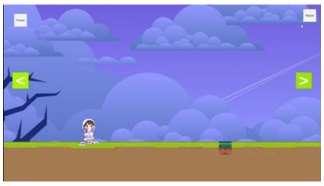


Figure 3. Interface of the game “Escape World”

step-by-step solutions to rebuild the given pattern. Thus, it is focusing on algorithm design CT components at each level.

The third game, “Escape World” (Figure 3) is a 2D platform-based game. The main character got lost in an unknown world, trying to find his way back home by overcoming the challenges and obstacles that he has to face while moving forward in the game. The player movement consists of basic two-dimensional physics-based movement. In order to advance forward in the game, the player has to go through checkpoints and solve problems that are based on computer programming concepts and CT components. It includes programming concepts such as if-else conditions, loops, etc. This game provides a fun and engaging experience to introduce computational thinking to young learners. At each checkpoint, the child has to perform certain tasks or program various objects that are found in the game by creating an instruction sequence using a block coding interface. A demonstration video of the platform is readily available for viewing in the appendix section, under reference 3.

3. RESULTS

3.1. *Background Survey Results*

For the survey, a total of 219 respondents participated. Out of these participants, one respondent's response was rejected and the grade 6 respondents (total of 38 respondents) were omitted from the results as they are not familiar with the programming or related concepts according to their ICT curriculum. This brings the total evaluated respondents to 180. Out of the total evaluated respondents, only half of the respondents were familiar with the terms ‘computer programming’.

Through the survey responses, we got to know the different modes of delivery Sri Lankan schools use to deliver ICT curriculum such as blackboard, whiteboard, video/audio, that could be considered as traditional modes of delivery.

Even Though the Ministry of Education and National Institute of Education (NIE) who are responsible for the ICT curriculum in Sri Lanka suggested the Scratch (Resnick, 2007) tool for students from grade 7 upwards to learn programming concepts, only a few students were familiar with Scratch. Moreover, majority of students were not familiar with Scratch, raising concerns about schools not conducting the lessons properly due to certain practical issues (Figure 4). The teachers and students stated several issues such as the initial complexity of understanding the platform and needing thorough guidance from the teachers while students are engaged in the learning platform.

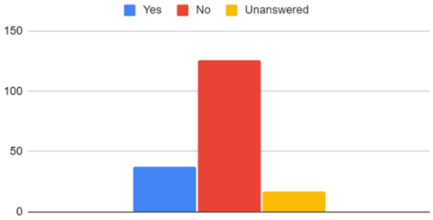


Figure 4. Students’ response to whether they are familiar with the Scratch tool

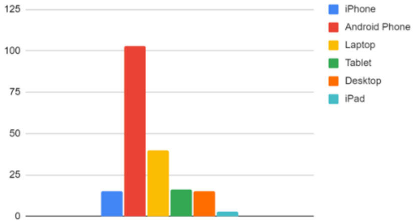


Figure 5. Student’s response to the personal devices they possess

Through the survey, we were encouraged to develop a mobile application as the students had a clear preference for mobile devices over any other electronic device (Figure 5).

3.2. *Student Feedback Survey Results*

After the development of the platform “SELLAM”, we conducted another survey to get feedback about the platform from students. We acquired 100 responses from students representing different schools in Sri Lanka. As we were aiming for a fair evaluation of our platform, we have also used Scratch and Code.org platforms to compare the performance of our platform under certain criteria by giving all the 3 platforms to the survey respondents. We maintained anonymity regarding the development of "SELLAM." to get unbiased results.

With the gathered feedback from the students, the majority of the students (70%) believe that “SELLAM” would be helping them to understand complex concepts in ICT textbooks easily (figure 6).

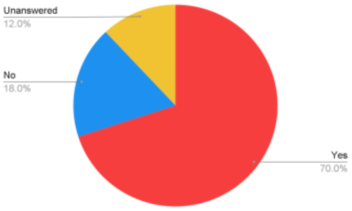


Figure 6. Students’ preference about the platforms that would help them to understand complex concepts in ICT textbooks easily

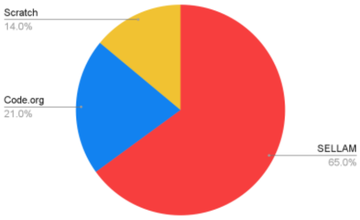


Figure 7. Students' preference about the platforms after trying out multiple levels in each

After trying out multiple levels in each platform, the "SELLAM" turned out to be the platform they liked the most, leading with 65% majority, and Scratch and Code.org received 14% and 21% respectively.

According to the student feedback, "SELLAM" was able to lead in several aspects compared to the other two platforms such as the most fun and enjoyable platform to learn programming concepts (50%), the platform with the simplest instructions that are easy to understand (55%), the easiest platform to handle (56%), most eye-catching graphics (69%) and most pleasant color blending (70%). When considering the student preference to use the app repeatedly, Scratch leads "SELLAM" by 8% meanwhile Code.org had the least preference. Students mostly preferred to use Scratch to learn ICT subjects (42%) meanwhile "SELLAM" was the second choice (37%). We can argue that this decision might be biased since Scratch is currently used in the Sri Lankan ICT curriculum (Appendix 2).

"SELLAM" received positive responses for its gradually increasing complexity, simple control buttons, engaging game characters and graphics, and challenging tasks that require logical thinking. They found Scratch and Code.org interesting due to the sequential codes, sound effects, and movements, but the excessive use of blocks and the difficulty in understanding the games, were the drawbacks.

"SELLAM" platform is an effort to address the limitations of the current existing platforms. It addresses the issue of most platforms not focusing on educating about programming languages. Though "SELLAM" is also not explicitly educating about programming languages, the "Arrange Beads" game tries to incorporate variables, direct keywords used in programming language and mathematical logic used in simple programs such as odd and even number finding. Each game in "SELLAM" is arranged into different levels with each level introducing a new programming concept as well as visible difficulty compared to previous levels, providing the users a challenge to conquer each level. The content in "SELLAM" was designed and developed after consulting well-experienced educationalists in the ICT field who had years of experience in teaching students about the domain. Therefore, the content was kept at a sufficient level for students to learn programming concepts without adding overwhelmingly large content that is unable to be grasped by students. The interfaces of "SELLAM" were developed under the consultation of experts as well as thoroughly tested with users for their feedback. The interface was structured in a way such that it was engaging as well as easy to handle by the users.

DISCUSSION

This paper introduces a new game-based interactive learning platform for kids to learn computer programming concepts by targeting multiple CT components. The developed platform addresses the existing limitations of currently available platforms and the feedback gathered from the students shows that our platform addresses several limitations that we identified from existing learning tools. We have received feedback which enabled us to identify certain limitations within our platform. These limitations comprise a restricted coding space, arduous navigation on small mobile devices, and the imperative for more levels to be incorporated. As future work, those limitations would be addressed in the "SELLAM" platform to make it more user-friendly.

4. REFERENCES

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of scratchjr in Europe. *European Journal of STEM Education*, *3*(3). https://doi.org/10.20897/ejsteme/3868

Educational Publications Department. (2018). Educational Publications Department. Retrieved April 10, 2023, from http://www.edupub.gov.lk/

ICT Industry Skills Council. (2021). *Skill Gap Assessment of the ICT sector in Sri Lanka* (pp. 1-82) (Sri Lanka, ICT Industry Skills Council). Colombo, Western: ICT Skill Sector Council

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883–908. https://doi.org/10.1007/s11191-021-00202-5

Resnick, M. (2007). *Imagine, program, share*. Scratch. Retrieved April 13, 2023, from https://scratch.mit.edu/

U.S. Bureau of Labor Statistics. (2022, September 8). *Computer and Information Technology Occupations : Occupational Outlook Handbook*. U.S. Bureau of Labor Statistics. Retrieved April 13, 2023, from https://www.bls.gov/ooh/computer-and-information-technology/home.htm.

Zhang, S., Wong, G. K., & Chan, P. C. (2022). Playing coding games to learn computational thinking: What motivates students to use this tool at home? *Education and Information Technologies*, *28*(1), 193–216. https://doi.org/10.1007/s10639-022-11181-7

5. APPENDIX

1. https://docs.google.com/document/d/1Q9UM51rar6ZKKRG8HIN4Egn3lB5oZRS2HVarV0jutdI/edit
2. https://docs.google.com/document/d/1ba9FXigtxoAP4Swg0GPLL5CDU-7NUJJyW1rCiJEfvyI/edit?usp=sharing
3. https://youtu.be/JIUjLippwd0

Integrating Engineering Design Process into After-Class STEM Learning in Secondary Education: A Case Study of Two Winning Groups

Biyun HUANG[1]*, Morris Siu-Yung JONG[1,2], Chun-Yu HOU[1]
[1] Centre for Learning Sciences and Technologies, [2]Department of Curriculum and Instruction,
The Chinese University of Hong Kong, Hong Kong
lucyhuang99@cuhk.edu.hk, mjong@cuhk.edu.hk, violettahou@cuhk.edu.hk

ABSTRACT

Learners' engagement in the engineering design process allows them to develop problem-solving, communication, and other 21st century skills. However, there is a need for more research to explore how the engineering design process can be integrated into K-12 STEM education, particularly in the development of open-ended design products. In response, we conducted a case study to probe into what design activities students would experience when developing a prototype 2.0 (i.e., similar to a real product) and how they would perceive their learning experience. Analysis of student learning journals revealed that students engaged in various activities such as planning, searching, making, and simulating. These activities offered students a comprehensive understanding of the engineering design process and stimulated their strong interest correspondingly. It is suggested that, in the course of STEM learning, more opportunities should be provided for empowering students to actualize design solutions and unleash their interest and potential in design.

KEYWORDS

STEM, engineering design process, case study, learning journal, maker activities

1. INTRODUCTION

Engineering design is a process of using intellect to design solutions (Dai et al., 2023), where the designer evaluates the problem, proposes a solution, defines the product specification, and, through iterative testing and adjustment, comes up with a solution that satisfies the user (Dym et al., 2015). This process often involves complex decision-making activities and close collaborative interaction within the team (Brown, 2008; Lee & Ostwald, 2020). Incorporating the engineering design process in K-12 classrooms would advance students' skills in problem-solving, communication and teamwork, and other 21st century skills (Koh et al., 2015; Lin et al., 2020). Students would develop the essential sets of skills through the activities of analytical thinking, visualization, and testing (Arık & Topçu, 2020; Dam & Siang, 2020). Despite the growing interest in integrating the engineering design process into K-12 classrooms, literature on how to implement it in in-class or after-school programs is limited (Crismond & Adams, 2012; Simarro & Couso, 2021).

2. LITERATURE REVIEW

More empirical research is needed to understand how engineering design activities can be effectively incorporated into K-12 STEM education (Long et al., 2020). More case studies are in demand to develop a deeper understanding of how students engage with the activities and perceive their experiences (Shé et al., 2022). In recent years, Zhou et al. (2017) organized a two-week toy design workshop for secondary school students and found that students improved their self-efficacy in engineering design. English (2019) introduced engineering design practice into an elementary class where students use sketches to design their shoes and found that students improved their knowledge and design capacity after engaging in design activities. In our study, we will focus on the prototype stage, during which students transform their design ideas into prototype 2.0 (i.e., similar to real products). We intend to understand what students "do" and "know" (Kolodner, 2002; Yata et al., 2020) and to understand how they perceive their experience in the prototyping process. Based on students' learning journals, the following research questions are proposed:

(1) What were the main activities students engaged in during the engineering design process?

(2) What was the main technical knowledge students learned during the engineering design process?

(3) How did students perceive their experience in the engineering design process?

3. RESEARCH CONTEXT

There are varied models of the engineering design process, but their core elements are similar, such as generating ideas, making prototypes, and making iterative improvements to the prototypes (Arık &Topçu, 2020). In our case, students would experience three learning phases (see Figure 1). In the first phase, all grade 8 students participated in the ideation activity, during which the students learned disciplinary knowledge and proposed design ideas for helping improve the lives of community citizens. In the second phase, selected students with good performance in the first phase developed their idea into prototype 1.0 with cardboard and circuits. In the third phase, the winning groups of the second phase had the opportunity to advance their design solutions into prototype 2.0. Phase 3 learning was conducted in the format of an after-class program, where students developed their product with the support of mentors and industry experts.

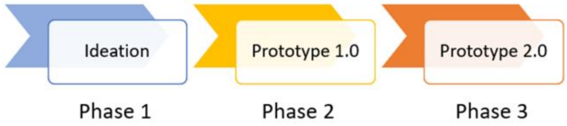


Figure 1. The three learning phases

In this study, we selected two winning groups as case studies (Yin, 2002). To answer the research questions, we collected and analyzed the contents of students' learning journals. In the learning journal, students recorded their progress in prototyping and their feelings about participating in the activities. The learning journals were analyzed using qualitative content analysis (Patton, 2002). Consent for participating in the research has been obtained prior to the study. The two winning groups consist of three students each. One group designed an air purifier that can be used in small spaces such as cars and sub-divided rooms. The other group worked on smart diapers that can be used in homes with infants and nursing homes.

4. RESULTS

RQ1 What were the main activities students engaged in during the engineering design process?

