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

CTE-STEM

2023.6.7-6.9

Zhongli, Taiwan

Proceedings of International Teacher Forum

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 2737-565X

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

Keynote Speakers

Prof. Yasmin Kafai
Teaching, Learning, and Leadership Division –
University of Pennsylvania

Title: Preparing the Next Generation of Computational Thinkers

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

TEACHERS FORUM PAPER

A Critical Review on the use of Computational Thinking (CT) in General Studies to Enhance Students' Interest of Learning

Tin Yee CHAU[1*], Cho Ting CHOW[2], Yik Man HUEN [3]
[123] Yuen Long Government Primary School
chautyc@netvigator.com, chotingzhou@hotmail.com, huenyikman@outlook.com

ABSTRACT

This study examines how one primary school implements Computational Thinking (CT) in General Studies and evaluates student making use of self-assembled and coded devices to enhance interest of learning.

The facilitator and Panel head reported on how CT was being implemented and a group of 9 students who was selected from a class of around 40 students. These students attended a focus group to evaluate the use of self-assembled and coded devices in teaching General Studies, Primary 4.

The purposes of the study were to

(1) investigate the potential of Computational Thinking (CT) have a positive impact on the learning of General Studies in primary school,

(2) views on the self-assembled and coded devices can bring forth to teachers and learners in the education process.

A qualitative case study research method was chosen because it provided thick and rich descriptions of how teacher perceived the future of teaching. Students were invited to attend the discussion session to ensure that the technology at used on them are worth to. The size of the focus group was restricted to 9 students, as focus groups allow a depth of interaction, which is difficult to achieve in larger groups. With a smaller group, all students had chances to participate in the discussions.

KEYWORDS

Computational Thinking (CT), coded, CoolThink@JC

1. PURPOSE OF THE STUDY

This study aims to look for insight as to how Computational Thinking (CT) be successfully incorporated into classrooms and the challenges and demand for changes that it poses to the teachers.

There are two key questions in this research which are as followings:

1)"Will the use of Computational Thinking (CT) materials have a positive impact on the learning of General Studies in primary school?"

2)"What changes will self-assembled and coded devices bring forth to teachers and learners in the education process?"

It is the aim of this research to investigate for some possible answers to these questions through a review of relevant scholarly literature and an analysis of the views of teachers who currently use this technology.

2. QUESTIONS FOR THE FOCUS GROUP STUDENTS

Discussion focused on the following topics:

Topic 1: Interface and Content of the coded devices.

- How about the design and the interface of the coded devices, are they understandable and helpful?

Topic 2: The Level of the Motivation.

- How much do you feel when teacher used self-assembled and coded devices as an integral part of a teaching?
- Where / How often to use or assemble it, in school or at home?

Topic 3: Learning Value and Perceived Value of the Materials.

- How far do you think such self-assembled and coded devices can lead you learn more?

3. SCHOOL BACKGROUND

The school was selected for this study because of their recognition as leaders in the application of information and communication technology in teaching across the curriculum. The selected primary school had sixty-one teachers and offered 950 students aged from 6 to 11.

4. LITERATURE REVIEW

Restructuring schools is a never-ending process. The first goal for restructuring school is to change the philosophy so that education starts with the needs of the child. The second goal for restructuring the schools is to change the learning environment. (Collins, 1991)

Learning environments for children today are very different from what they were ten years ago. One of the major changes is the increased integration of technology into learning in school. Another change is the increase of student's learning opportunities outside regular schooling. Recent researches on learning also indicate that technology plays a critical role in changing classroom environments and restructuring schools to promote more engaged and powerful learning.

On 7th May 2014, the Government announced "The Fourth Strategy on Information Technology in Education (ITE4) Consultation Document". ITE4 is formulated to unleash the learning power of all our students to learn to learn and to excel through realizing the potential of IT in enhancing interactive learning and teaching experiences. With IT-rich school environment, schools' professional leadership and capacity, and the support from community partnerships, we aim to strengthen students' self-directed learning, problem-solving, collaboration and computational thinking competency, enhance their creativity and innovation, and even entrepreneurship, as well as to nurture the students to become ethical users of IT for pursuing life-long learning and whole-person development through leveraging technology and the capacity of IT. (Quality Education Division, 2014). Indeed, some evidence even suggests that technology may accelerate the restructuring process, especially if it focuses on using learning resources outside the classroom (Pisapla et al., 1993; Jones et al., 1994). Besides, educational technologists argue that the computer is ideal for the development of students' thinking skill because it can access, store, sort, and present information quickly in a variety of ways (Jonassen, 1996; Jonassen, Peck & Wilson 1999).

In Hong Kong, Computational Thinking (CT) is mainly cultivated through coding teaching in computer lessons. CT is considered as an essential problem-solving skill in today's society with rapid technological development. CT skills are unquestionably important in the twenty-first century, and such skills are clearly best taught through the actual use of computers or tablets. However, it is important that technology must be immersed in a diversity of school subjects and throughout the curriculum, and not simply used to impart technology-related knowledge and skills. More specifically, CT involves several imperative thinking skills including abstraction and decomposition, thinking recursively, problem reduction and transformation, error prevention and protection, and heuristic reasoning which are needed to solve universal complex problems, not limited to software problems.

In Hong Kong, CT is mainly cultivated through coding education, eg. CoolThink@JC, It is an educational project focusing on the primary students' CT cultivation, which includes a framework for guiding the design of K-12 CT curriculum (Kong 2016). CoolThink@JC which include on use of Scratch and App Invertor, they are considered as an essential problem-solving skill in today's society with rapid technological development. CT education has been immersed in a diversity of school subjects in Hong Kong. From the report entitled "Learning to Learn - The Way Forward in Curriculum Development" (Hong Kong Curriculum Development Council, 2001), teacher's role is to help students learn to learn, which involves developing their independent learning capabilities, leading to whole-person development and life-long learning. Thus, students should learn to be objective. There are many alternatives and one of them students could use was online CT project. By using the online coding project and hands-on connecting the electronic devices which offers step-by-step directions to explain just what students must do in order to meet the expectations for the learning activity. But, most educators complain that using online materials suffers from a low "signal to noise ratio" - the confusing, weak and unreliable information (noise) outweighs and threatens to drown out the information most worthy of consideration. Teachers want to see students putting their energy into interpretation rather than wandering. While the results of earlier studies on coding education indicated a positive impact on the overall performance of mathematics and general studies subjects (Wong et al. 2015).

5. METHODOLOGY AND METHODS SELECTED

Qualitative researchers often work with small samples of people nested in their context and study in-depth (Miles & Huberman, 1994). Qualitative samples are often purposeful rather than randomly selected. In this case study, participants were selected because of their integration of information and communication technology. Thus, the Class teacher and the Panel Head were asked whether the return on the investment had been worthwhile. It was found that searching for the answer to this question was difficult. The difficulty lied in how to judge the effectiveness of computer used in schools and the impact of implementing CT on teaching and learning were more difficult to detect, though arguably the most important, about CT existed within the school. We were going to deploy the actually happening in classrooms and did it reflect the reality of teacher attitudes towards the technology. In this way CT implementation in a school can be treated as a human endeavor that, in a Vygotskian sense, is mediated by social–cultural–historical tools and practices (Vygotsky, 1978).

In order to facilitate the process more deeply and had better result, round tables and discussions, focus group observation were carried out to gathering more information about the CT implementation. This would help providing a more subjective analysis.

6. FINDINGS ON THE USE OF COMPUTATIONAL THINKING (CT) IN GENERAL STUDIES (FOCUS GROUP STUDENTS)

6.1. Topic 1: Interface and Content of the Materials

How about the design and the interface of the coded devices, are they understandable and helpful?

Students found the use of the coded devices and CoolThink@JC online demonstration were generally satisfactory and particularly helpful in facilitating distance learning access, although there were some difficulties which were related to the use of a network for delivery. Students found the materials motivating in the context of its role as a support for the lecture. Specifically, they found that it filled in informational detail which was necessarily missing from lectures, that it filled in material which was missing from textbooks and that students could review their learning in their own time and at their own pace. This was found helpful in clarifying concepts, and the ability to repeat material which was difficult (not practical in lectures) was appreciated.

Some aspects of the material were seen as having particular advantages over lectures. The interactive animated demonstrations from CoolThink@JC were especially highly regarded, with an agreed view that illustrating concepts in this way added something to the learning experience which could not be provided adequately in a lecture setting. The fact that the demonstrations were truly interactive and students could provide their own data made their use particularly valuable and motivating as a learning medium.

6.2. Topic 2: The Level of the Motivation

How much do you feel when teacher used self-assembled and coded devices as an integral part of a teaching?

Where/How often to use or assemble it, in school or at home?

In general, students thought that the self-assembled and coded devices increased their motivation, and they used the material frequently outside of scheduled time (e.g. at home). They thought that lectures could be integrated with the material by putting coded devices used in lectures on-line. This would increase the signposting within the material as well as provide a handy revision cross-reference. The strongest motivating factor was universally agreed to be the CoolThink@JC animated demonstrations.

6.3. Topic 3: Learning Value and Perceived Value of the Materials

How far do you think such self-assembled and coded devices can lead you learn more?

Students in this study felt that they probably did learn more on using the materials, and the ability of the system to reinforce concepts greatly aided retention. This was particularly true of the demonstrations and tutorial questions. Most students reported trying each question several times if they got it wrong initially, rather than just looking at the answer right away.

7. CONCLUSIONS AND REFLECTIONS

Programming can be difficult as educators to bring across to students, it requires the use of logic and concepts that are not present in any other subject students have in their curriculum. However, we are confident — in both the educating front of teachers and the receiving end of students — that programming could be effectively executed in the classroom.

To enhance understanding and interest of students during lessons, intuitive and practical examples can be provided to help them better compute novel ideas. For example, simple online simulations or applied logic into real life situations aid the fundamental understanding of the code they are writing. By creating relatable situations, students no longer have to make huge leaps in logic from what they already know, diminishing barriers to entry for pupils who are new to programming, making them more eager to contribute in lessons. When students are lacking in confidence or motivation, it is then crucial as educators to not make individuals feel isolated. Group work is a great method to ensure that this can be avoided, as students commonly face similar difficulties, they are more motivated to work through problems together instead of feeling that they are falling behind on their own. Therefore, for programming to get across to students easily, intuitive examples must be used in combination with group work so that efforts by educators are made the most effective.

Programming may sometimes seem pointless to students, as content that is taught early on — such as drawing a square may be nothing of note and easily done by hand. However, to then follow up and re-using these basic skills with another basic skill such as loops, the square suddenly becomes a tessellating pattern that looks far more complex to the eye, its accuracy virtually impossible to be accomplished by hand. By showing students the power that even the basics of programming possess, they quickly understand why it is useful in real life and its importance. Then, educators could also display to students the omnipresence of code in their daily lives, emphasising the importance of programming. For example, students may be inspired by how their favourite apps and games are built all based on the same building blocks they are using. Teachers may then use this opportunity to show how what seems like vastly different end products, such as mobile games and websites, can all be produced through programming. Through this, students can begin to find creativity and fun in programming, incentivising them to work through more complex procedures in programming. If students do not develop a deeper interest or passion, teachers can provide achievable tasks that gradually increase in difficulty to spark students' curiosity and find joy in the ultimately satisfying problem solving process. Hence, in order to motivate students to be more open to programming — which they are all well capable of, educators are encouraged to aid learners to find their intrinsic motivation in programming, achieving this through igniting curiosity in progressively challenging tasks or displaying what great applications they could possibly create.

