Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020

19 - 21 August 2020

Computational Thinking Education

Created and Funded by

香港賽馬會慈善信託基金
The Hong Kong Jockey Club Charities Trust
同心同步同進 RIDING HIGH TOGETHER

Co-created by

香港教育大學
The Education University of Hong Kong

Massachusetts Institute of Technology

香港城市大學
City University of Hong Kong

CoolThink@JC

Proceedings of International Teacher Forum on

International Conference on

Computational Thinking Education 2020

19-21 August 2020

Hong Kong

Funded and Created by

The Hong Kong Jockey Club Charities Trust

Co-created by

The Education University of Hong Kong

Massachusetts Institute of Technology

City University of Hong Kong

Publication of The Education University of Hong Kong

10 Lo Ping Road, Tai Po, New Territories, Hong Kong

ISBN 978-988-77034-7-1

CTE 2020

Editors

Siu-cheung KONG
The Education University of Hong Kong, Hong Kong

Ting-chia HSU
National Taiwan Normal University, Taiwan

Rong-huai HUANG
Beijing Normal University, China

Chee-kit LOOI
Nanyang Technological University, Singapore

Marcelo MILRAD
Linnaeus University, Sweden

Ju-ling SHIH
National Central University, Taiwan

Hyo-Jeong SO
Ewha Womans University, South Korea

Valentina DAGIENE
Vilnius University, Lithuania

Preface

CTE2020 is the fourth international conference organized by CoolThink@JC, which is created and funded by The Hong Kong Jockey Club Charities Trust, and co-created by The Education University of Hong Kong, Massachusetts Institute of Technology, and City University of Hong Kong. Organized along with the conference, CTE2020 International Teacher Forum is first introduced this year. The Forum aims to set a stage for K-12 CT teachers worldwide to share best practices and key challenges of implementing computitional thinking education in different countries/ regions.

With the support from Coordinating Scholars, Review Panel Members and K-12 CT teachers from different countries/regions, the Forum received a total of 28 submissions by 56 authors from 8 countries/regions (see Table 1).

Table 1: Distribution of Paper Submissions for CTE2020 International Teacher Forum

Country / Region	No. of Authors	Country / Region	No. of Authors
China	1	**Singapore**	1
Finland	1	**South Korea**	12
Hong Kong	21	**Sweden**	2
Lithuania	3	**Taiwan**	15
		Total	**56**

The Review Panel for the Forum is formed by 32 Members and 8 Coordinating Scholars worldwide. Each paper with author identification anonymous was reviewed by at least two Review Panel Members. Coordinating Scholars of each participating country/region then conducted meta-reviews and made recommendation on the acceptance of papers based on the reviews provided by the Review Panel Members. With the comprehensive review process, a total of 24 accepted papers are presented (see Table 2) at the Forum.

Table 2: Paper Presented at CTE2020 International Teacher Forum

Track	Number of Paper
China (via Prof. Ronghuai HUANG)	1
Hong Kong (via Prof. Siu Cheung KONG)	8
Lithuania (via Prof. Valentina DAGIENE)	3
Singapore (via Prof. Chee-kit LOOI)	1
South Korea (via Dr. Hyo-Jeong SO)	1
Sweden (via Prof. Marcelo MILRAD)	2
Taiwan (via Prof. Ting-chia HSU and Prof. Ju-ling SHIH)	8
Total	**24**

On behalf of the Conference Organizing Committee and CoolThink@JC, we would like to express our gratitude towards all the partners and participants for their contribution and participation to CTE2020 International Teacher Forum.

We sincerely hope everyone enjoy and get inspired from CTE2020 International Teacher Forum.

With Best Wishes,

Prof. Siu-cheung KONG

The Education University of Hong Kong, Hong Kong

Conference Chair of CTE2020

Principal Tsz-wing CHU

St. Hilary's Kindergarten and Primary Schools, Hong Kong

Conference Chair of CTE2020

Table of Contents

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Pedagogical Practices in Teaching Scratch & Micro: bit Programming for Computational Thinking Development: Developing Diet Related Games

Yuen Ting CHENG
St. Edward's Catholic Primary School, Hong Kong
secps056@stedward.edu.hk

ABSTRACT

In the 21st century, STEM education has grown rapidly in order to let children develop computational thinking skills and practices. Programming activities with well-designed curriculum is one of the ways to apply in STEM education. There are different softwares and platforms from the Internet for teachers to use as teaching tools. This is an easy way for children to use these tools through a computer but without real object to create and observe. Educational robotics is another tool for children to learn. Although in the last two decades, robots have started their incursion into the formal educational system and researchers have stressed the learning potential of robotics, the slow pace of their development due to the cost of the robot kits (Yiannoutsou, Nikitopoulou, Kynigos, Gueorguiev, & Fernandez, 2016). Recently, the cost of robots has decreased, whereas their capabilities and the availability of supporting hardware and software has increased. Therefore, using robots as teaching tools have become more appealing to schools. This study shares pedagogical practice in teaching Scratch & Micro: bit for Computational Thinking.

KEYWORDS

computational thinking, Scratch, Micro: bit, pedagogy, primary school students

1. INTRODUCTION

Computational thinking skills include a wide range of mental tools and concepts from computer science and related to STEM (Science, Technology, Engineering, Mathematics). Computational thinking skills help people to solve problems, think logically and establish a design thinking mindset. However, there are different kinds of tools for students to learn Computational thinking skills including Scratch, Micro: bit, App Inventor, Makey Makey and mBot etc. The aim of this study was to shares pedagogical practice in teaching Scratch & Micro: bit for Computational Thinking.

Teaching Computational thinking to primary school students in Hong Kong has become increasingly important and ubiquitous. It is due to educators finding that only focusing on primarily languages and calculations is not adequate for students to face the challenges in the 21 st century. Recently, Science, technology, engineering and mathematics (STEM) education for primary school students became more important. Students acquire related knowledge help them to establish problem-solving skills, logical thinking, mathematical thinking skills and being innovative. This essential for understanding problems and to enable effective solutions to be developed (Kazimoglu, Kiernan, Bacon, & Mackinnon, 2012). Also, new technology learning standards and best practices for integrating technology into early childhood education have been developed (Barron et al., 2011; International Society for Technology in Education (ISTE), 2007; NAEYC & Fred Rogers Center for Early Learning and Children's Media, 2012; U.S. Department of Education, 2010). The objective of Computational thinking is not just to make students proficient in computer sciences, but also to allow students to apply their computational thinking skills in other classes. Logic is the most important skill humans employ for solving problems, and further developing this skill with the aid of computers and other digital tools has become an important aspect of our daily life and work (Barr et al., 2011)

2. PEDAGOGICAL PRACTICES IN PROGRAMMING DIET RELATED GAME

2.1. Scratch Group

Scratch-based game activities were applied to this Study Group. In this context, the teacher teaches computational thinking skills (concepts and practices) through Scratch by creating a game (with examples). Students were asked to design a simple game using Scratch. After the lesson, the project will be evaluated by teachers. For the computational thinking skills taught in this lesson mainly focus on Variables and Video Sensing. Students learnt the concepts with examples of Variables and Video Sensing. Then they grouped into pairs and use Scratch as the tool to do programming activities. After finishing the task, they are allowed to modify the game or create another game by using Scratch with Variables and Video Sensing skills. Students uploaded their work to Scratch Studio and evaluated by teachers and peers.

The lesson materials can be found: http://bit.ly/38ndf65

Figure 1. Lesson Slide of Scratch-Learning Objectives

Figure 2. To teach-Definition of Variables

Figure 3. To teach-Explanation of Video Sensing

In this study group, teacher used Nearpod as an e-learning tool in order to provide more interactive activities for students to understand the CT concepts and more guidance for students to work on their programming activities.

2.2. Micro: bit Group

Micro: bit -based project was applied to another Group. In this context, the teacher will teach Computational thinking skills (concepts and practices) through Micro: bit by creating a project (with examples). Students were asked to design a mini project using Micro: bit. After the last lesson, the project will be evaluated by teachers. For the computational thinking skills taught in this lesson is similar with Scratch Group, the CT concepts are Variables and Broadcasting. Changed from Video Sensing to Broadcasting is due to the limitations and differences of Scratch and Micro: bit platform. But both are coding skills with interaction effect. This lesson uses a robotic tool- Micro: bit as the teaching and learning tool. The coding platform of micro: bit is also used for programming activities and the platform is similar to Scratch. Students work on the programming activities then experience and observe from the micro: bit. Students also grouped into pairs and use micro: bit as the tool to do programming activities. After finishing the task, they are allowed to modify the game or create another game by using micro: bit with Variables and Broadcasting skills. Students uploaded their work to Google Drive and evaluated by teachers and peers.

The lesson materials can be found : http://bit.ly/2tLhwkC

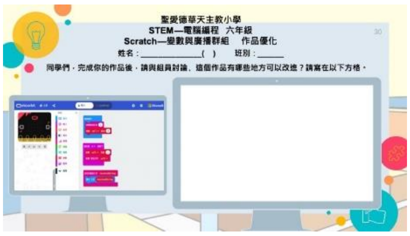


Figure 4. After code-Optimize Worksheet

In micro: bit group, the teacher also used Nearpod as an e-learning tool in order to provide various interactive activities for students to understand the CT concepts.

3. REFLECTION AND CONCLUSION

Teaching computational thinking through Scratch and Micro: bit able to raise students interests and have positive perceptions towards STEM education. These tools are user-friendly and interesting for students to code. The code blocks are easy to understand with computational thinking concepts. Both tools allow students' ideas to be put into practice.

On the other hand, the teaching methods and teaching plans are important. They will influence students' understanding of computational thinking concepts and also students' perceptions towards STEM education. The "To play, to think and to code" pedagogy is one of the useful ways to teach computational thinking concepts as it allows students to think through play and start to understand CT concepts with coding.

Students' perceptions towards STEM education are also important and valuable to concern. Students' perceptions will strongly affect their motivation to learn CT skills. Different teaching tools including robotics are worth exploring and study as different tools, pedagogies and curriculum in STEM education will all affect students' perceptions. Future studies should consider evaluating the effectiveness of students in learning CT skills through different kinds of robotics with programming activities and their perceptions towards STEM education.

4. REFERENCES

Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. (2011). Take a giant step: A blueprint for teaching children in a digital age. *The Joan Ganz Cooney Center at Sesame Workshop.*

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

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play. *Procedia Computer Science*, *9*, 522-531.

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.

Resnick, M., & Robinson, K. (2018). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. Cambridge, MA: The MIT Press.

Wong, G. K., & Cheung, H. (2018). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 1-13.

Yiannoutsou, N., Nikitopoulou, S., Kynigos, C., Gueorguiev, I., & Fernandez, J. A. (2016). Activity Plan Template: A Mediating Tool for Supporting Learning Design with Robotics. *Robotics in Education Advances in Intelligent Systems and Computing*, 3-13.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Educational Benefits of an Integrative Approach Combining STEM, Project Based Learning, Robotics and Programming in Schools

Emma EKLUND
Region Kronoberg, Secondary school
Växjö, Sweden
emma.eklund@skola.vaxjo.se

ABSTRACT

In Autumn 2017, the Swedish National Agency for Education, introduced programming in the curriculum for almost every subject in school. Changes are significant including programming and computational thinking as tools for thinking in order to acquire problem solving skills. One of the aims of the Swedish school system is to educate creative, communicative and digitally competent students, able to innovate and collaborate. One important question arises, *how can schools work with programming in the classroom in order to meet all these demands*?

During the last decade, I have been systematically working with programming in schools with more than 400 students in about 10-15 classes in order to provide some answers to the question above. The different activities include working with programmable LEGO-EV3 robotics and computer-aided learning in *an integrative approach bringing together several subject matters*. Second, the chosen approach gives them the opportunity to compete as a team in the world's largest Robotics competition; The *FIRST LEGO League*.

In their digital work, students have reached heights in knowledge about STEM, programming, problem-solving strategies and synergy effects that teachers never thought were possible, to the extent that they reach a salient *social*, *personal* and *intellectual* growth. Their learning process goes far beyond how to solve problems utilizing programming and constructing robotics. Their skill sets develop beyond all goals in the curriculum. Added-value pedagogy in project-based packaging brings them meaning to the “digital” work in a whole new dimension. The remaining of the paper provides a more elaborated description of these ideas.

KEYWORDS

STEM, robotics, added-value pedagogy, student-based learning, integrated curriculum

1. PROGRAMMING AND DIDACTICS

Schools often let every student to work alone. It is very hard to introduce and implement *soft core values* (cooperation, equality, democracy) in such a system. Even if the curriculum demands innovators and problem solvers, at the same time the system prevents it. As already mentioned, digitalization and programming are now incorporated in the curriculum in almost every subject in school. But the latest is done in fractions and boxed into the ordinary lesson routine. Teachers face the dilemma of first attaining knowledge themselves on their spare time, a problem Rolandsson (2015) addresses. At the same time as the old knowledge stays in the curriculum, a lot of new knowledge is to be taught in the same amount of time.

Will a lesson or two in math, then an app in physics and then a Scratch-lesson next week in science-class add up to innovation, or is it fractions of testing things out? Selwyn (2017) points out the dilemma when even though using new and innovative digital approaches, due to the system, the work gets reduced from challenging to “same old traditional learning” in which the “*what”* had changed, but not the “*how”* and “*why”*.

Resnick (2010) elaborates on what schools are able to offer *“through social processes that include participation in certain forms of high-demand learning”* (Resnick, 2010, p.186) instead of as an *“entity”* or as something that people have a fixed amount of time to accomplish (Resnick & Gall, 1997). Resnick suggests a new way of school organization where teachers´ instruction and professional development should aim for and secure a *“thinking curriculum”* with high cognitive demands on conceptual learning, reasoning, explaining, and problem solving (Resnick, 2010).

2. PROGRAMMING TO INTEGRATE DIFFERENT STEM-RELATED TOPICS

In my classes, every student in seventh grade, learns the basics of the programmable LEGO EV3 robotics in groups of 3-4 students. We discovered that they developed problem-solving strategies with this material from the focus on social aspects of their work. The technological material itself is not the evident object of learning in the pupils’ experiences. How can we as teachers use this to accelerate their learning and programming skills? One possible answer to that question is working *“outside of the box”* and use programming to integrate different STEM- related topics/subjects, instead of working inside the box with STEM in all different subjects.

Once the students become familiar with the LEGO EV3 materials, they can sign up for the school team to compete in FIRST LEGO League - FLL. This is an age-integrated and entrepreneurial teamwork of about 25-30 students working very hard during 10 weeks, in school and as extracurricular activity. Synergy effects rise when they get to work with robotics towards a common goal – the competition. The added value pedagogy creates a core value in these activities, namely *engagement*. Every year they work about a theme; a real-life problem about equality and sustainability, which engages them further more.

The technical part of the competition is to construct both an advanced robot and clever tools to solve a specific number of physical tasks/missions on a challenge-table. The robot and the tools are then programmed to solve as many missions as possible in two and a half minutes. The students merge knowledge of all STEM subjects, programming and computational thinking in speed and depth that far excel what you achieve in the ordinary classroom lesson. For example, we discuss proportional-integral derivate and end ratio of gears, we compare theory versus reality, errors and margins, force transmission, practical application of math because they want to know more!

Thereafter, they will be collaborating with *other* teams from all over the world, sharing ideas and discussing technical solutions (Brennan et al., 2010). Through Skype, students find others alike, a *WOW-moment*, with great emotional impact. These different interactions require language skills like programming also does later in life. When teachers no longer are able to keep up, they get in contact with *real programmers* which mentors and talks about programming, which very few ordinary teachers master.

In this competition, they have to reach out to social media, their own school, other schools and the local municipalities. They have to contact local press, radio, companies trying to sell their solution and get them to sponsor their work in the competition. The marketing aspect make them believe in themselves. They discover traits they didn't knew existed, find new abilities and learn they can accomplish much more together in true teamwork. They develop the "grit", and the problem-solving attitude that are essential for the scientist/innovator, needed for the future, never giving up until they have a solution. The figures below illustrate some of these activities.

Figure 1. Students working hard on regional competition in the robotics challenge

Figure 2. The teams innovative Boxrobot with its award-winning Gearbox and example of My_block programming

Leadership and self-confidence, are achieved only by student-guided teaching. Together they construct, program, solve problems, write scientific reports, market the team and their solution, seek out sponsors, build a science prototype and creates a stand for a summary display of their work, and of course a lot knowledge and experience on programming techniques.

3. CONCLUSIONS

Creating a greater context through interdisciplinary work and an integrated curriculum is one possible approach that nurture students´ digital competencies and boost their digital creativity. A project like the one described in this paper prepares students to be the future "learners" and "problem identifiers" that our society requires. Engaged through meaningful "real-life" connections, they show an increased rate of learning. Connecting different areas of study emphasizing on the unifying concept of computational thinking, programming and STEM-related topics are the benefits of an integrated curriculum, to become more successful in all different but still complementary and related subjects. By doing so, I hope we can continue to promote and develop new pedagogical approaches about how to integrate computers and programming into technology education. Three years in a row, teams from our school have qualified to the FLL World Championship (90 best teams in the world out of 44 000 teams), Detroit, USA. Our students have reached deep knowledge in computational thinking, technology, engineering, programming and problem solving. The outcomes of this project are a very good example of how to work with digital tools and techniques in school in novel ways. Before the robotics projects in 2012, about 3-4 students per 9th grade class chose technical gymnasium. 2019 we hit a record with 8-9 students per class of which many had been involved in FLL, competing or indirectly by supporting the team. In 2020 we have qualified to Open International in Greece. Perhaps the record will be beaten?

4. REFERENCES

Brennan, K., Resnick, M., & Monroy-Hernandez, A. (2010). Making projects, making friends: Online community as catalyst for interactive media creation. *New Directions for Youth Development, 2010* (128), 75-83.