The analysis of students' learning journals showed that students were involved in a wide range of activities. As shown in Table 1, under the category of "do," students performed a series of activities, such as "plan," "search," "purchase," "use/apply," "think," "draw," "make," "assemble," "test," and "discuss." Students can develop a more integrated view of the typical industrial manufacturing processes through these sessions. They understood how a product's development progresses from "zero" to "one." Students had different focuses at various stages of product development. In the early stages, they considered the conceptual model of product development, and in the middle stages, they considered specific implementation details, such as the placement of screws. In the later stages, they thought about further improving the product, such as reducing the air purifier's noise. See Figure 2 for an example of the conceptual model and final product.

Table 1. Main activities and knowledge involved in the engineering design process

Categories	Sub-categories	Examples
Do	Plan	e.g., plan the blueprint
	Search	e.g., search for the required materials
	Purchase	e.g., purchase the filter
	Use/apply	e.g., use the breadboard
	Think (conceptualize; compare & decide)	e.g., complete the preliminary conceptual model e.g., consider the placements of screws e.g., consider how to make the fan less noisy
	Draw	e.g., 3D drawing e.g., circuit diagrams
	Calculate, measure	e.g., calculate the lengths and distances e.g., calculate the position of parts
	Make	e.g., make a bracket to support the UV lamp
	Estimate	e.g., estimate the position of the UV lamp
	Assemble/ connect	e.g., assemble and connect the circuit
	Simulate (integration)	e.g., simulate the working process through software
	Test	e.g., test the operation of each part of the product
	Discuss	e.g., discuss how to improve the product e.g., discuss the pros and cons of different placements of the components
	Redo	e.g., redraw 3D drawings
	Improve	e.g., improve the 3D drawing e.g., improve the appearance
Know	3D drawing	e.g., know how to do 3D drawing
	Electronic & engineering knowledge	e.g., know the AC and DC circuits e.g., know the voltage of each electrical appliance e.g., know how to connect circuits

Figure 2. Example of conceptual model and final product

RQ2 What was the main technical knowledge students learned during the engineering design process?

Under the category "know," students learned knowledge on "3D drawing" and "electronic and engineering knowledge." For example, they mentioned that they learned how to connect electric circuits. See Table 1. As this phase was more about hands-on and learning by doing, students talked more about the "do" part and less about the "know" aspect.

RQ3 How did students perceive their experience in the engineering design process?

The students described their moods using keywords and short sentences in the learning journal. At the beginning stage, after learning the components of a circuit diagram and searching for the product materials, one group wrote the keywords "enjoy, look forward to," which indicates that the group liked the activity and was full of expectations for the subsequent activities. In the middle stage, when the group completed the 3D drawing of the internal design and started to develop the product's outer shell, they described it as an "excited, flow state." When the product was further improved, and the students assembled the filter and switch, they said, "It was a new look." Moreover, they shared, "If you have persistence, the iron pillar will become a needle." At the final stage of product completion, the students wrote, "I am overjoyed." They further described, "It is not about getting a job done; it is about getting a job well done." Overall, the students were delighted and enthusiastic about all the learning activities. They not only completed their prototype 2.0 but also developed their cognitive capacity. They appreciated the process of growth.

5. CONCLUSION AND DISCUSSION

The study explores integrating the engineering design process into K-12 after-class programs. By analyzing the learning journals of two winning groups, we found out what the students went through in the product development process and how they perceived the experience of each session. Through this hands-on session, students generally explored and witnessed how their product ideas were gradually shaped and transformed. Through this process, students understood the process of engineering design, learned the knowledge needed to develop products, and accumulated experience in analysis and thinking (Dam & Siang, 2020). In the future, we can provide more opportunities for students interested in engineering design and those who do not know much about this field with more hands-on opportunities and discover their interests and potential.

6. ACKNOWLEDGEMENTS

The work was substantially supported by The Hong Kong Jockey Club Charities Trust (Project Title: Jockey Club Community Care and STEM in Action [Project S/N Ref: JC 2019/0112]).

7. REFERENCES

Arık, M., & Topçu, M. S. (2020). Implementation of engineering design process in the K-12 science classrooms: Trends and issues. *Research in Science Education*. https://doi.org/10.1007/s11165-019-09912-x

Crismond, D. P., & Adams, R. S. (2012). The informed design teaching & learning matrix. *Journal of Engineering Education-Washington*, 101(4), 738 -797.

Dai, Y., Liu, A., Qin J., Guo, Y., Jong, M. S. Y., Chai, C. S., & Lin, Z. (2023). Collaborative construction of artificial intelligence curriculum in primary schools. *Journal of Engineering Education, 112*(1), 23-42.

Dam, R. F., & Siang, T. Y., (2020). *What is design thinking and why is it so popular*?. Interaction Design Foundation.

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of engineering education*, *94*(1), 103-120.

English, L. D. (2019). Learning while designing in a fourth-grade integrated STEM problem. *International Journal of Technology and Design Education*, 29(5), 1011-1032.

Koh, J. H. L., Chai, C. S., Wong, B., & Hong, H. Y. (2015). Technological pedagogical content knowledge (TPACK) and design thinking: A framework to support ICT lesson design for 21st century learning. *The Asia-Pacific Education Researcher*, 24(3), 535–543.

Kolodner, J. L. (2002). Learning by Design™: Iterations of design challenges for better learning of science skills. *Cognitive Studies*, 9(3), 338–350.

Lin, L., Shadiev, R., Hwang, W. Y., & Shen, S. (2020). From Knowledge and skills to digital works: An application of design thinking in the information technology course. *Thinking Skills and Creativity*, https://doi.org/10.1016/j.tsc.2020.100646

Long, N. T., Yen, N. T. H., & Van Hanh, N. (2020). The role of experiential learning and engineering design process in K-12 STEM education. *International Journal of Education and Practice*, 8(4), 720-732.

Patton, M.Q. (2002). *Qualitative research and evaluation methods*. Sage.

Shé, C. N., Farrell, O., Brunton, J., & Costello, E. (2022). Integrating design thinking into instructional design: The# OpenTeach case study. *Australasian Journal of Educational Technology*, 38(1), 33-52.

Simarro, C., & Couso, D. (2021). Engineering practices as a framework for STEM education: A proposal based on epistemic nuances. *International Journal of STEM Education*. https://doi.org/10.1186/s40594-021-00310-2

Yata, C., Ohtani, T., & Isobe, M. (2020). Conceptual framework of STEM based on Japanese subject principles. *International Journal of STEM Education*, 7. https://doi.org/10.1186/s40594-020-00205-8

Yin (2002). *Case study research: Design and methods* (3rd ed.). Sage.

Zhou, N., Pereira, N. L., George, T. T., Alperovich, J., Booth, J., Chandrasegaran, S., Tew, D. M., & Ramani, K. (2017). The influence of toy design activities on middle school students' understanding of the engineering design processes. *Journal of Science Education and Technology*, 26, 481-493.

Computational Thinking in Elementary Classrooms: Teachers' Understanding of Computational Thinking Practices and Integration

Victoria MACANN
Massey University, New Zealand
macannvi@msu.edu,

Ceren OCAK
Georgia Southern University, USA
cocak@georgiasouthern.edu,

Aman YADAV
Michigan State University, USA
ayadav@msu.edu

ABSTRACT

Computational thinking is viewed as an integral part of the education system as it will be instrumental in almost all future professions. However, there is still a need to support teachers in understanding what CT is, as well as how to integrate it into their everyday teaching and current curriculum. This study was conducted as part of a larger NSF-funded professional development project to support elementary teachers to integrate CT into their mathematics and science classrooms. In this study, we worked with 22 teachers who taught grade levels ranging from grade two to grade five. Our findings revealed how in-service teachers understand five major CT practices and how they connect those practices to their current practice.

KEYWORDS

Computational Thinking, teacher professional development

1. INTRODUCTION

Computational thinking (CT) integration throughout curriculum areas prepares students not only for the future workforce, but also increases their understanding of disciplinary skills, creativity and general problem-solving (Angeli et al., 2016, Ketelhut et al., 2020; Pollock et al., 2019; Mishra & Yadav., 2013; Yadav et al., 2017). Although there has been recent push on how to integrate CT into teaching practice for pre-service teachers in teacher education programmes (e.g., International Society for Technology in Education (ISTE), the American Computer Science Teachers Association (CSTA), the Next Generation Science Standards (NGSS), we are just starting to understand how in-service teachers see the role of CT; the need is even more prominent regarding how in-service teachers connect CT to their current practice (Rich et al., 2019; Yadav et al., 2017). In this study, our purpose is to identify teachers' understanding of five major CT practices and their perceptions of how CT connects to their current teaching practice. Therefore, we asked: How do elementary teachers see connections between CT integration and their disciplinary teaching at the elementary level?

2. PERSPECTIVES

According to Angeli et al., (2016) and Ketelhut et al., (2020) how to think computationally (content knowledge) and how to teach to think computationally (pedagogical content knowledge) are vital in teachers' decisions for how they integrate CT into their teaching practice. Similarly, Hug et al., (2018) argue that teachers are more willing to integrate CT when their CT knowledge increases.
A challenge for CT integration is the continued lack of understanding about CT content knowledge and skills teachers need (Rich et al., 2019), which should be addressed in teacher professional development (PD) courses (Caskurlu et al., 2021). In order to develop PD programs that can support teachers to integrate CT into their classroom instruction, we need to understand how teachers understand CT and make connections between CT and their classroom practice. To that end, we designed and developed pre-professional development (PD) to provide elementary teachers with support that they need on CT integration, feeding their thoughts and ideas about computing integration back into the design of the PD in the recursive meetings of PD. We initially collected data from teachers' reflections in an asynchronous Google classroom that included three CT units that focused on three CT practices (algorithms and debugging, abstraction and decomposition, and pattern recognition) and engaged in conversations of how our world, disciplines, and students intersect with CT. For the purposes of this study, we analyzed participant's online responses in an asynchronous reflection thread, where the teachers commented on the five CT practices and their perceptions of how CT connects to their current practice.

3. METHOD

3.1. Study context

The study was a part of a larger NSF-funded professional development project to support elementary teachers to integrate CT into their mathematics and science instruction. Twenty-two elementary teachers from a Midwestern school district participated in the PD that included a combination of synchronous and asynchronous work to develop teachers' foundational understanding of CT as well as seed some initial ideas about CT integration in elementary classrooms. Teachers participated in face-to-face synchronous workshops to co-design CT-integrated math and science lessons. The teachers were also invited to join a month-long asynchronous Google classroom work in May 2022 and completed the units of work as described in Table 1. To help teachers develop foundational CT knowledge, the Google classroom included one CT unit that presented what CT is, and three CT practices units (algorithms and debugging, abstraction and decomposition, and pattern recognition). The final unit provided teachers with CT integration examples they could explore to see CT in action in a disciplinary context. Given our focus on how teachers see the relevance of CT to their classroom instruction, our analysis focused on teacher reflections in units two, three, and four.

Each unit also had an unplugged and plugged activity for the participants to put their new learning into practice. Once the teachers completed the activities from the units, they were asked to reflect on their understanding as comments added to each section. We added the teachers' comments to a spreadsheet and compiled key themes from all the comments from each unit. The key findings from each unit are presented in the results section. All data were kept confidential, and the participants' comments were presented anonymously.

Table 1. Elementary Teachers' CT Professional Development on Google Classroom

Unit	Tasks
What is Computational Thinking?	This unit focused on getting familiar with different CT definitions that exist in the current literature and how these different perspectives could be leveraged in K-12 education. Participants had the opportunity to discuss existing perspectives towards using computers to teach problem-solving through online discussions. The materials in this section helped teachers consider the core values that might inform the design of their integrated computational thinking activities such as equity & social justice, economic & workforce development.
Algorithms and Debugging	This unit focused on defining algorithms, where algorithms exist in current K-12 content, and understanding the difference between computational thinking and algorithmic thinking. Participants learned how they can find and fix errors in an algorithm or a computer program to ensure the plan of action runs as smoothly as possible.
Abstraction and Decomposition	This unit introduced how to think abstractly and allow complex ideas to come together without getting tied up in specific details. Decomposition is about breaking big challenges into smaller ones. The material in this section provided participants the opportunity to use decomposition in the context of their classrooms.
Patterns	Patterns are used to identify and utilize information, particularly with respect to how computer scientists can leverage these repetitions in the context of their coding solutions.
CT from a Disciplinary Perspective	This unit provided our teachers with discipline-specific CT integration examples such as CT in STEM, CT in Social Science, ELA, and the Arts. Through these different CT integration examples, our teachers had the opportunity to tailor and create their own CT integrated lessons so that we could build our conversations around how to bring computational thinking practices to diverse disciplinary areas.

3.2 Participants

There were 22 teachers involved in this study: two male and 20 female. The teachers taught grade levels ranging from grade two to grade five. The participants were based out of the Kentwood school district and were digital leaders in their school. This meant that they taught/supported other teachers with implementing integrated computational thinking activities in their classrooms.

3.3 Procedures

The participants completed five asynchronous Google classroom units of work. As stated previously, this study only focuses on teachers' reflections on the three units that covered the CT practice given our research goal. In the Google classroom we asked the teachers to reflect on their understanding of CT practices and how these ideas might play a role in your instruction. The participants addressed the following questions for each unit.

Below are the questions for unit two.

- Do you see ways to utilize algorithms or debugging in your instructional practices?
- How might you enhance your instruction through the inclusion of algorithms and debugging?
- Share your thoughts on your learning experience during the plugged activity?