We believe that programming is very useful for all students — For ones who are less interested, it gives them valuable basic information as to how machines around them in their daily lives operate; For ones who are more interested, programming can inspire possible career paths for students very early on in their education, giving them a huge head start for their futures. We believe that every student is capable of programming as it is merely an extension of problem solving and creativity that is so often used in every person's daily life. To inspire the creativity needed in programming, groupings in peer work where there is a more capable student acting as a 'leader' in each group can be used. These more capable members can get a conversation started, then spark other students to think and express their ideas more openly, continuing said conversation in succession. Furthermore, programming gives students valuable transferable skills of problem and critical thinking. Because code relies so much on procedures where slight difference can make or break their final products, it educates students to be more thoughtful and tackle troubles they face with more maturely even at a younger age. This skill of problem solving can be attained by reflecting upon their own flaws in logic and practice in programming. Therefore, we believe programming produces more collected students, and can only be beneficial to the plastic growing mind.

From our estimations of our students learning programming, we believe that the vast majority of students have the ability to learn programming and finish the activities in the curriculum. While we agree that fewer would be able to further develop and program on their own, this may be due to the fact that programming requires more abstract thinking that is harder to apply in concrete situations as students' progress. Concepts become increasingly challenging and students' ability eventually come down to differing speeds in which abstract thinking develop. This is particularly true in the case of the P4 students, only aged 10, that we used in our school. We also found that students generally preferred to work in groups as they found it easier to open up and solve problems together. When students found the content difficult and are beginning to lose focus, we saw the use of teacher demonstrations or watching videos online the most effective method to motivate our pupils as they provide clear signals and reassurance to whether students are on the right track or may need to seek further aid from teachers.

8. REFERENCES

Collins, A. (1991). *The Role of Computer Technology in Restructuring Schools: Center for Technology in Education*. New York: Rochester.

Jonassen, D.H. (1996). *Computers in the Classroom. Mindtools for Critical Thinking*. Englewood Cliffs, NJ: Merrill.

Jonassen, D.H., Peck, K.L., & Wilson B. G. (1999) *Leaning with Technology: A Constructivist Perspective*. Upper Saddle River, NJ: Merrill.

Jones, B.F., Valdez, G., Nowakowski, J, & Rasmussen, C. (1994). *Designing Learning and Technology for Educational Reform*. Oak Brook, IL: North Central Regional Educational Laboratory.

Kong, S. C. (2016). *A Framework of Curriculum Design for Computational Thinking Development in K-12 Education.* Journal of Computers in Education,3(4), 377–394.

Miles, M. B., & Huberman M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. London: Sage.

Pisapla, J., Schlessinger, J., & Parks, A. (1993). *Technology Review of the Literature.* VA: Virginia Department of Education.

Quality Education Division. (2004*). Information Technology in Education – Way Forward.* Hong Kong: Government Logistics Department.

The Curriculum Development Council (CDC). (2001). *Education Commission's (EC's) Report on Learning for Life, Learning through Life - Reform Proposals for the Education System in Hong Kong.* Hong Kong: Government Logistics Department.

Wong, G., Ching, M., Tang, J. K. T., Zhang, L., Cheung, H. Y., & Chui, H. L. (2015). Impact of Computational Thinking Through Coding in K-12 Education: A Pilot Study in Hong Kong. *Proceedings of the 11th International Conference on Technology Education in the Asia Pacific Region.* Hong Kong: Hong Kong Polytechnic University.

Vygotsky, L. (1978). *Mind in Society: The Development of Higher Psychological Processes.* Cambridge, MA: Harvard University Press.

Use of GeoGebra Augmented Reality in Teaching and Learning Calculus: Volume of Solids in Integration

Xiajuan YE, Yeow Hwee Joel TEOH, Ching Ming KU
NUS High School of Mathematics and Science, Singapore
nhsyx@nus.edu.sg, nhstyhj@nus.edu.sg, nhskcm@nus.edu.sg

ABSTRACT

Through years' of teaching the topic, teachers notice that students have difficulties in visualizing 3D objects in a 2D plane. Augmented reality (AR) is the integration of digital information with the user's environment in real time (Alexander S. Gillis, 2022). GeoGebra is a dynamic mathematics software for all levels of education that brings together geometry, algebra, spreadsheets, graphing, statistics, and calculus in one engine and it is free of charge. This paper shares the use of GeoGebra Augmented Reality in teaching and learning integration of volume of solids in calculus. The lesson was conducted to enrich students' learning experiences and help students visualize the solids generated to weave together their background knowledge in order that an image could be made in their mind's eye to match what they learn. Students were allowed to use either their handphones, tablets or laptops (which may not have the AR feature) during the lesson to make them feel comfortable in class when learning something new. Two perception surveys, namely, pre-lesson surveys were conducted before and after the lesson with about 98% of the students who responded to the post survey agree or strongly agree that they could visualize the 3D objects better with the use of ICT, namely GeoGebra in this context. A two-tailed paired t-test was also conducted to conclude that the AR tool enhances the students' learning of the topic.

KEYWORDS

Augmented Reality, GeoGebra, Calculus, Integration, Volume of solids

1. INTRODUCTION

In their years of teaching calculus, especially the volume of solids, the authors understand that students find it difficult to visualize 3D solids in integration. Even with the aid of conventional Graphing Calculators (GC), students still find it not easy to imagine the solids as they cannot 'feel' or 'touch' the solids.

GeoGebra AR serves as supplementary tool to enhance, rather than to replace, our traditional instructional practices to achieve students' learning objectives. Handphones and Personal Learning Devices (PLDs, for example, laptops or tablets) are portable and easily accessible by students. GeoGebra 3D calculator is a free application which could easily be downloaded and accessible by all PLDs.

Our school is a specialized independent school in Singapore. We have a unique curriculum that is relevant, rigorous and inspiring to students who have the aptitude in and passion for Mathematics and Science. The students in our school are considered high ability learners in the country. The lesson was conducted for 190 high ability learners from the Year 4 cohort. Among the students, there are 35 female students and 155 male students. Before the actual lesson, related concepts and theorems were taught and explained. The students were asked to download GeoGebra 3D calculator (preferably at home), and the applets developed were sent to students before the lesson.

2. LESSON ENACTMENT

A file with instructions for the students was used during the lesson to facilitate teaching and learning. The following three examples were explored to help students visualize the solids formed.

Example 28 The figure below shows a solid torus (shaped like a donut) with radii r and R. By revolving the area of the region bounded by the circle $(x-R)^2 + y^2 = r^2$ about the y-axis, prove that the volume of the torus is $V = 2\pi^2 r^2 R$.

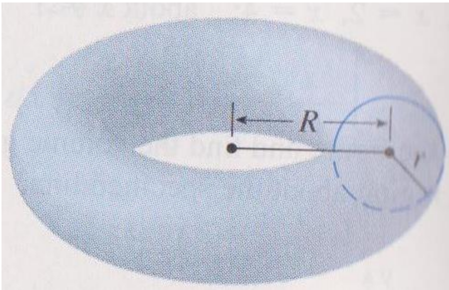


Figure 1. Solids formed by washers.

Screenshots from the corresponding applet can be found below:

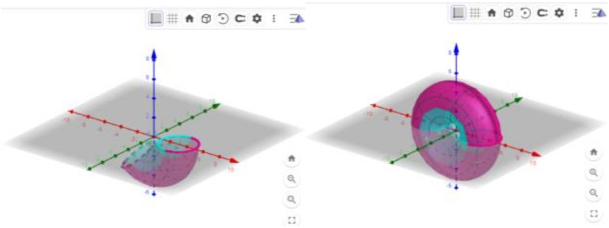


Figure 2. Solids formed by washers using GeoGebra.

Example 29 The figure below shows a solid with a circular base of radius 1. Parallel cross-sections perpendicular to the x-axis are equilateral triangles.

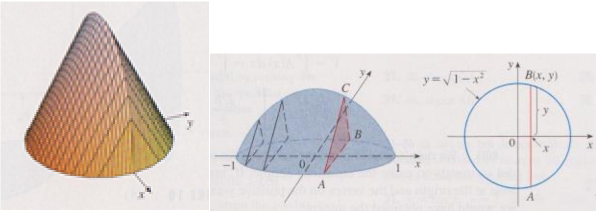


Figure 3. Solids formed with circular base and triangular parallel cross sections perpendicular to the x-axis.

(i) Find the area $A(x)$ of a typical cross-section at a distance x from the origin.

(ii) Hence, find the volume of the solid.

Screenshots from the corresponding applet can be found below:

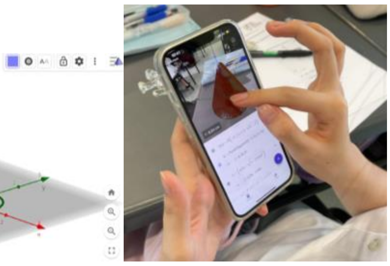


Figure 4. Solids formed with circular base and triangular parallel cross sections perpendicular to the x-axis using GeoGebra and its AR feature in classroom.

Example 31 Find the volume of the solid whose base is the region inside the circle $x^2 + y^2 = 9$ and the cross-sections taken perpendicular to the y-axis are squares.

No picture was given to students in the notes to help them visualize the solid formed.

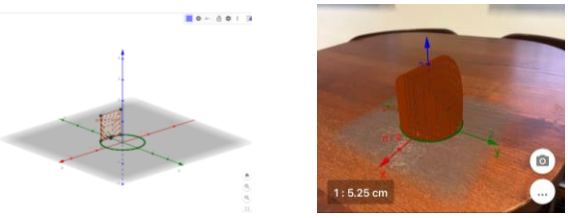


Figure 5. Solids formed with circular base and square parallel cross sections perpendicular to the y-axis using GeoGebra and its AR feature.

3. OBSERVATIONS AND FINDINGS FROM STUDENTS

When the hands-on session exploring AR feature in GeoGebra started, all students with no exceptions participated actively. Majority of the students used their handphones to visualize the solids formed, while some students used the GeoGebra applets in their PLD to visualize animation of how the solids were formed.

Two perception surveys, namely, pre-lesson surveys were conducted before and after the lesson. The following four questions asked.

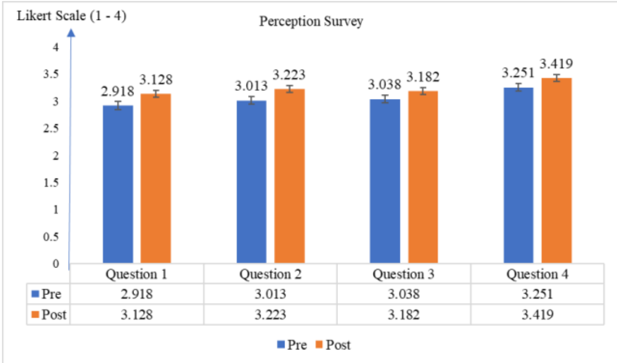
Q1: I learnt better with the use of ICT.

Q2: I enjoy using ICT in my learning process.

Q3: Lessons are more engaging and interesting when we use ICT.

Q4: I can visualize 3D objects better with the use of ICT.

A 4-point Likert scale was used with 1 presenting strongly disagree, 2 representing disagree, 3 representing agree and 4 representing strongly agree. The chart below shows the survey results.



From the chart above, we can clearly see that there is a good increase in the Likert scale when it comes to answering '*I learn better with the use of ICT*' after the lesson. In fact, about 90% of the students answers the post survey with agree or strongly agree. Also, more than 91% of the students enjoyed using ICT in their learning process. Nearly 90% of the students agree or strongly agrees that lessons are engaging and interesting when they use ICT. About 98% of the students who responded to the post survey agree or strongly agree that they could visualize the 3D objects better with the use of ICT.

When asked how the lesson could be improved students expressed their hope that the AR feature could be available in Android handphones. Some students were also eager to learn how to create their own examples with GeoGebra.

When asked what topics they think ICT will help them learn better, students listed out the following: transformation of graphs, complex numbers, vectors, and integration. The authors would like to include the use of ICT to enhance students' learning experiences in these topics in the coming years.