Rolandsson, L. (2015). *Programmed or Not. A study about programming teachers' beliefs and intentions in relation to curriculum.* Doctoral thesis, KTH School of Education and Communication in Engineering Science, Stockholm Sweden

Resnick, L.B. & S. N.-L. Gall (1997). Socializing intelligence. *Piaget, Vygotsky and beyond*. New York: Routledge, 145-158.

Selwyn, N., Nermorin, S., Buffin, S., & Johnson, N. F. (2017). Everyday Schooling in the Digital Age: High school, high tech? Left to their own devices; the everyday realities of one-to-one classrooms. *Oxford Review of Education, 43*(3), 289-310.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Teaching Computational Thinking in Lithuania's Primary Schools

Rūta FILONČIKIENĖ
Vilnius Gediminas Technical University Engineering Lyceum, Lithuania
rutaf@hotmail.com

ABSTRACT

Information and communication technology has more and more impact in Lithuania's schools. Along with new, improving technologies, a variety of teaching, learning and demonstration tools are coming to secondary schools, and even to primary schools. Teachers need to face new challenges, improve not only self and professional competences, but to be able to use new technologies, improve their kids digital literacy as well as computational thinking (Wing, 2006).

KEYWORDS

primary education, computational thinking, robotics

1. INTRODUCTION

Primary school teachers understand that knowing how to use and manage technologies in this digital world is essential as well as was teaching how to read, write and count. Educational experts discovered teachers and students needs not only to know how to use computer programs, but also how to create digital content.

If we think about Lithuania's primary schools pupils (age 7-11) a decade ago, we noticed the focus on critical thinking, creativity and imagination. The main ideas of modern school are – improving competence, problematic and computational thinking. Knowledge, skills and abilities, that will be needed in the nearly future, are becoming not so important. Without even noticing it, these changes tell us to improve and change process of education.

2. INFORMATION TECHNOLOGIES IN PRIMARY EDUCATION

Information technologies came to Lithuania's primary schools naturally, step by step. Its coming was dictated by environment, changing student's needs, and lifestyle. More often teachers started to integrate information technologies into their teaching subjects, but never named it as informatics.

In 2016 the Ministry of Education, Science and Sport gathered researchers, teachers, educational experts and entrepreneurs in order to prepare Information Technology General Curriculum for Primary education. Collaborating with educational community, business and public organizations, considering international experience, project was created for the Primary Education Informatics framework program. This project helped to educate primary schools pupils' computational thinking and digital literacy.

3. PROJECT "INFORMATICS IN PRIMARY EDUCATION"

In 2017 the project "Informatics in Primary Education" was implemented. The aim of this project was to develop the content of primary education informatics and to test it in the summer of 2018 in ten pilot schools. Teachers from selected schools developed and tested educational materials, made suggestions on the curriculum for informatics and the draft framework program, and shared best practices.

90 additional schools implementing primary education programs were selected from 2018 to 2022. Together with already involved group of 10 schools, experts, researchers and practitioners will test and develop the integrated content of information technology. From 2020, all Lithuanian schools will be provided with consultations on the content and implementation of the informatics program[2].

4. CURRICULUM

Framework of curriculum. The content of the Primary Education Informatics Framework Program covers the following areas:

- Digital content;
- Algorithm and programming;
- Problem solving;
- Data and information;
- Virtual communication;
- Safety and law

The curriculum does not include a separate lesson in informatics, it only appears in elementary school from grade 5 (primary school in Lithuania from grade 4), but schools have the possibility to choose how to teach information technologies. Some give a separate extra hour and have a lesson in information technology, others integrate it.

4.1. Educational Resources

It is a teachers' choice. Prepared lesson descriptions (plans and additional tools), different activities (tasks, textbooks "Thinking and creating..."), games ("Bebras" cards), a variety of digital teaching materials translated into Lithuanian that would be used to promote computational thinking, digital textbooks EMA that covers all the areas of informatics programs. The commentaries provide questions and insights that help us rethink how short Bebras-like tasks serve as scaffolding to engage children in computing (informatics). They help us to consider, in greater depth, computing concepts that need to be experienced by children in various ways - the solving of short tasks being one of them[1].

4.2. Approaches

How? It is taught by playing, creating a natural environment for children (Dagienė, 2019).

First and second graders learn to distinguish, differentiate between different devices, talk about safe online behaviour and emerging threats, become familiar with how computers work and the basic concepts of programming.

As they play, pupils learn how to move an object, create simple codes to complete a task, recognize sequences, repetitions, store information, process data, create digital content, and solve a variety of logical tasks.

5. CONCLUSION

"This is the best and the most interesting lesson", - say most of the pupils. Pupils intuitively understand what is needed and useful for them.

6. REFERENCES

Dagienė, V., Futschek, G., & Stupurienė, G. (2019). Authors' Response: Concepts of Computing in "Mind-size Bites". *Constructivist foundations. 14*(3), 413-415.

Dagienė, V. (2017). Online resource: Informatika pradiniame ugdyme. Retrieved Nov 18, 2019, from https://informatika.ugdome.lt/lt/apie/

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

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Approaches of Integrating Computational Thinking into Programming Lessons

Wei-sin HO
Maris Stella High School (Secondary), Singapore
ho_wei_sin@moe.edu.sg

ABSTRACT
Computational thinking (CT) (Wing, 2006) describes a set of thinking skills, habits and approaches that are integral to solving complex problems using a computer and widely applicable in the information society. It is widely recognized as an important skill in K-12 education. CT integration can be done through the computer science approach of teaching thinking processes for problem-solving during programming lessons. This paper shares a Singapore secondary school's integration of the KWS problem-solving strategy and Use-Modify-Create (Lee et al., 2011) pedagogical approach to develop students' CT skills, as well as improve their motivation to learn programming.

KEYWORDS
computational thinking, K-12, programming, KWS, Use-Modify-Create

1. INTRODUCTION
Under the Singapore education system, students who offer General Certificate of Education (GCE) O-Level Computing undergo a two-year course from Secondary 3 to 4. Computing is taught over three curriculum hours per week and students learn Python programming as part of the curriculum. The national assessment includes a theory and a practical paper.

At Maris Stella High School, Computing students take an international test on computational thinking, the Design Thinking with Robotics and Computational Thinking International Competition (DrCT) test, to cultivate their logical reasoning ability and computational thinking skills. This test provides students with the opportunity to develop the following four key aspects of CT:

- Decomposition: Breaking down processes or problems into smaller, manageable parts
- Pattern Recognition: Observing patterns, trends, and similarities
- Abstraction: Removing unnecessary details and generalizing principles from specific instances
- Algorithm Design: Developing the step by step instructions for solving this and similar problems

Based on teacher observations, students who experience challenges in programming exhibit the following attitudes (Gomes & Mendes, 2007):

1.1. Problem Understanding
Students try to solve a problem without completely understanding it. The problem stems from difficulties in interpreting the problem statement and anxiety in starting to write code without first reading the problem correctly.

1.2. Relating Knowledge
Students do not establish correct analogies with past problems and may not be able to connect patterns of new problems with prior ones.

1.3. Reflection about the Problem and the Solution
Students tend to delve straight into writing code without thinking carefully about the algorithm required. This often leads to them missing out on steps, and developing frustration while in the midst of creating a solution.

2. AIM OF THE STUDY
The aim of this study is to explore how the KWS problem-solving strategy and Use-Modify-Create pedagogical approach can be used to develop CT skills in secondary school students in the context of formal Computing programming lessons.

3. APPROACH
Lesson design centred on the KWS strategy and the Use-Modify-Create (Lee et al., 2011) pedagogical approach. Figure 1 below summarizes the teaching process over four lessons. Each lesson duration is one and a half hours.

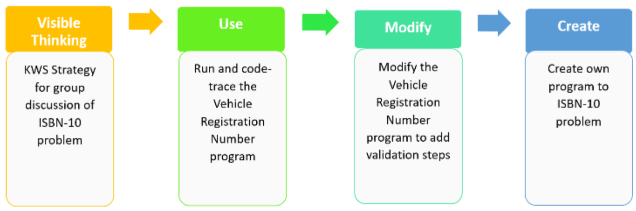


Figure 1. Summary of Teaching Process

The KNWS problem-solving strategy (Phonapichat & Wongwanich, 2013) has been used to solve Mathematical word problems, in particular to address problem-reading difficulties among elementary school students. N which stands for What information do I NOT need in order to solve this problem? was omitted as the problem contexts encountered at the secondary level mostly do not include irrelevant details.

Vygotsky views interaction with peers as an effective way of developing skills and strategies (Vygotsky, 1978). With this in mind and the need to guide and extend learners' thought processes, the lesson design included opportunities for collaborative learning and practicing of visible thinking routines using the KWS strategy.

In groups of four, students discussed their interpretation of a programming problem and their responses to the eight guiding questions in the KWS graphic organizer. The questions made links to the Input-Process-Output (IPO) model, which is commonly used to describe the structure of a computer program or system.

Table 1 below shows the mapping of the KWS strategy to the relevant CT skill and IPO structure.

Table 1. Mapping of KWS Strategy to CT Skill

Strategy	Description	CT Skill	IPO
K	What do I KNOW from the information stated in the problem?	Decomposition, Pattern Recognition	Input(s)
W	WHAT does the problem want me to find?	Decomposition	Output(s)
S	What STRATEGY or OPERATION will I use to solve the problem?	Decomposition, Pattern Recognition, Abstraction	Process(es)

In the first lesson, students were introduced to the ISBN-10 problem for group discussion using the KWS strategy. This helped to resolve issues of different interpretations and also aided students in making connections to similar problems.

Peer learning also took place as they were able to listen to different viewpoints and methods of their peers as they refined their solutions.

After the group discussion, students complete a set of activities based on the Use-Modify-Create pedagogical approach over the second and third lessons. This approach is based on the premise that scaffolding increasingly deep interactions will promote the acquisition and development of CT.

3.1. *Use*

A similar problem was shared. The Python code for validating a Singapore-registered vehicle number was given to students as a code-tracing and comprehension practice. Tracing code improves programming skills (Lister, 2011). Students worked in pairs to run and trace the code. They had to match each code block to a description. From this activity, students gained a first-level understanding of how a similar problem is being decomposed.

3.2. *Modify*

Next, the students were challenged to modify the program code to include data validation for new input. Modification of this necessitates an understanding of some level of abstraction contained within the program.

3.3. *Create*

Using the schema acquired, students had to submit a working Python program on the ISBN-10 problem.

Students were not restricted by the number of submissions as well as when to submit, as long as it was done by the end of the last lesson. Majority of students submitted at least two versions of the program code.

4. FINDINGS

A survey was conducted for 38 Computing students in September 2019. As shown in Figure 3, 60.5% of them strongly agreed that the KWS strategy helped them to develop the CT skill of problem decomposition.

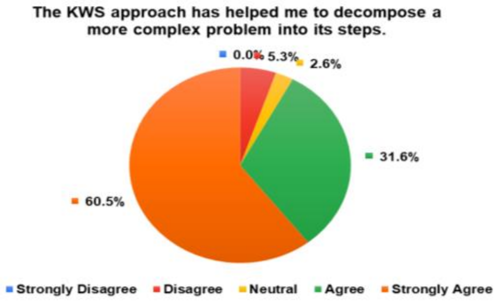


Figure 4. Survey Question: The KWS approach has helped me to decompose a more complex problem into its steps.

Besides the survey findings, an interesting empirical observation was made. Some students demonstrated comparatively higher levels of abstraction generalization in the later versions of their program code as compared to their first versions while there were no significant changes for some.

5. CONCLUSION

In this study, the survey results reflected that most students were more motivated in problem-solving and gained greater confidence in a group setting than individually.

While empirical observations and survey results reflected that most students were able to learn and develop some form of decomposition and pattern recognition skills, future studies could also include the use of a more measurable instrument for a more conclusive evaluation of the level of CT skills gained.

6. REFERENCES

Gomes, A. & Mendes, A.J. (2007). Learning to Program - Difficulties and Solutions. *Proceedings of the International Conference on Engineering Education – ICEE 2007*, 283-287.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2, 32-37.

Raymond, L., Colin, F., & Donna, T. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *Proceedings of the Conference on Integrating Technology into Computer Science Education* (*ITiCSE*), 161-165.

Phonapichat, P., Wongwanich, S. & Sujiva, S. (2014). An Analysis of Elementary School Students' Difficulties in Mathematical Problem Solving. *Procedia - Social and Behavioral Sciences*, 116, 3169-3174.

Vygotsky, L.S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.

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

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Fostering 21st Century Literacy and Computational Thinking Through Robotic STEM Design Competition Game

Hsin-yin HUANG[1*], Ju-ling SHIH[2*]
[1]Tainan Municipal Sinhua District Naba Elementary School, Taiwan, ROC
[2]Graduate Institute of Network Learning Technology, National Central University, Taiwan
franch28@gmail.com, juling@cl.ncu.edu.tw

ABSTRACT
Problem-solving, Computational Thinking (CT), STEM and 21st Century skill are essential skills to achieve in 21st century. To comprehend the learning consequent in this study, problem-solving including five CT dimensions, STEM and 21st Century skill attitudes were examined through questionnaires. There are about forty-six 4th,5th-grader primary school students who will be involved in the Robotic STEM Design game which is designed to enhance the effectiveness of the problem-solving. It was found that CT dimensions was correlated to the STEM and 21st Century skill aspects. Robotic STEM design competition game has positive effects to the students, and would conduct the students to have better performance in the complex real-life situations. This game in the research could improve students problem-solving skills.

KEYWORDS
computational thinking, STEM, 21st century skill, robotic game

1. INTRODUCTION
A number of robotic games have been created to help students programming. Many of these games focus on improving problem solving skills. Robotic is a unique learning agent which could offer interesting activities. Therefore, the interdisciplinary education of CT, STEM and 21st Century skill has become the world trend. We consider that computational thinking is a universal skill that can foster students' problem-solving skills across the primary school curriculum. But the students with good grade play the critical role to control even win the game without team spirit. The students with low grade sometimes could not show their talents in the game. For enhancing students' motivation to improve their CT, STEM and 21st Century skill performances and promote their self-confidence, we would design a robotic game with LEGO EV3 to enable student to apply their existing knowledge, CT and problem-solving skills to go through all stages of the robotic game.

2. RELATED WORK

2.1. Game-Based Learning & Robotic Game
Game-based learning (GBL) use the game as a learning activities in order to get the instructional goal, and learners will learn skills and construct knowledge through games. (Zainuddin, 2018). GBL is an effective learning method that can improve problem solving skills and provide learners with a sense of achievement(Kuo & Hsu, 2019; Qian & Clark, 2016). Students design, construct, and program robots in robotic game who are exposed to computational concepts such as decomposition, abstraction and algorithm during this process of game (Chalmers, 2018).

2.2. Computational Thinking & STEM
CT apply fundamental concepts that derive from computer science in general and computer programming/coding in particular, including real life activities and to solve daily life problems(HUANG, HUANG, SHIH, TSAI, & LIANG, 2019). CT is a problem-solving process that includes Decomposition, Evaluation, Abstraction, Algorithm and Generalization. Finally, students could generalize and transfer the problem-solving process to the other problems and real-life(Leonard et al., 2016). All stage students should learn and practice these problem-solving skills. Teachers have to strengthen the elements of CT in interdisciplinary curriculum (Huang, Shih, Huang, Liang, 2019).

2.3. STEM & 21st Century Skills
STEM is an interdisciplinary and cohesive learning paradigm which based on real-world application (Afari & Khine, 2017). STEM could improve students' higher order thinking skills and technological literacy, making them be better problem solvers, innovators and inventors (Stohlmann, Moore, & Roehrig, 2012). STEM learning is an integrated learning way for learners and connected to the real world of business and research. In 21st century, Students should be required a new set of competences and skills such as critical thinking, creativity and innovation and problem-solving skills. The skills are so important to students and they should learned early in primary school levels both in formal and informal learning settings(Usart, Schina, Esteve-Gonzalez, & Gisbert, 2019). Students have the opportunity to work together to foster collaboration, using robot, problem-solve, and think critically and innovatively (Eguchi, 2017).

3. RESEARCH DESIGN
Students are divided into defenders' and attackers' groups. The defenders set up the context of the game such as terrain, obstacle, route, detector and the traps. The attackers use strategies and design robots to break through the defenders' towers. This game works like a tower defense game. Before the game, students have to learn to use and create with LEGO EV3 blocks and sensors. First, they have to learn how to assemble the blocks for the suitable structure to complete the task such as throw a ball to attack the defenders' bunker or attackers' vehicle, or to preclude the barrier. Secondly, students have to learn how to write computer programs to control the sensors and decide in what situation to use the tools. Third, they have to learn block-coding to control the LEGO EV3 robots and receive information from the sensor in game context map and game environment to decide where to go and what to do. Students have to cooperate with each

other and apply their interdisciplinary knowledges and computational thinking skills to achieve their goals of the game. Finally, the attackers and defenders set up the game context and build the robots to play the game. The evaluation to the students' performance in the game uses the evaluation rubrics to assess students' attitudes, product, and design process. In every stage of the product development, students' discussions about the robot design, their prototype drafts, and the prototypes are collected and analyzed. So they assemble the blocks to complete their robot design products, and the combat outcomes were also recorded for cross analysis of students' performance. After the game was finished, questionnaires of 21st Century skill, STEM and CT was conducted to know their attitudes. It's also important to conduct Semi-structured interview to understand that what students learn and what's their opinion?