4. RESULTS

Our analysis of participant's responses from Google classroom revealed how teachers' understood five CT practices. The results are detailed as follows and summarized in Table 2:

Algorithms and Algorithmic Thinking. Participants understood that algorithms need to be a clear set of instructions for students to follow, and that those instructions also need to be in a particular order to complete the task. The participants had a general view that algorithms do not always require a computer, and are mainly used in mathematics. Several participants mentioned that although algorithms seem initially intimidating, they are more approachable when broken down into individual steps and used in sequence for problem-solving.

Debugging. Participants understood that debugging is both finding and fixing problems, and debugging practices need to be explicitly taught by carefully analyzing each step one at a time rather than simply throwing out the whole problem. The participants valued debugging as a teaching strategy for those students who are not natural problem-solvers, and can be used as a powerful structure to see where we have failed, and to encourage students that mistakes are key for learning.

Abstraction. Several participants reflected on the key element of abstraction - focusing on only what matters and eliminating excess information. Participants also recognized that they already use abstraction in their classrooms across curriculum areas. An interesting point raised was recognizing when to use abstraction and when not to in order to encourage challenge/struggle for students.

Decomposition. Breaking a problem down into subproblems was perceived as a key learning strategy. The participants also reflected on how when using decomposition, encouraging students to keep track of each subproblem of similar ‘size’ is important as it provides a parameter when students think about breaking up the task.
Patterns. Several participants commented on how patterns are important in real-life/professions and not solely something learned at school. One participant provided an example of when she had used pattern recognition in spelling: a) What attributes do you notice that these words have in common? b) What do you think is the spelling rule that these words follow? c) Look at these words and use what you know about the special attributes/features of our words to see if they match or don’t match the pattern.

Table 2. Conceptions of CT Practices

CT Practice	Summary of Response
Algorithms and Algorithmic Thinking	• Algorithms are instructions that need to be precise and clear. • Algorithms need to be in a particular order to complete the task correctly. • Algorithms do not just apply to computers; they can be used everywhere. • Algorithms can be viewed as intimidating, but when explained as individual steps, students realize how approachable algorithms can be.
Debugging	• Debugging is finding as well as fixing problems. • We need to teach our learners to debug by carefully analyzing each step one at a time rather than simply throwing out the whole problem. • Convincing students to debug or refine their work is a challenge. • Debugging seems like an important term and a key strategy, especially for those who aren't natural problem-solvers. • Debugging can be used as a powerful structure to see where we have failed, and to encourage students that mistakes are key for learning.
Abstraction	• The key element of abstraction - focusing on only what matters and eliminating excess information. This may be separated down into smaller chunks of information too - what is important right now to solve this step of the problem? • Already using abstraction in your classes - across curriculum areas (e.g., guided reading, mathematics). • One aspect raised was when to use abstraction, and when not to (and encourage a little struggle!) • Several people mentioned the counterpoint of whether abstraction detracts from curiosity.
Decomposition	• “When using decomposition, we need to try to keep track of all the parts and that they are of similar ‘size’.” (Participant quote). “This provides more of a parameter when having learners think through breaking tasks down.” (Participant quote) • Six participants agreed that “the age that a child is introduced to computational thinking compared to waiting until he/she is an adult, greatly affects his/her success with understanding CT.”
Patterns	• Patterns are important in real-life/professions and not solely something learned at school. • There were many examples of using pattern recognition in mathematics activities: “Going from repeated addition to multiplication, “find the rule” problems with students, and

	also spelling and "Students need to learn and observe patterns in syllable patterns and letter sounds to become more efficient decoders so that they can focus on comprehension." (Participant quotes)

5. SCHOLARLY SIGNIFICANCE

Our results suggest that after being introduced to CT practices, elementary teachers made connections between CT integration and their classroom practice. Once the teachers made explicit links to CT vocabulary, they understood where CT practices were occurring currently in their own teaching practice. The teachers also saw how CT vocabulary can be utilized with their students as a way to encourage systematic problem-solving across a variety of disciplines and support their instruction. Although links were made across curriculum areas/disciplines, the teachers also appeared to be more comfortable with the application of CT in the context of mathematics due to a similar vocabulary. For example, algorithms and debugging are often thought of in mathematics as demonstrated by participant comments "I feel like we mostly hear the term algorithm in math. Interesting to think about how it applies to all subjects," and, "this reminds me of times that the Math in Focus book gives an example of a student that has made a mistake and asks the learner to identify the mistake and fix it (or debug!)"

The potential influence of integrating CT across curriculum areas has implications for future CT research and professional development for elementary teachers. A focus on developing teachers' understanding of CT integration into subjects other than math, would help clarify the degree of problem-solving inherent to CT across all curriculum areas. In addition, teacher reflections suggest that they see CT as being a valuable metacognitive strategy to support their disciplinary instruction as well as student learning. A recent article also suggested that CT could be used as a pedagogical tool to support teachers to explicitly teach metacognitive strategies, which are often not taught in elementary classrooms and are important for student learning (Yadav, Ocak, & Oliver, 2022).

6. REFERENCES

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society*, *19*(3), 47-57. https://www.jstor.org/stable/10.2307/jeductechsoci.19.3.47

Caskurlu, S., Yadav, A., Dunbar, K., & Santo, R. (2021). Professional development as a bridge between teacher competencies and computational thinking integration. In Computational Thinking in Education (pp. 136-150). Routledge. https://doi.org/10.4324/9781003102991-8

Hug, D. S., Eyerman, D. S., Cota, R., & Pontelli, D. E., (2018). *Embedding K12 professional development through co-teaching experiences: Sustaining computational thinking in interdisciplinary courses.* 2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology, RESPECT 2018 - Conference Proceedings. https://doi.org/10.1109/RESPECT.2018.8491708

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology*, *29*(1), 174-188. https://doi.org/10.1007/s10956-019-09798-4

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Mishra, P., & Yadav, A. (2013). Rethinking Technology & Creativity in the 21st Century. *TechTrends, 57*(3), 10-14. https://doi.org/10.1007/s11528-013-0655-z

Pollock, L., Mouza, C., Guidry, K. R., & Pusecker, K. (2019). *Infusing Computational Thinking Across Disciplines: Reflections & Lessons Learned.* Proceedings of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN, USA. https://doi-org.ezproxy.massey.ac.nz/10.1145/3287324.3287469

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education, 27*(2), 165-205. https://www.learntechlib.org/primary/p/207487/

Wing, J., M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717. https://doi.org/10.1098/rsta.2008.0118

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE), 14*(1), 1-16. https://doi.org/10.1145/2576872

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer. https://doi.org/10.1145/2994591

A Study of Students' Computational Thinking and Complex Problem-Solving Skills in the Summit Game

Yu-Tang Hung[1], Ju-Ling Shih[2*], Tsung-Yen Chuang [3]
[1,3] Department of Information and Learning Technology, National University of Tainan, ROC (Taiwan)
[2]Graduate Institute of Network Learning Technology, National Central University, ROC (Taiwan)
[1] tonycafa@hotmail.com, [2*]juling@cl.ncu.edu.tw,
[3]chuangyen@mail.nutn.edu.tw

ABSTRACT

Many people, nowadays, suppose that the structure of a computer system is the simulation of the mental activity or process of thinking of human beings. Throughout history, most problems are set about not with the algorithm or computer system but via anthropogenic procedures such as looking for the crux of the problem, trial and error, verification of the solution and so on. However, with the growing development of information technology in recent years, many educational institutions have cultivated the act or process of thinking in the computational thinking (CT) courses for learners in only computer science and information engineering; they ignore the fact that we need to solve many problems in the social issues through CT and better teamwork to figure out the most appropriate solutions.

This study aims to integrate CT into the course design and practice it in complex problem-solving for the high school learners. Meanwhile, the researchers also explore the demonstration of CT skills by investigating the difference in the students' performance of problem-solving after the instruction of CT. The online concept mapping serves as a tool to guide the students' CT structure, and by following the steps, they are expected to solve the social problem in a certain scenario. The result indicates that with a CT-structure as a scaffold, these learners can easily spotlight the cause and the effect of the social issue, then communicate with their counterparts in a rational way, and thus create a win-win strategy for solving the social problems.

KEYWORDS

computational thinking, social issue, concept map

1. INTRODUCTION

CT is an essential competency and shall be looked upon as one of the elemental skills to learn in our century. (Barr et al., 2011; Curzon, 2016) Most of the time, when people encounter the challenges, difficulties, problems, or just the office routine, they usually take a series of measures such as analyzing the straits, finding resources, consulting someone's opinion, prioritizing the steps, making the decision, investigating the feasibilities, or just adhering to the rule of thumb. As a result, the computer scientists generalize how humans deal with problems and extract all the possibilities into the functional procedures, which is CT; then they formulate these procedures into codes for the machine to follow and work efficiently and precisely. In that case, after learning CT, students are expected to solve the problems efficiently and precisely as well.

Therefore, an integration of CT into social issues was developed in this study. Furthermore, students need to conduct a group discussion first about the issue embedded in the assigned scenario, then deliver a declaration on the group decision and problem solution in front of all the classmates. To reply to the declarations issued, all the groups have to reiterate their stances, respond to the stakeholders' appeal, and then interact with their counterparts again.

It was expected that the learners would solve the problem more methodically after the integration of CT into the social study. Hence, a course was conducted for answering the research question: What is the difference in the students' performance of problem-solving before and after the integration of computational thinking?

This study combined creative problem-solving with CT steps into the multilateral communication in a game summit. The conceptual foundation of this research is supported by CT and the creative problem-solving approach (CPS).

2. LITERATURE REVIEW

The purpose of the study is to investigate students' computational thinking patterns in terms of international communication from their online concept maps.

2.1. Creative problem-solving (CPS)

The theory of creative problem solving, drawing upon the theory of the German physicist Hermann Helmholtz and the interview of several creative experts, was developed by Graham Wallas in 1926. The four steps of the creative thinking manner, first proposed by Graham Wallas consist of: preparation, incubation, illumination, and verification. The period of preparation is to collect information about the problem and try to develop solutions related to the problem. The period of incubation is to try out various aspects of the idea in order to find a solution, and also to think deeply about it. The next stage is illumination, which facilitates the key to the solution of the problem with a clear guidance. The period of verification is for the final method to be certified to determine the reasonableness and feasibility of this conclusive solution.

Many researchers in succession to Graham Wallas have subdivided and supplemented those processes with creative problem-solving procedures in detail (Bogen, Joseph E., & Glenda, 1999). With years of practical experiences and academic efforts, some scholars have revised creative thinking into the creative problem-solving model with six phases: mess-finding (MF), fact-finding (FF), problem-finding (PF), idea-finding (IF), solution-finding (SF), and acceptance-finding (AF) (Isaksen & Treffinger, 2000).

2.2. CT Integrated Complex Problem-Solving (CTICPS)

According to the Four Cornerstones of Computational Thinking on BBC Bitesize website, the four specific techniques of CT include: 1. Decomposition: Breaking down a complex problem or system into smaller, more manageable parts; 2. Pattern recognition – Looking for similarities among and within problems; 3. Abstraction – Focusing on the important information only, ignoring irrelevant detail; 4. Algorithms - Developing a step-by-step solution to the problem, or the rules to follow to solve the problem. All cornerstones carry equal significance and can be compared to the legs of a table.

From the above-mentioned literature of CT and CPS, the researchers synthesize and combine the four techniques of computational thinking (CT) and the six phases of creative problem-solving (CPS) into the CT-Integrated Complex Problem-Solving (CTICPS) approach. The researchers summarize the similarities between CT and CPS with four steps in order, which serve as the theoretical basis for this study. These steps include defining the problem, analyzing the challenge, reaching the consensus, and finding the solution. Based on the aforementioned theoretical foundation, the researchers illustrate here the incorporated model in detail as follows (See Table 1):

Table1. CT-Integrated Complex Problem-Solving (CTICPS)

Model	CT	CPS
Defining the problem	Decomposition	MF, FF, PF
Analyzing the challenge	Pattern Recognition	FF, PF
Reaching the consensus	Abstraction	IF, SF
Finding the solution	Algorithms	SF, AF

3. METHOD

3.1. Participants

The participants of this study were 22 10th-grade students attending a selective course of intercultural communication (IC) in a senior high school in southern Taiwan. It hasn't been time for these 10th graders to choose their academic clusters in the semester and that means they are still open to different majors in the future. None of them had ever learned any programming language, flow charts for algorithms, or concept-mapping prior to their participation in this study.

3.2. Course design and the measuring tool

For this pragmatic study, the students need to internalize the conceptual knowledge of CT, and then apply the CPS model to solve crises in the intercultural communication course.

In order to provide the students with a scaffolding aid, the teacher first guided them to use an online concept mapping platform for two weeks and allowed them to practice concept mapping on different projects. Following the consecutive two-hour weekly courses, the researchers spent two weeks on the content of Intercultural Communication (IC) and another two weeks asking them to have group discussions on recent international crises. After a thorough discussion and a mock international negotiation in the first summit game, the national leaders of all groups were required to make a declaration about their respective diplomacy and compromise; then, the students drew pre-concept maps on how to deal with the situation, indicated in their final declaration. After that, the teacher spent two weeks on briefly explaining the concepts of CT and the six stages of CPS. During these sessions, the instructor also gave an introduction to the CT-Integrated Complex Problem-Solving (CTICPS) approach, consisting of defining the problem, analyzing the challenge, reaching the consensus, and finding the solution, as well as to how to apply CTICPS to negotiate with the stakeholders, who might be allies or opponents.