A paired t-test was conducted for the pre and post survey. The results are shown in Table 1 in the appendix. Based on the results of the t-test, there is a statistically significant difference between the means of the two samples, with a 95% confidence interval for the difference respectively. We can conclude that the AR tool enhances the students' learning of the topic.

4. CONCLUSION AND REFLECTIONS

Augmented Reality allows real and virtual objects to coexist in the same space, which helped our students in creating, feeling, and touching the solids which are otherwise impossible. The survey results show that most students gained better experience and enjoyed the lesson.

Research studies have identified AR as having immense potential to enhance learning and teaching (Billinghurst & Duenser, 2012; Dede, 2009). The use of AR in the classroom has shown to increase student motivation (Wernhuar Tarng & Kuo-Liang Ou, 2012; Johnson et al., 2011)

Even though we allow students to use their PLD to visualize the 3D solids formed if their phones could not work, we do believe that AR technology does have its advantage. It is of no doubt that AR technology overlays virtual objects onto the real world, creating a more immersive and contextual experience than a simple 3D visualization tool. This allows students to 'feel', 'touch' and 'make' the virtual objects in a more realistic way, making it a more engaging and dynamic experience than a simple 3D visualization tool. Furthermore, AR technology is often mobile. It can be used on smartphones, tablets, and other portable devices. This allows students to access AR experiences on-the-go and try it out wherever and whenever they would like to explore or learn.

Things could have gone through more smoothly if we had known that the GeoGebra AR feature was not available in some of the Android hand phones. Students who did not download GeoGebra 3D calculator beforehand used web version GeoGebra to observe the animation of how solids were formed. However, due to the slow Wi-Fi, they could not enjoy fully the wonderful visualization. On top of this, the teachers must develop the applets from scratch to cater to the curriculum. This is because there are not many resources available.

5. ACKNOWLEDGMENTS

We would like to thank our school for providing us the platform and the opportunity to explore innovations in teaching. We would also like to express our gratefulness to our colleagues who conducted the lessons and helped in collecting students' responses.

6. REFERENCES

Alexander S. Gillis (2022). *Augmented Reality.* Retrieved on Jan 28, 2023, from https://www.techtarget.com/whatis/definition/augmentedrealityAR#:~:text=Augmented%20reality%20(AR)%20is%20the,overlaid%20on%20top%20of%20it.

What is GeoGebra. Retried on Jan 28, 2023, from https://www.geogebra.org/about

Personal Learning Device. Retried on Jan 28, 2023, from https://www.moe.gov.sg/news/parliamentary-replies/20201102-personal-learning-device

Mark Billinghurst & Andreas Duenser. (2012). Augmented Reality in the Classroom. *IEEE: Computer, 45* (7), 42-49.

Chris Dede, et al. (2009). Immersive Interfaces for Engagement and Learning. *Science: 323, 66* (9), 66-69.

Wernhuar Tarng & Kuo-Liang Ou. (2012). A Study of Campus Butterfly Ecology Learning System Based on Augmented Reality and Mobile Learning. *2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education*, 62-66.

Yuen, S. C.-Y., Yaoyuneyong, G., & Johnson, E. (2011). Augmented Reality: An Overview and Five Directions for AR in Education. *Journal of Educational Technology Development and Exchange, 4(1), 119-140.*

Videos resources and instructions for students are available at: shorturl.at/ntFV3

What is Augmented Reality in Education? https://www.fingent.com/blog/augmented-reality-in-education-training-use-cases-and-business-benefits/

7. APPENDIX

Table 1. results of two-tailed paired *t*-test (post-pre)

	Q1	Q 2	Q 3	Q 4
P value	0.0024	0.0033	0.0459	0.0278
mean	0.21	0.21	0.14	0.17
95% confidence interval	[0.07,0.35]	[0.07,0.35]	[0.00,0.29]	[0.02,0.29]
intermediate values used	t=3.0566 df=305	t=2.9663 df=305	t=2.0043 df=305	t=2.2107 df=305
standard error	0.069	0.071	0.076	0.076

The Effect of Visual Programming Environments on the Development of Computational Thinking and the Influence of Self-Regulating Ability in Upper Primary School Children

Maxim VANSTEENKISTE[1*], Nardie FANCHAMPS[2]
[1]Sint-Godelievecollege, Belgium
[2]Open University, The Netherlands
vansteenkiste.maxim@outlook.com, nardie.fanchamps@ou.nl

ABSTRACT
To develop computational thinking (CT), a variety of programming environments can be selected from. Apart from a focus on developing specific skills, flanking pedagogical conditions can play an equally important role. The aim of our study is therefore to investigate, among ninety upper grades primary school students in Belgium, the effect of learning to program in a visual programming environment on the development of CT, and to what extent this effect is influenced by self-regulatory ability. To investigate this, a quantitative quasi-experimental research design was adopted using two intervention groups that applied either a visual programming environment with tangible output or a visual programming environment with on-screen output. To determine the difference in the development of CT and the influence of self-regulatory ability, a pretest-posttest design was used. Our initial indications show that significant differences in CT development are noticeable between the two programming environments, but that learners' self-regulatory ability had a weak but no significant impact on the development of CT. Based on the results obtained, it is recommended to further investigate the importance that can be attributed to regulating thoughts, feelings, abilities and behaviour, and that cooperating with the environment and handling feedback when developing CT are essential prerequisites.

KEYWORDS
programming, computational thinking, self-regulating ability, tangible output, visual output

1. INTRODUCTION
Due to increased technologisation, our society has changed considerably in the 21st century. As a result, the importance of conceptual and metacognitive skills in the fields of communication, collaboration, socio-cultural awareness and digital literacy has grown significantly (Thijs et al., 2014). As a consequence, sufficient attention must be paid to the development of the required skills in education in order to help students grow into fully-fledged participants in 21st-century society (Voogt & Roblin, 2010). One of the fundamental skills of digital literacy is computational thinking (CT). CT is the process of reformulating problems in a way that allows Information and Communication Technology (ICT) applications to solve them (Wing, 2006). It stimulates the ability to reason, abstract and solve problems (Wing, 2006). A suitable learning activity to develop CT is programming (Lye & Koh, 2014). To teach the basics of programming, visual programming environments are often used in primary education due to their accessibility for beginners (Lye & Koh, 2014). Visual programming environments can differ in terms of output, distinguishing between a) on-screen output where the output is only displayed on a computer screen or tablet, or b) tangible output that allows learners to programme physical objects (Bocconi et al., 2016). In the latter variant, programming activities are made very concrete for learners and they are encouraged to touch, explore and test objects. Recent studies (Zheng et al., 2018) indicate that the self-regulatory abilities of students can influence the extent to which ICT-skills are acquired in digital environments. Students who have difficulties to self-regulate their learning process will experience more difficulties in digital learning environments. Self-regulation means that a learner is able to act independently and take responsibility when performing learning tasks (Zimmerman, 2008).

2. PURPOSE OF THE STUDY
This research aims to investigate the effect of the type of programming environment and the associated characteristic differences on the development of computational thinking and self-regulatory abilities. The goal is to explore to what extent self-regulatory skills influence the growth of computational thinking (CT skills) in upper elementary school students (9-12 years).

3. METHOD
A quantitative, quasi-experimental study was conducted to determine the potential effects of the type of programming environment on the development of CT and self-regulatory abilities. Therefore a pretest-posttest design was applied. Primary schools in Flanders, Belgium were approached to participate in the study. This research used existing class groups with the aim to keep the existing class dynamics intact. The class groups were assigned into one of the two intervention groups: visual programming with on-screen output or tangible output. Children from both experimental groups were offered six programming lessons of one hour each as an intervention. To determine the level of CT, the validated Computational Thinking Test (CTt) was used (Román-González et al., 2017). To measure self-regulatory abilities, the Self-Regulatory Tool was used (Vandevelde et al., 2013).

4. MATERIALS
Regarding the programming intervention, the visual programming environments Lego WeDo 2.0 with tangible output and Code.org with on-screen output were applied. Lego Wedo 2.0 is a robotic system for designing and building programmable artefacts and can be programmed

using its own application by dragging and dropping visual command blocks in a worksheet. Code.org is a visual programming environment based on Javascript with purely an on-screen output. It can be programmed by dragging and dropping definable code blocks to program. Concepts practised with both programming environments are: sequences, loops, conditions, functions, variables, decomposition, pattern recognition, abstraction and algorithmic thinking.

5. FINDINGS

Table 1 displays the pretest-posttest results. From the data, it can be deduced that students, based on applying both visual programming environments Lego WeDo 2.0 and Code.org, show higher average CT scores on the posttest in a comparison with the pretest.

Table 1. Means and Standard Deviations of CT

Variables		Pretest		Posttest	
	n	*M*	*SD*	*M*	*SD*
Total CT-test	74	11.41	3.11	13.55	3.61
Visual output	44	12.77	2.67	15.36	3.10
Tangible output	30	9.40	2.62	10.90	2.51

Note. CT = computational thinking; *M* = mean; *SD* = standard deviation.

Table 2 displays the differences in CT development between both experimental conditions. It is shown that there is a significant difference between the two intervention groups regarding the overall development of CT. Students who programmed in a visual environment with visual output show a significant increase in the total score of the CT-test used. Regarding the development of CT-subskills, we indicate that the group who applied visual programming with visual output shows an increased development in terms of 'sequencing' and 'completion', but these differences are not significant. The group who applied visual programming with tangible output shows an increased development in 'debugging', but this difference is also not significant.

Table 2. Development CT-(sub)skills pre- and posttest

	VO		TO				
Variables	*M*	*SD*	*M*	*SD*	*t*	*p*	*d*
Total CT-test	2.59	2.11	1.50	2.19	2.15	.03	.51
Sequencing	2.18	1.32	1.50	1.83	1.75	.09	.44
Completion	.18	1.69	-.40	1.83	1.41	.16	.33
Debugging	.23	1.01	.40	1.45	-.60	.55	-.14

Note. CT = computational thinking; VO = visual output; TO = tangible output; *M* = mean; *SD* = standard deviation; *t* = t-value; *p* = p-value; *d* = effect size

Regarding the development of self-regulatory abilities, we deduce that the differences between the two experimental groups are not significant $t(72) = 1.94$, $p = .08$. The results of a correlation-regression analysis show that there is a weak correlation between the development of self-regulatory abilities and the development of CT skills, r (73) = .06, $p = .31$. Subsequently, it was examined whether the development of the self-regulatory abilities could be a good predictor for the development of CT skills. The results showed that self-regulatory abilities are not a good predictor for the development of CT skills, $R^2 = .004$, $F(73) = .256$, $p = .61$.

6. CONCLUSION

Our research indicated that visual programming environments can play a prominent role in the development of CT, where an on-screen output can lead to a significant development on CT in a comparison to environments with a tangible output. There was a weak, but not significant, correlation found between self-regulatory ability and the development of CT.

Due to a number of limitations, the results of this study should be approached with some caution. For example, it was expected to involve 90 students in the study. Due to the fact that not every student received permission, the number of participants was limited to 74. This also had implications for the distribution of the intervention groups. Fewer students were given permission in certain classes, as a consequence the intervention groups were not evenly distributed. As a result, more data was collected from the group that programmed with a viusal on-screen output. Working with existing class groups also means that certain characteristics of the group may have influenced the results of the study. A recommendation for follow-up studies is therefore to replicate this study with larger research groups that are more evenly distributed. It may be a possibility to divide pupils non-randomly, for example according to level and to test to what extent cooperation has an influence on the development of these skills.

A second limitation can be found in the CTt used to measure the development of CT skills. This test measures the concepts of CT and is essentially task-based. This test thus takes less account of the wider development of CT, such as creativity and inquiry-based learning of the student. This may have contributed to a bias in favor of the students who learned to program with a visual on-screen output. However, the CT test used in this study is one of the few validated tests suitable for assessing CT skills among the target population studied. Focus on the development of measuring instruments that reflect the broader characteristics of CT, such as creativity and exploration is therefore an absolute necessity for further research.