Table 1. Evaluation rubrics

<table>
<tr><th></th><th>Structure</th><th>Sensor& Movement</th><th>Block-coding</th><th>Game-Task</th></tr>
<tr><td>Design diagram</td><td colspan="4">Collaborative</td></tr>
<tr><td>Prototype</td><td rowspan="3">Creative
Practical</td><td rowspan="3">Suitable
Work</td><td rowspan="3">Correct
Efficient
Skills</td><td rowspan="3">Movement
Get card</td></tr>
<tr><td>Product</td></tr>
<tr><td>Adjustment</td></tr>
<tr><td>Literacy</td><td colspan="4">Questionnaire: 21st Century skill, STEM, CT
Semi-structured interview</td></tr>
</table>

4. RESULT AND DISCUSSION

In each stage of the game, students have to finish the mission with discussion. All design diagram, prototype, and products should be different by different groups and stages. Students need to learn how to draw design diagram and build up the prototype. The 21st Century skills show out such as critical thinking and team work. One man could not overcome the mission so students could learn collaboration, CT and the spirit of team work. With the students' feedback in each learning stage, teacher could observe their learning situation and show the hint to students.

5. CONCLUSION

This game could improve students' CT skills, attitude of STEM and 21st century skills and help students to integrate and apply the interdisciplinary knowledge and skills (Plass, Homer, & Kinzer, 2015). Students learn how to win the game with discussion in the complexity learning context. The pedagogical approach of involving the students that contributed to students' motivation and success in learning the course (Barak & Assal, 2018). Although LA students cannot achieve as much as those with HA students but they were highly motivated in their problem-solving tasks even without extrinsic rewards and scores. Teachers should to help them to get better Algorithm skills so that they can accomplish more in the strategic game and problem-solving tasks, and can have better performance in general.

6. REFERENCES

Afari, E., & Khine, M. (2017). Robotics as an Educational Tool: Impact of Lego Mindstorms. *International Journal of Information and Education Technology, 7*(6), 437-442.

Barak, M., & Assal, M. (2018). Robotics and STEM Learning: Students' Achievements in Assignments According to the P3 Task Taxonomy—Practice, Problem Solving, and Projects. *International Journal of Technology and Design Education, 28*(1), 121-144.

Chalmers, C. (2018). Robotics and Computational Thinking in Primary School. *International Journal of Child-Computer Interaction, 17*, 93-100.

Eguchi, A. (2017). Bringing Robotics in Classrooms. *Robotics in STEM Education*. Springer, 3-31.

Huang, H. Y., Huang, S. H., Shih, J. L., Tsai, M. J., & Liang, J. C. (2019). Exploring the Role of Algorithm in Elementary School Students' Computational Thinking Skills from a Robotic Game. *Proceedings of International Conference on Computational Thinking Education 2019*, 217-222.

Huang, H. Y., Shih, J. L., Huang, S. H., & Liang, J. C. (2019). Effects of the Interdisciplinary Robotic Game to Elementary School Students' Abilities of Computational Thinking and STEM. *Proceedings of 27th International Conference on Computers in Education*. Asia-Pacific Society for Computers in Education.

Kuo, W. C., & Hsu, T. C. (2019). Learning Computational Thinking Without a Computer: How Computational Participation Happens in a Computational Thinking Board Game. *The Asia-Pacific Education Researcher*, 1-17.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *Journal of Science Education and Technology, 25*(6), 860-876.

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist, 50*(4), 258-283.

Qian, M., & Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior, 63*, 50-58.

Stohlmann, M., Moore, T. J., & Roehrig, G. H. (2012). Considerations for Teaching Integrated STEM Education. *Journal of Pre-College Engineering Education Research (J-PEER), 2*(1), 4.

Usart, M., Schina, D., Esteve-Gonzalez, V., & Gisbert, M. (2019). Are 21st Century Skills Evaluated in Robotics Competitions? The Case of First LEGO League Competition.

Zainuddin, Z. (2018). Students' Learning Performance and Perceived Motivation in Gamified Flipped-class Instruction. *Computers & Education, 126*, 75-88.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Computational Thinking Unplugged in CS Middle Class Lessons in Lithuania

Eglė JASUTĖ
Vilnius University Institute of Educational Sciences, Vilnius Jesuit High School, Lithuania
egle.jasute@mif.vu.lt

ABSTRACT
The paper deals with computational thinking in the basic school classes at Vilnius Jesuit High school. The resources used for developing computational thinking of 10-15 years students are presented: CS Unplugged and Algorithms unplugged. Some examples of unplugged activities are described in this paper.

KEYWORDS
CS unplugged, basic education, computer science, computational thinking

1. INTRODUCTION

Computational thinking (CT) is a fundamental life skill of today person. An active promoter of CT prof. Jeannette M. Wing (2012) believes that CT is the thinking processes involved in formulating a problem and expressing its solution in a way that a computer – human or machine – can effectively carry out. In the context of Computer Science (CS) education, one of the most important goals is the development of CT and problem solving skills. CT skills are closely related to the CS concepts and approaches. On the one hand, these skills cover fundamental programming concepts such as task decomposition, abstractions and generalization, data structures and algorithms. On the other hand, CS education challenges, promote the search for new educational tools, methods, activities and resources (Burbaite et al, 2018). Problem solving, inquiry based, and collaboration methods are used to develop students skills needed for CT development. There are plenty of new materials and tools to help students to improve their CT in the middle classes of CS: CS Unplugged, Code Monkey, Code.org, Scratch, Robotics, Bebras Content and others.

The idea to strengthen students' CT skills without computer is presented in this paper. The unplugged computing activities are making significant positive change of students' CT skills (Delal and Oner, 2020; Conde et al., 2017) and the inclusion of unplugged activities in the instruction are beneficial taking into account CT, motivation and gender (Olmo-Muñoz et al., 2020).

Vilnius Jesuit High school is a school for the gifted students located in the old town of Vilnius. CT is integrated in all subjects such as Mathematics, Nature Science or CS. Some examples of CT unplugged activities for the CS lessons are described in this presentation.

2. CS UNPLUGGED ACTIVITIES FOR 10-11 YEARS STUDENTS

CS Unplugged activities developed by the Computer Science Education Research Group at the University of Canterbury in New Zealand are intended to give a feel for what the subject is. The subject can be learned by doing. Presented activities give students deeper understanding how computer works when it is solving problems. So students become computer thinkers and get the understanding of how to prepare a problem for computer to solve (CS Unplugged, 2019). CS Unplugged can be used to achieve positive results, helping both students and teachers to explore computer science in a meaningful and engaging way (Bell, 2018). The activities are performed using worksheets, explanations, videos and offer more complicated tasks for more engaged students.

Example 1. When data are stored on a disk or transmitted from one computer to another, we usually assume that it doesn't get changed in the process. But sometimes things go wrong and the data are changed accidentally. This activity uses a magic trick to show how to detect when data have been corrupted, and to correct it.

1. You will need a pile of identical, two-sided cards.

2. Choose a student to lay out the cards in a 5 × 5 square, with a random mixture of sides showing.

3. Casually add another row and column, "just to make it a bit harder". These cards are the key to the trick. You must choose the extra cards to ensure that there is an even number of coloured cards in each row and column.

4. Get a student to flip over one card only while you cover your eyes. The row and column containing the changed card will now have an odd number of coloured cards, and this will identify the changed card. Can the students guess how the trick is done?

Extensions of activity. It can be used with other objects. Anything that has two 'states' is suitable. For example, you could use playing cards, coins (heads or tails) or cards with 0 or 1 printed on them (to relate to the binary system).

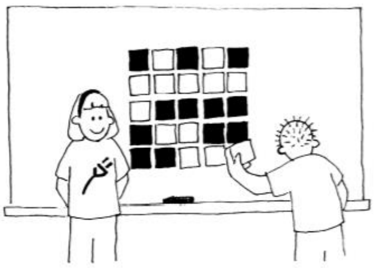


Figure 1. Magic Trick from CS Unplugged

Example 2: Computers are often used to put lists into some sort of order, for example names into alphabetical order, appointments or e-mail by date, or items in numerical order. Sorting lists helps us find things quickly, and also makes extreme values easy to see.

If you use the wrong method, it can take a long time to sort a large list into order, even on a fast computer. Fortunately, several fast methods are known for sorting. In this activity students will discover different methods for sorting and see how a clever method can perform the task much more quickly than a simple one.

Aim: To find the best method of sorting a group of unknown weights into order.

You will need: Sand or water, 8 identical containers, a set of balance scales.

What to do:

1. Fill each container with a different amount of sand or water. Seal tightly.

2. Mix them up so that you no longer know the order of the weights.

3. Find the lightest weight. What is the easiest way of doing this?

Note: You are only allowed to use the scales to find out how heavy each container is. Only two weights can be compared at a time.

4. Choose 3 weights at random and sort them into order from the lightest to the heaviest using only the scales. How did you do this? What is the minimum number of comparisons you can make? Why?

5. Now sort all of the objects into order from the lightest to the heaviest. When you think you have finished, check your ordering by re-weighing each pair of objects standing together.

One method a computer might use is called a selection sort. This is how selection sort works. First find the lightest weight in the set and put it to one side. Next, find the lightest of the weights that are left, and remove it. Repeat this until all the weights have been removed. Teacher can watch or show to students video, how a selection sort works: (https://www.youtube.com/watch?v=cVMKXKoGu_Y&feature=emb_logo.)

3. ALGORITHMS UNPLUGGED FOR 13-15 YEARS STUDENTS

Algorithms are very important component of CT, which focus on a problem solving process. Algorithms specify the way computers process information and how they execute tasks. The collection of the tasks developed by prof. Valentina Dagienė and her team at Vilnius University is one of the opportunities for students to make their learning more interesting, engagement and more understandable.

Bebras tasks contain concepts of about nearly all areas of informatics – half of them are devoted to algorithms. Bebras tasks present a motivating way to introduce CS concepts to students as well as developing CT skills and can be used within the school curriculum to promote CT and provide teaching materials (Dagiene, 2016).

Example 1: Beaver wants to try a new underwater metro. Starting at home stop A, he wants to visit every other stop only once! Beaver is using the Nearest Neighbour Algorithm: in every stop he will choose the nearest stop from all of unvisited stops yet (the shortest distance in kilometres). From the last stop go home.

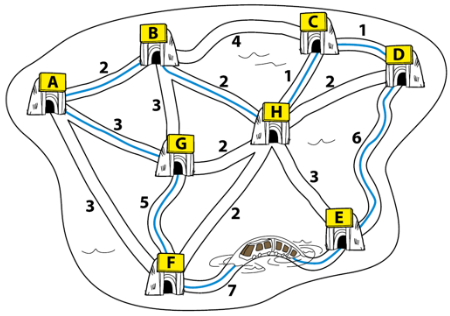


Figure 2. Task of Computational Thinking: Choosing the Cheapest Is Not Always Cheap

Beaver's route using the Nearest Neighbour Algorithm is: ABHCDEFGA and it is 27 km long.

Prove this route is not the shortest one.

The Nearest Neighbour Algorithm: Start at a random point and repeatedly visit the nearest points until all have been visited. Then come back to the start point.

4. REFERENCES

Burbaitė, R., Drąsutė, V., & Štuikys, V. (2018). Integration of Computational Thinking Skills in STEM-driven Computer Science Education. *Proceedings of 2018 IEEE Global Engineering Education Conference (EDUCON)*, 1824-1832.

Bell T., Vahrenhold J. (2018). CS Unplugged—How Is It Used, and Does It Work? *Adventures Between Lower Bounds and Higher Altitudes*, *11011*, 497-521.

Conde M. A., Fernández-Llamas C., Rodríguez-Sedano F. J., Guerrero-Higueras A.M., & Matellán-Olivera M. (2017). Promoting Computational Thinking in K-12 Students by Applying Unplugged Methods and Robotics. *TEEM 2017: Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality*, 1–6.

CS Unplugged. *Computer Science Without a Computer*. Retrieved August 5, 2020, from https://csunplugged.org/en/

Dagiene, V. (2016). It's Computational Thinking! Bebras Tasks in the Curriculum. *ISSEP 2016: Informatics in Schools: Improvement of Informatics Knowledge and Perception*, 28-39.

Delal H., Oner D. (2020). Developing Middle School Students' Computational Thinking Skills Using Unplugged Computing Activities. *Informatics in education*, *19*(1), 1-13.

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*.

Wing, J. M. (2012). Computational Thinking. *Microsoft Asia Faculty Summit.* Tianjin, China.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Teaching MIT App Inventor Programming for Computational Thinking Development in Grade 6 with 'To play, to think, to code' Approach: A Two-button Game

Wai-kei LUK
Maryknoll Fathers' School (Primary Section), Hong Kong
vickyluk02721@gmail.com

ABSTRACT

The main focus of this essay is to illustrate a pedagogical method used in an MIT App Inventor programming lesson. A class of 26 students in Grade 6 will be used as an example to demonstrate the pedagogy - a Two-button Game. It is a game which requires two players to play on separate devices. One point will be awarded for each press of button. Whoever scores the most points wins. The approach, 'To play, to think, to code' (Kong, Lai & Sun, 2020), was adopted. To captivate students and arouse their interests, they were asked to play the game in the beginning. Then some unplugged activities and discussions were carried out to stimulate students' thinking. This is also where computational thinking (CT) was embedded. Not only did students learn the way to code, but they also learned the way to think and create. Throughout the activities, students were provided with some necessary concepts as foundation. After that, students would start to code in pairs. The construction of a complete app did not simply rely on a single flow mentioned above. The contents of this unit were divided into different sections according to the difficulty and complexity of the learning focuses. At least one learning focus was included in each section. The same teaching approach went on in each section. Scaffolding was constructed in building students' knowledge bits by bits, from easy to difficult. With this approach and suitable guidance, students could successfully build a mobile app, Two-button Game, from scratch to hatch.

KEYWORDS

computational thinking, MIT App Inventor, pedagogy, primary school students, programming

1. INTRODUCTION

Coding education has become a global trend. Governments across the globe have gradually included computer programming in their national curriculum, including Hong Kong (Rich & et al., 2017). The importance of learning programming is not 'primarily about equipping the next generation to work as software engineers, it is about promoting computational thinking' (Crow, 2014). Equipping students with CT is essential because it is 'a fundamental skill for everyone' that 'involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science' (Wing, 2006). Besides, CT is the skill that plays an important role in the learning process of children because it forms a better collaboration between human and machine (Martín, 2016). Therefore, coding paves the way for teachers to teach and students to learn CT.

2. THE DEVELOPMENT OF A TWO-BUTTON GAME

Students made use of the MIT App Inventor platform to develop a mobile app. Two-button Game is the app that allows students to battle with each other on two devices. Through the development of the app, students learnt new blocks for CloudDB components. The knowledge about 'Events', 'Parallelism', 'Repetitions' and 'Conditionals' were reinforced. The layout of Two-Button Game is shown in Figure 1.

Figure 1. Layout of Two-button Game

3. PEDAGOGICAL PRACTICES IN PROGRAMMING THE APP

The lessons were divided into four main parts. First, it was about forming blocks for the start buttons and the scores. CT concepts about 'Events' and 'Parallelism' were covered. The second part was about forming the clock, which focused on 'Conditionals'. Then, the third part was about linking two devices with CloudDB. The final part was a challenging one about 'Parallelism'. Students worked in pairs throughout the whole lesson.

According to a survey of 357 teachers in England done by Sentence and Csizmadia in 2015, when teachers were asked about the most effective way to teach coding and CT, teachers 'emphasised unplugged, hands-on, contextualised activities and the importance of lots of practice'. With regard to this, an approach 'To play, to think, to code' was applied in order to achieve an effective lesson.

In the first and second part of the lesson, students were allowed to play a semi-final version of the game. This time, both students played the game on the same device. Playing the game in advance did not just serve the purpose of arousing students' interests, but it also allowed students to understand the flow of the game. To transfer an abstraction into an algorithm, students were asked to complete a flow chart worksheet in pairs. Details of the worksheet are shown in Figure 2.

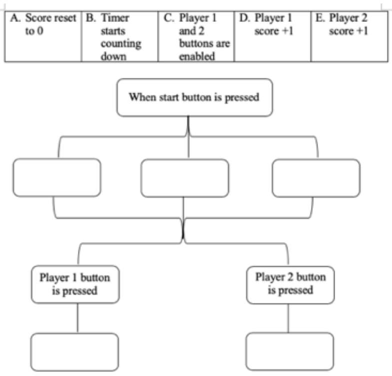


Figure 2. Worksheet for Thinking about the Flow of the App

After that, students were introduced to the layout of the app. Most of the interface components were given to them so as to help them focus on the thinking process and code-building. Then each pair of students was given some printed block pieces and asked to select suitable components for each 'Event'. Some distractors were also given to test their understanding. This activity prepared the students for coding because it encouraged them to think about the parts they need and steps they would take in programming the game. An example is shown below for building the start button.

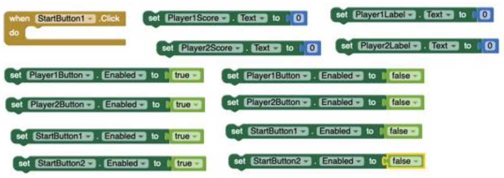


Figure 3. Printed Blocks for Students to Build the Start Button

For the third part of the lesson, students would play the final version of the game on two devices and complete a flow chart similar to the one in the previous stage. It helps students gain a basic idea of how CloudDB works. Following that, a more interactive role-play activity was carried out to facilitate students' learning.

The whole class was divided into four big groups. Each group consisted of three pairs of students. They would act as Player 1, Player 2 and CloudDB respectively. Ten to fifteen minutes were given to the whole class to discuss and prepare for the presentation showing how CloudDB gets, stores and transfers the data from one device to another. Each group would have three minutes for the presentation. A final reveal of the answer would be done after all groups had shared their ideas. This activity brought great fun to students, and it deepened the understanding of students towards CloudDB. Through the discussion and role-play planning process, students did not passively receive knowledge from the teacher. They, however, participated actively in exploring and puzzling out the operation of CloudDB. The abstract 'get, store and transfer' concept was presented in a more interesting and interactive way.