With the aforementioned CTICPS, the teacher led them to run the integrated model.

1. Defining the problem: The complex problem was not easily conceivable to get a whole picture and the students had to clearly define the problem and to break down the problem into smaller, more workable parts. It was also associated with information-collecting, intelligence-gathering, dispute-querying, and identifying key stakeholders' purposes. To incorporate CPS, the students have to verify the source of information, to focus on the truth and to break away from turmoil in discussion for a deep understanding of this problem. Meanwhile, the learners also need to consider the IC aspect and probe the different perspectives of the allies, opponents, and rivalries to find a common ground and shared goals;

2. Analyzing the challenge: In this step, the students used CT skills to pick up any information that may have been neglected and to reassemble any incomplete pieces of puzzle to have a more comprehensive knowledge from the scenario. They can also use CPS such as fact-finding (FF) and problem-finding (PF) to increase their findings from visualization and imagination. As a result, any possibilities would not be omitted in this stage;

3. Reaching the consensus: In this step, the learners would come up with many possible solutions to the problem. After checking one direction at a time from these findings, the learners would focus on a range of options and evaluate them based on their feasibility, impact, and conceivable outcomes. In addition, through a series of massive data collection and intelligence analysis, the students could also gradually summarize the pros and cons into a consensus by analyzing the strengths, weaknesses, opportunities, and threats.

4. Finding the solution: The final step consists of testing, refining, implementing, and monitoring the solutions for evaluation. Compared with algorithms of CT, the step of solution-finding also needs to be gone through by examining, executing, identifying, and verifying the validation and effectiveness of the final solutions. Also, the learners would use CPS techniques such as solution-finding (SF) and acceptance-finding (AF) to respond to the impact. In addition, they were supposed to employ negotiation and communication skills of IC to address any arising challenges or objections with the allies and opponents.

In this study, the pre- and post- concept mappings on problem-solving were required before and after the CTICPS-instruction. The concept maps drawn before CTICPS-instruction were the pre-test data; those drawn after the instruction were taken as post-concept maps. Moreover, for investigating the learners' authentic voice, an open-ended interview would be given as well. The analyses of the concept maps resorted to their additional clarification in the interviews.

4. RESULTS

In order to probe the difference in the students' performance of problem-solving before and after the CTICPS approach, the researchers analyzed the number of the nodes on the concept maps and the completion of the four-step skills of CT to investigate the performance of the assigned problem on the learners' concept maps. A graphical organizer known as a concept map is used to generate ideas and visually depict the connections between concepts.

In this study, many students drew shapes of idea and marked related nodes in their pre-concept maps; however, most of them could only draw the concept maps regarding some important knowledge they acquired in the class; they could not effectively apply the knowledge and information they found for the problem-solving to the issue. Take the pre-concept map of Participant A as an example (See Fig. 1).

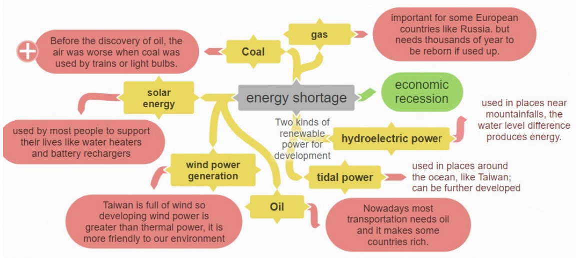


Figure 1. The Pre-concept-map of Participant A

He was aware of different types of energy, for example, the would-be-exhausted (coal, gas, and oil) and the renewable (wind power, solar energy, hydroelectrical power and tidal power). But instead of defining the problem of energy shortage first, he directly decomposed the issue of energy shortage into finding types of energy as well as giving the consequence of the shortage, namely, economic recession. Many nodes in their pre-concept maps were excluded for the invalidation of CT. After excluding the invalid nodes of the pre-concept maps, the average score of the students who mentioned about problem-solving was 3.32. After the instruction of the CTICPS, the post-test data in table 2 indicates the average score of valid nodes raised to 8.00. The obvious increase in nodes occurred (See Table 2) when most of the learners employed CTICPS in the IC course and put more efforts in discussing problem solving and in having domestic and international negotiations among group members in the second summit game.

Table 2. Number of Nodes of the Concept Maps

	Number of Nodes	N	Mean
Pre-test	73	22	3.32
Post-test	176	22	8.00

On the post-concept map shown as Fig. 2, participant A did the task of problem solving for the issue of energy shortage by utilizing the CTICPS model, which includes four elements: defining the problem by decomposition, analyzing the challenge by pattern recognition, reaching the consensus by abstraction, and finding the solution by algorithm. The four elements are analyzed as follows.

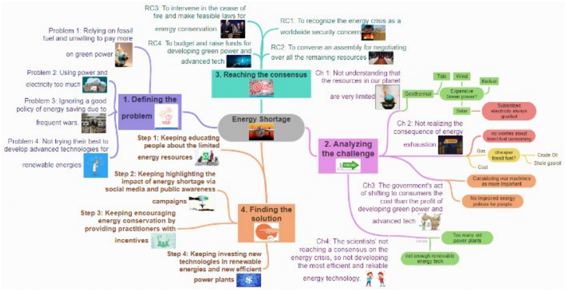


Figure 2. The post-concept-map of Participant A

First, he defined the problem of energy shortage by decomposing it into four small problems (See Fig. 6):

1. The present problem of energy usage and its reason: People rely on the fossil fuel rather than green power for its smaller expense.
2. The reason for the problem of energy shortage: People use power and electricity too much.
3. The cause and effect of not solving the problem of energy shortage: Due to frequent wars, the government is busying doing other businesses than a good policy of energy saving.
4. The fact of lacking solutions to the problem: People do not try their best to develop advanced technologies.

Participant A's good performance of defining the problem resorts to his considering each aspect of energy shortage, and this becomes the key point that decides whether his problem solving would succeed.

From figure 7, the researchers observed that participant A, by taking the step of pattern recognition, proceeded to analyze four challenges underlying the problem of energy shortage. He claimed that people stayed unalert to the exhaustion of energy in our planet, that is, they did not feel the urgent need to save energy, thus they refused to use green power because of its greater expense. From the fragmented information about different types of green power (i.e. expensive green power and cheaper fossil fuel) as well as the political and professional factors in not developing green power and advanced technology, participant A identified three patterns of challenges respectively from people, the government, and the scientists. The potential reasons for the three groups' inaction reside in the consideration to the cost, the war, the lack of improved policies and techniques, as well as the existence of old power plants. After the correlation among the four challenges was found, he continued to decide which aspects are critical to the problem solution.

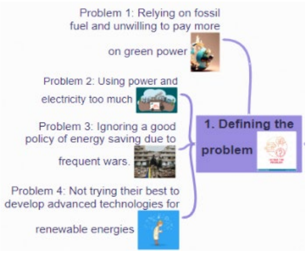


Figure 3. Step 1 of CTICPS on Participant A's map

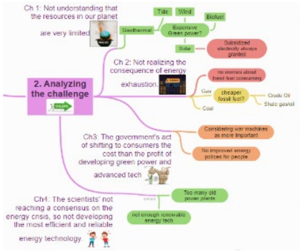


Figure 4. Step 2 of CTICPS on Participant A's map

The third step is to reach the consensus by abstracting the important concepts regarding solving energy shortage from the detailed ones such as the information about the types of energy and the underlying causes of not solving the problem. To target at the former challenges in the second stage of the CTICPS, he reached a consensus for four aspects, including recognizing the energy crisis as worldwide security concern, convening an international assembly to scrutinize and negotiate over the remaining resources for all human beings, conciliating the wars among the hostile countries and legislating for a better energy policy, and having more investment budget in developing renewable energy (See Fig. 8). His focus is on people's awareness, international cooperation, practical policies, and financial assistance. He has great achievement in the job of abstraction.

In the last step of CTICPS, the students coalesced the scattered concepts, partial pieces of information, and plausible ideas to a qualitatively valid direction. Here, they must examine whether a complex problem can be successfully resolved by virtue of the systemized problem-solving skills in the previous three stages. In figure 9, the researchers noticed that Participant A made a suggestion to raise people's awareness by education of power sources, provoke international cooperation by focusing on public attention through global campaigns, execute practical policies by offering substantial benefit projects to people, and provide financial assistance by injecting research and development into leading-edge energy. His solution concentrated on the feasibility of awakening people's consciousness of energy crisis so that people would supervise the government to all kinds of worldwide agreement. Moreover, under pressure from the people, the administrators, for better budget management, are willing to make concessions to raise energy funds and investment for the general public and scientists, instead of allowing monopoly by a few energy suppliers and plutocrats. Seen from the whole concept map, he demonstrated a great performance of problem-solving, and his maneuvers can undergo repetitive verification to similar complex problems in social issues.

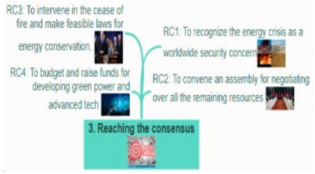


Figure 5. Step 3 of CTICPS on Participant A's map

Figure 6. Step 4 of CTICPS on Participant A's map

In addition, the statistics in table 3 shows that the students' best performance in the pre-concept maps generally fell in the aspect of Pattern Recognition and the worst performance fell in their Abstraction skills. That describes the students can find some partial information from the Internet and past learning experiences that can be used as some choices of the solutions, but they cannot synthesize the choices into a completer and more systematic one. More characteristically, their consensus to the solution lacks consistency and that makes their problem-solving skills in the pre-concept maps appear to be few and weak.

Table 3. Number of Steps in CT of the Concept Maps

Step in CT	De-composition	Pattern Recognition	Abstraction	Algorithm
Pre-test	22	33	6	12
Post-test	40	43	46	47

The item of Algorithm is less than half of the item of Decomposition. On the contrary, the data in Table 2 shows that the students' CT skills generally improved after the CTICPS, especially in the feature of Abstraction. Although the improvement in the item of Pattern Recognition was less pronounced, it was possible that the students' ability to search for information was capped because the data-questing skills were not included in the CTICPS model. However, the students' performance in Decomposition shows a multiplication, and more importantly, the students' performance of problem-solving significantly increased four times compared to the pretest. Thus, students' problem-solving skills did improve significantly.

5. DISCUSSION AND CONCLUSIONS

To sum up, the CTICPS activity was found effective and significant in enhancing the students' CT and facilitating them to come up with feasible solutions. The researchers found that most of the participants were interested in social issues but were unable to think of solutions; their attitudes towards disputed issues or problem solving were also mostly negative. By interviewing the participants, the researchers learned that they were not interested in international affairs and international conflicts, and that they did not perceive the relevance to and impact of the international events. However, via the introduction of CT, students can first break down the assigned scenario into smaller parts that are understandable and easy to find data.

Consequently, by following the CTICPS model, it is relatively easy for the students to carry on group discussion and focus on the topic of the discussion—the social issue and strategies of solution. They would search for the information on the Internet with systematic integration and application of the knowledge and logic they have learned in class. After the CT-Integrated course, the students learned that they could use computational thinking for future programming courses or utilize the concept to deal with many daily problems in their lives. Therefore, they took a more positive attitude in formulating more practicable solutions and creating better concept maps by following the computational thinking procedures.

6. REFERENCES

Barr, D., Harrison, J., & Conery, L. (2011). *Computational thinking: A digital age skill for everyone*. Learning & Leading with Technology, 38(6), 20-23

Bogen, J. E., & Bogen, G. M. (1999). Split-brains: interhemispheric exchange in creativity. *Encyclopedia of creativity*, 2, 571-5.

Curzon, P. a. M., Peter W (2016). *Future Thinking. In The Power of Computational Thinking (pp. 1-6)*. WORLD SCIENTIFIC (EUROPE). https://doi.org/doi:10.1142/9781786341853_0001

Isaksen, S. G., Dorval, K. B., & Treffinger, D. J. (2000). *Creative approaches to problem solving: A framework for change*. Kendall Hunt Publishing Company.

The Importance of Creativity in Developing Computational Thinking in Primary Education

Nardie FANCHAMPS^{1*}, María ZAPATA-CÁCERES^{2}, Estefanía MARTIN-BARROSO^{2}, Emily VAN GOOL^{3}
^{1}Open University, Netherlands
^{2}Universidad Rey Juan Carlos, Spain
^{3}Fontys University of Applied Sciences, Netherlands
nardie.fanchamps@ou.nl, maria.zapata@urjc.es, estefania.martin@urjc.es, evangool57@gmail.com

ABSTRACT

Applications for developing Computational Thinking (CT) are currently in the spotlight. This is notable since recently published theories and preliminary research conducted suggests that forms of creative thinking and acting are at least as important as the mental skill to apply concepts, methods, problem solving techniques, and the ability for logic reasoning for developing CT. Our research provides additional indications of the importance of creativity in CT development. We collected and analysed both quantitative and qualitative data to enlighten the importance of creativity and the feeling of having been creative when designing and solving tasks in developing CT. Our findings indicate that there is a clear interaction between being creative, acting creatively and using creative skills in developing CT. Subject to relevant influence of creativity, a development of sub-characteristics of CT could be demonstrated, elucidating that learning CT goes beyond a mere focus on concepts derived from computer science.