A third limitation is the limited time span of the study, which may have been too short to determine the long-term effects of both programming environments. The difference between pre- and post-test was only two months. In this period we already see a remarkable improvement of the CT skills compared to the pretest, but it is possible that this short duration favored the group that learned to program with the visual on-screen output. Due to the explorative and experimental characteristics, the effects of the programming environments with a tangible output may only be visible in a long term time span. If we provide students the opportunity to work with both programming environments for a full academic year, the results may be different. This is also confirmed by the debriefing conversations with the involved facilitators who have indicated that the group that programmed with a tangible output was catching up in the later sessions and that they may have mastered some of the strategies of CT in a more effective way. However, we cannot confirm this suspicion on the basis of these results. An important recommendation may therefore be to extend the period between the pre-test and post-test, and to offer students more time on task. In this regard, the most ideal scenario would be if students could practise for a full academic year.

A fourth limitation is the self-regulation test that was used to map the self-regulatory abilities of the students. Students had to assess themselves and it is possible that this was still somewhat difficult for students aged 9-12 to do this accurately. In the post-test we've noticed that a number of students rate themselves remarkably lower than in the pre-test. The pre-test was administered at the beginning of the academic year, which may have made it more difficult to correctly answer each question and it may have been the case that certain students have overestimated their abilities. This may have influenced the results of the study. We suggest further research with larger ande more equally divided groups. We also recommend further research into developing validated CT-tests which have multiple approaches to solving a problem, which can also stimulate the creative aspect of CT.

7. REFERENCES

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site). https://doi.org/10.2791/792158

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

Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? criterion validity of the computational thinking test. *Computers in Human Behavior, 72*, 678-691. https://doi.org/10.1016/j.chb.2016.08.047

Thijs, A. M., Fisser, P., & van der Hoeven, M. (2014). *Digitale geletterdheid en 21e eeuwse vaardigheden in het funderend onderwijs: een conceptueel kader*. SLO, Enschede. http://downloads.slo.nl/Repository/21e-eeuwsevaardigheden-conceptueel-kader.pdf

Voogt, J., & Roblin, N. P. (2010). 21st century skills. *Discussienota. Zoetermeer: The Netherlands: Kennisnet*, *23*(03), 2000.

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

Zheng, L., Li, X., & Chen, F. (2018) Effects of a mobile self-regulated learning approach on students' learning achievements and self-regulated learning skills. *Innovations in Education and Teaching International, 55*(6), 616-624. https://doi.org/10.1080/14703297.2016.1259080

Zimmerman, B.J., (2008). Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American educational research journal*, *45*(1), pp.166-183. https://doi.org/10.3102/0002831207312909

透過自主學習策略運用線上「編程數位說故事」支援以英語為外語或第二語言的小學生學習英文寫作：由資訊科技科入手

蒙韋綸[1]，黃智仁[2*]，馬韵斯[3]
[1,2]鳳溪第一小學，中國香港
[3]香港教育大學，中國香港
mungsir@yahoo.com.hk，astomwong@gmail.com，yma@eduhk.hk

摘要
本文是一篇以小學四年級的同學為對象，透過自主學習策略使用線上編程平台 Scratch 進行「數位說故事」學與教活動，實現英文科及資訊科技科跨學科活動。這個跨科單元課程在正規課堂進行，包括英文課堂及資訊科技課堂，分別於英文課堂教授英語語法、寫作，於資訊科技課堂動手製作數位故事，引導學生提升自主學習能力及學習成效，探索創新教學法，一方面發展四年級學生的英文寫作能力，另一方面激發他們對學習編程以發展運算思維的興趣和信心。

關鍵字
運算思維/計算思維；編程教育；自主學習；數位說故事；英文寫作

1. 前言
後疫情時代，自主學習是一項在面授課堂、網上課堂都同等重要的學習策略。本研究以小學四年級英文科為本，由資訊科技科協作，完成為期約四星期的英文寫作課程。英文科、資訊科技科教師與學術研究共同設計課程並開發教材，教授小四英文科「My Favourite Festival 我最愛的節日」此一單元，在英文課堂上學習相關詞彙、語法及寫作技巧，並在資訊科技課堂上利用 Scratch 網上編程平台完成編程和製作動畫，創作「My Favourite Festival」數位小故事，最後在英文課堂上向同學展示自己的作品，完成同儕點評及自評 (Ma et al., 2022)。在自主學習策略和教師的指導下，學生在過程中逐步學習制定學習目標、安排時間、選取學習策略、聽取聽取同儕及教師的意見、尋求協助、溫故反思，逐步按計劃完成學習目標 (Zimmerman, 2011)。

近十年來，香港學界致力於資訊科技科內推動運算思維能力發展。近年更強調編程教育應與其他科目練繫以解決現實生活中的問題 (EDB, 2020)。隨著資訊科在運算思維編程的教學日趨成熟，學校開始探索並推動跨科合作以產生學科之間的協同效應。此外，過去三年的疫情推動了 K-12 網上學習 (online learning) 混合學習 (blended learning)，「自主學習」亦成為促進學生學習效能的熱門話題。如何透過資訊科技科與英文科結合的運算思維課程提升同學的「自主學習」能力成為了一個項值得探討的方向。

2. 文獻回顧

2.1. 自主學習 (Self-regulated Learning)
Knowles 於 1975 年提出了「自主學習」的概念，並於近年來成為了其中一個熱門的研究議題。其定義是學習者於他們的學習過程中扮演主導者的角色。而主導者需要選取和運用適當的評估並了解自己的學習需要，並根據結果訂立學習目標。

2.2. 數位說故事 (Digital storytelling)
研究發現，透過利用多媒體平台進行數位說故事活動，可以幫助學生學習英語，提升學習動機，創意 (Liu et al., 2018)。利用編程數位說故事，能激發學生在編程創作活動中學習正規英文寫作的潛能(Burke & Kafai, 2010)。

2.3. 運算思維 (Computational Thinking)
推行跨學科課程的理念是希望同學透過體驗英文科與資訊科的跨科課程，掌握整個設計思驟步驟。另一方面亦期望讓參與課程的同學實踐於資訊科課堂上學習到的運算思維知識(CT Concept)、運算思維技能(CT Practices) 及運算思維視野 (CT Perspectives)並展示他們的學習成果(Kong & Lai., 2022)。修畢課程後,同學們便有能力運用已有知識把創新想法利用數碼創意轉化並製作出作品的原型。課程對象為四年級的同學,他們在課程前並不需要掌握任何編程的技巧和知識。

3. 課程目標及預期課程學習成果
本課程的目標是為了透過英文科與資訊科的跨科合作，培養學生於資訊科技範疇的興趣、技能和態度以提升同學的數碼創意。同時亦希望同學在完成作品的過程中產生跨科學習的協同效應以提升同學學習英文科和資訊科知識的興趣，推動自主學習的發展。

同學於修畢本課程後應該具備以下六項學習成果:可以展示出對於運算思維、編程技巧和動畫製作的基本原理知識和相關技能;對於設計思維步驟有基本的認識;可以開發簡易動畫作品;遇上各種困難和問題時,仍然可以使用成長形心態面對;可以有效地應用和整合語文知識、溝通技巧、運算思維技巧、創意思維技巧及解難能力以處理專案開發的過程;及根據課堂的內容而自主於家中學習更深入的內容。

4. 編程說故事平台 – Scratch

Scratch 是一個「所見即所得」的線上免費編程平台，以大量的圖像化介面配合方塊式編程簡化以往繁複的作品製作過程。教師可以針對他們的教學目標使同學具像化地即時了解編程的效果，提升同學學習編程的興趣和動機。使用平台時，教師需要對平台比較熟悉對各個課題有較深入理解，才能保證教學效能。學生透過 Scratch 使用簡單的編程模塊，便創作不同的互動動畫、遊戲或故事，甚至能夠應用人工智慧的技術在他們的作品中。在本次教學實驗中，學生利用 Scratch 創作自己的動畫故事。

圖 1 學生作品範例

5. 量測學生學習成效的方法

為了可以更全面地評估學生的學習成效，跨科課程評估方式包括了總結性評估和進展性評估。而進展性評估所佔的比重會較總結性評估多是基於課程的教學策略希望透過動手動腦的課堂方式，結合自主學習概念，培養學生於資訊科技範疇的興趣、技能和態度以提升同學的數碼創意。

5.1. 課堂前測與後測(總結性評估)

跨科課程通過前測與後測了解學生於課堂上的的需要和對課堂上的知識掌握程度。前測與後測主要可分為兩大部分：第一個部分主要是讓教師了解學生於英文學習、自主說故事和運算思維賦權三大範疇分數為何；第二個部分是英文寫作測試，讓教師了解學生在看圖說故事和英語文法兩大範疇的掌握程度。

5.2. 口頭回饋（進展性評估）

學生於最後一節的英文課中演示他們的創意寫作及互動說故事成品。教師會為同學演示表現作出改善的回饋和評分，讓他們可以在將來再次準備成果匯報時可以有更好的表現。

5.3. 自評及同儕互評（進展性評估）

除了教師為學生的表現作出即時的回饋外，同學需要反思自己在課堂學習經歷中的表現和可以改善的部分。因為他們需要認識自己在學習方面的不足才能提升自主學習的技巧和動機。

另一方面，同學亦需要互相評估。讓他們在同儕互評的過程中掌握改善學習方法和客觀地了解自己的表現。

6. 教學設計

6.1. 課程主題： My Favourite Festival

以「我最喜愛的節日」為跨學科課程的主題是因為學生們都擁有經歷不同節日的體驗。他們對於不同節日都有各自的感受和回憶，因此選取緊貼生活經驗的「我最喜愛的節日」為主題，以提升同學學習動機。

6.2. 教學設計模式

課程統整的方式至今仍然沒有統一的歸納方法和系統。話雖如此，我們還是可以從「學科界限」的角度為課程統整的方式分為「單一學科」、「跨學科」、「科際融合」和「超學科」四大類（游家政，1999）。

由於本課程透過「我最喜愛的節日」的主題統整其自然連結了英文科和資訊科的課堂和教學活動，但是學科之間的界限清晰。英文科負責教授同學與課題相關的英語知識、文法和創意寫作的部分，資訊科負責教授學生把創意寫作轉化為互動說故事作品。

7. 教學流程

跨科課程先以英文科「My Favorite Festival」單元教授英語詞彙、語法和寫作技巧，再配合資訊科並透過「數位說故事」的單元教授同學如何使用 Scratch 把英文寫作轉化為有趣生動的互動故事。最後再於英文課堂上演示他們的作品，並收集同儕及教師的意見，讓同學透過英文寫作和編程向大家分享他們最喜歡的節日。

8. 教學反思與建議

8.1. 教學挑戰

本課程結合了各種不同的理論和框架以設計課程的流程、體驗和內容。從課堂上同學的反應和課堂後的作品便可以了解到同學對於可以參與課堂是十分高興的。完成課堂後，同學亦可以展示課堂最初要求的能力標準及知識概念。總括而言，課堂十分順利地完成了它的目標。縱雖如此,推行跨科課堂的過程並非一帆風順。當中亦遇上很不少的困難和挑戰,幸好課程最後把這此問題都逐一解決。

8.1.1 疫情下教學策略 – 混合式學習 (Blended Learning)

由於本課程結合了英文科與資訊科的內容，因此編程課的課堂時間較以往更為有限。為配合兩科的教學進度並深入地照顧每位同學的學習差異，所以同學可以於 Edmodo 上預先下載學生材料(例如: 教學筆記、簡報、編程檔案等等)並按照自身進度預先了解課堂的內容及嘗試編程。上課時,同學便可以一起討論在編程時遇到的問題並以更深入的角度進行教學。