In the final stage, individual guidance would be offered to each pair of students depending on their progress. Some challenging tasks would be introduced to them with appropriate guidance and support.

Debugging is very important in programming. It was hoped that by putting students in pairs they would support each other, especially for the 'testing and debugging' process. Students were encouraged to test their game at each stage throughout the whole coding process. Whenever they found a problem in their game, they had to go back to the codes, spot the problem and think of a solution with the CT knowledge they had acquired. It could ensure that they could build their app in an incremental and iterative manner.

4. A REFLECTION ON THE TEACHING FOR COMPUTATIONAL THINKING DEVELOPMENT

Coding is not an one-off practice. It gears students up with a life-long thinking skill. Teaching coding is not only about teaching students the way to code, but the way to think. The goal of it is to help students apply and incorporate CT into other aspects of their learning, and even their daily lives.

5. CONCLUSION

The 'To play, to think and to code' pedagogy shows a great success in this lesson. Students benefited from the unplugged activities because they ensured that students had sufficient knowledge, skill and confidence before coding. Hence, this pedagogy and the unplugged activities will be the pillars for the lessons in the future.

6. REFERENCES

Crow, D. (2014). Why Every Child Should Learn to Code. *The Guardian.* Retrieved February 3, 2020, from https://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick

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

Martín, J. (2016). Por qué debemos enseñar a los niños a programar. Retrieved February 3, 2020, from http://futurizable.com/debemos-ensenar-los-ninos-programar

Rich, P., Browning, S., Perkins, M., & Shoop, T., Belikov, O., & Yoshikawa, E. (2017). *Coding in K-8: International Trends in Computing Education with Primary-aged Children.* DOI:10.13140/RG.2.2.29782.14409/1

Sentence, S. & Csizmadia, A. (2015). Teachers' Perspectives on Successful Strategies for Teaching Computing in School. A. Brodnik & C. Lewin (Eds.) (2016), *IFIP TC3 Working Conference "A New Culture of Learning: Computing and next Generations": Preliminary Proceedings,* 201-210.

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

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Using Iteration Concept in Teaching MIT App Inventor Programming in Primary 5-6 students for Computational Thinking Development

Kin Chung William LUK
The Hong Kong Baptist University Affiliated School Wong Kam Fai Secondary and Primary School, Hong Kong
wkcluk@hkbuas.edu.hk

ABSTRACT

This paper is illustrating a pedagogical practice in teaching MIT App Inventor programming. Serval classes of Grade 5-6 were learning to develop various apps. To promote self-directed learning, the Teacher tried to use the idea of "iteration" and MVP (Minimum Viable Product) to lead students to start and complete a basic and simple version of the product first, by having more iterations, they keep enhancing their products. In contrast to the code-by-code or function by function practice, the iteration practice let student has his own product according to ability, this positively facing the learning diversity and developing their self-directed skills.

KEYWORDS

iteration, minimum viable product, computational thinking, MIT App Inventor, K-12 coding

1. INTRODUCTION

Since Scratch came to Hong Kong in 2010, coding training began to play a crucial role in the primary school computer subject curriculum. The CoolThink project in 2016 changed the focus of the coding training in primary schools, from a focus on syntax, adjusted to the development of computational thinking.

As most of the computer subject teachers don't have the necessary training of programming concepts, teachers always just teach coding in their own ways.

My students tried to learn coding in a new way in the past two years. Students were found to be more engaged and feel more satisfaction.

2. THE TEACHING METHODS THAT COMMONLY ADOPTED PREVIOUSLY

2.1. Learn from model code listing code-by-code

Teachers always expect their students to complete the tasks in the same way as their model answers, so they always use the model codes to teach in the form of code-by-code explanations.

2.2. Students build the program function-by-function

Some teachers try to break down a complicated task into several small parts through function or component. Students then develop their app function-by-function, according to the workflow.

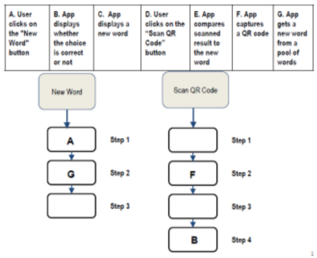


Figure 1. Worksheet for Thinking about the Flow of the App

3. THE ITERATION CONCEPT

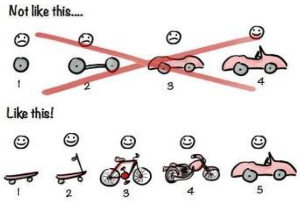


Figure 2. The Diagram Shows the Idea of Iteration

I met the idea of iteration in the field of Product Design, and I tried to apply the concept in my coding lessons.

Different from code-by-code, or function-by-function, students complete various iterations from simple to complicated, from raw to details, every iteration should be a complete task, known as the minimum viable product.

Example 1 My piano (from single to multi)

• Layout • Buttons • Coding	• C note • Other notes • Layout
function-by-function	by iteration

Figure 3. The Table Contrasts the Two Different Lesson Flows

If the teacher follows the code-to-code method, some students may be unable to follow, and some may stick at the layout arrangement in the very beginning and thus have no chance to learn to code.

If we go by iterations, students will start from a single button first. By reuse and remix, a standard practice in Computation Thinking, most of my students can build a raw piano that was playable, although not everyone can handle the layout, students felt satisfactory in the activity because everyone can play with the app they built.

Example 2 Control ball movement by the accelerometer (from basic to combined)

Ask students just to follow the teacher's model answer is simple, but usually, they seem able to read the codes, they still don't know how to generate the codes. Like the codes list below: why does the value of Ball1.X can add together the value of yAccel equals to the new position of the Ball?

Figure 4. The Codes Calculate the New Location of the Ball according to the Tablet Position

However, by using the 'iteration' learning method, we just consider a single direction in one dimension first, then other dimensions. Students catch up quickly, and by reuse and remix, at last, most of the class can develop the four directions by themselves:

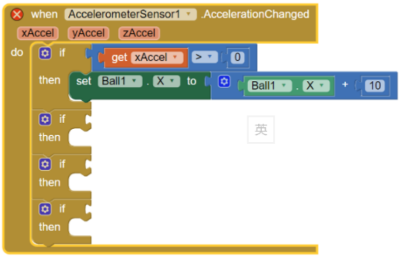


Figure 5. The Codes Calculate the New Position of the Ball according to the Tablet Position in Four Directions

Find the x-y coordinates of the ball from the output of the accelerometer in two equalities	**Determine the x-y coordinates of the ball from the output of the accelerometer in +ve horizontal direction** **in -ve horizontal direction** **in +ve vertical direction** **in -ve vertical direction**
code-by-code	**By iteration**

Figure 6. The Table Contrasts the Two Different Lesson Flows

Example 3 Find all factors (from specific to general)

In this example, students have to write a simple program to list out all factors of a whole number inputted by the user. Instead of asking students to build an app of finding factors of any given numbers at the very beginning, I followed the concept of iteration, this time, from specific to general.

I ask my students to find the factors of a fixed number first. This step is critical but straightforward. I found that it was more natural for the students to understand how it works if we fix a number. Then just use the mix and reuse practice and variable concept, most of the students, can extend the usage of the app to any whole numbers.

Compare with the code-by-code method, by the iteration approach, more students able to complete the final task by themselves Use a table to find out the factors of a fixed number (8) Waite an app list factors of any number	Use a table to find out the factors of a fixed number (8) Write an app list factors of a fixed number (8) Waite an app list factors of any number
code-by-code	**By iteration**

Figure 7. The Diagram Contrasts the Different Lesson Flows

Example 4 Two buttons Game with CloudDB (from simple to more complex)

In this example, I tried to use the iteration method by going from simple to complex, instead of the usual way function by function.

two buttons, score fairer mechanism CloudDB transfer completed	a local two buttons game game support two players by CloudDB game support two players by CloudDB with a fairer design
function-by-function	**By iteration**

Figure 10. The Diagram Contrasts the Different Lesson Flows

In the function-by-function method, the intermediate product is not a completed program, and it cannot run, test, or debug directly until the completion of the whole program.

If students build the program from simple to complex, I found that it was easier for them to understand in small steps, and they could test and debug a shorter program, so they felt more satisfaction, and thus more students could complete the task.

4. CONCLUSION

The "iteration" pedagogy is a promising approach that nurtures student's problem-solving skills and promotes their self-directed learning. I shall continue to apply this pedagogy in my teaching practices in computational thinking education and share with my teaching peers to enhance this pedagogy.

5. REFERENCES

Kniberg, H. (2016). Making Sense of MVP (Minimum Viable Product) – and Why I Prefer Earliest *Testable/Usable/Lovable.* Retrieved Jan 10, 2020, from https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Ewha Hackathon Program for Improving Elementary Students' Computational Thinking Based on Design Thinking Process

Ju Yeon PARK[1], Hye Young CHUNG[2*], Sung Hee KIM[3], Su Bin CHO[4], Young Mi LEE[5], Yoo Kyung LEE[6], Hye Sun YOON[7], Jee Eun PYO[8], Jae Ho LEE[9], Won Kyung LEE[10], Jung Ah LEE[11], Eun Bi KIM[12]
[1~12]Ewha Womans Elementary School, Seoul, Korea
pjy32622@gmail.com, hychung@ewha.ac.kr, kstarhee@sen.go.kr, josubin@sen.go.kr, gngngnt@sen.go.kr, lyk2601004@sen.go.kr, suba1996@sen.go.kr, jepyo407@sen.go.kr, redline80@naver.com, ranny00@sen.go.kr, junya730@sen.go.kr, eunbbb@gmail.com

ABSTRACT
This case study illustrates the design and application of the Ewha hackathon program based on the design thinking process to improve computational thinking for the 4th-6th grade students in Korea. This study explores the possibility of applying the Hackathon program and provide information on the design of the hackathon program as a collaborative problem-solving process.

KEYWORDS
hackathon, design thinking, computational thinking, collaborative problem solving, group CT

1. INTRODUCTION
In Korea, Software (SW) education has been implemented for 17 hours in the 6th grade from 2019 to improve Computational Thinking (CT) in elementary schools. However, to improve CT through creative and convergent problem solving experiences, it is necessary to provide practical experiences that students can directly solve the problem, and provide them with a project and enough time to concentrate on. One alternative is to use hackathon. The word hackathon is combined from the hacking and marathon, that gives you the opportunity to immerse yourself in solving a problem(Briscoe, 2014). Therefore, this study designed and implemented a Hackathon Program coupled with the design thinking process suitable for elementary school students in Korea.

2. DESIGN THINKING PROCESS
Design Thinking is a problem-solving process and thinking method that began in the design field. This method is a creative idea model that emphasizes empathy on problems and visualization of thoughts (Brown, 2011; Carroll et al., 2010). The Design Thinking process at D-School(2009), Stanford University consists of five steps: 1) Empathize: a process to understand a problem; 2) Define: describes problems that can be realized through observation from a user-focused perspective; 3) Ideate: an exploration process that focuses on expanding rather than focusing, exploring a wide range of solutions through large amounts of ideas; 4) Prototype: the process of making prototypes quickly and inexpensively; and 5) Test: improves the prototype produced and gives the opportunity to go further.

As another framework, the Design Thinking process proposed by IDEO(2011) has five steps 1) Discovery: understand the challenge, prepare research, gather inspiration; 2) Interpretation: tell stories, search for meaning, frame opportunities; 3) Ideation: generate ideas, refine ideas; 4) Experimentation: make prototype, get feedback; and 5) Evaluation: improve the prototype, move on further with the prototype.

The Ewha Hackathon process implemented in this study was established in five stages: 1) Empathy: be aware of the problem situations, discover the problems; 2) Defining the problem: narrow down the problem to solve; 3) Developing Ideas: create ideas by team, use brainstorming, come out with as many ideas as you can, and avoid judging others; 4) Prototyping: implement the idea in SW, and 5) Test: test the completed program and fix errors.

3. DESGIN OF THE EWHA HACKATHON PROGRAM
The topic of the hackathon program was "Coding Together, Ewha Smart School". The topic was chosen to reflect the main idea of the SW education policy for a 'fun' and 'safe' school life. The topic is to collect ideas that can improve the situation and environment of the school, which is the closest and the most practical problem for the students. The participants were 36 students at Ewha Elementary School, and each team consisted of four members from 4th to 6th graders. The mixed-grade grouping in the team composition was used to increase communication and collaboration skills.

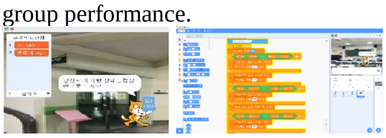
Two orientation sessions were provided to help students understand the program. The first session introduced the meaning of Ewha Hackathon, the schedule and instructions for the competition, and the topic presented in a video. In the second session, students learned about the SW tool and Design Thinking process.

The program was supported by 11 teachers, mentor teachers (16 college student volunteers who majored in Computer Science), and a professor of Computer Science and the principal of our school as judges. Mentors only provided technical advice and support to help students solve problems with their ideas. They did not provide ideas or do any coding for them.

4. IMPLEMENTATION OF THE EWHA HACKATHON PROGRAM
In Ewha Hackathon, it consists of five steps and seven detailed activities based on design thinking (see Table 1).

Table 1. Activities in Each Step

Step	Activities
Empathy	Meet the task - Watch a video about a task from a Computer Science professor

	<Task> *….Dear children, have you ever thought of ideas for a fun, safe and smart school? New and cool ideas are not far away from us…. Think about the places you go in school every day. Today, your ideas can be fun, safe and smart. I look forward to your wonderful ideas…* (Activity I) Write down wishes / what they would want to have, dissatisfaction / improvement, and what to do.
Defining the Problem	Collect data to accomplish the task (Activity II) Discuss and record the elements necessary to solve the problem in the activity sheet, collect related materials to find more ideas, and record in the post-it. 
Developing Ideas	(Activity III) Share ideas on problem solving, choosing the best idea, choosing a role for the task. (Activity IV) Sketch simply by drawing and writing what you want to make. 
Prototyping	Finding solutions to problems based on discussions Implement the task solution in SW (Activity V) Realizing ideas using Scratch, Entry, Hamster Robot, etc.
Test/ Feedback	(Activity VI) Test the work, correct errors and make up. (Activity VII) Present and evaluate group performance.  <Evaluation Elements> 1) Problem Solving 2) Creativity 3) SW Utilization Ability 4) Collaboration Ability 5) Presentation Skills <Award Name> -Idea, Work Completion, Collaborative, Bonus

5. CONCLUSION

The necessity of designing an Elementary Hackathon project was identified through the Design Thinking process. In a survey (on a 5-point scale) conducted after participating in the Hackathon program, the students scored high on the awareness of the importance in SW, their interest in SW education, their willingness to continue SW education, their satisfaction with Hackathon, the value of collaboration realized through Hackathons, and the confidence in Computational Thinking (see Figure 1).

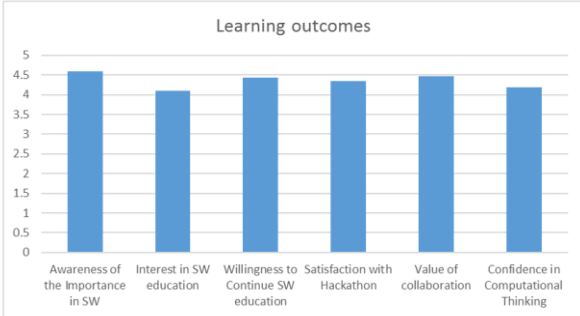


Figure 1. Learning Outcomes

Each team experienced solving problems collaboratively. In a follow-up questionnaire on how many students liked it, they provided the comments as:

'It was good to be able to solve the problem with my seniors.'

'It was a good process to complete the program by collaborating and sharing roles.'

In addition, students stated what they had learned as the following:

'I found that co-operation has more power.'

'I could feel the need and fun in SW'

In summary, through the hackathon program, students realized how important the thinking and collaboration process in Design Thinking was to improve their Computational thinking. Also this study is meaningful in that students' individual Computational Thinking are collaboratively gathered to produce new outputs. In addition to the improvement of individual Computational Thinking, collaborative Computational Thinking will be further studied in future research.

6. REFERENCE

Briscoe, G. (2014). Digital Innovation: The Hackathon Phenomenon. *Creativeworks London Working Paper, 6* , 1-13.

Brown, T., & Katz, B. (2011). Change by Design. *Journal of Product Innovation Management, 28*(3), 381-383.

Carroll, M., Goldman, S., Britos, L., Koh, J., Royalty, A., & Hornstein, M. (2010). Destination, Imagination and the Fires within: Design Thinking in a Middle School Classroom. *International Journal of Art & Design Education, 29*(1), 37-53.

Stanford D. School (2009). *Bootcamp Bootleg.* Hasso Plattner: Institute of Design at Stanford.

IDEO (2011). *Design Thinking for Educators*. Retrieved Dec. 28, 2019, from https://designthinkingforeducators.com/toolkit/

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Computer Science Teaching in Vilnius Lyceum, Lithuania

Bronius SKŪPAS
Vilnius Lyceum, Lithuania
Bronius.Skupas@licejus.lt

ABSTRACT

In this paper there is presented teaching computational thinking in Vilnius Lyceum, Lithuania. The paper discusses the main challenge of CT teaching – dropping student interest in CS courses. Efforts to make CS teaching more attractive and less disappointing are also presented.

KEYWORDS

CS popularity, reasons for CS dropout, effort for CS attractiveness

1. ABOUT VILNIUS LYCEUM

Since 1995 I have been working as a Computer Science teacher in Vilnius Lyceum – one of the best secondary schools in Lithuania.