KEYWORDS

Computational thinking, creativity, motivation, unplugged activities, education

1. INTRODUCTION

Currently, variations in approaches regarding the development of computational thinking (CT) are prominent. Creativity appears to be an important parameter concerning the development of CT. Previous studies show that creative thinking and creative action can spur extraordinary discoveries in solution processes for challenging problems. Therefore, it seems almost certain that creativity can have an important impact in the (further) development of CT.

Previous studies indicated interrelationships between CT and creativity (Israel-Fishelson & Hershkovitz, 2022), and indicate positive effects of creativity via application of games regarding CT development (Román-González, Pérez-González, & Jiménez-Fernández, 2017). There is also evidence of a link between creativity and CT, with exhibiting originality early in a game being associated with relatively easy success at this stage but being negatively associated with progressing further in a game (Hershkovitz et al., 2019). In addition, research also distinguishes between two types of creativity, namely: creative thinking and computational creativity, where creative thinking should be seen as the innovative process of solving challenging problems, and where computational creativity is characterised by applications of computer technology to emulate, study, stimulate and enhance human creativity (Israel-Fishelson et al., 2021).

Our research focuses on the role of creative thinking and how and to what extent it contributes to the development of CT. Participants were 89 primary school students and where the skill of CT was measured using the Beginners Computational Thinking Test (BCTt) (Zapata-Cáceres, Martín-Barroso, & Román-González, 2020) targeted to students from 4 to 10 years old, and the Computational Thinking Test (CTt) (Román-González et al., 2017), targeted to students from 10 to 16 years old.

2. METHODOLOGY

This study was carried out in schools in The Netherlands and Spain, focusing on the last cycle of primary education (grades 4, 5 and 6), with 89 pupils aged 9-13 years, as shown in Table 1.

Table 1. Participants.

Sub-sample	n	Country	Grade	Age
A	53	Spain	4	9 to 10
B	36	Netherlands	5, 6	10 to 13

In the case of sub-sample A, the aim was to relate students' skills in creativity-related subjects to CT skills. Students were administered the age-appropriate test BCTt to collect their scores in order to obtain their current level of development in CT. In addition, their scores were collected in the creative subjects such as music and art.

In the case of sub-sample B, a mixed-methods study was conducted to investigate whether the use of unplugged board games influenced CT development, and to what extent creativity played a characteristic role in this. The unplugged activities consisted of 8 different board games including Froggit, Cube Duel, Checkers, and BattleShip (see Fig.1). To determine the effect of the intervention on the development of CT, quantitative data was collected by administering the age-appropriate test CTt (pre-test) to capture the previous level of CT, then had 5 sessions with unplugged board games for CT development for 1 hour each session, and then administered the CTt again as a post-test.

In both cases, qualitative data were collected regarding both aspects of CTt and students' creativity.

In sub-sample A, students were asked about their motivations and interests in programming, CT and creativity. They were asked about their favourite subjects, as well as about their preferences in the professions they would like to pursue in the future. They were also asked whether they thought they could learn these skills while playing. In sub-sample B, to determine to what extent creativity had a defining effect regarding CT development, interviews were conducted after playing different board games.

Figure 1. Some of the unplugged board games used.

These interviews included the following four main questions (1) “What would you like to share about working with board games?”, (2) “What did you learn from it?”, (3) “Have you been creative?”, and (4) “How would you feel about working with board games at school?” Subsequently, in-depth questions were asked: -“What did you find fun, difficult, boring about board games?”, -“Have you been creative, and how?”, -“What would you like to do differently?”, -“What do you know now? What can you do better?”, -“What helped you the most or least?”, -“How would you prefer to do it?”.

3. RESULTS

In sub-sample A, quantitative data indicate a highly significant correlation (at the 0.01 level, 2-tailed) between art subject skills and BCTt results (N = 53, Pearson Correlation = .391, Sig. = .004), as well as between music subject skills and BCTt results (N = 53, Pearson Correlation = .408, Sig. = .002).

Qualitative data show a preference of 16.98% of students for subjects related to creativity, these students have an average BCTt score of 21.8, higher than the sub-sample average of 21.5. However, students who wish to pursue professions related to creativity (20.75% of the sample) have, on average, a BCTt score of 21, lower than the sub-sample average. In terms of students' expectations of being able to learn CT through play and encouraging creativity, just over half, 56.6% of the students, think so.

In sub-sample B, regarding qualitative data, from the 36 students involved, 29 students show positive growth on CT (difference pretest: 53.57% of correct answers - posttest: 63.09% of correct answers). Further qualitative analysis by the interviews conducted reveals that when asked, "Have you been creative, and how?", pupils indicate: -“See how to solve challenges in the smartest way” (1), -“Make other combinations” (1), -“Know how to make figures move” (1), -“Think more about strategies” (3), -“I can handle my losses better now” (3), -“I have become better in class” (3), -“I can focus better” (1), -“I can better predict what the other person is doing” (2), -“Seeing carefully in steps, then trying” (1), -“Flexible thinking when dealing with challenging situations” (1). Deeper questioning reveals that pupils who played the several different unplugged activities rated them 64% as (very) fun, 25% as difficult and 10% as boring.

A further analysis indicated that 75% of the pupils who applied these unplugged activities articulated their growth in knowledge and skills more specifically and therefore most closely aligned with the development of their CT skills. Answers included: "Being able to think about strategies better" (25%), "Possibilities for classroom application" (25%), and "Being able to predict better what the other person is doing" (25%).

4. DISCUSSION AND CONCLUSIONS

Our findings indicate that creativity is an important parameter to develop CT, as one of our most relevant findings is the very high correlation found between the children's development on CT and their actual creativity skills shown in the school subjects. Moreover, when measuring CT development while enrolling with unplugged CT activities, students show high motivation on creative problem-solving which reflects in a higher development on CT. Moreover, our research emphasises that creativity is a thinking ability for computational problem solving in many innovative ways.

Furthermore, the interests of students with high BCTt scores in traditionally creative subjects may also indicate that there is a motivational relationship between interests in creativity understood as art or music, and the development of CT. However, the results indicate that students do not perceive this relationship, as it is not reflected in the professions they choose for the future and neither in the students' opinion as to whether creativity is necessary for the development of CT.

There are still discrepancies and areas where further research is needed in order to determine how creativity is impacting on the development of CT, as well as whether students' perception of the use of creativity when programming or when working in CT also influences their motivation in developing this skill. In addition, more research is needed on the specific CT skills on which creativity has more or less impact, on students' culture or perception of what creativity is and how it relates to CT, as well as on how to adequately measure the relationship between creativity and CT.

5. REFERENCES

Hershkovitz, A., Sitman, R., Israel-Fishelson, R., Eguíluz, A., Garaizar, P., & Guenaga, M. (2019). Creativity in the acquisition of computational thinking. Interactive Learning Environments, 27(5-6), 628-644.

Israel-Fishelson, R., & Hershkovitz, A. (2022). Studying interrelations of computational thinking and creativity: A scoping review (2011–2020). Computers & Education, 176, 104353.

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The associations between computational thinking and creativity: The role of personal characteristics. Journal of Educational Computing Research, 58(8), 1415-1447.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, 72, 678-691. doi:10.1016/j.chb.2016.08.047.

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. Paper presented at the 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal.

Designing Coding Problems and Unplugged Activities in Science to Develop Computational Thinking in Sixth Grade Students

Meghna SINGH[1*], Arti Singh[2], Archana KAPOOR[3]
[1,2,3]Dayalbagh Educational Institute, India
meghna23kairay@gmail.com, artisingh.agra@gmail.com, dr.archana.kapoor@gmail.com

ABSTRACT

Computational thinking (CT) is now widely recognized as an essential skill for the students to help solve problems which could benefit the masses. India has also accepted the importance of developing this skill in the students right from the childhood stage, as is evident in the new National Education Policy 2020. The new curriculum framework based on the Education Policy is yet to be presented. Therefore, it is a crucial time to look for efficient ways and means that could serve as potential references in deciding the CT curriculum for young children. This short paper presents the examples of subject-specific coding problems and unplugged activities as a method to integrate CT in regular Indian classrooms.

KEYWORDS

Computational thinking, Coding problems, Unplugged activity, CT in Science, Middle school students

1. INTRODUCTION

In 1980, Seymour Papert introduced Computational thinking in his book "Mindstorms: Children, Computers, and Powerful Ideas" wherein it was discussed as a thinking which students employed while programming with Logo (Papert, 1980). The idea of Computational thinking in K-12 Education was rekindled by Jeannette Wing (Wing, 2006). Computational thinking as defined by Cuny, Snyder and Wing, "is the thought process involved in formulating problems and their solutions, so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Wing, 2011). The information-processing agent here could be a human or a machine. Aho (2012) defined Computational thinking as, "the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms". Krauss and Prottsman (2016) states Computational thinking as a thought process to reformulate and solve problems utilizing problem decomposition, pattern matching, abstraction, and algorithms.

There is no consensus on the definition of Computational thinking till date, which hampers the development of uniform learning outcomes, teaching content and forms of assessments. This limitation however makes CT a dynamic concept which gives the freedom to the practitioners to choose from a variety of frameworks and definitions to suit the needs of their learners keeping in mind their stage of growth and cognitive development.

2. PURPOSE OF STUDY

India aims to leverage the power of new technologies like Artificial Intelligence and Machine Learning to drive growth and development in the coming years. The achievement of this aim depends on the educational system of the country. Therefore, Computational thinking was added to the sub-head "Curricular Integration of Essential Subjects, Skills, and Capacities" in paragraph 4.25 of the National Education Policy 2020 (Ministry of Human Resource Development, 2020). The new National Curriculum Framework based on the policy that would guide the introduction of this new skill in K–12 classrooms have not been rolled out yet. It puts the onus on education professionals to explore for efficient ways to incorporate CT in typical Indian classrooms without mentally straining both teachers and students. The authors have therefore taken on the current work to construct subject-specific coding problems and unplugged activities that could be utilized to develop CT in sixth-grade children.

3. OUR WORK

Our research focuses on the development of Computational thinking in the sixth-grade students using the Coding and Unplugged approaches. The coding approach utilizes the computers or other programmable electronic devices to develop Computational thinking in the students, whereas no such requirement of computers is necessary in the Unplugged approach (Brackmann et al., 2017).

3.1. Coding problems

For the Coding approach, we are designing problems in class 6 Science subjects to be used in the Computational thinking module. The problems are in line with the Science syllabus for class 6, currently prescribed by the ISCE (Indian School Certificate Examinations) Board, which is followed in most parts of India. The examples from each of the sub-subjects are given as under:

Example 1 is taken from the sub-theme 'Measurement of Temperature' under the theme 'Physical Quantities and Measurement' prescribed for class 6 Physics.

Table 1. Coding problem in Physics

Programming environment: Scratch
Q. Develop an algorithm that is capable of: **(i)** asking user to input a temperature in Fahrenheit scale, **(ii)** convert the temperature from Fahrenheit to Celsius scale, **(iii)** print the result.

```
when clicked
say Hello! for 2 seconds
say I will help you to convert the temperature from Fahrenheit to Celsius for 5 seco
ask type the temperature in fahrenheit and press enter and wait
think Hmm... for 2 seconds
think 5 / 9 * answer - 32 for 2 seconds
```

type the temperature in fahrenheit and press enter

212

Figure 1. Code snippet for the Physics problem

Example 2 is taken from the sub-theme ‘Food and Chemistry’ under the theme ‘Importance of Chemistry’ prescribed for class 6 Chemistry.

Table 2. Coding problem in Chemistry

Programming environment: Scratch
Q. Develop an algorithm that is capable of: **(i)** defining Agro-chemicals, **(ii)** state types of Agro-chemicals, **(iii)** ask the user to input the name of any Agro-chemical, **(iv)** identify the type of Agro-chemical that the user has fed and print it.

```
when ⚑ clicked
say (Agro chemicals are the chemicals which are helpful to the agriculture) for (4) seconds
say (There are different types of agro chemicals) for (4) seconds
say (They are fertilizers, pesticides, insecticides and preservatives) for (4) seconds
ask (Type an agro-chemical and I will tell you to which category it belongs) and wait
if <[fertilizers ▾] contains (answer) ?> then
    say (It is a fertilizer) for (2) seconds
if <[insecticides ▾] contains (answer) ?> then
    say (It is an insecticide) for (2) seconds
if <[pesticides ▾] contains (answer) ?> then
    say (It is a pesticide) for (2) seconds
if <[Preservatives ▾] contains (answer) ?> then
    say (It is a preservative) for (2) seconds

add (DDT) to [insecticides ▾]
add (BHC) to [insecticides ▾]
add (sodium benzoate) to [Preservatives ▾]
add (sodium metabisulphate) to [Preservatives ▾]
add (salicylic acid) to [Preservatives ▾]
add (urea) to [fertilizers ▾]
add (sodium nitrate) to [fertilizers ▾]
add (potash) to [fertilizers ▾]
add (ammonium phosphate) to [fertilizers ▾]
add (calcium nitrate) to [fertilizers ▾]
add (malathion) to [pesticides ▾]
add (parathion) to [pesticides ▾]
add (aldrin) to [pesticides ▾]
```

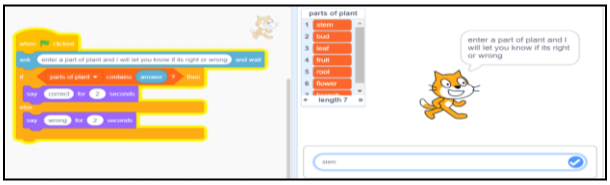
potash

Figure 2. Code snippet for the Chemistry problem

Example 3 is taken from the theme ‘Plant Life’ prescribed for class 6 Biology.