因香港疫情影響,學生由以往可以全日在校改為只能半日在校。所以,學生的課外延伸活動安排受到嚴重影響。同時,教師接觸同學的時間亦大幅減少。這意味著教師可以安排個別小組進行相關研討及工作的時間都會較少,進而削弱了相關教學的力度。

8.1.2 照顧學習者差異

眾所周知,資訊科課堂上佔課時比例最多的部分便是協助同學制作課堂相關的作品了。因為同學需要設計作品的內容並跟據教師的指示完成所有編程步驟。但是由於並非每位同學家中都有足夠的資源和支援協助同學。因此製作成品所需要的時間亦較以往更長。

8.1.3 資訊科與英文科的跨科配合

推行課程的其中一個難點便是不同科組之間的教學進度需要互相配合，因為各班上課的時間各不相同，有時候會遇上某班同學的資訊科或英文科剛好遇上假期而引致各班的教學進度不一。

8.2. 教學建議

8.2.1. 積極使用和配合電子教學平台

針對教學困難部分提出的第一項問題的建議是在教學時積極使用和配合電子教學平台。因為同學在面授課時減少而促使課程的效果減弱。若於課程規劃時便已經使用和配合電子教學平台，除了讓教師可以更有效地運用面課堂的課時外，亦可以培養同學自動在網上尋找課程相關的資料以更有效地推動同學發展「自主學習」的技能。

8.2.2. 為同學提供更多的製作成品的資源庫

雖然在推行課程前已為同學準備好參考圖庫，但在實踐課程的時候發現圖庫的存量可以有更多的選擇讓同學減少尋找需要資料的時間。由於課程對象的同學年紀尚幼，需要花費更多的時間於網上尋找資源以完成他們的作品。同學特別在尋在角色圖片、動畫背景中花費的時間較多，而老師於課程期間亦發現同學們有很多有趣的創意。他們會非常在意圖像是否能準備地表達他們的意思，因此在選圖方面同學表現得很嚴謹。

8.2.3. 為學生的作品加上更具體的框架

解決第二個教學難點的另一個方向便是為同學最終成品加上更具體的框架使同學可以在有限的時間內完成他們的作品。因為同學有很多有趣的點子想要實踐，因此他們在製作成品的過程中不停加入新的點子，使成品的製作時間不停增加。因此可以參考語文科作文時的寫作限制，讓同學可以在有限的時間和框架內發揮他們無限的創意。

10. 参考文獻

Burke, Q., & Kafai, Y. B. (2010, June). Programming & storytelling: opportunities for learning about coding & composition. In *Proceedings of the 9th international conference on interaction design and children* (pp. 348-351).

EDB (2020). *Computational Thinking – Coding Education: Supplement to the Primary Curriculum*

Kong, S. C., & Lai, M. (2022). Computational identity and programming empowerment of students in computational thinking development. *British Journal of Educational Technology*, *53*(3), 668-686.

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.

Kong, S. C., & Ma, Y. T. (2022). Principals' Perceptions of Online Teaching and Learning in School After the Outbreak of the Pandemic. *International Journal of Distance Education Technologies (IJDET)*, *20*(1), 1-18.

Liu, K. P., Tai, S. J. D., & Liu, C. C. (2018). Enhancing language learning through creation: The effect of digital storytelling on student learning motivation and performance in a school English course. *Educational Technology Research and Development*, *66*(4), 913-935.

Ma, Y. T., Kong, S.C., & Sun, D. (2022, November). Using Digital Storytelling on Scratch to Support Primary School EFL/ESL Students' Writing: A Self-regulated Learning Approach. In S. Iyer, et al. (Eds.), *Proceedings of the 30th International Conference on Computers in Education ICCE 2022* (pp. 714-718). Kuala Lumpur, Malaysia: Asia-Pacific Society for Computers in Education.

Zimmerman, B. J., & Schunk, D. H. (2011). *Handbook of self-regulation of learning and performance*. New York: Routledge.

游家政（1999）。學校課程的統整及其教學。課程與教學季刊，3(1)，19-38。

Using Scratch Digital Storytelling through Self-Regulated Learning Strategies to Support Primary EFL/ESL School Students in Learning English Writing: From the Perspective of Information Technology Subject

Wai Lun Mung [1], Chi Yan Wong[2*], Yunsi Tina Ma[3]
[1,2]Fung Kai No.1 Primary School, Hong Kong, China
[3]The Education University of Hong Kong, Hong Kong, China
mungsir@yahoo.com.hk，astomwong@gmail.com，yma@eduhk.hk

ABSTRACT
This article is to share the experience of promoting the school-based interdisciplinary course of computational thinking by teaching the interdisciplinary course of computational thinking to primary four students to develop their English proficiency, computational thinking, design thinking and positive thinking. All classes are based on the 7 curriculum principles proposed in the K-12 curriculum design framework for the development of computational thinking and the 7-step theory of the TPACK teaching method, and are compiled in conjunction with the content of the school-based English curriculum. The focus of classroom teaching is to coordinate the information subject with the teaching progress of the English subject and help students master the skills of independent learning to improve their learning motivation.

KEYWORDS
Self-regulated learning approach, information and technology education, English education, learning and teaching strategy, computational thinking

運算思維教育的教學反思：

運用編程結合運算思維提升學生的數學學習興趣與表現

許文星 [1*]，黃玉茹 [2]
[1,2] 英皇書院同學會小學第二校，香港
mshsu@kcobaps2.edu.hk, yywong@kcobaps2.edu.hk

摘要

本文闡述如何引導領袖服務生運用運算思維設計互動學習程式以提升能力較弱學生的數學學習興趣與表現的實踐例子。從四名 10-11 歲「數學特攻隊」服務生的經驗作引入，他們發現數學學習表現較弱的同學普遍未能熟練掌握「九九乘數表」，如何有效解決當前的數學學習困難是首要考量。他們希望透過運用編程結合運算思維設計應用程式「乘法遊戲王」來推動同儕學習，提升學習興趣和表現。設計小組藉著老師引導，運用 MIT App Inventor 2 設計趣味導向的「九九乘數表」應用程式，通過運算思維實踐解決問題，從而鞏固運算思維的概念，並提升學會學習的能力。

關鍵字

運算思維；MIT App Inventor；編程；數學；乘法

1. 前言

2018 年的一份研究文件指出考試導向的文化嚴重影響了學生的學習經歷，亦大大降低了他們的學習興趣(Chan et al.，2018)。然而，二十一世紀是科技躍飛的年代，若學校課程能運用資訊科技的優勢，教導學生有效地運用資訊科技以解決問題，摒棄考試導向並利用趣味導向(interest-driven)幫助學生進行學習，使他們不只是資訊科技的使用者，而是積極的參與者，相信不但能提升學生的學習興趣，更能提升他們的學習表現。於課程發展方面香港教育局亦甚具遠見，教育局課程發展處於 2017 年就建議在小學階段引入編程教育來培養學生的計算思維能力，希望通過適當設計的學習活動，為學生提供獲取和應用計算思維和編程技巧的機會（課程發展議會，2017）。而學者亦認為運算思維(computational thinking)課程應按照「趣創者理論」(Interest-driven Creator Theory)設計，把學生培養成解決數碼問題的創造者（Kong，2016）。故此，本文將探討如何利用運算思維教育引導學生發現和分析問題，並利用創意製作一個有趣互動的應用程式以解決學習中所面對或發現的問題。

2. 教學實踐過程

2.1. 從身邊發現和分析問題

學校在小息時段設有「數學特攻隊」服務，設立目的是協助數學老師幫助成績稍遜的同學解決學習數學的困難，希望透過朋輩互動協作，來推動同儕學習。於學期末的檢討會，其中四名 10 - 11 歲的學生透過對數學老師的訪問並綜合他們的服務經驗，他們發現這些需要輔助的同學，不論是哪個年級，往往因為計算速度緩慢而影響學習進度及表現，在學習上大打折扣。他們大多數未能掌握或不熟練數學基本功「九九乘數表」，因而大大影響了他們在數學科各方面的學習表現。

2.2. 資料搜集和尋求協助及意見

在日常運算思維的教學中，老師經常鼓勵學生除了從身邊事物或日常生活中觀察和發現問題外，也要透過閱讀報章、時事新聞主動搜集資料，為自己的觀察及發現加以佐證，以確保解決方案循正確的方向發展。

在香港大學主修數學的前立法會主席曾鈺成曾於 2017 年在報章中撰文說明乘數表的重要性，文章中他更強調連一直不提倡背誦文化的英國政府，也決定要把乘數表重新納入小學課程中(曾鈺成，2017)。透過上述報導，學生確立了掌握「九九乘數表」對數學學習的重要性。另外，通過訪問及尋求數學老師的指導，學生得悉小學生於二年級開始學習乘數表課程，且受訪老師反映普遍低年級學生都認為背誦乘法口訣枯燥乏味，對這課題的學習興趣普遍不高。

2.3. 針對需求，創設方案

透過綜合分析整體搜集的資料及老師的意見後，其中四名服務生決定組成設計小組，認為可以透過 MIT App Inventor 2 設計一個應用程式，將它作為「數學特攻隊」服務生的工作工具，以解決所發現的問題：如何使同學更容易掌握「九九乘數表」。設計小組認為該應用程式需要具備以下特點方能更有效地協助他們服務及幫助同學學習:

1. 應用程式需要趣味導向(interest-driven)，並具備獎勵機制以吸引同學進行學習。
2. 應用程式能夠引導同學建構出乘法的概念。
3. 應用程式需要有計分功能，讓服務生和受支援的同學都能清楚了解學習進度。
4. 應用程式需要協助同學進行自學，以減輕「數學特攻隊」服務生工作時的負擔。

2.4. 編寫應用程式及實踐意念

設計小組把應用程式定名為「乘法遊戲王」，並把玩法設計成「有獎問答比賽」的形式，以吸引受支援的

同學能利用應用程式積極學習。在編程的過程中，學生需要應用不同的運算思維概念(CT concepts)，例如<事件 event>、<運算子 operator>、<條件 conditional>等設計應用程式；通過運算思維實踐(CT practices)解決問題，包括<測試及除錯 testing and debugging>、<反覆構思及漸進編程 being incremental and iterative>、<算法思維 algorithmic thinking >等，確保應用程式能夠順利運作，並有明確的規則及步驟清楚表達問題的解決方法(Kong, 2016)。學生在應用程式的主頁可以按自己的能力和學習需要選擇乘法課題，進入該課題後，透過作答乘法題目獲取分數，當累積到指定的獎勵分數後，程式就會隨機在 5 個小遊戲中選一個作為獎勵派給同學遊玩。如果同學忘記了該課題的乘數表，他們可以按「我要幫助」按鈕，程式就會顯示該課題的乘數表以幫助同學。於熟記乘數表後，他們可以按「關閉幫助」按鈕，以隱藏有關乘數表。學習過程中，學生隨時可以按「回選單頁」按鈕，回到目錄頁，並按自己需要選擇其他課題繼續學習。應用程式利用了「等於、不等於、和」、「如果 – 則」等條件程式碼，而計分功能則利用「取得初始值」程式碼把分數分享到應用程式的不同介面中。「乘法遊戲王」的介面及功能可參閱圖 1 和 2。