Founded in 1990, Vilnius Lyceum is a state coeducational day school enrolling students in Grades 9 to 12. The total number of students is 609, including 75 students in the International Baccalaureate Diploma Program.

Vilnius Lyceum is a modern, receptive to innovation educational establishment with a dynamic, creative, open-minded community ready to accept academic challenges and act with integrity and honesty. The school aims to provide stimulating environment for talented and highly-motivated students so that they can pursue quality education, develop and manifest their individual skills and abilities, maintain communication based on democratic values, and increase their civic, cultural, patriotic and international awareness. (https://licejus.lt)

In our school CT is covered mostly in Information Technology (IT) and International Baccalaureate Diploma Program (IB DP) Computer Science (CS) classes. It is well known that Computational Thinking has close affinity to Computer Science (Curzon & McOwan, 2017)) Grade 9 students have the compulsory IT course which involves C++ programming basics and the practical course of office tools. Grade 10 students have an optional technology course which elaborates different IT tools and have to make an IT product. Grades 11-12 have two options: national program or IB Diploma Program. Both have optional courses for computer programming. National program students have Further programming courses with C++ and IB students are taught Computer Science OOP module with Java basics. Both courses have an option to empower course to a higher level by adding extra 50% lessons.

From 2016 Vilnius Lyceum joined the network of STEAM schools of Lithuania and in 2018 we opened FabLab in our school. From 2018 we started organizing national contest TechHero, which is targeted to promote technical creativity and programming skills between 9-12'th graders (https://licejus.lt/Techhero,2019).

2. CT TEACHING CHALLENGES

Vilnius Lyceum has a long tradition to have strong students in programming. However, our school profile slightly changes and more and more students are changing their interest to humanitarian sciences. This is rather challenging for CT teaching as most teaching is done in elective IT or CS courses. It is interesting as similar problems like dropout in CS1 courses are discussed even at the university level (Kinnunen & Malmi, 2006).

I tried to collect feedback why students do not choose CS for higher grades in Vilnius Lyceum:

- Students declare that programming is complicated and difficult therefore they prefer easier options.
- Growing country economics and safety changes student interests – they are not frightened to select professions with lower incomes.
- The stereotype of a programmer in popular culture is not attractive for further life projection. A lot of students think that most of programmers do not have social life.
- More girls than boys are admitted to our school through entrance exams. Typically, there are fewer girls in noncompulsory CS classes. Still a lot of girls believe that CS is not required for their future.
- Students of current generation would like to get quick result without much effort. It seems that students do not have the patience to get the result.
- Internet and app stores are crowded with solutions for virtually any problem. It is not easy to find good idea for your own project.
- Quality of student made projects is so different from those available in Internet and app stores.
- Most of the programs created by students do not have any practical value.
- Students would like to learn other subjects and CS does not fit into their timetable.

3. BEST PRACTICES TO GAIN POPULARITY BACK

We have been observing these problems already for several years. Some of the mentioned factors we are trying to tackle by making adjustments in our classes.

3.1. A Stereotype of a Programmer

Some adjustments in our curriculum are targeting to change the stereotype of a programmer coming from popular culture. We are organizing visits to local IT companies, making connections to Lyceum Alumni organization (http://licejausalumni.lt/). Visits of former students, working in IT companies demonstrate that the actual job is not related to "geek style", and female are working there as well. Discussions and open question sessions with young specialists demonstrate possibilities and wide selection of job options in IT companies.

3.2. Barrier Breakers

One of the most complex problems in CT is to make barrier for programming basics lower. Nowadays the most popular idea is to use block style programming languages. There is a big set of them now like Scratch, MIT App Inventor, Snap, Google Blockly, BBC Micro:bit IDE, LEGO Mindstorm IDE with integrated LabView, MakeBlock IDE's for robots or Dobot IDE for robotic arm programming. However, our grade 9 - 12 students are a bit too old to "play" with block languages. They are still not strong in programming, but already know, what kind of tools is used by professionals.

On other hand, most of them still need help with understanding simple programming constructs like looping or program decomposition. My selection for this is set of tools based on ProcessingJS. Typically I start with https://www.khanacademy.org and https://editor.p5js.org/.

Why I chose them? There is a set of reasons.

- Student gain practice with the language, which is almost C++ syntax (this helps to explain, that we are gaining useful practical knowledge).
- It does not require any installation.
- Short code examples with quick nice graphical feedback.
- Huge amount of "spin-offs", which have been created by other users. The tool promotes creativity.
- Quick integration of mouse.
- Easy step to JavaScript web programming world.
- It is a very good tool to explain some constructs like loops, arrays in visual way.
- For most students it seems not as complex as C++.
- These lessons are loved by most girls.

Typical approach to graphical tasks is a very short and simple formulation like "Paint a grid and graphics of some function", "Paint regular polygon", "Paint some type of snowflake or star". Usually no object size, data input or anything else is provided. Students must show creativity in decisions they making. Pre-knowledge is restricted to line function and general understanding of loops and conditionals.

Graphical tasks at least have some feeling that the solution gives you some benefit. If a task like grid painting fails, student usually receives interesting pictures, which are difficult to paint without an algorithm. Real time program running sometimes messes up the screen, but after that the students aren't afraid to make errors – sometimes the results of the script with errors are even nicer than the expected outcome.

3.3. Project Based Learning

Grade 11-12 IB DP Computer Science students should create at least one project with full documentation. In the national IT syllabus, we do not have such a requirement. However, typically the students who chose a higher level also have projects which take much more time than one lesson to implement.

It is a complicated problem to select the topic and the end user for a project. Students typically overestimate their capabilities. They want to create programs with nice GUI and complicated functions as they are used to seeing in applications or apps in their devices. In the end the students sometimes become disappointed by the final result of their project.

In the national syllabus it is no requirement to have a personal project. So we suggest creating a bigger project in a group. This sometimes involves group projects with version control systems like GIT and public servers (Github.com or Bitbucket.org). Quality of the projects with several students involved lasting for and several weeks is much higher and the students are learning to work together, to split the project into smaller parts and the final result is less disappointing.

Rather nice example of such a project of my students is Battleship game for human or artificial intelligence agents. The idea is to run a contest between AI agents created by the students. This project gave a lot of possibilities to discuss deconstruction of system to backend and frontend and even smaller parts. Interesting discussions were taken when selecting OS for operating, data storing model, database structure, graphical game presentation for observers, etc. Students were happy to learn working in groups and to understand different roles of workers in IT companies.

Some projects involve popular hardware like Arduino, Micro:bit, RaspberryPI, NodeMCU and sensors, Wifi network. Nice example was temperature logging app for Android with data from the websites and a specific data collecting device made by students.

3.4. Competitive Learning

Other direction to interest of the students is related to different competitions. I can mention National Olympiads in Informatics, which consist of several rounds and take place at school at least 3 times a year, the international Bebras contest (bebras.org), our organized TechHero for Lithuanian gymnasium teams, Ventspils IT Challenge (Latvia) and other local competitions. To summon all these competition results I can proudly say that most successes in some sort of competition usually increase the interest of the students in to CS subject.

4. REFERENCES

Curzon, P., & McOwan, P. W. (2017). *The Power of Computational Thinking: Games, Magic and Puzzles to Help you Become a Computational Thinker.* London, UK: World Scientific Publishing Europe Ltd.

Kinnunen, P., & Malmi, L. (2006). Why Students Drop Out CS1 Course? *Proceedings of the Second International Workshop on Computing Education Research (ICER '06).* ACM, 97-108.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

"Unplugged" Programming - A Way to Learn the Basics of Programming

Berit SVENSSON
Elementary School Lessebo, Sweden
berit.svensson@lessebo.se

ABSTRACT

In this paper I describe our approach to introducing programming in our school with young pupils 6 to 8 years old. Our initial experimentation with Bee-Bots and using code.org in the Ipads indicated that these hands-on activities are not sufficient if we want children to get deeper knowledge in this field. Thus, it is crucial to challenge students in order to understand the importance of thinking in multiple stages in relation to programming. Collaborative unplugged programing activities have been performed by the students in order to gain an understanding of different concepts and methods used in programming. The initial results of these efforts indicate that these activities provide a solid ground for introducing young school children to coding.

KEYWORDS

"unplugged" programming, computational thinking, coding, collaboration

1. INTRODUCTION

When the National Agency for Education in the Autumn 2018 introduced the topics of *digitization* and *programming* as part of the curriculum for compulsory schools from first grade, there were many teachers who did not know how to handle this change. Several colleagues began to think about what the change meant and how teaching activities should be designed. The curriculum states that "The school should stimulate students' creativity, curiosity and self-confidence as well as their willingness to try and put ideas into action and solve problems. Students should be given the opportunity to take initiative and responsibility and develop their ability to work both independently and together with others" (Skolverket, 2017, p. 9).

Since I and many with me had no prior knowledge in programming, we had to start with what we already had access to. At the school where I work there were already some Bee-Bots (programmable robots) and Ipads. In year 1, we started by building different courses which we then programmed the Bee-Bots to run. We also did code.org exercises on the Ipads. Soon enough, I discovered that most students did not understand what they were doing. They pressed and pressed the arrow buttons until they reached some results but they did not understand why they should press certain buttons and that they needed to think several steps ahead for it to be correct. Thus, I decided to take a step back in order to rethink and plan my teaching based on the basic concepts that I thought were important for the children to bring with them to complete the assignments. We started then with *"unplugged"* programming exercises (Bell at al., 2009). Grover and Pea (2017) have defined a number of key concepts and methods that are important to address if we want to promote *Computational Thinking (CT)*. Some of these key concepts include the following ideas: a) Logic and Logical Thinking, b) Algorithms and Algorithmic Thinking, c) Patterns and Pattern Recognition, d) Abstraction and Generalization, e) Evaluation and Automation. But not only concepts are important but also several methods are crucial to support CT education. The same authors refer to *Problem Breakdown Items, Testing and Troubleshooting, Iterative Replacement and Collaboration & Creativity.* Many of these key concepts and methods have been used in our school activities.

2. "UNPLUGGED" PROGRAMMING

Based on what I described in the previous section our goals related to working with "unplugged" programming were:

- Students should gain an understanding of what programming means.
- Students should learn terms and concepts used in programming.
- Students should learn that by coding and coding each other, they should gain a good understanding that the more elaborated code they give, the better results they get.

As a teacher, I also wanted to find programming exercises that could be used in several different subjects. Moreover, the work with "unplugged" programming should lead to being able to further programming in scratch, code.org and different kinds of robots and providing an understanding of how we should go about and why we do it.

Some of the theoretical ideas guiding my work are inspired by Mannila (2017, p. 123) where she states that digital competence is not just about digital technology. It's not just about doing, but also about understanding. She also writes that programming is not just about arriving to an end product but also that children have to go through several phases such as analyzing the problem, idea or assignment and designing a solution. She further writes that "So-called unplugged programming, that is, programming without a computer, serves as an easily accessible and adaptable way to introduce programming" Manila (2017, p. 235). Thus, I started by making a sorting mat (see figure 1 below) that I encounter during a competence development day. On the sorting mat, children should sort different cards with numbers or pictures by advancing on the mat by following different lines. We started by sorting the numbers up to 10 and animals by the number of bones.

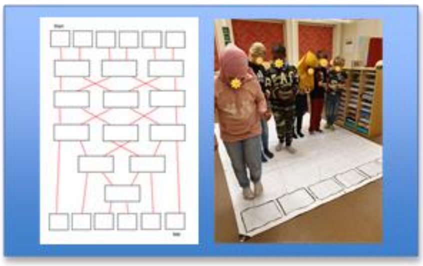


Figure 1. The Sorting Mat

These activities worked very well and the students immediately realized that they had to wait for each other at a new box to compare their cards before they could move on to get it right in the end. It was not easy to find ready and good tasks that matched my goals so much I had to find or change myself. I painted set up a programming box out in the school yard where students were allowed to program each other past obstacles and up to various taxes. Here, too, they soon discovered how important it was to give each other accurate instructions / codes so that they could understand each other.

After two years, I have now completed many exercises where we practice the different words and concepts of programming by individually solving several coding tasks or finding bugs. For some of the exercises, children can work individually and some of them require students coding each other in pairs or in smaller groups, for instance coding with recycling and paint boxes (see figure 2 below).

Figure 2. Coding with recycling and paint boxes.

3. RESULTS AND REFLECTION

A major challenge has been to come up with exercises that are not only a fun activity but also provide knowledge and lessons that children can take with them to upcoming programming exercises. Many exercises the students themselves have been able to develop further by coming up with more ways to complete or solve the tasks. The programming exercises have not only strengthened the students´ knowledge of terms and concepts used in programming, but they have also been very good as collaborative learning exercises. The students have unknowingly trained to give each other feedback by talking to each other about how they perceived the codes to work. In cases the code was not sufficiently clear, children were required to perform multiple iterations to improve it in order to reach the target. During the time *"unplugged" programming* have been used in our teaching, I have seen how students through physical work have become increasingly confident in their use of the words and concepts we have practiced. By training programming collaboratively and in different ways, they have understood the importance of having clear instructions to achieve the expected results. The training with *"unplugged" programming* has led them having a greater confidence and knowledge on how and which steps they should think about when programming a robot such as Edison. The latest is an important knowledge gain that I have not seen before if compared with students who haven't worked with *"unplugged" programming.*

Programming tools and methods develop all the time, so it is important that students get a solid foundation in how programming works. In addition to working with problem solving, seeing patterns, creating algorithms and programming each other, we have also used block programming for instance with Edison robots. By programming in different ways, children get very good foundations for the understanding of programming and how it can be used in different contexts. I have received very positive responses for the materials I developed from other teachers who have used them. I visited other schools, had workshops for teachers and I shared my experiences in a teacher pod. All these materials are available on the web, so teachers from our region can used them in their classes.

4. REFERENCES

Bell, T., Alexander, J., Freeman, I, & Grimley, M. (2009). Computer Science Unplugged: School Students Doing Real Computing without Computers. *New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20-29.

The Swedish National Agency for Education. (2017). *Curriculum for primary school, preschool class and leisure home.* Stockholm: The National Agency for Education, 9.

Mannila, L. (2017). *Teaching programming in school: Why, what and how?* Lund: Student Literature.

Grover, S & Pea, R. (2017). *Computational Thinking: A Competency Whose Time Has Come*. Retrieved December 9, 2019, from https://www.researchgate.net/publication/322104135_Computational_Thinking_A_Competency_Whose_Time_Has_Come

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

The Effectiveness of Interdisciplinary Computational Thinking Education for Middle School Students: A Case of Integrating IT in Technology Domain and Mathematics Domain

Chuen-Cheng CHEN
Taipei Municipal Long Men Junior High School, Taiwan
doniface@gmail.com

ABSTRACT

The purpose of this study is to explore how to integrate the knowledge and skills and computational thinking acquired by middle school students in various domains. In terms of mathematics, finding factors has been learned in elementary school. However, students can already use other ways to find out by using prime factorization. The concept allows the skills of finding factors to improve efficiency and accuracy. The subtlety is that the method of finding factors for middle school information subjects still rely on the elementary school. It should be confirmed and examined that the some domains learned by students should be systematically integrated, so the instructional experiment expected to transform the learned mathematical knowledge into actually algorithmic steps through the depiction of flow charts, and integrate mathematical knowledge and computational thinking. In order to avoid the need to be familiar with software environment or user interface learning, this interdisciplinary instruction will try to use unplugged computational thinking learning activity in this course, so that students can focus on those strategies of problem solving and show them into flow charts.