Table 3. Coding problem in Biology

Programming environment: Scratch
Q. Develop an algorithm that is capable of: **(i)** asking user to input a part of plant, **(ii)** identify if the input fed by the user is a part of plant or not **(iii)** print the result in the form: correct/wrong.



```
when ⚑ clicked
ask (enter a part of plant and I will let you know if its right or wrong) and wait
if <[parts of plant ▾] contains (answer) ?> then
    say (correct) for (2) seconds
else
    say (wrong) for (2) seconds
```

parts of plant

1	stem
2	bud
3	leaf
4	fruit
5	root
6	flower

length 7

enter a part of plant and I will let you know if its right or wrong

stem

Figure 3. Code snippet for the Biology problem

3.2. *Unplugged activities*

For the unplugged approach, the lesson plans will be developed integrating CT components in the subject topics. The examples from each of the sub-subject are given as under:

Example 4 is taken from the sub-theme ‘Characteristics of Solids, Liquids and Gases’ under the theme ‘Matter’ prescribed for class 6 Physics.

Table 4. Unplugged activity in Physics

Topic: Matter
Activity: Develop a model to depict packing of molecules in solid, liquid and gas using clay and other stationery items.

Example 5 is taken from the sub-theme ‘Arrangement of atoms and molecules in Solids, Liquids and Gases’ under the theme ‘Matter’ prescribed for class 6 Chemistry.

Table 5. Unplugged activity in Chemistry

Topic: Composition of Matter
Activity: Develop a model to depict arrangement of Hydrogen and Oxygen atoms in a Water molecule using clay, match sticks, and other stationery items.

Example 6 is taken from the sub-theme ‘Circulatory system: Process of circulation of blood in human heart’ under the theme ‘Human Body’ prescribed for class 6 Biology.

Table 6. Unplugged activity in Biology

Topic: Circulation of blood in human heart
Activity: Develop a model to represent circulation of blood in human heart, using clay, chart paper and other stationery items.

4. FUTURE RESEARCH

We plan to implement the two approaches in Indian classrooms to investigate their effectiveness in developing Computational thinking in sixth grade students.

5. REFERENCES

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.

Brackmann, C. P., Moreno-León, J., Román-González, M., Casali, A., Robles, G., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *ACM International Conference Proceeding Series*, 65–72.

Krauss, J. & Prottsman, K. (2016). *Computational thinking and coding for every student: The teacher’s getting-started guide*. Corwin Press.

Ministry of Human Resource Development. (2020). *National Education Policy 2020*. Retrieved January 22, 2021 from https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf

Papert,S. (1980).*Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books

Wing, J. (2011). *Research notebook: Computational thinking—What and why?* The Link Magazine. Retrieved August 22, 2022, from https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Wing, J. M. (2006). Computational thinking. *Communications of The ACM*, 49(3),33–35.

Learning About STEM Concepts in Primary Education by Using Highly Visualised ICT Learning Environments

Nardie FANCHAMPS[1*], Renée-Claire BELTING[2], Paul HENNISSEN[3], Stefan MANDOS[3], Juliette PASSARIELLO-JANSEN[4], Renato ROGOSIC[4], Krist SHINGJIERGJI[1], Bart BEAUMONT[3]
[1]Open University, Netherlands
[2]Erasmus University Rotterdam, Netherlands
[3]Fontys University of Applied Sciences, Netherlands
[4]Maastricht University, Netherlands
nardie.fanchamps@ou.nl, belting.renee@gmail.com, paul.hennissen@zuyd.nl, stefanmandos@hotmail.com, j.passariellojansen@alumni.maastrichtuniversity.nl, renato.rogosic@maastrichtuniversity.nl, krist.shingjergji@ou.nl
b.beaumont@denieuwstepabo.nl

ABSTRACT

Highly visual ICT learning environments offer excellent opportunities to enable meaningful learning about Science, Technology, Engineering & Mathematics (STEM). Application of such environments can lower the level of abstraction and can increase imagination, allowing fundamental concepts underlying STEM to be taught from an early age. Through stimulating inquiry-based learning, curiosity and by meaningful play, such ICT learning environments can ensure that primary school pupils develop a positive learning attitude and greater motivation for beta science. Our research project describes the application of Flui.Go as an experimental, high-visibility ICT STEM learning environment for pedagogical improvement regarding teachers in training to make pupils discover more about STEM. To analyse the application of Flui.Go, three different conditions are compared (Flui.Go versus traditional teaching materials and science boxes). Our research indicates that both teachers and primary school pupils develop a more positive attitude towards STEM through the application of highly visual ICT learning environments in which play-based learning takes place. An increase in the learning of underlying STEM concepts can be demonstrated compared to more traditional approaches.

KEYWORDS

STEM, ICT learning, teacher professionalisation, visual learning environments, digital learning technologies

1. INTRODUCTION

In primary education, there is an increasing emphasis on Science, Technology, Engineering, & Mathematics (STEM) education to prepare children for current societal challenges (Bybee, 2013). STEM knowledge and digital practice are crucial skills to navigate the 21st century (Slangen, 2009). Therefore, STEM-education is a necessary yet challenging development for students (White, 2014). The inherent complexity of STEM and the lack of attuned pedagogy results in a substantial knowledge- and attitude gap of students who report being deprived of seeing good examples in practice (Şenyiğit et al., 2021). Although the STEM knowledge-gap is a growing concern, research has paid little attention to the effects of different types of educational environments (Sumida, 2015). An experiment was performed to compare Flui.Go, a highly visual toy-kit featuring transparent blocks with internal channels and digitally operated pumps, a conventional science box, and traditional teaching methods to determine which learning environments are most effective in improving children's attitude towards technology.

2. METHOD

We conducted an explorative study to provide information on the effects of highly visualized learning and the possibility of their use as tools for integrated STEM education. Quantitative data was obtained through a pre-test-post-test design to examine the research question and the associated hypotheses; these included (a) Children's Attitude toward Technology Scale (CATS) questionnaire, (b) Fascination for Science, (c) Value of Science, and (d) Science and Physics Concepts quiz. Furthermore, qualitative data were collected during the experimental sessions where Flui.Go and commercial science boxes were used to describe the observable behaviour of students in the classroom while conducting experiments concerning colour mixing, pH value, and density. The research was conducted among pupils aged 11-12 from grade 8 (n = 50; 40% males, 60% females) of a primary school in the Netherlands in 5 sessions of 1 hour. Two experimental groups (n= 16; n = 16) and one control group (n = 18) were formed, guided with the same type and level of teacher-intervention.

3. RESULTS & DISCUSSIONS

3.1 Quantitative Data

Quantitative data from pre- and post- test questionnaires were analysed to determine whether inquiry-based learning adds value in comparison to traditional teaching settings. Results from the Children's Attitude towards Technology Scale (CATS) questionnaire are presented in Table 1. Average score (M) in the post-test was higher for students who participated in the experimental environments. Students using either the Flui.Go kit or the commercially available science box scored better than students in the control group who followed the regular course curriculum. These results are indicative of a more positive attitude toward technology. Therefore, the improvement in the scores between the pre- and post- test is likely due to the use of a visual STEM learning environment. One-way ANOVA variance analysis was used to compare the three groups and test the assumption of equal variances. A statistically significant difference between at least two groups was found (F (2, 97) = 6.519, p = 0.002). Tukey's honest significance difference (HSD) test for multiple comparisons assessed the significance of the differences between pre-and post- tests. Cohen d's determined the magnitude of the effect size, considered large above 0.8. The Flui.Go kit group had a significantly different mean value with a measurable large effect size in comparison to the control group (*p* = 0.002, 95% C.I. = [-0.28, -0.06], *d* = 1.00). There was no statistically significant difference between the commercial science box group and the control group (p = 0.081), nor between the groups that used Flui.Go and the commercial science box (*p* = 0.368).

Table 1. Children's Attitudes toward Technology Scale (CATS).

Variable	Pre-test CATS (26 items)				Post-test CATS (26 items)			
	M	*SD*	*Range*	*Mdn*	*M*	*SD*	*Range*	*Mdn*
Flui.Go (*n* = 16)	2.75	0.13	0.54 - 2.96	2.77	2.80	0.18	2.54-3.23	2.81
Science box (*n* = 16)	2.63	0.26	2.19 - 3.31	2.60	2.74	0.24	2.50-3.42	2.70
Control group (*n* = 18)	2.68	0.27	2.12 - 3.38	2.69	2.63	0.16	2.35-2.92	2.67

Note. M = average; SD = standard deviation; range = spread in measurement; Mdn = median.

3.2 Qualitative Observations

Observations in the classroom indicated that students involved in either one of the two experimental groups were enthusiastic and demonstrated a higher drive to learn about STEM. When comparing both experimental environments, it was found that the usage of the Flui.Go kit had several benefits over the commercially available science boxes.

3.2.1. Engagement

The Flui.Go kit is fundamentally different from what students are generally exposed to, resulting in an evident feeling of novelty that enhanced the excitement and curiosity towards their assigned STEM learning. When performing experiments, students seemed intrigued by both the usage of the controllable pumping system as well as the visual results in the transparent building blocks. The engagement of the students was high throughout the experiment sessions, which not only resulted in maintained focus but also in an intrinsic drive to learn more. This was evident when an experiment of colour mixture drove the students to continue investigating even after the experiment was done and the basic colours had been mixed.

3.2.2. Creativity

The students created their own combinations while playing with all the substances in the reaction. These instances repeated over the different experimental sessions. Students consistently showed responsibility to complete the prepared experiments correctly and stayed longer after completion to change the different variables the kit provides. An increase in creativity can be linked to this internal initiative as they tried to add substances to the experiments that were neither mentioned nor suggested by the guidebooks. For example, in an experiment about density with water and oil, deciding to add milk to see how it would affect the observed layers; or even changing the suggested building structure to create alternatives that would ultimately reach the same experimental result as was observed by one group that attempted to layer the transparent building blocks vertically instead.

3.2.3. Understanding of Complex Processes

The commercially available science box, although user-friendly with a lower learning curve for understanding how to use the components, constrained any modifications to the pre-set experiments. The experiments were procedurally defined and could only be performed once without the possibility of extra variable control to affect the output. As a result, the length of engagement was shorter; students would perform the experiments, obtain the results, and directly move on to the next activity. Instead, the modularity of the Flui.Go kit allows for multiple solutions to the same problem. Students had control over the compositions of the ingredients and fluid ratios via the ICT system, which could modify the flow processes and cause changes that are immediately visible through the transparent building blocks. This allowed students to create a link between the changed variables and their effects. The added layer of complexity increased satisfaction when completing activities and cooperation when finding solutions together.

3.2.4. Benefits for Educators

Interviews with teachers provided insight from a different perspective. Teachers reported, with a surprised undertone, that they had identified new talents in their own students. Teachers gained enthusiasm to deviate from traditional STEM teaching methods, resulting in an increased appreciation for the pedagogic value of trial and error. The research inspired teachers to foster an environment in which students can exhibit skills that may have gone unnoticed.

4. CONCLUSION

A highly visual ICT learning environment proved essential for the deepening of STEM learning since the ability to vary, monitor, and control the variables within the experiments was the key difference between the two learning environments tested. Students, although initially unfamiliar with all the electronic and programmable operations of the pumps, were able to understand their functioning and achieve control of every aspect of the activity. The freedom to explore encouraged an intrinsic motivation to continue experimenting, supporting the implementation of visual STEM learning environments with a meaningful added value in comparison to traditional teaching settings.

5. REFERENCES

Bybee, R. W. (2013). The case for STEM education: Challenges and opportunities.

Senyigit, Ç., Önder, F., & Silay, I. (2021). An Inquiry-Based Learning Approach for Effective Concept Teaching. ie: Inquiry in Education, vol. 13(1), 10.

Slangen, L (2009). *Techniek, leren door doen. Didactiek en bronnen voor de pabo*. Baarn, Netherlands: HB Uitgevers.

Sumida, M. (2015). Kids science academy: Talent development in STEM from the early childhood years. In *Science Education in East Asia* (pp. 269-295). Springer, Cham.

White, D. W. (2014). "What is STEM education and why is it important?," *Florida Association of Teacher Educators Journal*, vol. 1, no. 14, pp. 1-8.

Learning Assessment of Computational Thinking Digital Game <Captain Bebras>

Yan-Ming CHEN*, Ju-Ling SHIH
Graduate Institute of Network Learning Technology, National Central University, Taiwan
*peter880118 @gmail.com, juling@cl.ncu.edu.tw

ABSTRACT

A digital game with computational thinking elements was developed in this study, inspired by the internationally known Bebras Challenge. The digital game was used to develop students' computational thinking skills and to explore students' performance of computational thinking before and after the game. The purpose of this study was to develop computational thinking skills gradually through thinking and analysis between tasks to make students more motivated in the game. The results of the study showed that students' computational thinking skills had improved remarkably.