圖 1 「乘法遊戲王」課題介面及功能

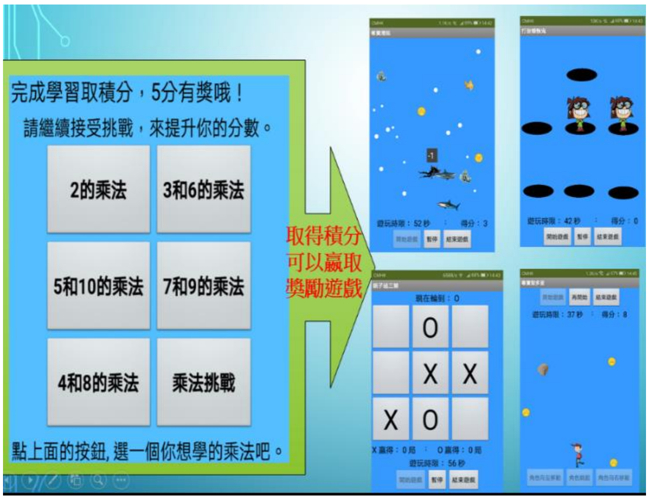


圖 2 透過學習取得積分，贏取趣味遊戲時間

2.5. 設計配合數學課程編排，學習事半功倍

根據數學老師提供的意見，學生學習「九九乘數表」若能依課程的編排，將更有效地幫助他們熟記九九乘法，而將來學習除法時，也就更容易產生學習遷移，對提升學生數理能力影響深遠。乘法課題經過特別的編排以便幫助同學們更有效地掌握所學，因為不同數字的乘數表是有所關聯的，例如「3 和 6」、「4 和 8」、「5 和 10」以及「7 和 9」。要令使用者在按下該課題的按鈕後，則只出現有關數字的乘法表，設計小組需要為應用程式寫下特別條件的程式碼。

例如當按下「7 和 9 的乘法」課題按鈕時，設計小組需特別編程，使應用程式先執行設定乘法題目的程式碼，即產生被乘數和乘數的數字。要使兩個數字的其中一個成為符合課題的數字，即兩個數字的其中一個必需是「7 或 9」，學生想到只需使程式先設定第一個（被乘數）和第二個數（乘數）為 1 - 9 的隨機數字，然後再透過條件程式碼「如果 – 則」反復檢查所產生的這兩個數是否符合條件以組成指定課題的乘法題目。「乘法遊戲王」程式碼示例可參閱圖 3。要成功產生符合選擇課題「7 和 9 的乘法」的題目，主要考慮以下三個情況：

- 情況 1. 如果第一個數字是 = 7 或是 = 9，則產生乘法題目。
- 情況 2. 如果第二個數字是 = 7 或是 = 9，則產生乘法題目。
- 情況 3. 如果第一個數不是 7 和 9，以及第二個數也不是 7 和 9，則程式會再執行整個設定乘法題目程序一次，直至產生的隨機數字中有 7 或有 9 的乘法題目為止。

圖 3 「乘法遊戲王」產生符合選擇課題題目的程式碼

2.6. 實際測試

服務生在執行「數學特攻隊」工作時，受助學生協助測試了他們所設計的「乘法遊戲王」應用程式。在實際應用時，他們發現了一些問題，並收集了使用者的若干回饋，總結如下：

1. 應用程式因為所使用的介面較多，開啟過多視窗，運作時間越長越遲鈍。

2. 部份能力較高的學生很容易就能取得獎勵積分，故此要考慮為程式加入可以調整獲獎積分的設定。

3. 使用者希望應用程式可以有帳戶登入的功能，並在每次登入後可以記錄及顯示他們的個人積分。

4. 使用者希望應用程式可以讓他們自由選擇獎勵遊戲，而不是得到指定積分後隨機派發獎勵遊戲。

3. 反思

完成這個應用程式和相關活動後，老師鼓勵同學對製成品和活動過程進行反思。設計小組都覺得實際測試的過程讓他們獲得了最多的啟發，因為透過觀察程式的實際運作、使用者的反應，以及收集使用者的意見，使他們產生了更豐富和多元化的意念，從而能作出進一步的改善計劃。由此可見，設計者與用戶之間的理解是創新設計的基礎（Kelley & Kelley, 2015）。

在運算思維學習的過程中，老師的適當引導和鼓勵有助提升學生的學習效能。在發現問題階段，四名服務生都能綜合他們整年的經驗提出各方面的問題，雖然資料豐富，然而欠缺聚焦，老師在這個階段適當地介入和引導學生把各問題作分類與甄選，實在有助他們找出癥結所在，並確保他們稍後討論的解決方案循正確的方向發展。在實際測試並取得使用者的回饋後，學生難免會感到失望和沮喪，因為現實與期望總是有落差的，面對使用者的回饋和意見往往會令學生對自己的編程能力產生質疑，老師在這階段的鼓勵和引導尤為重要。

老師應該讓學生明白，雖然他們現階段的能力未必能即時為應用程式作出適切的修訂，以回應使用者的回饋及需求，但運算思維教育的重點並不是教授學生製造出完美的製成品，而是在學習過程中培養學生<反覆構思及漸進編程>的態度，從而養成持續審視問題的習慣，學會檢討製成品的優缺點及提出改善方法，也要懂得反思整個過程中的所學，從而提升學會學習的能力。

雖然實際測試讓設計小組學生知道他們設計的應用程式並非完美作品，而且仍有很多需要改善的地方，然而他們也觀察到應用程式的確可以提升受支援學生的數學學習興趣。相較過去沒有使用「乘法遊戲王」時，現在的學習過程中充滿了喜悅和歡笑聲，而這些受支援的學生不但更能準時出席活動，且他們對學習也表現得更積極和主動。然而，若能紀錄和統計受支援學生的前後成績表現差異，將能更清晰和具體地反映出應用程式對學生的實際幫助，這亦是設計小組在改善應用程式的同時，未來將要執行和跟進的研究方向。

4. 總結

乘法在數學學習領域中，擔當著十分重要的角色，因為它與除法、因數、倍數等課程皆息息相關。透過運算思維設計數學應用程式「乘法遊戲王」，將遊戲融入學習活動中，不但能提升學生的學習興趣，更能幫助他們在乘法學習上奠立穩固的基礎。另外，設計小組在合力製作應用程式的過程中擴闊了運算思維視野，表達自己對問題的想法及實踐，正面地運用運算知識及技能獲得運算身份認同，他們能夠更有信心地運用數碼科技來處理身邊的挑戰，並結出數學教育及運算思維創意解難甜美的果實！

5. 參考文獻

Chan, T.-W., Looi, C.-K., Chen, W., Wong, L.-H., Chang, B., Liao, C. C. Y., Cheng, H., Chen, Z.-H., Liu, C.-C., Kong, S.-C., Jeong, H., Mason, J., So, H.-J., Murthy, S., Yu, F.-Y., Wong, S. L., King, R. B., Gu, X., Wang, M., Wu, L., Huang, R., Lam, R., & Ogata, H. (2018). Interest-driven creator theory: Towards a theory of learning design for Asia in the twenty-first century. *Journal of Computers in Education*, *5*(4), 435-461.

Kelley, D., & Kelley, T. (2015). Creative Confidence: Unleashing the Creative Potential within Us All. USA: Harper Collins.

Kong, S.C. (2016). A framework of curriculum design for computational thinking development in K-12 education. Journal of Computers in Education, 3(4), 377-394.

課程發展議會（2017）。**《計算思維─編程教育：小學課程補充文件》**。香港：課程發展議會。

曾鈺成 (2017)。**《考乘數表》**。香港：am730 報章。

Pedagogical reflections on Computational Thinking:

Applying the integration of programming and computational thinking to enhance the students' interest and performance in mathematics learning.

Hsu Man Sing[1*], Yuk Yue Vicky WONG[2]

[1,2] King's College Old Boys' Association Primary School No.2, Hong Kong

mshsu@kcobaps2.edu.hk, yywong@kcobaps2.edu.hk

ABSTRACT

The purpose of this paper is to illustrate the practical example of how to apply the integration of programming and computational thinking to enhance the students' interest and performance in mathematics learning. Based on the experience from four members of the "Mathematics Special Service Team", they found that most of the students with poor mathematics performance generally have a weak foundation in the "multiplication table". The "Mathematics Special Service Team" members take an oath to tackle the problem and hope to help the students with mathematics guidance needs. The four members formed a design team, by integrating programming and computational thinking, they designed an APP named "The Multiplier Game of Thrones". With the guidance of the teachers, the design team used MIT App Inventor 2 to design an "interest-driven" application program to enhance the students' interest and performance in the learning of multiplication table. The team members tackled the problem by applying the computational thinking practices, they also built up a solid foundation for computational thinking concept during the self-directed learning process.

KEYWORDS

Computational thinking, MIT App Inventor, coding, Mathematics, Multiplications

香港小學運算思維教育發展：個案實踐

羅金源[1*]，鄺靈俊[2]
[1,2]香港教育大學賽馬會小學，香港
kylaw@edujcps.edu.hk，lckwong@edujcps.edu.hk

摘要
本文描述一所香港小學發展運算思維教育的進程，學校參與香港賽馬會運算思維教育計劃，經過 6 年的發展，形成具校本特色的運算思維的課程。校本延伸部分具 2 個導向，即 X(橫軸) 導向及 Y(縱軸) 導向。X(橫軸) 導向意思指運算思維與科目聯乘，而 Y(縱軸) 導向意思指深化運算思維運用於 STEAM 及人工智能中。文中舉出運算思維課程伸延至中國語文、英國語文、常識、音樂、體育、視覺藝術、STEAM 的校本例子。讓讀者對香港運算思維教育發展有更深的認識。

關鍵字
運算思維教育；香港；小學；校本；學科

1. 前言
本校自 2017 年起積極發展運算思維教育，成為香港賽馬會運算思維教育計劃的先導學校。並於 2000 年起續三年成為計劃中的五所資源學校之一，負責協助推展運算思維的發展，過往 3 年已協助 30 所小學發展校本運算思維教育。學校積極投入發展，每星期均設有 2 節共 1 小時綜合科技課教授 4-6 年級學生運算思維。為了照顧個別差異，課節安排 2 位教師共同協作教授。透過共同備課、教師培訓及師徒制的方式，能教授運算思維的教師人數由最初的 3 位教師增至現時 19 位。現時的教師團隊除了教授運算思維外，更主要任教不同科目，包括中國語文、英國語文、數學、常識、音樂、體育及視覺藝術。而團隊亦包括各科科主席，讓運算思維教育能更容易融合於各科目發展。

2. 本校的運算思維教育課程

2.1. 核心部分
本校的運算思維課程是依照香港賽馬會運算思維教育計劃的教材修訂校本而形成的。整個課核心透過編程 Coding 的載體引領學生掌握運算思維，運算思維教育是一個教授思維策略的一種教學法，學生能夠掌握基礎編程知識，運用並解決問題技巧，內化提升數碼身份的認同及動機。

根據江紹祥(2016)運算思維發展 K-12 課程設計架構，當中提出的 7 個課程原則：

1. 利用運算思維知識作為課程的基礎

2. 把運算思維實踐作為課程的核心

3. 透過實際的經驗發展運算思維視野

4. 採用自上而下的策略以解決複雜運算思維的問題

5. 利用個人專案(final project)策略以檢視學生於不同層次時掌握運算思維的情況

6. 利用趣創學習(Interest-driven) 的策略來孕育趣創學習者

7. 使用評估標準和個人專案(final project)等級來培養創造力

這七項原則包括長中短的目標策略。按照原則(1) 先以一個課節為單位，教授運算思維知識。按照原則(2)，利用多節的組合單元，把運算思維實踐成為一種習慣。按照原則(3)透過不同的單元及實際的運作經驗，把運算思維視野成為一種態度。按照原則(4)，把問題簡化，讓學生的學習形成算法思維。按照原則(5)，評估學習的學習狀況。按照原則(6)，讓學生樂於創造，達到自我主導學習及終學學習的目標。按照原則(7)，透過個人專案，重視創造力的發展。

因此課程依照以上原則，以單元形式教授，以個人專案作階段性的總結。四年級的學生以 Scratch 為學習工具，界面較容易掌握，適合初學習編程的同學。而五六年級以 App Inventor2 為學習工具，讓學生有機會接觸雲端、物聯網等應用，擴闊學習視野。