KEYWORDS

flow chart, factor, prime factorization, Interdisciplinary instruction, unplugged computational thinking learning

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

中學生跨域運算思維學習成效研究：以導入科技領域資訊科技與數學領域為例

陳春成
臺北市立龍門國民中學，台灣
doniface@gmail.com

摘要

本研究旨在探討如何整合中學生各領域之間所學得的知識技能與運算思維，數學科在小學階段已學到基本的因數找法，到了中學階段，學生能透過質因數分解的概念，讓尋找因數的技巧獲得效率的提升，微妙的是中學資訊科對因數的找法仍停留在小學的基礎上；研究者認為領域之間應有系統性的整合，因此設計透過流程圖的描繪，將數學已學得的知識轉化成實際的演算法步驟，整合為不插電運算思維學習活動，從觀念基礎上對數學的知識以及運算思維作整合。讓學生可以專注於問題解決的策略上，並將策略以流程圖的樣貌展現。

關鍵字

流程圖；因數；標準分解式；跨領域教學；不插電運算思維學習

1. 研究背景

數學科對因數的概念是從小學階段開始建立，而小學介紹因數是基於對因數概念的認知，因此明定被分解數的因數，在扣除 2、3、5 或其次方的部分後、只剩一因數（十二年國民基本教育課程綱要, 2019），對因數的定義也僅限於可以整除另一數的概念，然而國中階段對因數的要求除了意義上的理解之外，也需要熟練計算，並能運用到日常生活解決問題。但是一般在資訊課程中教授的程式設計或演算法，如果要找任一整數的因數，依舊是將該數除以小於該數的整數，若能整除則表示其為因數，直到結束的重複結構。

國中數學的內容可透過質因數分解的標準分解式，用於求因數及倍數，在數學科，學生也已經能使用這樣的概念解題，可在短時間內更精確地找到所有因數，但是資訊課沿用小學的基礎知識，是有其不足之處。

因此，研究者希望可以在中學生已經習得流程圖的基礎上，輔助數學科中已經進一步認識找因數的知識與方法，應用在資訊課程的解題（演算法）策略。

2. 文獻探討

流程圖最早可追溯到 1949 年，當時為了明確規劃電腦程式的運作順序所設計的一套法則與標準（Goldstine & von Neumann, 1947）。美國國家標準協會在 1960 年代開始制定流程圖及標準符號。在 1970 年代國際標準化組織採用其方案，並且逐漸成為電腦科學設計各種演算法的模式，現時通用的版本是 1985 年大致底定。

流程圖在早期程式語言設計不易提供給非設計者閱讀的結構下，可以讓其他群體的共同設計者更能理解設計者的思維與設計脈絡，惟程式語言近年的發展已經朝虛擬碼（pseudocode）的方向前進，不僅在學習與撰寫程式語言上有較低的門檻，也同時能兼顧演算法設計的展示，在程式語言的結構表示變得越來越容易閱讀下，流程圖的需求也變得式微，儘管如此，在跨足不同程式語言的對話上，設計者仍需透過理解流程圖，藉此共通性才不致於誤解整體運算邏輯（Bohl & Rynn, 2007），因此能夠設計流程圖結構仍是學習現代程式設計與演算法的基礎。

因數的定義根本必須由乘法架構討論，明確定義因數僅討論正整數的前提下，若可將某數寫成兩個整數相乘，則這兩個整數均為某數的因數（factor）。

將任何一個正整數寫出它所有質數因子的乘積表達式。這個過程稱為質因數分解（Prime factorization），而將其表達出來的式子稱為標準分解式。

由標準分解式的結構，可得任一正整數均可表示為：

$$N = x^{\alpha} y^{\beta} \ldots\ldots z^{\gamma}$$

3. 教學設計

本課程受教學生即為七年級國中生，資訊科技已完成認識流程圖的辨識，數學完成因數與倍數與標準分解式表示等概念及計算，所需知識技能均已完備。

研究者的主要焦點將著重在如何將數學科所學概念應用於資訊科技的演算法，目前在求學生對此概念上的演算理解，並且期望學生能有不同的探究，故將採不插電運算思維的課程模式，避免耗費時間在程式語言的編寫與除錯或研究如何使用流程圖軟體拖曳與放置等過多技術操作上，為了讓學生更快觀察出因數與標準分解式的關聯，另行準備平板電腦並由設計者從許多 App 中找出免付費且功能最為單純的安裝，供各組學生自由運用。

研究者認為，藉由數學科已學習到的知識引入到資訊科技科目上，並結合資訊科透過程式或計算機快速驗證的優勢，期望學生可以更加融會貫通所學知識的相關應用，而非科目屬性不一，無法跨域交互參考。

為能有效推展活動，研究者製作學習單，並由各組主動掌握活動進度，研究者則將依照學習活動的時間，適時協助未能跟進的組別。

結合學習目標，能了解標準分解式的結構，並能做出拆解，同時使用科技工具減縮操作與計算時間，是學習單前二個階段預期達成的目標。

為避免研究者個人於課堂實施過程有所疏漏，協請相關領域教師協同觀課，並於課堂中詳實進行觀課紀錄，以利事後對研究者課程與教學上的回饋與建議。

表一 課堂學習單

「想不到」的因數學習單

第 組 組員名單：______________________

階段 1:標準分解式隱藏的秘密

	20	180	2520
有哪些因數			
標準分解式			

- 請觀察各數的標準分解式, 是否都隱藏著他們的因數?
- 是否可以得出以下結論: 任何一個數的因數都存在於標準分解式內?
- 完成以下填空:

數	20	180		2520
任取一因數	5	45	90	630
完整標準分解式	$2^2 \times 5$	$2^2 \times 3^2 \times 5$	$2^2 \times 3^2 \times 5$	
任取因數的分解式	1 x 5	$1 \times 3^2 \times 5$		

階段 2:回歸本質找因數，前 3 題請使用計算機，後 4-10 題使用 App

4. 結果與討論

4.1. 研究者個人觀察與觀課教師群回饋建議

預期導入的科技工具有發揮功能，所有組別的學生都可以在略作介紹功能的情形下即能立即運用於解題上，大大縮減初期必須個別找出因數的冗長時間。

經由觀課確實體現許多設計過程並未發現的狀況，較為關鍵的觀察建議有：（1）男女學習氛圍明顯不同，在不同單元分組方式能否有不同的學習效果產生？（2）流程圖尚須從確定數值運算 再成為代數呈現，也就是具體到抽象（3）可以嘗試用卡片讓學生排列，以限制條件的方式來完成（4）學習單上可考慮有詳盡的說明，方便學生理解任務。都是研究者於設計與實施課程過程中未能預先設想的方向。

圖一 學生操作 app 順暢/研究者同仁協助觀課

4.2. 學生課堂表現結果分析

所有組別都能完成前四個階段任務，但第五階段的挑戰題僅有二組達成，且只有一組的流程圖邏輯合理。可知課程深度無法於設計的時間區間來完成，合理推估需要兩節課的時間，完成度應已足夠。

學生對數學概念可以與資訊課結合頗有興趣，因此整個課堂的推展過程頗為順利，幾乎所有的組別都能專注並投入解題的過程。

圖二 多數學生均能專注於解題上

原先設計採用不插電的形式也證明可以讓學生心無旁騖專注於討論過程，而不易受到電腦操作分心或於操作過程中產生困擾等外在因素影響。

4.3. 學生課後回饋分析

研究者於課後請學生以無記名方式填寫回饋單，期能得知學生針對此項跨域課程的感想

實施課程共有 32 位學生，實際回收有效問卷為 31 張，無效問卷 1 張，問卷前六項描述表示同意（或非常同意）者比例均超越 80%，並以第 5 項：老師安排的上課方式讓我更努力學習、專心聽講達 86%，以及第 6 項：我都能從輕鬆學習中，獲得專業知識達 86%最高。

第 7 項描述是唯一表示同意（或非常同意）比例未達 80%者，其中有幾位學生表示未同意原因為：「資訊課老師並沒有特地為我加強流程圖的部分」。

5. 結論與建議

本課程以不插電的活動型態，用於實施數學科與資訊科跨領域模式，實現國中知識概念與技能結合的課程設計，無論研究者、觀課者或受教學生對課程本身的反饋都能保持非常正面的肯定與接受度，也間接證明並非教學領域之間必須各自獨立，或資訊課程的內容就必須於資訊教室內實施，以及流程圖的練習或繪製必須透過軟體實現。

整體而言，學生課後對此課程的回饋滿意度超越 80%，並且極大多數均對研究者的教學模式均能支持與肯定，一直以來，資訊教師普遍都認為資訊課程必然要在有資訊設備的前題下方能實施，應是對傳統資訊課程的一種迷思，若能不透過電腦設備，讓學生回歸基本的問題拆解、模式識別、抽象化、演算法推演，同樣能構成運算思維，且更有機會做好完整的問題解決能力，也將會是研究者未來願意繼續不插電課程的最大動力。

課程實施過程唯一美中不足之處在於研究者將此課程的時間限縮得太短，若能將課堂延伸為兩節，則學生將能有更充裕的時間將自己思考的演算法推展完成，並有機會與其他組同學分享，進行更多完整的討論與互動。

6. 參考文獻

教育部。（2019）。十二年國民基本教育課程綱要。2019 年 8 月 10 日，取自 https://www.naer.edu.tw/files/15-1000-14113,c1594-1.php

Bohl, M., & Rynn, M. (2007). *Tools for structured and object-oriented design*. Pearson India.

Goldstine, H. H., & Von Neumann, J. (1947). Planning and coding of problems for an electronic computing instrument Part II. *Institute for Advanced Study (1)*, 1-77. Princeton, New Jersey.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

4D Immersive Experience Space: Spatial Impacts on Students' Coding Skill

Chi Cheung CHING[1*], Siu Lung HUI[2], Ka Kui TSANG[3], Kei Yu CHAN[4]
[1,2,3,4]Fukien Secondary School Affiliated School, Hong Kong
ccching@fssas.edu.hk, slhui@fssas.edu.hk, laurencetsang@fssas.edu.hk, kychan@fssas.edu.hk

ABSTRACT

Coding education is increasingly important in STEM education as it equips students with essential 21st century skills, including problem-solving, logical thinking, computational thinking and design skills. Students are required to master the technological mechanism and operation, as well as how technology can be innovated to create opportunities. The mastery of these skills enables students to think logically and critically to break down problems into smaller pieces, identify errors and come up with solutions. Most importantly, these skills are transferable that they can apply in real life circumstances.

On grounds of the significance of coding skills, the implementation of coding education and its improvement is inevitable. An idea of using 4D immersive experience space is proposed in this paper, given that its spatial benefits combined from a traditional classroom and computer room could potentially enhance students' learning efficacy in coding education. A control experiment is applied to measure students' performances and interest in both conventional computer room and 4D immersive experience space. The results indicated the change of spatial setting improves students' concentration while learning and allows them to locate bugs in an efficient manner by visualizing the program effects.

KEYWORDS

4D immersive experience space, spatial impacts, coding education

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

4D 沉浸體驗空間：多維空間元素對運算思維教學的影響

程志祥 [1*]，許少龍 [2*]，曾嘉駒 [3*]，陳麒宇 [4*]
[1,2,3,4]福建中學附屬學校，香港
ccching@fssas.edu.hk，slhui@fssas.edu.hk，laurencetsang@fssas.edu.hk，kychan@fssas.edu.hk

摘要

隨科技發展迅速，運算思維教育對培育廿一世紀人才的重要性亦大大提升。運算思維不僅涵蓋計算機科學的知識涵養，而且延伸運算思維技能所培養的解難能力、計算思維、邏輯思維及創新思維。這些技能轉移在不同領域上的應用更是運算思維教育的主要目的。為提升運算思維教育的教學效能，本文聚焦在進行運算思維教育時提供的空間元素。考慮本港小學運算思維教育以計算機主導的教學環境設定，及參考以實體工具進行教學的環境，本文旨在探討空間元素對運算思維教學的影響，透過量度學生的學習表現和興趣，4D 沉浸體驗空間所提供的空間元素有助學生確定程式錯處，亦能提升學生的專注力。

關鍵字

4D 沉浸體驗空間；空間性影響；編程教育

1. 前言

在資訊科技發達、互聯網盛行的年代，無論是對社會和個人發展都有重大的影響和改變。運算思維教育對培育和裝備學生成為廿一世紀的人才是非常重要的。課程發展議會（2017）定義「運算思維」為將一個原始方案或計算問題，經過一個過程後，引出一個可執行的程式（頁 4）。 學生需具備有關的基本知識素養，包括建構概念及操作軟件、硬件的技能。通過掌握基礎編程的技巧，學生得以製作計算成品，當中包含設計、編碼、測試、除錯等步驟。Wu et al. （2019） 指出在編程等過程中，學生需要運用解難能力、邏輯思維、計算思維和創新思維去將問題拆細成可管理的小組件，辨識錯處，然後制定解決方案。這些技能是具轉移性質，能夠應用在不同層面的情境之中，這亦是運算思維教育的價值所在。

運算思維教育是提升未來人才競爭力的重點教育，為了讓學生有效地學習，本文參考了不同運算思維的教學模式，利用嶄新的 4D 沉浸體驗空間，研究空間元素對其教學上的影響。為甚麼我們選取除錯活動為是次研究重點，經過所有電腦科老師商討教學難點時，均認為學生的除錯能力不是太高，可能受制於他們使用的電腦屏幕的大小及方向（大多是直立式），我們嘗試以 4D 沉浸體驗空間改變以上的環境條件，再看看能否有效改善學生於除錯活動的表現。然後，通過對照實驗檢測學生課堂表現，指出空間元素對運算思維教學的影響。

2. 運算思維教學環境設定

運算思維教育的教學模式可以是多元化的，由單純利用積木、模塊等的實體組件，到複雜的計算機方程也可以是運算思維教學模式。

傳統上，本港的運算思維教育大多定位在資訊科技教育科的課程架構內，所以教學環境通常設定在電腦室。課程發展議會（2014）指出教育局早在 1998 年推行資訊科技教育，並推動學校實踐「範式轉移」，即是由課本主導和以教師為中心的教學模式，轉變為互向和以學生為中心的學習模式（頁 1）。經過這策略的推動，電腦室是上課的常規教室，注重建構知識之餘，亦重視為學生提供實踐的機會。

除了電腦編程的教學，在外國早至五至六歲的學童已經要學會基本算法，利用簡單工具創造程式和為程式除錯。作為幼稚園教師的 Nelson (2018) 認為學習編碼是一種語言，細至每個字母或任何工具都可以代表不同的數值或公式，讓他們了解科技運作的基礎原理和操作系統。這類運算思維教育利用實體工具和環境進行教學，切合尚未掌握電腦系統操作的學生之需要。同時，這些實體工具能夠將編程抽象的概念得以形象化和具體化，有助學生掌握基本運算思維的概念。

運算思維教育雖能培育編碼技能，但往往容易產生令學生專注力不足的問題，例如學生會在使用電腦時瀏覽其他網站。另外，複雜的電腦編碼難以形象化的問題會令學生無法將程式錯處定位。而實體編程教育則未能夠編寫電子編碼，僅能提供基礎的編程培訓。考慮到兩者之利弊，本文提倡利用 4D 沉浸式教室互補以上兩種教學模式的不足。

3. 4D 沉浸體驗空間

4D 沉浸體驗空間的設計是啟發自「沉浸式教學模式」—一種強調以內容導向打造語言意識文化（Content-based Instruction），即在教學時，將內容和語言同步進行教學（Snow, 1988）。這種源於加拿大的法語學習課程希望透過沉浸在非母語的學習環境之中，潛移默化地吸收知識。4D 沉浸體驗空間應用了這種教學模式的環境沉浸元素，將 4D 劇院的配套應用在教學習而上。所謂 4D 是指視覺、聽覺、觸覺及嗅覺上的觀感體驗，通常配合 3D 的視象概念，加入風、燈光、味道、煙霧、震動等的物理效果技術。此技術可以將課本上的知識形象化，模擬任何場景，一反傳統課堂的設定。透過 4D 的感觀刺激，學生能夠按課程設計，體驗及更了解所教授的主題。

理論上，4D 沉浸體驗空間能夠綜合傳統教室和電腦室在運算思維教育的好處，既能運用 4D 科技將複雜的編

碼具體地呈現，又能讓學生專注地利用電腦學習編程知識和技巧。

4. 研究方法

為探究 4D 沉浸體驗空間所提供的空間元素對編程教育的影響，本課堂研究利用對照實驗比較學生在傳統電腦室和 4D 沉浸體驗空間的學習表現和興趣。運算思維教學其中一個目標在於培育學生的解難能力，當中透過對程式進行「除錯」，能有效提升學生的閱讀、理解、測試及解難能力。本課堂的設計乃根據「賽馬會算思維教育」課程「級別一」「L1U8.3 以 Scratch 發迷宮遊戲」 所設計。本課節重點在於培育學生的「除錯能力」（Debugging）。學生於前一教節學會如何使用 Key pressed, If …Then…, 使用 Variable 計分及轉換場景等功能。學生於本課節中需對應一個有問題的程式（當中包含四個錯處，分別是有關 1. Key pressed, 2. If …Then…, 3. 使用 Variable 計分及 4. 轉換場景），找出錯誤的地方，再進行修正，正確地修正每個錯處可得 25 分，全部完成共得 100 分。他們完成隨錯活動的時間會被記錄。另外，學生須講出他們隨錯時的思考步驟。是次研究於本校其中一班四年級學生進行，他們隨機編為兩組，其中一組學生於傳統電腦室進行課堂，而另一組學生於 4D 沉浸體驗空間內進行學習。於 4D 沉浸體驗空間內學生使用整個教室牆身及地面作為不同空間的學習工具（如右圖所示）。

是次研究的教材以自學的形式編排，教師於課堂中只是輔助角色。於課堂前後均訪問學生對運算思維學習的興趣。

5. 數據分析

是次課堂研究，會將學生隨機分為兩組，再收集了以下數據：

1. 學生完成「除錯」活動所需時間
2. 學生於「除錯」活動所得分數
3. 學生能否說出「除錯」的程序
4. 學生於「除錯」活動後，對編程學習的興趣

整體而言，學生於 4D 沉浸體驗空間內進行活動的成效較於傳統電腦的成績較為理想。他們於 4D 沉浸體驗空間內所得的平均成績較傳統電腦的高約 11 分（100 分為最高分數），當中包括四個改錯點。而他們的整體平均所需時間則較為略短。再者他們於完成活動後相對傳統電腦室有較多學生能清楚說明他們的「除錯步驟」（8 人：4 人）。而對編程活動的興趣而言，學生於 4D 沉浸體驗空間內進行活動有顯著提升。

6. 結語

我校已於電腦課程上運用 CoolThink 所開發的運算思維教育課程進入第三個年頭。基本上所有電腦科教師已接受了由教育大學提供的教師培訓，他們均掌握各項教導運算思維的技巧，並於這兩年取得不錯的成績。於過去兩年，學生於傳統電腦室均能正常發揮。

但學校期望進一步提升學生在解難活動上的能力，是次課堂研究設計主要在於:

1. 自主學習能力
2. 「除錯」—解難能力

當中為提升學生的自主學習能力，我們以「興趣」主導設計，以卡通人物貫穿整個活動，而活動以「偵探」形式設計。教師於課堂只是輔助角色。

而針對提升學生解難活動的能力方面，課堂設計時集中提升學生的閱讀程式能力、除錯能力及測試能力。當中更期望學生能歸納出一個屬於他們自己的「除錯循環」。

綜合所有數據而言，於 4D 沉浸體驗空間進行運算思維課堂，因著空間元素，確能有效提升學生的學習興趣。而其他各方面也有所提升。是次課研仍基於學校校本課程，如何延伸至其他學校運用都有待考究。

期望日後能多運用 4D 沉浸體驗空間作運算思維教學研究，使之更了解如何於這範疇能更好培育我們的下一代。

7. 參考文獻

課程發展議會（2014）。**基礎教育課程指引—聚焦・深化・持續**。擷取自網頁 https://www.edb.gov.hk/attachment/tc/curriculum-development/doc-reports/guide-basic-edu-curriculum/becg_2014_Full.pdf

課程發展議會（2017）。**計算思維—編程教育**。擷取自網頁 https://www.edb.gov.hk/attachment/tc/curriculum-development/renewal/CT/supplement_CT_chi_draft.pdf

Snow, M. A., & Brinton, D. M. (1988). Content-based Language Instruction: Investigating the Effectiveness of the Adjunct Model. *TESOL quarterly*, *22*(4), 553-574.