KEYWORDS

Bebras, Computational Thinking, Digital Game

1. GAME DESIGN

Computational thinking is the ability to think through a problem and then come up with a solution or to understand the problem in a more comprehensive way (Selby et al., 2014). The Bebras Challenge is an international computational thinking test that divides computational thinking skills into eight themes." Abstraction", "Logic", "Data Analysis", "Decomposition", "Algorithms", "Simulation", "Systematic Evaluation", and "Generalization". (https://www.bebras.org/)

A successful teaching process requires not only the appropriate materials but also the active participation of learners to produce an enhanced learning outcome. Digital games have been shown to have strong motivational factors that can motivate learners (Laine & Lindberg, 2020).

Therefore, this study designed a digital game <Captain Bebras> based on the subjects of the Bebras Challenge 2016 and 2017. <Captain Bebras> has five levels of sequential tasks based on the background of The Age of the Great Voyage (Chen & Shih, 2022). Designing a comprehensive virtual world background allows players to quickly adapt and learn during the game. For example, Frossard et al. (2019) used 13 game scenarios to form a physical teaching environment, and Moreno-Ger et al. (2007) used a "video game storyboard" to create a series of virtual stories to simulate reality.

Each level of the game has a task to be solved, while to make the game challenging, the levels are set in order from easy to hard (Taylor et al, 2019). For example, the first level (錯誤! 找不到參照來源。) is an "easy" level problem from the Bebras 2017 quiz, corresponding to CT's "Algorithm and System Evaluation" theme.

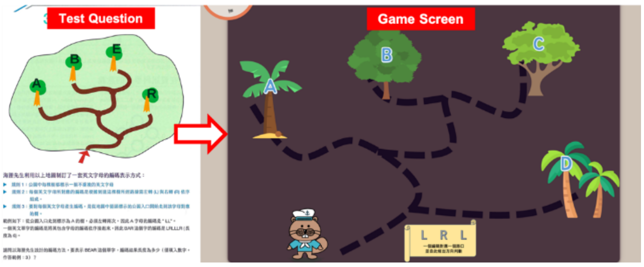


Figure 1. Game One

The second level (*Figure 2*) combines two "easy" questions from the Bebras quiz, deepening the difficulty of the task, and the corresponding CT themes are "Logic, Algorithms, Simulation, Abstraction, Generalization, and System Evaluation". The task was set to read and rearrange the order of the spices on the scroll and to find the correct spice from the four mazes.

Figure 2. Game Two

The third level (*Figure 3*) required students to use the maze from the previous level and, given the limited number of gold coins, this level focused on challenging students' prioritization and sequencing. The corresponding CT is "Abstraction, Decomposition and Systematic Evaluation".

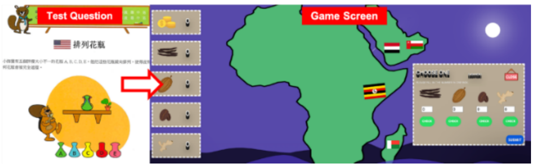


Figure 3. Game Three

Each level has a corresponding computational thinking skill, and as the difficulty of each level gradually increases, motivating the player to gradually develop computational thinking skills in the game and improve their CT skills without noticing (Chen & Shih, 2022). If the player answers incorrectly in a certain level, the player will need to repeat the same task until he solves the problem. Finally, when the player completes all the tasks, he will develop all eight computational thinking skills. (*Table 1*)

Table 1. Game and CT

Game One	**Game Two**	
Algorithms	Logic	Abstraction
Systematic-Evaluation	Algorithms	Generalization
	Simulation	Systematic-Evaluation
Game Three	**Game Four**	**Game Five**
Abstraction	Algorithms	Simulation
Decomposition	Systematic-Evaluation	Data Analysis
Systematic-Evaluation		Systematic-Evaluation

2. ACTIVITY DESIGN

2.1. Activities

This study was conducted as a one-day camp with 33 junior high school students aged 12 to 14 years old, 24 males and 9 females. It was held at the National Central University in Taiwan. The whole activity lasted for two hours in total. According to the activities, the students were divided into two classes, one in the morning and one in the afternoon, and the content of these two classes was exactly the same.

At the beginning of the activity, students were given a quiz with seven computational thinking questions to determine their prior knowledge level. After completing the quiz, the students began playing <Captain Bebras>, which lasted approximately 30 minutes. At the end of the game, the students will be given a quiz again. Similar to the previous test, there are seven computational thinking questions to determine their computational thinking skills after the game. Finally, the teacher will guide the students to review the whole activity and obtain feedback on the game and their motivation through a questionnaire. (*Figure 4*)

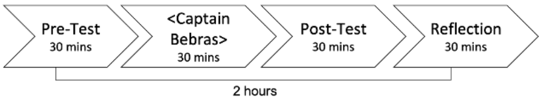


Figure 4. Course Activities Flow

2.2. Instructions

In class, the teacher simply explains the process and how to use the system to the students. Students can complete the pre and post-tests in the <Captain Bebras> system. After completing the pre-test, students will be directed to the game scene, during which the teacher will not give tips to students to pass the levels, and students can complete all the learning stages by themselves through the in-game hints. (*Figure 5*)

The pre-test and post-test questions are displayed on the computer screen and students are required to write their answers on the answer sheet in order to analyze the student's solution process. Both the pre-test and post-test questions were the same type of question, but with different answers and options. Students were required to write down their thinking process and answers for each question, and receive a point for getting both correct.

Figure 5. Student Classroom Performance

3. RESULTS

A total of 33 pre-tests and 33 post-tests were collected from 33 students in this study.

Based on the students' pre-test and post-test, the average score of the pre-test was 46.36 while the average score of the post-test was 53.64, showing a remarkable improvement (*p=.005*) (*Table 2*). This indicates that the <Captain Bebras> digital game used by the students in this study was effective in developing computational thinking skills and was demonstrated in the results of testing with the Bebras questions.

Table 2. Paired Samples T-Test for Pre and Post-test

T-test	N	Mean	SD	t	p
Pre-test	33	46.364	21.624	-2.988^{**}	.005
Post-test	33	53.636	18.844		

$^{**}p < .01$, $^{***}p<.001$

4. Acknowledgements

This study is supported in part by the National Science and Technology Council (previously known as Ministry of Science and Technology) of Taiwan, under MOST 108-2511-H-008 -016 -MY4.

5. References

Chen, Y. M., & Shih, J.-L. (2022, June). Bebras in the Digital Game< Captain Bebras> for Students' Computational Thinking Abilities. *In CTE-STEM 2022 conference.* https://doi.org/10.34641/ctestem.2022.455

Chen, Y. M., Shih, J.-L., & Chen, S. W. (2022). Design Methodology of Bebras Thematic Game. *The 30th International Conference on Computers in Education.*

Frossard, F., Frutos, M. B., & Trifonova, A. (2012). A learner-centred game-design approach: Impacts on teachers' creativity. *Digital Education Review*, 21, 13-22.

Laine, T. H., & Lindberg, R. S. (2020). Designing engaging games for education: A systematic literature review on game motivators and design principles. *IEEE Transactions on Learning Technologies*, *13*(4), 804-821.

Moreno-Ger, P., Sierra, J. L., Martínez-Ortiz, I., & Fernández-Manjón, B. (2007). A documental approach to adventure game development. *Science of Computer Programming*, *67*(1), 3-31.

Selby, C., Dorling, M., & Woollard, J. (2014). Evidence of assessing computational thinking.

Taylor, M., Baskett, M., Reilly, D., & Ravindran, S. (2019). Game theory for computer games design. *Games and Culture*, *14*(7-8), 843-855.

Examining Learner Engagement in Robot-Assisted Language Learning among Third Graders

Yu-Ching Lin[1], Vivien Lin[2*], Nian-Shing Chen[3]
[1,2] Graduate Institute of Children's English, National Changhua University of Education, Taiwan, R.O.C.
[3]Program of Learning Sciences, National Taiwan Normal University, Taiwan, R.O.C.
m0946004@gm.ncue.edu.tw, vivienster@gmail.com*, nianshing@gmail.com

ABSTRACT

The use of educational robots in classrooms has become more prevalent. With well-designed scripts installed, robots can serve such roles as teaching assistants, smart classroom managers, and learning companions. In traditional English as a Foreign Language classrooms, limited time is spent on individual or pair-based oral practice among learners, leading to the need to create more oral practice tasks. With an interdisciplinary effort, this study engaged eight Taiwanese third graders in oral interactions with an educational robot as well as with each other using activity scripts developed on an open-source visual programming platform. A humanoid robot acted as a facilitator to help participants engage in English oral practice activities. This study examined the effectiveness of robot-augmented textbook-based English learning activities on third graders' English oral interaction skills. Preliminary findings based on video analysis of learner engagement patterns showed that the robot-augmented textbook learning mode effectively led to robot-learner interactions. The researchers conclude that robot-augmented textbook learning using educational robots and sensing technology can be conducive for oral language development among young foreign language learners.

KEYWORDS

Educational robots, English learning, cooperative learning, human-robot interactions, oral competence

1. BACKGROUND

Educational robots feature humanoid appearance, repeatability, flexibility, and emotional expressions in situated learning (Hong, 2013). Robot's special features are suitable for supporting students with oral skill development and can be a great teaching tool for teacher to use in class. Specifically, Robot Assisted Language Learning has been implemented to facilitate oral interaction (Lin et al., 2022a&b) among young language learners (Han, 2019; Cheng et al., 2020). When young learners are learning how to speak a word or having conversation with others, they need lots of practices. Robots offer ample speaking practice opportunities because they allow learners to speak many times without fatigue or negative emotions, and without any difficulty in fluency and accuracy in the target language. Therefore, robots are good language learning companions for young learners.

Under the circumstance that many families in Taiwan are dual-income, parents may opt for buying robots for their children to practice English at home. It is therefore important to explore the effectiveness of using robots for oral language practice. The purpose of this study was to explore third-graders' engagement in robot-augmented textbook learning that focused on the learners' English oral interaction competence, especially their turn-taking skills. The research question raised was:

RQ: Can robot-augmented textbook learning effectively engage learners in oral interactions conducive to language improvement?

2. THEORETICAL UNDERPINNINGS

In recent years, communicative competence not only focuses on linguistic and grammatical knowledge such as phonological, semantic, syntactic and discourse but also pays more attention to pragmatic skills. Sociocultural theory is important to human language development. Sociocultural theory emphasized on the interaction between people can improve students' language skills (Panhwar et al., 2016). Vygotsky (1978) asserted that children's thinking ability was constructed by social interaction. The concept of the zone of proximal development (ZPD) and the scaffolding theory are therefore applied in this study. ZPD refers to the gap between the ability with which children can achieve a learning goal by themselves and the ability after children were assisted by scaffolds in the learning environment. Through robot-assisted oral tasks, target learners were expected to interact with their peers, which the researchers believed would helped them develop their language competence phonologically, semantically, syntactically, morphemically, and pragmatically.

3. THE STUDY

This is a multiple-case study using the qualitative research method based on video analysis. The robot-augmented textbook learning system consisted of an educational robot in the humanoid form, a tablet, a textbook embedded with QR codes, and sensing devices (See Fig. 1). The teaching material was adapted from a Taiwanese third-grade English textbook, and the contents included wh- questions, yes/no questions, and simple dialogue. Eight third-graders participated in this study. The research divided the experiment into two phases. In the first phase, individual guided learning was implemented (See Figs 2 & 3). The robot gives instructions to students for them to follow and give answer individually. The second phase consisted of cooperative learning in a pair-based activity. Two learners engaged in turn-taking English to complete the activity.