2.2. 延伸部分
建構於核心部分的課程發展校本延伸部分，校本延伸部分發展作 2 個導向，即 X(橫軸)導向及 Y(縱軸)導向。X(橫軸) 導向意思指運算思維與科目聯乘，運算思維可以與各科配合。透過不同科目老師聯乘，把運算思維結合於其他學科中，以科目學習知識為題，教授運算思維為本，編寫教育程式為果。而 Y(縱軸) 導向意思指深化運算思維運用的延伸，把運算思維融合於 STEAM 的學習中，利用編程強化 STEAM 的學生創作，讓成品更具智能及多樣化。而人工智能的應用，可深化 STEAM 的創作成品，更具智能及人性化，創作的空間更廣闊。此外，更可讓學生接觸有關初階的人工智能算法，讓學生有能力透過已有的認知探索更多的領域。

本校按照核心部分的結構，製作了以下單元，結合中國語文、英國語文、常識、音樂、體育、視覺藝術，亦把課程延伸至 STEAM 的抽離課程。學生於課堂中進行編程，並把製作成果給予低年級學生遊玩提升學習興趣。單元名稱可參閱圖 1。

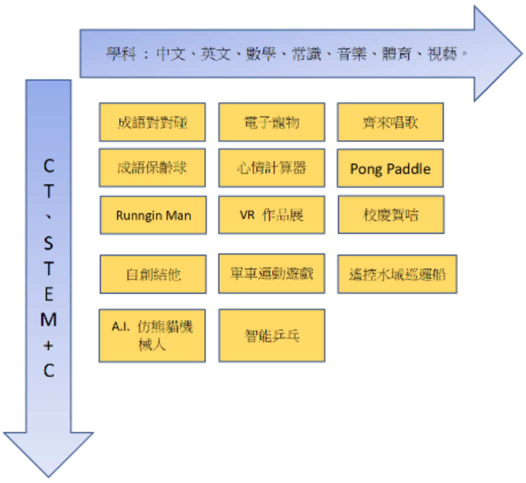


圖 1 校本延伸課程：X(橫軸)導向及 Y(縱軸)導向

2.2.1 X(橫軸）運算思維與科目聯乘：中文科

「成語對對碰」及「成語保齡球」課程設計利用 Scratch 平台，讓學生運用「序列」、「事件」、「碰到」、「重複」、「條件」、「變數」及「同步發生」等運算思維概念，亦運用運算思維實踐進行測試及除錯，以確保程式運作如期。

學習內容與中文科緊密聯繫，學生能借此兩個程式重温已學的成語知識：「成語對對碰」讓學生根據提供的成語選出缺少的漢字，並加以填充，組成完整的成語；「成語保齡球」則是等待學生完成：「成語對對碰」之後，再根據角色說出的句子，選出正確的成語，表情達意。「成語對對碰」及「成語保齡球」中文課程設計可加強學生對所學成語的記憶和運用成語的能力。

圖 2 中文科

2.2.2 X(橫軸）運算思維與科目聯乘：英文科

「電子寵物」設計利用 Scratch 平台，讓學生創作電子寵物，透過英語環境表述，加強學生英語運用。單元焦點教授學生運算思維概念：「事件」、「廣播」、「重複」及「變數」。同時，學生能夠在課堂中實踐反覆構思及漸進編程的運算思維，測試及除錯，以發展個人的運算思維。學習內容與生活緊密聯繫，無論是學生親身餵養寵物時的互動經驗，亦或是在其他電子設備平台玩虛擬餵養遊戲的體驗，都可運用在其個人的編程設計中，使他們盡情地發揮創意，創造出獨一無二的 Scratch 電子寵物。

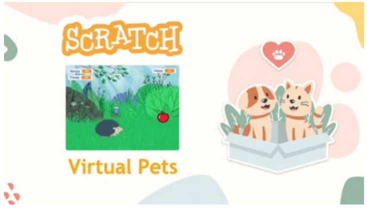


圖 3 英文科

2.2.3 X(橫軸）運算思維與科目聯乘：常識科

「心情計算器」課程設計利用 App Inventor2 平台，讓學生在課後能掌握以下運算思維概念：「事件」、「序列」。同時，讓學生在編寫計算心情的程式時，進行運算思維的實踐 — 「嘗試測試及除錯」，以發展個人的編程思維。課程的內容亦與學生的生活息息相關，學生可以透過程式紀錄自己一天經歷的心情，讓他們可以透過心情紀錄分析自己情緒的來源，加深對自我了解及認識。

圖 4 常識科

2.2.4 X(橫軸）運算思維與科目聯乘：音樂科

「自創結他」課程參考香港賽馬會運算思維教育計劃（階段二）「自創鋼琴」課程，改編並延伸成為校本課程。設計利用 App Inventor2 平台，讓學生在綜合科技課堂中應用運算思維實踐中的「反覆構思及漸進編程」、「重用」及「測試及除錯」，以發展個人的運算思維。吉他的和弦根據學校校歌的和弦編寫，讓學生能邊唱校歌，邊以「自創吉他」伴奏，增加他們對編寫程式的學習動機。並讓他們嘗試將已學的編程重用及整合，編寫屬於自己的程式。

圖 4 音樂科

2.2.5 X(橫軸) 運算思維與科目聯乘：體育科

「Running Man」課程設計利用 Scratch 平台，讓學生能利用視訊偵測技術及藉着動作跟程式互動，讓學生在課後能掌握以下運算思維概念：「序列」、「事件」、「重複」、「條件」、「變數」及「同步發生」。過程中，學生進行測試及除錯，以確保程式運作如期。學習內容與體育科緊密聯繫，學生能重溫已學的運動知識，例如田徑比賽中的起跑指令。另一方面，在體驗的過程中，學生能透過不斷擺動身體而提升心肺耐力及肌耐力等體適能。

圖 5 體育科

2.2.6 X(橫軸) 運算思維與科目聯乘：視藝科

Pong Paddle 彈力球課程利用 Scratch 平台讓學生透過簡單又有趣的彈球遊戲了解「廣播」、「變數」及「事件」等運算思維概念。學生可以運用「變數」及「條件」反覆構思用編程進行不同變化：例如當回力球碰到邊緣或底板(條件)會有不同的結果，更可以進階運用變數去增加遊戲難度改變球的移動速度。

2.2.7 Y(縱軸)運算思維運用融合於 STEAM：單車運動

單車運動遊戲課程設計利用 Scratch 平台配合 Micro:bit 內置的 accelerometer，讓學生在課堂中掌握以下運算思維概念：「事件」、「重複」及「變數」，亦在課堂中實踐「反覆構思及漸進編程」的運算思維，嘗試「測試及除錯」，以發展個人的編程思維。學習內容與生活緊密聯繫，疫情期間，除了能夠鼓勵同學做運動外，同時能夠觀賞不同地方的景點，增加使用健身單車的樂趣。

圖 6 單車運動

2.2.8 Y(縱軸)運算思維運用融合於 A.I：仿大熊貓機械人

仿大熊貓機械人有兩大用途，包括記錄珍貴的野生大熊貓片段及搜集非法捕獵大熊貓的證據，以追緝盜獵者，同時亦會監察非法砍伐林木與破壞野生環境的行為。在課程中學生會運用到運算思維概念的「序列」、「事件」、「條件」、「重複」，在運算思維實踐時，同學要運用測試及除錯，使熊貓能保持平衡地移動，同學亦要學習運用 A.I.人臉辨析技術去辨認工作人員及盜獵者，讓同學在學習運算思維時，亦能關注環保議題。

圖 7 仿大熊貓機械人

3. 回顧與展望

回顧過去 6 年運算思維於本校的發展，運算思維改變的不只於學生思維上的改善，更重要的是教師教學方式。從傳統的「重溫已有知識→引起動機→學習活動→解說→評估→鞏固」，發展至運算思維教育提倡的「To Play→To Think→To Code→To Reflect」教學方式。這些新的思維讓教師更靈活地突破以往的框架。我發現經過運算思維教學的教師，當他們回到任教不同科目時，明顯教學方式比以往更加靈活，教學更有條理，更細化學習的難點，讓學生更容易吸收。此外，教師更表示他們多了信心運用科技進行教學。

展望將來，本校的運算思維教育發展將以 X(橫軸)導向及 Y(縱軸)導向發展，透過 2 個導向的交錯將形成具學科內容拼以 STEAM 形成展示的學習單元。例如運算思維課程的「迷宮遊戲」，配對合適中文字詞，設計 micro:bit 成為遙控器以控制 scratch 遊戲。這些具趣味的學習內容，正正是 Chan, (2018) 提出「趣創者理論」，以興趣驅動愉快學習，有助學生建立核心能力，如思考及解難能力，以應對社會變遷與新科技發展。

此外，本校亦著重運算思維與其他思維之間的連繫，例如中文科重組句子，當中的排列與運算思維提出的「序列」概念相似。數學科的分步計數其實亦是運算思維提出的「反覆構思及漸進編程」。若能訂定出各學科與運算思維重疊概念的部分，就能讓運算思維教學立體呈現於各學科，學生學習更能具體掌握及關連，運算思維更能應用於生活中。

4. 參考文獻

Chan, Tak-Wai, Looi, Chee-Kit, Chen, Wenli, Wong, Lung-Hsiang, Chang, Ben, Liao, Calvin, . . . Ogata, Hiroaki. (2018). Interest-driven creator theory: Towards a theory of learning design for Asia in the twenty-first century. *Journal of Computers in Education*, 5(4), 435-461.

Kong, S. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, 3(4), 377-394.

Development of Computational Thinking Education in Primary Schools in Hong Kong: A Case Study

Kam -yuen LAW[1*], Ling-chun KWONG[2]
[1,2]The Education University of Hong Kong Jockey Club Primary School, Hong Kong
kylaw@edujcps.edu.hk，lckwong@edujcps.edu.hk

ABSTRACT

This article describes the development of computational thinking education in a Hong Kong primary school, which participated in the Hong Kong Jockey Club Computational Thinking Education Programme, and after 6 years of development, formed a school-based curriculum of computational thinking. The school-based extension has two guides, namely X (horizontal axis) guide and Y (vertical axis) guide. X (horizontal axis) orientation refers to the joint multiplication of computational thinking and subjects, while Y (vertical axis) guidance refers to deepening computational thinking applied to STEAM and artificial intelligence. The article gives school-based examples of computational thinking courses extended to chinese language, english language, general studies, music, physical education, visual arts, and STEAM. To enable readers to have a deeper understanding of the development of computing thinking education in Hong Kong.

KEYWORDS

Computational Thinking Education, Hong Kong , Primary School, school-based curriculum, subjects

Plugged-In And Unplugged Chemistry Computational Thinking Through Engineering Design Process

Norhaslinda ABDUL SAMAD[1], Kamisah OSMAN[2*], Nazrul Anuar NAYAN[3]
[1,2]Faculty of Education, National University of Malaysia, Malaysia
[3]Faculty of Engineering and Built Environment, National University of Malaysia, Malaysia
P107708@siswa.ukm.edu.my, kamisah@ukm.edu.my, nazrul@ukm.edu.my

ABSTRACT

In this Fourth Industrial Revolution driven by advances in computing, employees everywhere need to think creatively about how they perform their jobs. Therefore, students must prepare to be competent workforces with computational thinking. The primary purpose of this study was to suggest an inquiry-based chemistry education method that makes use of computational thinking paradigm. This strategy encourages the use of unplugged and plugged-in computational thinking activities to assist students build essential skills in the context of the engineering design processes. Past research has demonstrated that teaching computational thinking in traditionally "hard" subjects, such as chemistry, has improved student achievement and instilled necessary competencies in the classroom. The computational thinking skills include decomposition, pattern recognition, abstraction and algorithmic thinking. This study aimed to be a resource for chemistry instructors to design and conduct engineering-oriented lessons in secondary school classrooms, thereby enhancing students' understanding of chemistry.

KEYWORDS

Computational thinking, unplugged, plugged-in, chemistry, engineering design process.