Wu, L., Looi, C. K., Multisilta, J., How, M. L., Choi, H., Hsu, T. C., & Tuomi, P. (2019). Teacher's Perceptions and Readiness to Teach Coding Skills: A Comparative Study Between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific Education Researcher*, 1-14.

Nelson, S. (2018). 5 Reasons why Coding is Important for Young Minds - Learning Resources Blog. Retrieved December 5 2019, from http://blog.learningresources.com/5reasonskidscoding

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Experience and Reflections in Teaching Computational Thinking at School

Chi-kwan CHONG[1*], Chi-kei LO[2]
[1,2]Sheng Kung Hui Chi Fu Chi Nam Primary School, Hong Kong
Chong31415@skhcfcn.edu.hk, bonnierejoy@skhcfcn.edu.hk

ABSTRACT

This paper mainly discusses the school's progress and experience in implementing programming education with CoolThink@JC's pedagogy in the past three years. The purpose of teaching programming in the school, the difficulties encountered in the implementation, the students' experience in learning programming and their achievements are mentioned in the paper. Personal thoughts and opinions of teachers and students are shared in this paper so that the experiences of the school can set forth the future reference for schools and educators in implementing coding education.

KEYWORDS

computational thinking, coding, CoolThink@JC

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

在校推行運算思維教育的經驗及感受

莊志鈞 [1*]，盧梓琪 [2]
[1, 2] 聖公會置富始南小學，香港
Chong31415@skhcfcn.edu.hk，bonnierejoy@skhcfcn.edu.hk

摘要

本文主要論述本校在三年間推行 CoolThink@JC 編程教育的過程及經驗，文中將提及本校在推行編程教育的目的、推行過程遇到的得著及困難、學生在學習編程的感受、以及他們在編程創作上的成就。

本文章是由教師及學生提供的個人感受及意見，藉此希望能分享親身經歷，讓各教育者可以參考以更了解在學校推行編程的要點。

關鍵字

運算思維；編程；CoolThink@JC

1. 前言

本校藉著有幸成為 CoolThink@JC 的先導學校，在推行運算思維上，CoolThink@JC 所給予的支援很全面，包括從一開始的 78 小時教師培訓，以及不同的交流團，一年 TA 人員到校支援，令本校更順利地推行運算思維，故藉此分享來回饋當中的謝意，讓更多學生在編程教育中更認識運算思維，成為未來數碼世界的前鋒。

2. 推行編程教育的目的

在香港 STEM 教育興起的同時，其他國家已把編寫電腦程式成為課程範圍，本校教師便開始討論將資訊科技科加入編程，以幫助學生提升邏輯思維，又能讓學生創作屬於自己的作品，故此在參與 CoolThink@JC 前，本校已經有教授 Scratch，進行編程相關之課外活動如 Lego coding 班等。獲得 CoolThink@JC 的支援後，本校除了有更多資源教授編程外，更能把運算思維的重點教授學生。如台灣學者綜合各種不同的定義，將運算思維定義為「能有效應用運算方法與工具解決問題之思維能力」（林育慈和吳正己，2016）。我校盼望藉推行編程教育，培養學生運算思維，以強化分析與處理生活問題的能力。

在推行編程教育後，學生都很期待資訊科技之課堂，當他們在編程時出錯誤，一開始都會舉手問老師，但當他們開始熟悉後，教師都會建議他們先自己思考，再與同學一同討論，再解決不了才問。這種思考模式在課程緊密的課堂很難實踐，而學生亦有相當強的學習動機，令他們很享受編程過程。運算思維的思考模式很難作評估，但在近年發現學生在課堂時，開始會以「如果…那麼…」去討論問題，亦在談吐間提到「用平板測試，看看那裡有錯誤…」，甚至用流程圖去畫他們的設計，都令本人覺得學生在學習編程上是有作得著的。

如果提到編程教育的目的，必須考慮到學生在課堂有否得著。看著他們有學習動機，願意多嘗試自己去解決問題，與同學共同面對問題等，這便是本人希望達到的目的。

3. 推行過程的得著及困難

成為 CoolThink@JC 先導學校後，本校共有四位教師參與 78 小時培訓班，令我們教師對編程認識更深入。以本人為例，本身並不是教授電腦科，經過培訓後，對編程教育便開始有認識。而與同事一同合作亦是推行編程教育非常重要的一環，由於在編程教學時，會出現很多技術上的問題，本校教師們便在共同備課時一同討論，或分享自己作設計的程式及教學方法，讓教學更流暢。

在推行編程教育的過程中，本校也遇到不少困難，如在教學進度上，由於大部分的課堂時間會被安排為編程教學，在本校一星期只有一節資訊科技課的情況下，課程上需要大幅度修改，以免出現顧此失彼的情況。

已在進行教學期間，最大困難是硬件技術問題，例如學生一開電腦時會出現 window 更新，或者進入帳號時打錯多次密碼導致不能進入，用 App Inventor 2 時要使用同一個 Wifi 等，雖是一些瑣碎的事，但亦是令教學上帶來很多不便。而 CoolThink@JC 在本校教編程的第一年，安排了兩位教學助理幫忙，令本校編程教學更流暢。故此剛開始推行編程教育時，有教學助理幫忙可以幫助教師更集中教授編程。

故在推行編程前，可考慮教授編程的教師團隊有否足夠時間備課及設計教學內容是非常關鍵的，教師需要有充裕的時間先揣摩及設計不同的程式，使他們更了解 Scratch 及 App Inventor 2 等軟件的運作。其次需要設計合適學校課程內容的進度，如有需要可考慮將進階的課程轉於課外活動教授。最後建議剛開始教授時，可編排更充裕時間讓學生學習，除了降低因趕課程而教學質素下降，更可以讓學生有足夠時間去摸索。

4. 學生學習編程的感受

本校學生在學習編程的過程中，都顯出有濃厚的學習動機，例如在本人教授四年級 Scratch 後，有不少同學在家中亦會設計程式，在學校分享自己的作品，這都令教師感到很大鼓舞。以下是兩位學生的分享：

「我在學校學習編程，除了學懂如何寫程式外，還明白到隨著科技日益發展，電腦程式也愈來愈重要，編寫程式可以令自己心中的設計在電腦實現，還可以設計一些天馬行空的應用程式。」（六年級學生 - 高靖茵）

「在學習編程中，我學到團隊的重要性，如果沒有隊名的互助下，很難設計出具特色的作品；另外，編程背後原來可以幫助人，令我更有動力去設計更多程式。」（六年級學生 – 劉冠言）

5. 學生在編程創作的成就

本校在上學年才開始參加編程比賽，老師們發現學生開始有編程概念及有能力創作自己的作品時，便開始讓他們參加比賽，與學校的資優教育部門合作，參與九龍樂善堂舉辦的「全港創新科技設計大賽」。為使學生的作品更富意義，老師先安排學生與本校社工進行面談，認識學校學生的需要，亦透過此機會讓學生認識進行面談的程序及技巧。隨後，教師再帶學生到香港基督教服務處兒童發展中心與治療師進行面談，讓學生更了解所設計產品之目標對象，了解產品用家的特性及需要，使設計產品的推動力大大提升。參賽過程中有幸獲得評判認可，學生亦從中領悟到設計產品的最終目的是惠及他人。

經歷第一次之參賽及獲獎後，本校繼而參加了 CoolThink@JC 舉辦的比賽，有幸獲評判賞識獲得亞軍。比賽經驗讓我們深深體會到學生最大的得著是經驗當中的過程。學生在設計產品前，是需要藉著訪問了解身邊人的真正需要，以達到以人為本的原則，故此他們在訪問後轉化為問題，在他們十多個問題及解決方案之中，由老師伴同下選出來。當中讓他們初步學習到設計思維（Design Thinking），雖未能完整地讓學生進行有系統的過程，但由於對設計對象有更深認識，令他們有更多的動力去做。

兩次比賽令學生成長很多，包括他們會想以設計作品去幫助他人，亦了解訪問要注意的細節，以及上台演講的技巧，這些都是他們在比賽中所最大獲益的地方。

6. 推動學生自主學習

本校認為要在學校發展運算思維，是需要讓學生在家亦有自學的方法，以及分享自己作品的機會。故此我們會分享其他自學資源網站給學生，如當他們設計程式有困難時，引導他們如何以關鍵字在瀏覽器找教學，例如有學生在家中設計「加法遊戲」，當時他不懂如何讓程式計出答案，本人除了會以口頭教授他如何設計外，同時介紹自學資源上一個類似的教學網站給他，讓他多由自己去找尋答案，解決自己的問題。

本校亦會在不同的機會下，讓學生分享他們的作品，如過去參辦的科學日中，讓學生去展示他們的作品給家長及同學，例如學生設計了 Micro:bit 鋼琴、水果音樂等，學生在分享之中，更因自己的作品得到欣賞而獲得很大的滿足感，他們之後便有很大動機繼續去學習編程及設計程式。

7. 總結

雖然未能輕易將運算思維以評估去量化，但運算思維能力卻是我們學生必需具備的能力。這種思維模式對學生很有幫助，以裝備他們在未來的數碼世界中，更快應對當中的挑戰，解決生活上的問題，並能為未來社會作出貢獻。

8. 參考文獻

林育慈和吳正己（2016）。運算思維與中小學資訊科技課程。**國家教育研究院教育脈動電子期刊，6**，5-20。

圖 1 學生在 CoolThink@JC 比賽上被訪問，讓他們有機會分享自己的成品

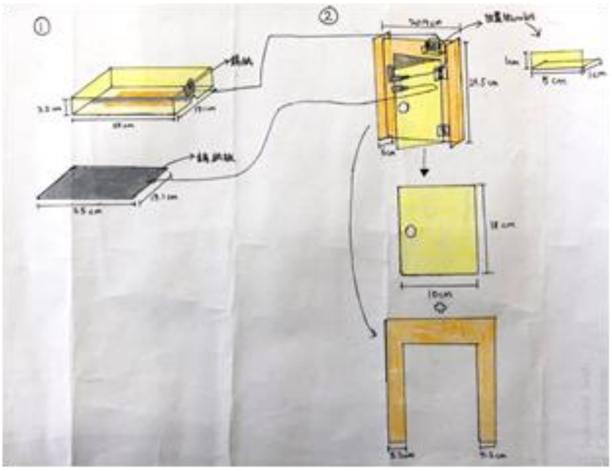


圖 2 由學生畫的作品設計圖

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Inquiry-based Learning by Role-Playing in Computational Thinking

Chien-Wen CHUANG[1*], Chia-Liang CHEN[2*]
[1,2] Taipei Municipal Nan-Gang Elementary School, Taiwan
00320@nkps.tp.edu.tw, 00122@nkps.tp.edu.tw

ABSTRACT

Computational thinking capability is an important skill for the 21st century. In order to enhance children's computational thinking ability for the information age, this research proposes an inquiry-based role-play model to provide comprehensive concepts as the instructional scaffolds to the students before coding. Students learned logical thinking and coding concepts through observation, exploration, and discussion in the physical learning gaming scenarios which were later applied in the Scratch block coding activities. From the empirical comparative study with focus group interviews, we found that through this model, students not only have a clearer understanding of the overall programming concept, but also feel more relaxed and comfortable in learning computational thinking.

KEYWORDS

computational thinking, role-playing method, inquiry-based learning

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

探究式角色扮演之運算思維

莊茜雯 [1*]，陳家亮 [2*]
[1,2] 臺北市南港區南港國民小學，臺灣
00320@nkps.tp.edu.tw，00122@nkps.tp.edu.tw

摘要

現今社會中，資訊能力是一項很重要的技能，為提升現今學童邏輯思考能力，本研究提出「探究式角色扮演之運算思維的教學模式」，讓學習者透過運算思維的模式，經由角色扮演的方式，藉由觀察與探索來理解程式設計裡的邏輯概念。最後經由焦點團體訪談可得知，透過本研究所提出的教學模式，發現學習者不但對整體程式設計的概念更加清楚明瞭，也對於學習程式設計課程更感到輕鬆與自在。

關鍵字

運算思維；角色扮演法；探究式學習

1. 前言

現今社會中，資訊能力是一項很重要的技能，大多數學生，通常只會照本宣科的按照課本上的內容照做，缺少思考的能力。而「運算思維」是一種用電腦的邏輯來解決問題的思維。它是運用電腦拆解與解決問題的思維方式，來幫助人們理解問題、規劃解決方式，以及進行決策（Wing, 2008）。

「角色扮演」最早是作為團體輔導及心理諮商的重要技術，後來被應用在教學上，用以探討學習者在學習時的情感、態度、人際關係與問題解決策略。因此，本研究提出「探究式角色扮演之運算思維」的教學模式，讓學習者透過運算思維的模式，經由角色扮演的方式，藉由觀察與探索來理解程式設計裡的邏輯概念。

2. 文獻探討

2.1. 角色扮演法（Role-Playing Method）

角色扮演（Role Playing）源自於 J. D. Moreno 於 1920 年左右所創立心理劇（Psychodrama）。後來 Shaftel 和 Shaftel（1967）則將角色扮演運用在教學上。其主要的目的為提供個人學習角色扮演的機會，使之能設身處地去扮演一個原本不屬於自己的角色，得以嘗試和體驗不同的行為模式，增加角色的學習，擴展生活中的知覺，而理解不同角色的想法與感受。

2.2. 探究式學習（Inquiry-based Learning）

探究式學習主要是希望能培養學生獨立思考以及自主解決的能力，由學習者的好奇心開始，加上本身已具備的知識，透過探索、推敲、反思以及驗證的互動過程，來培養學生的溝通能力。

2.3. 運算思維（Computational Thinking）

Jeannette M. Wing（2006）開宗明義地表示運算思維代表一個普遍適用的態度和技能，不僅僅是計算機科學家，任何人都將渴望學習和使用。而 Google 也提出了四個 CT 的核心能力（如圖 1 所示），並表示一個具有 CT 核心能力的人，可以運用現有的知識或工具，找出解決艱難問題的方法。

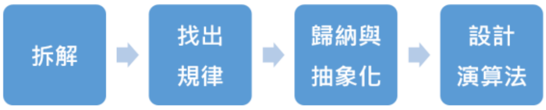


圖 1 運算思維的四大核心能力

3. 研究方法

3.1. 研究對象

本研究所實施的研究對象為臺灣臺北市某國小五年級學生共四班，其中兩個班為實驗組，共 48 人；兩個班為對照組，共 51 人。

3.2. 研究工具

本研究搭配該所小學所使用之教科書、Scratch 3.0 線上網站，以及 Google 所提供之 Google Classroom 進行課程設計。並於課程單元結束後，利用研究者所設計之訪談大綱，對學生進行焦點團體訪談。

3.1.1. 教科書

課堂中所使用的課本為宏全資訊所出版的「Scratch3 小創客寫程式」，依照課本的教學順序，進行程式遊戲設計教學。

3.1.2. 訪談大綱

為了解本研究所提出之「探究式角色扮演之運算思維模式」設計的成果是否能夠幫助學習者邏輯思考能力的提昇，因此，研究者列出訪談大綱（共 3 題，如下所示），於課程單元結束後，對學生進行焦點團體訪談。

- 本單元透過「探究式角色扮演之運算思維」的教學模式，跟之前學習程式的方式有什麼不同？
- 你覺得「探究式角色扮演之運算思維」的教學模式與之前的學習程式的優缺點分別是什麼？
- 如果讓你選擇，你會希望之後的程式設計課程用「探究式角色扮演之運算思維」的教學模式來學習，還是之前學習程式的方式來學習？為什麼？

3.3. 教學模式

本研究所提出的「探究式角色扮演之運算思維模式」（如圖 2 中的實驗組所示），於課堂一開始時，讓學生先觀察教師所要求最後的任務遊戲。

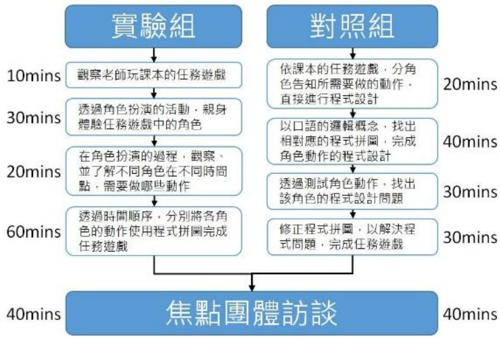


圖 2 實驗流程圖

接著，藉由角色扮演的方式，請學生親自來體驗當遊戲中的主角，透過教師所給予他們的一些指令，請學生演出教師所要求的指令（如圖 3 所示），並同時觀察不同指令所要求的差異性，理解不同程式積木的概念。接著，讓學生進入 Scratch 線上版，進行積木程式撰寫（如圖 4 所示）。最後，透過 Google Classroom 讓學生繳交完成的程式積木，全班一起討論同儕之間利用不同程式積木所建置出的任務遊戲，其差異與優缺點。