Figure 1. The robot-augmented textbook learning system

Figure 2. Guided learning in the individual learning mode

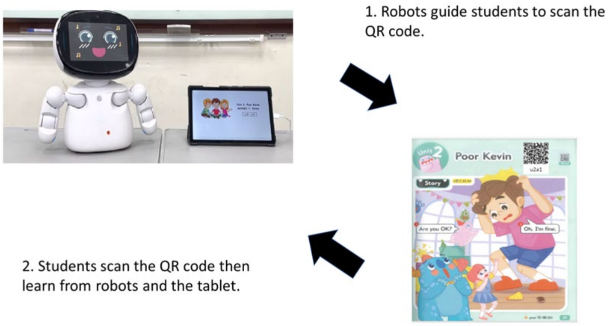
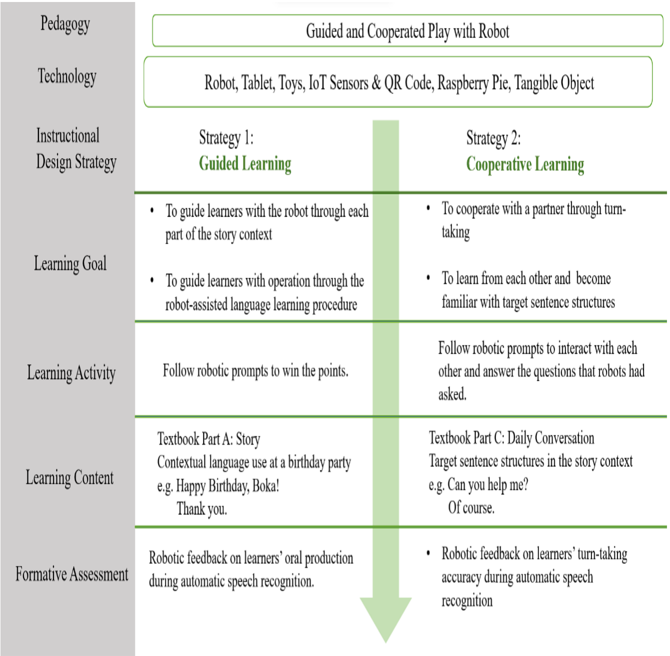


Figure 3. The instructional design with two strategies



Pedagogy	Guided and Cooperated Play with Robot	
Technology	Robot, Tablet, Toys, IoT Sensors & QR Code, Raspberry Pie, Tangible Object	
Instructional Design Strategy	Strategy 1: **Guided Learning**	Strategy 2: **Cooperative Learning**
Learning Goal	• To guide learners with the robot through each part of the story context • To guide learners with operation through the robot-assisted language learning procedure	• To cooperate with a partner through turn-taking • To learn from each other and become familiar with target sentence structures
Learning Activity	Follow robotic prompts to win the points.	Follow robotic prompts to interact with each other and answer the questions that robots had asked.
Learning Content	Textbook Part A: Story Contextual language use at a birthday party e.g. Happy Birthday, Boka! Thank you.	Textbook Part C: Daily Conversation Target sentence structures in the story context e.g. Can you help me? Of course.
Formative Assessment	Robotic feedback on learners' oral production during automatic speech recognition.	• Robotic feedback on learners' turn-taking accuracy during automatic speech recognition

Video analysis divided learner engagement patterns into three dimensions – behavioral, cognitive, and emotional engagement (Philp, J., & Duchesne, 2016). Then, the patterns were grouped into *strengthening* or *inhibiting* learner engagement. Specific codes were identified after iterative video viewing for each learner in each phase.

4. PRELIMINARY FINDINGS

The preliminary engagement analysis results focusing on the individual learning activity showed that learner engagement included three dimensions – behavioral, cognitive, and emotional engagement in descending order in terms of the frequency of occurrence (See Table 1). Strengthening Behavioral Engagement involved codes such as Paying Attention to the Robot, Interacting with the Robot, Repeating the Robot's Instruction, Imitating the Robot's Action; while Inhibiting Behavioral Engagement included such codes as Looking around and Talking to a Partner. Strengthening Cognitive Engagement involved such codes as Saying Something Related to the Task, Responding to the Robot's Prompts, and Nodding to Signal Understanding; whereas Inhibiting Cognitive Engagement included codes such as Relying on Teacher's Translation, Hesitating and Not Responding until the Robot Gave Correct Answer, and Saying Something Unrelated to the Task. Finally, Strengthening Emotional Engagement included such codes as Feeling Happy as indicated by learners' facial expressions, whereas Inhibiting Emotional Engagement included codes such as Feeling Confused or Feeling Frustrated as shown by learners' facial expressions or utterances.

Table 1. Learner engagement patterns found in the robot-augmented textbook learning

Engagement Type	Nature of Engagement (# instances)	
Behavioral	Strengthening (917)	Inhibiting (11)
Cognitive	Strengthening (218)	Inhibiting (77)
Emotional	Strengthening (28)	Inhibiting (11)

5. CONCLUDING REMARKS

This qualitative study shows sufficient HRI in individual learning. The robot-augmented text learning mode can effectively lead to robot-learner interactions and learner responses behaviorally, cognitively, and even emotionally. The researchers will proceed to analyze learner-learner interactions as well as robot-learner interactions in the pair-based activity as the next step of this research.

6. REFERENCES

Philp, J., & Duchesne, S. (2016). Exploring engagement in tasks in the language classroom. Annual Review of Applied Linguistics, 36, 50-72.

Han, J. (2019). Robot-Aided Learning and r-Learning Services. In Human-Robot Interaction; Chugo, D., Ed.; IntechOpen: London, UK.

Lin, V., Yeh, H. C., Huang, H. H., & Chen, N. S. (2022a). Enhancing EFL vocabulary learning with multimodal cues supported by an educational robot and an IoT-Based 3D book. System, 104, 102691.

Lin, V., Yeh, H. C., & Chen, N. S. (2022b). A systematic review on oral interactions in robot-assisted language learning. Electronics, 11(2), 290.

Cheng, Y. W., Wang, Y., Yang, Y. F., Yang, Z. K., & Chen, N. S. (2020). Designing an authoring system of robots and IoT-based toys for EFL teaching and learning. Computer Assisted Language Learning, 34(1-2), 6-34.

Panhwar, A. H., Ansari, S., & Ansari, K. (2016). Sociocultural theory and its role in the development of language pedagogy. Advances in language and literary studies, 7(6), 183-188.

Proposals for Hands-on and Heads-on Authentic Learning Strategies in New Education Normal

Tosh YAMAMOTO [1], Kazuya TAKEMATA[2*], Akiyuki INAMIDE[3], Ayako TANAKA[4] ,
[1,4]Kansai University of International Studies, Kobe, Japan
[2,3]Kanazawa Institute of Technology, Japan
soetosh@gmail.com, takemata@neptune.kanazawa-it.ac.jp,
minamide@neptune.kanazawa-it.ac.jp, aya-tanaka@kuins.ac.jp

ABSTRACT

This paper proposes a new authentic educational paradigm in New Education Normal. As a showcase for an integrated curriculum incorporating lifestyle, society, cultural values, regional history, STEAM, and entrepreneurship, active learning in the realm of the regional society involving elementary school through university students is demonstrated. In such active learning, learning begins with awareness of being a member of society, and then, the team-based PBL proceeds. The team conducts field research to identify grave problems or issues to prevent sustainability of the society and then tries to work on optimal solutions to share with other teams. In this educational model, the essential function of transcending values and cultures is guaranteed through the active learning mindset of all learners. In this way, the authenticity of learning is guaranteed and transcends cultural values and wisdom to the future generation.

KEYWORDS

authentic learning, academic integrity, New Education Normal, STEAM, gamification, innovative and creative learning

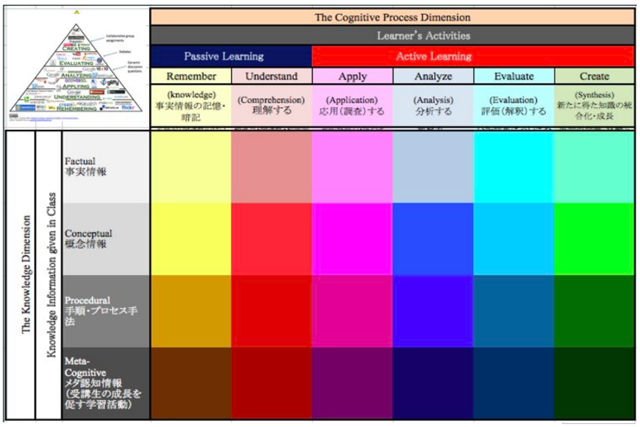
1. INTRODUCTION

Since the advent of SDGs into the realm of education, STEM has been incorporated into K-12 education. The educational paradigm has developed from the cognitive model to the constructive model and then to the connectivism model. This paper proposes a new authentic educational paradigm in New Education Normal. The proposed curriculum integrates lifestyle, society, cultural values, regional history, STEAM, disruptive and innovative learning, as well as social entrepreneurship in order to transcend to a future society filled with wellness.

In what follows, active learning is defined in terms of Bloom's Taxonomy. Further, future skills needed in future education are elaborated. In the end, an authentic learning activity for the elementary school level is showcased.

2. ACTIVE LEARNING

New Education Normal choreographed around ICT and Digital Transformation is a must. So is active learning. In Figure 1, while traditional education is centered around the top left corner where memorization and understanding are the major learning activities, learning activities in the New Education Normal will go beyond such boundaries, where exploration of acquired knowledge is the key. In the matrix, the learning activities such as applying, analyzing, evaluating, and creating are considered active learning.



The Cognitive Process Dimension

Learner's Activities

Passive Learning | Active Learning

The Knowledge Dimension

Knowledge Information given in Class

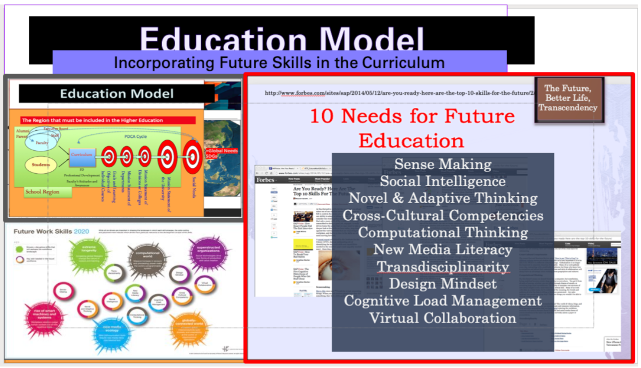
	Remember	Understand	Apply	Analyze	Evaluate	Create
	(knowledge) 事実情報の記憶・暗記	(Comprehension) 理解する	(Application) 応用(調査)する	(Analysis) 分析する	(Evaluation) 評価(解釈)する	(Synthesis) 新たに得た知識の統合化・成長
Factual 事実情報						
Conceptual 概念情報						
Procedural 手順・プロセス手法						
Meta-Cognitive メタ認知情報 (受講生の成長を促す学習活動)						

Figure 1. Bloom's Taxonomy Matrix

The realm of active learning is the major playing ground for the New Education Normal. And yet, such an endeavor is not an easy task. In order to nurture the fields of active learning, we must define the future skills needed for the New Education Normal, which is seen in the next section.

3. FUTURE SKILLS

Institute for the Future defines future skills as given in Figure 2 to nurture innovative and creative skills to survive and contribute to the future society.



Education Model

Incorporating Future Skills in the Curriculum

10 Needs for Future Education

Sense Making
Social Intelligence
Novel & Adaptive Thinking
Cross-Cultural Competencies
Computational Thinking
New Media Literacy
Transdisciplinarity
Design Mindset
Cognitive Load Management
Virtual Collaboration

Figure 2. Future Skills Defined by IFTF

Note that such future skills are not included in traditional education. In other words, referring to Figure 1, learning activities must range from the top left corner of the matrix to the entire area of the matrix. After all, authentic learning resides in the entire area of the matrix. Even in traditional education, we have been using thinking tools such as graphic organizers to enhance learning and critical thinking skills. In the Post pandemic era, we must employ innovative and creative thinking tools to nurture all future skills in the entire area of the matrix.

4. SHOWCASE

It is demonstrated as a showcase that university students and elementary school children work together with the local sweet shop owners to create the local sweets exploration map, which was later transformed into a sweets-hunting game designed and programmed by the students. The elementary school children walked around the neighborhood area and researched sweets shops, finding out their location, history, opening hours, products, most-sold popular sweets, and service types, among others. And then they created a trifold to include all information, as shown below.

Figure 3. Research Results shown in infographic

Upon completion of the trifold, the school children worked on designing a sweets-hunting game. The concept of the game is to obtain designated items by maneuvering a robot car. The entire process of this series of learning offers the school children to understand the local neighborhood area.

Figure 4. Game Board for Robot to Complete Tasks

In this way, the authenticity of learning can be reflected in the learning contents and resources from the regional segments of living, which include: lifestyle, regional culture (tangible and intangible cultural heritage), regional industry, economy, and local history, to name a few.

Figure 5. Scene of Completing a Task with Robot

5. CONCLUSION

Active learning in the New Education Normal implies the combination of traditional school learning with innovative, creative, and social entrepreneurial learning with gamification. With a future-oriented mindset, young minds can learn from the findings in the surrounding world and then share them with others in rich media. Education is not solely dependent on the endeavor of the school teachers, but of the entire stakeholders who run lives there to the future, as a result, offering a lifelong learning mindset for life to younger generations..

6. REFERENCES

Feaster, Y., Segars L., Wahba, S. and Hallstrom J. (2011). *Teaching CS unplugged in the high school (with limited success*). Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, 248-252.

Messac, A. (1996). *Physical programming: effective optimization for computational design*. AIAA Journal, 34 (1), 962-974.

Kimura, T. D., Choi, J. W. and Mack, J. M. (1990). *Show and Tell: a visual programming language*. Glinert, E.P. (Ed.), Visual Programming Environments: Paradigms and System, IEEE Computer Society.

Minamide, A. and Takemata, K. (2019). *Development of new programming with stickers for elementary school programming education. International Conference on Education and New Learning Technologies.* (Edulearn 2019), 4796-4800.

Minamide, A., Takemata, K. and Fujishima, S. (2021). *The practice of computational thinking education with teaching materials associated with the community*. International Conference on Education and New Learning Technologies (Edulearn 2021), 9565-9571.

Minamide, A., Takemata, K. and Fujishima, S. (2022). *The practice of computer thinking education with the cooperation of university students in elementary school: proposal of class management method under COVID-19.* International Conference on Education and New Learning Technologies (Edulearn 2022), 2056-2061.

Yamamoto, Toshiyuki. (2022). "A Proposal for Implementing Authentic Assessment Enhanced with Academic Integrity in New Education Normal." *Journal for Research Institute for Education*. Kansai University of International Studies. *Vol. 15*. 149-16

The Asia-Pacific Society
for Computers in Education
(APSCE)

National Central University