1. INTRODUCTION

The contemporary workforce must adapt to the era of the Fourth Industrial Revolution (IR 4.0) defined by computer-based technologies that are driving innovation transformation in global industry. Careers in the digital age require a wide range of abilities, including the ability to think critically and analytically, ability to solve complex problems, and ability to analyze and evaluate systems (World Economic Forum 2018).

Computational thinking (CT) is an approach to solving difficult problems that is inspired by the concepts of computer science (Wing 2006). Algorithmic thinking, abstraction, decomposition, and pattern recognition are all emphasized as important CT skills. CT encourages the growth of problem-solving abilities and techniques, whether or not a computer is used. The application of CT skills in the classroom can be accomplished via both unplugged and plugged-in activities spanning a wide range of subject areas (Mensan et al. 2020). Plugged-in activities make use of computer software and instructional programming or learning programs that are either freely available online or can be purchased, whereas unplugged activities do not involve the use of any digital programming or instruments.

2. PROBLEM STATEMENT

In the era of IR 4.0, it is more important than ever to train skilled professionals in the fields of science, technology, engineering, and mathematics (STEM), and chemistry in particular. Students' inability to grasp the underlying ideas and principles that drive chemistry is the most significant barrier to their success in learning the subject (Dewi et al. 2021; Prokša et al. 2018).

To compete on a global scale, today's students need 21st century skills, such as CT, to excel in STEM fields, particularly chemistry. Furthermore, the level of CT competence is now considered moderate (Durak et al. 2019). This is because students have less desire to learn CT due to the limitations they face, such as the need for constant access to the internet in suburban schools and the essentially passive nature of text-based programming languages (Threekunprapa & Yasri 2020). Moreover, there is a shortage of qualified and experienced teachers in the fields of computer science (CS) (Ling et al. 2018). The majority of current CS teachers have a limited understanding of CT and often hold false beliefs about the field (Alfayez & Lambert 2019; Fessakis & Prantsoudi 2019; Bower et al. 2015).

Previous research has determined, based on gender comparisons, that the level of CT skills between male and female students varies and is not equal. According to Korkmaz and Bai (2019), male students have a greater level of CT in terms of critical thinking skills than female students. However, according to Israel-Fishelson et al. (2021), female students were found to have a higher level of computational creativity than male students. In LEGO robotics program, male students focused more on the operational aspects of robot assembly and coding, whereas female students emphasized group dynamics (Ardito et al. 2020).

CT ideas have been utilized in numerous fields, and the ability to think computationally is vital for all fields. For example, its integration with the field of pure science for systematic problem solving should also be revealed (Chongo et al. 2020).

In conclusion, a learning approach is needed to enhance current instruction methods by integrating the Engineering Design Process (EDP) into both plugged-in and unplugged activities, with the goal of stimulating students' critical and creative problem-solving abilities in chemistry.

3. PLUGGED-IN AND UNPLUGGED COMPUTATIONAL THINKING

CT is the foundation of computer science, but it is also applicable to many other fields of science and engineering (Peel et al. 2021). Recently, it has become possible to teach CT skills in the classroom through both unplugged and plugged-in activities in various disciplines. The ideal approach for students to use their CT skills is, without a doubt, through plugged-in activities (Brackmann et al. 2017; Weintrop & Wilensky 2017; Lye & Koh 2014). However, unplugged activities are also an effective means of introducing CT in the classroom (Caeli & Yadav 2020; Delal & Oner 2020; Saxena et al. 2020; Kuo & Hsu 2020; Rich et al. 2020; Threekunprapa & Yasri 2020b; del Olmo-Muoz et al. 2020). The CT levels of students who participated in both unplugged and plugged-in activities prior to the latter were found to be greater than students who participated in plugged-in activities only. The effect of an unplugged strategy on CT skill acquisition is determined by the relationship between sensory motor elements and higher-level cognitive processes (Città et al., 2019).

It is common practice to have elementary school children engage in unplugged activities of CT (Saxena et al. 2020; Mensan et al. 2020; Kuo & Hsu 2020; Rich et al. 2020; Relkin et al. 2020). The majority of classroom supplies, including paper, pens, and cards, are sufficient for most unplugged activities (Delal & Oner 2020).

In the author's chemistry class, students designed pseudocode and a flowchart for a procedural algorithm based on methods for preparing salt for an experiment. This activity encourages students to think algorithmically as they created a flowchart and pseudocode.

CT is commonly connected with computing and programming, thus it seems logical that plugged-in computational tools, such as Scratch and Python, will help develop relevant skills. Scratch is a visual programming environment, where students can create and share their own video games, animations, simulations, and interactive storytelling (Resnick et al. 2009). The Scratch programming language is block-based, which is similar to Lego bricks. Scratch aims to be a highly interactive platform, where students only need to assemble blocks and make modifications to the arrangement of the blocks in stages and frequently. In the author's chemistry class, Scratch was used by students to design simulations for scientific experiments and games for classroom use.

4. ENGINEERING DESIGN PROCESS

Systematic plugged-in and unplugged design activities can be performed using EDP. The EDP's theoretical underpinning is based on constructionism theory. EDP is a strategy defined by a set of phases that help engineers reach a solution. Since an engineer may need to begin the procedures at any suitable level, the model is cyclical and does not provide a starting point. Hill-Cunningham, et al. (2018) suggested five steps of EDP, namely ask, imagine, plan, create, and improve.

Figure 1 Engineering Design Process (Hill-Cunningham et al. 2018)

In the Ask phase, students formulate questions to better understand and categorize design problems. The second step asks the students to generate ideas, and the third step asks them to select the most promising ideas and create a list of what they need. The Create step asks students to construct a working prototype and put it through its paces before moving on to the next stage, where they will resolve any issues they find (Hill-Cunningham et al. 2018).

5. EXAMPLE OF LESSON PLAN

Constructivism theory forms the theoretical foundation for lesson plans prepared utilizing 5E Model, which consists of five phases, namely Engagement, Exploration, Explanation, Elaboration and Evaluation (Bybee 2009). Table 1 shows the 5E instructions of Salt topic based on the author's chemistry class.

Table 1 5E Instruction of Salt topic

Phase	Explanation
Engagement	• Students are shown a short video with recipes and instructions for cooking Korean cheese chicken. • Students discuss based on the video how the recipe can be altered if there are insufficient ingredients. • Students discuss that there are steps involved in performing an experiment, as well as factors that can vary depending on the circumstances.
Exploration	• Collaboratively, students conduct experiments of salt preparation according to the pre-determined station.
Explanation	• Students share the experimental findings with peers and teacher. • Teacher explains the concept of salt preparation method.
Elaboration	• Students are required to collaboratively design pseudocode and flowcharts for the experimental

<table>
<tr><td></td><td>algorithm that has been executed.
• Students should apply EDP to design pseudocode and flow chart.
<table>
<tr><th>Step</th><th>Explanation</th></tr>
<tr><td>Ask</td><td>Students identify the salt preparation process, and design flow chart and pseudocode in groups.</td></tr>
<tr><td>Imagine</td><td>Students brainstorm ideas and sketch flowchart and pseudocode.</td></tr>
<tr><td>Plan</td><td>Students plan and choose the best flowchart and pseudocode of the group.</td></tr>
<tr><td>Create</td><td>Students draw flowchart and pseudocode that have been selected.</td></tr>
<tr><td>Improve</td><td>Students share flowchart designs and pseudocode with peers and teachers and get feedback from them.</td></tr>
</table>
</td></tr>
<tr><td>Evaluation</td><td>• Students are required to assess the experience gained from this activity.
• Students check the extent to which their own understanding, abilities and competencies have changed.</td></tr>
</table>

6. CONCLUSION

The integration of CT into STEM education has the potential to increase student engagement through CT and scientific inquiry. Reflecting the fact that CT is a cross-disciplinary skill, the integration of CT is now more focused on facilitating the acquisition of subject matter (Yang et al. 2021). In this study, EDP was integrated with CT and applied in chemistry learning to enhance students' knowledge and skills.

A limitation of this study was the time constraints for students to be involved in designing activities. Compared to conventional learning, the time required to design artefacts or products is quite extensive.

In future studies, students should be allowed to design artefacts or products outside the classroom as project work. Thus, knowledge and skills can be acquired through unplugged or plugged-in CT design activities.

7. REFERENCES

Alfayez, A. A., & Lambert, J. (2019). Exploring Saudi computer science teachers' conceptual mastery level of computational thinking skills. Computers in the Schools, 36(3), 143-166.

Bower, M., & Falkner, K. (2015). Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. ACE, 160, 37-46.

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In Proceedings of the 12th workshop on primary and secondary computing education (pp. 65-72).

Bybee, R. W. (2009). The BSCS 5E instructional model and 21st century skills. Colorado Springs, CO: BSCS, 24.

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. TechTrends, 64(1), 29-36.

Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., ... & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, *141*, 103613.

Chongo, S., Osman, K., & Nayan, N. A. (2020). Level of Computational Thinking Skills among Secondary Science Student: Variation across Gender and Mathematics Achievement. Science Education International, 31(2), 159-163.

Delal, H., & Oner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. Informatics in Education, 19(1), 1-13.

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. Computers & Education, 150, 103832.

Dewi, C., Pahriah, P., & Purmadi, A. (2021). The urgency of digital literacy for generation Z students in chemistry learning. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(11), 88-103.

Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. Contemporary Educational Technology, 10(2), 173-197.

Fessakis, G., & Prantsoudi, S. (2019). Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. Informatics in Education, 18(2), 227-258

Hill-Cunningham, P. R., Mott, M. S., & Hunt, A. B. (2018). Facilitating an elementary engineering design process module. School Science and Mathematics, 118(1-2), 53-60.

Kuo, W. C., & Hsu, T. C. (2020). Learning computational thinking without a computer: How computational participation happens in a computational thinking board game. The Asia-Pacific Education Researcher, 29, 67-83.

Ling, U. L., Saibin, T. C., Naharu, N., Labadin, J., & Aziz, N. A. (2018). An evaluation tool to measure computational thinking skills: pilot investigation. National Academy of Managerial Staff of Culture and Arts Herald, 1, 606-614.

Lockwood, J., & Mooney, A. (2018). Developing Computational Thinking Test Using Bebras Problems. CC-TEL/TACKLE@EC-TEL.

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.

Mensan, T., Osman, K., & Majid, N. A. A. (2020). Development and validation of unplugged activity of computational thinking in science module to integrate computational thinking in primary science education. Science Education International, 31(2), 142-149.

Peel, A., Sadler, T. D., & Friedrichsen, P. (2021). Using unplugged computational thinking to scaffold natural selection learning. The American Biology Teacher, 83(2), 112-117.

Prokša, M., Drozdíková, A., & Haláková, Z. (2018). Learners' understanding of chemical equilibrium at submicroscopic, macroscopic and symbolic levels. *Chemistry-Didactics-Ecology-Metrology*, *23*(1-2), 97-111.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM, 52(11), 60-67.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. Journal of Science Education and Technology, 29(4), 482-498.

Rich, K. M., Yadav, A., & Larimore, R. A. (2020). Teacher implementation profiles for integrating computational thinking into elementary mathematics and science instruction. Education and Information Technologies, 25, 3161-3188.

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. The Asia-Pacific Education Researcher, 29(1), 55-66.

Threekunprapam, A., & Yasri, P. (2020). Patterns of Computational Thinking Development While Solving Unplugged Coding Activities Coupled with the 3S Approach for Self-Directed Learning. European Journal of Educational Research, 9(3), 1025-1045.

Threekunprapa, A., & Yasri, P. (2020b). Unplugged Coding Using Flowblocks for Promoting Computational Thinking and Programming among Secondary School Students. International Journal of Instruction, 13(3), 207-222.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE), 18(1), 1-25.

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

World Economic Forum Boston Consulting Group (BCG). (2018). Towards a reskilling revolution: A future of jobs for all. In World Economic Forum.

The Asia-Pacific Society
for Computers in Education
(APSCE)

National Central University