圖 3 角色扮演活動-Scratch 廣播功能概念

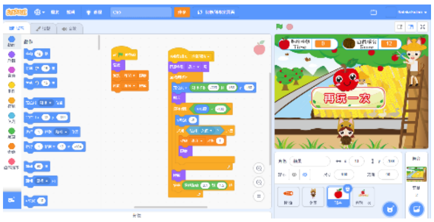


圖 4 完成任務遊戲-Scratch3.0 線上程式設計畫面

4. 研究結果

本研究透過「探究式角色扮演之運算思維」的教學模式，並藉由焦點團體訪談的方式，對學生進行質性訪談，並將其結果整理如下：

- 本單元透過「探究式角色扮演之運算思維」的教學模式，跟之前學習程式的方式有什麼不同？

團體 1：我覺得這樣的教學方式很有趣，能以角色扮演的方式，了解不同程式積木的功能。

團體 2：以前上課的時候，只要看課本把一樣的積木拼起來就好，這次上課玩的比較多。

- 你覺得「探究式角色扮演之運算思維」的教學模式與之前的學習程式的優缺點分別是什麼？

團體 1：這次的上課方式，可以讓我先體驗後，再慢慢想整個遊戲過程，但是我不太喜歡表演，如果可以直接讓我玩遊戲會更好！

團體 2：以前上課，不同角色的程式分開寫，有時候會不太知道跟其他角色的關係，這次上課後，覺得角色扮演這樣的方式，可以讓我比較清楚整個遊戲設計的流程。

- 如果讓你選擇，你會希望之後的程式設計課程用「探究式角色扮演之運算思維」的教學模式來學習，還是之前學習程式的方式來學習？為什麼？

團體 1：我希望能夠用角色扮演的方式來學程式設計，因為這樣感覺比較有趣，也比較能夠了解整個遊戲設計的概念。

團體 2：都可以！以前的上課方式，比較能了解單一角色的程式是怎麼寫的，這次的上課方式，則是比較能夠了解整個遊戲與角色彼此的互動關係，而且體驗過後的印象比較深刻。

授課教師亦透過本研究所提出之模式，表示：過去在教學上，總是一步一步帶學生了解不同角色該有的動作再逐一分別講解，而此次教學時，先讓學生了解整個遊戲的過程，並親身體驗過後，發現學生對於程式設計的迷失概念大幅降低，減少許多在授課時解釋迷失概念的負擔。

5. 結論

透過本研究所提出之「探究式角色扮演之運算思維模式」，讓學生對於整體程式概念的理解更加清楚明瞭，因此對於學習程式設計課程感到輕鬆與自在。未來，期望這樣的教學模式，能夠在更多的單元與班級進行，讓整個研究能夠加上量化的資料，進行更進一步的分析與探討。

6. 參考文獻

Costain, G., & McKenna, B. (2011). Experiencing the Elicitation of User Requirements and Recording them in Use Case Diagrams through Role-Play. *Journal of Information Systems Education, 22*(4), 367-380.

Pedaste, M., Mäeots, M., Siiman, L. A., Jong, T., Riesen, S. A. N., Kamp, E. T., Manoli, C. C., Zacharia, Z. C., & Tsourlidaki, E. (2015). Phases of Inquiry-based Learning: Definitions and the Inquiry Cycle. *Educational Reasearch Review, 14*, 47-61.

Shaftel, F., & Shaftel, G. (1967). *Role-playing for Social Values*. Englewood Cliffs, NJ: Prentice-Hall.

Shuchi. G., & Roy, P. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher, 42*(1), 38-43.

Wing, J. M. (2006). Computational Thinking. *Communication of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational Thinking and Thinking about Computing. *Philosophical Transactions of the Royal Society A, 366*(1881), 3717-3725.

Yablonsky, L. (1981). *Psychodrama Resolving Emotional Problems through Role-playing*. New York: Gardner.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Using Computational Thinking Skills to Learning on Musicals Animation Design

Chia-yen FENG[1*], Yu WANT[2] , Ying-rong CHEN[3]
[1,2] Affiliated Experimental Elementary School of University of Taipei, Taiwan
[3] Taipei Heping Experimental Elementary School, Taiwan
yeni0412@gmail.com, sher@esut.tp.edu.tw, katy630515@gmail.com

ABSTRACT
The purpose of this study was to applies computational thinking to the teaching musical animation, and explorer the effects of gender differences on learners' scratch animation. The object consisted of 28 sixth-grade students in one class. A quasi-experimental design was adopted and conducted a six-hour teaching experiment for six weeks. The results show that: the effects of gender on programming learning achievements are significantly different; learners using Scratch ultimately had more positive impact on learners' attitudes toward learning satisfaction.

KEYWORDS
computational thinking, gender difference, graphical music, musicals animation design

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

國小學生應用運算思維歷程設計音樂劇動畫之研究

豐佳燕[1*]，王瑀[2]，陳映蓉[3]
[1,2]臺北市立大學附設實驗國民小學，台灣
[3]臺北市和平實驗小學，台灣
yeni0412@gmail.com, sher@esut.tp.edu.tw, katy630515@gmail.com

摘要

本研究將運算思維歷程應用於「音樂劇動畫創作」教學中，探討不同性別在動畫創作的學習成效。研究對象為國小六年級兩班共 28 位學生，以實證研究進行為期 6 週共六小時的教學實驗，研究發現不同性別學生對故事創作向度有顯著影響，女生高於男生；「設計技巧」與「創意表現」面向則未達顯著差異；在「資訊科技融入音樂劇動畫創作」的自我評估均持正向態度。

關鍵字

運算思維；性別差異；音樂圖像化；音樂劇動畫

1. 前言

程式設計是一種運算式問題解決歷程，利用電腦科學概念，例如抽象、除錯、迭代與整合來解決問題（Brennan & Resnick, 2012）運算思維（Computational Thinking）是利用電腦科學概念進行解決問題、設計系統和人類行為理解的思維模式，不只侷限於電腦科學家而是每個人都應該擁有的基本能力，也是 K-12 學生的學習基礎（Wing, 2006）。

程式設計是創造運算作品的主要方式，透過程式學習可以建立運算思維概念的過程。動畫是科技與藝術的結合，因為學習者需要發展遊戲規則、創作角色和對話，以及設計視覺的效果（Denner, Werner & Ortiz, 2012）。

而在性別與學習方面，由於男女的生理差異會影響心理差異，對不同事物的興趣和注意力亦不同。不同性別學習者對不同領域的學習表現既有差異，在資訊科技融入音樂劇動畫創作是否也有所不同，亦是一個值得探究的問題。因此，本研究以運算思維歷程為鷹架，引導學生透過「音樂圖像化」課程發展音樂劇故事，再藉由 Scratch 工具創作音樂劇動畫，以探討不同性別製作音樂劇動畫創作的學習成效。

2. 文獻探討

2.1. 運算思維核心概念

自運算思維一詞被提出之後，許多學者提出運算思維所包含的元素有問題分解、模式識別、模式歸納與抽象化、演算設計（Wing, 2006; CSTA 2011; Grover & Pea, 2013; Selby & Woollard, 2014; Google, 2015; Shute, Sun & Asbell-Clarke, 2017）。本研究將運算思維定義四個核心步驟如下：

(1) 問題拆解：將一個問題拆分成數個步驟。
(2) 模式識別：預測問題發生的規律，並找出這個規律的模式來進行測試。
(3) 抽象化：找出導致這個模式的因素或準則。
(4) 發展演算法：建立一個能解決類似問題並能重複實施的做法。

本研究以「音樂圖像化課程」為學習鷹架，發展資訊科技融入音樂劇動畫創作課程之可行性，並探究對國小六年級學生音樂劇動畫之創作表現。

2.2. 程式設計學習與性別差異

學習者在電腦科學方面的性別差異備受爭議，大多數人認為男性具有更多的數學邏輯與計算能力，也經常因為電腦科學家較多男性而造成刻板印象（Margolis & Fisher, 2003）。然而，也有研究結果指出不同性別並不會影響程式設計的成效，女性會花費較多時間與同儕合作與溝通，而男性則在程式設計上花費較多時間（Bruckman, Jensen & DeBonte, 2002）。

由於世代的變遷，兩性觀念已成熟發展，兩性不論在資源或受教權上都趨於對等，即使現今在領域上仍有性別的差異，但在設計思維、運算思維等，目前未有研究指出男女有顯著的差異（Taylor & Baek, 2019)。

3. 研究方法

3.1. 研究對象

本研究所選取的對象是以研究者服務學校的六年級學生 28 名為研究對象，男生 11 名，女生 17 名。學生在四年級時的藝術與人文課程中已學習音樂圖像的概念，五年級時開始設計音樂圖像；在資訊方面，學生已有電腦繪圖、文書處理與簡報等資訊基礎能力。

3.2. 教學設計

本研究運用運算思維的四個步驟包含問題拆解、模式比對、抽象化、發展演算等歷程（圖 1），引導學生進行專題研究，以「奇幻旅程-波斯市場」為主題，製作音樂劇動畫。並以表 1 說明以運算思維歷程引導學生設計音樂劇動畫的過程。

圖 1 以運算思維步驟進行專題研究之歷程圖

表 1「奇幻旅程-波斯市場」音樂劇動畫之教學流程

教學單元	內容	運算思維
聽！音樂在說什麼？	聆聽音樂劇作品《波斯市場》，分析音樂概念。	問題拆解
畫·音樂	(1) 聆聽樂曲，找出音樂中的特色與規則。	模式比對

	(2) 將樂曲以圖像呈現，並練習圖像配對，形成模組。	
畫·音樂·故事	(1) 根據故事內容畫分鏡圖，並加上對話。 (2) 解釋音樂要素與故事內容之關係。	抽象化
音樂劇動畫	運用 Scratch 軟體製作動畫，並加入音樂與對話。	發展演算

4. 研究結果

4.1. 音樂劇動畫創作表現

本研究以自編「音樂劇動畫創作評分量表」分析學習者的創作表現。從表 2 得知「故事創作」面向達顯著差異，表示不同性別對故事創作向度有顯著影響，事後分析發現女生高於男生；「設計技巧」與「創意表現」面向則未達顯著差異，表示不同性別在「設計技巧」與「創意表現」向度上的表現無顯著影響。

表 2 音樂劇動畫創作表現單因子變異數分析摘要表

項目	變異來源	型III平方和	df	均方	F	p
故事創意	組間	26.714	1	26.714	10.214*	.004
	組內	68.000	26	2.615		
設計技巧	組間	1.513	1	1.513	.389	.538
	組內	101.166	26	3.891		
創意表現	組間	7.487	1	7.487	1.240	.276
	組內	156.941	26	6.036		

*p<.05

不同性別的學生在故事創意的表現有顯著不同，且女生高於男生。故事創意意指學習者在編製故事情節的語文能力與創意，女生在故事創意的表現高於男生，推論其原因女性在語文技巧上較男性為優，這是因為社會化歷程及大腦組織因交互作用造成的，但二者差異相當小。因研究範圍僅限於國小六年級學生，仍需擴大研究樣本資料以佐證（圖 2 為作品之舉例）。

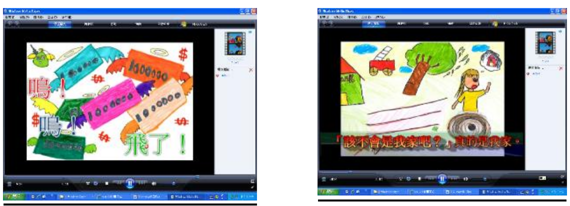


圖 2 小音樂劇動畫作品

4.2. 音樂劇動畫創作自我評估分析

以「音樂劇動畫創作學習與感受問卷」探討學習者的學習態度。從表 3 得知不同性別的學生自我評估音樂劇動畫創作的學習成效無顯著不同，從平均數來看，學習者均獲得良好的學習效果，表示資訊科技融入音樂劇動畫創作為學習成效帶來正面的影響。

表 3 音樂劇動畫創作自我評估單因子變異數分析摘要表

項目	變異來源	型III平方和	df	均方	F	p
熟悉度	組間	.095	1	.095	.160	.692
	組內	15.323	26	.589		
幫助度	組間	.110	1	.110	.258	.616
	組內	11.104	26	.427		
滿意度	組間	.097	1	.097	.146	.706
	組內	17.269	26	.664		

5. 結論與建議

本研究以運算思維歷程發展「音樂劇動畫創作課程」，提了學生完成音樂劇動畫創作歷程中的思考鷹架，輔助學生更有效的進行創作歷程的任務。此外，以「音樂圖像化課程」之鷹架引導學生運用 Scratch 創作音樂劇動畫之教學模式，揭示具體教學策略，可提供教學者參考與進與修正，讓相關課程的教學與研究更臻完善。

6. 參考文獻

Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual American Educational Research Association meeting*, 1-25.

Bruckman, A., Jensen, C., & DeBonte, A. (2002, January). Gender and programming achievement in a CSCL environment. *Proceedings of the conference on computer support for collaborative learning: Foundations for a CSCL community*. International Society of the Learning Sciences, 119-127.

CSTA (2011). CSTA K–12 Computer Science Standards. Retrieved December 15, 2019, from http://www.csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf

Denner, J., Werner, L. & Ortiz, E. (2012). Computer Games Created by Middle School Girls: Can They be Used to Measure Understanding of Computer Science Concepts? *Computers & Education*, *58*, 240-249.

Google (2015). *Exploring Computational Thinking*. Retrieved December 15, 2019, from https://edu.google.com/resources/programs/exploring-computational-thinking/

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

Selby, C. & Woollard, J. (2014). Computational Thinking: The Developing Definitions. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014. ACM.*

Shute, V. J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research Review*, *22*, 142-158.

Taylor, K., & Baek, Y. (2019). Grouping matters in computational robotic activities. *Computers in Human Behavior*, *93*, 99-105.

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

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Computing Thinking and Artificial Intelligence Curriculum Design: AI Power Recycling Box

Tai-Ping HSU[1], Ting-Chia HSU[2*], I-Chen HUANG[3]
[1,3]Taipei Municipal Rixin Elementary School, Taiwan
[2]National Taiwan Normal University, Taiwan
taipinshe@zhps.tp.edu.tw, ckhsu@ntnu.edu.tw, hij@mail.zhps.tp.edu.tw

ABSTRACT

This course develops an 18-period instructional material integrating the basic concept and application of Internet of Things, data science, and supervised machine learning in Artificial Intelligence. The course emphasizes the students' hands-on activities which carry out the process of doing, using, and thinking, and evolve them from users to designers. Based on the guidance of computational thinking phases, the students were instructed to design the AI power recycling boxes which can collect the left electricity in the battery. The problem solving process is integrated with the actual situation in daily lives so the literacy-based learning was achieved according to the new guideline of information technology discipline in Taiwan. The further concept that the students gained was to understand the possibilities of non-nuclear family from the interdisciplinary curriculum on energy education. This course presented the learning performance and content of natural science and the students did not learn the knowledge in fragment. They applied what they learned to solve everyday problem, highlighting the deepening value of literacy-based curriculum. The results of conducting this course in the primary school showed that the learning performance of the students on energy education was promoted. In addition, they could solve this real-life problems.

KEYWORDS

computational thinking, artificial intelligence, supervised learning, energy education

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

運算思維暨人工智慧教學活動設計：AI 電力回收盒

徐臺屏 [1]，許庭嘉 [2*]，黃怡真 [3]
[1,3] 臺北市大同區日新國民小學，臺灣
[2] 國立臺灣師範大學科技應用與人力資源發展學系，臺灣
taipinshe@ zhps.tp.edu.tw，ckhsu@ntnu.edu.tw，hij@mail.zhps.tp.edu.tw

摘要

本課程發展出完整的 18 節教學內容，環扣資料科學與人工智慧中的監督式學習，著重學生在課程中透過做、用、想等歷程來動手實作，讓學生從使用者進化到設計者。透過運算思維教學設計引領學生設計能回收電力與發電的 AI 電力回收盒，達成臺灣資訊科技教育新課綱所要求的生活出發素養導向學習，跨科整合能源教育探究非核家園的可能。整合自然科學領域學習表現與內容，學生學習的不是片面知識或概念，而是應用與解決生活問題的素養。本課程實施過程與結果發現能有效提升學生在能源教育的理解與分析能力，並能應用該能力解決生活真實問題。

關鍵字

運算思維；人工智慧；監督式學習；能源教育

1. 前言

臺灣乾電池一年的使用量近 9000 公噸，但其實民眾回收廢電池的量不到 4000 公噸。北市環保局表示，回收 3868 顆各型號電池，有將近六成的電池尚存一半以上的電量。本課程透過運算思維的拆解問題、模式識別、抽象化與演算法推演，並結合人工智慧（Artificial Intelligence）的監督式學習，引導學生設計出 AI 電力回收盒作品，試圖將一次性電池被丟棄前的餘電移轉到可充式電池上，以達到環境保護與廢物回收再利用的效能。

2. 文獻探討

2.1. 運算思維教學

運算思維（Computational Thinking）近年廣泛應用在中小學教育中。林育慈、吳正己（2016）統整各國資訊科技教育的研究發現，運算思維應包含抽象化（abstraction）、問題分解（problem decomposition）、模式化與模擬（modeling and simulation）及演算法思維（algorithmic thinking）。運算思維的教學強調從動手實作的具體經驗中學習拆解、模式識別、抽象化及演算法則的邏輯思考與問題解決能力（Google, 2019；林育慈和吳正己，2016）。

2.2. 人工智慧的監督式學習

曲建仲（2018）歸納機器學習包括下列四個步驟，分別是獲取數據、分析數據、建立模型與預測未來。而監督式學習（supervised learning）是機器學習的種類之一，機器在監督式學習下所有資料都有標準答案，可以提供機器學習在輸出時判斷誤差使用，預測時比較精準。

3. 教學設計

3.1. 教學對象

本課程之教學對象為臺北市某國小五、六年級之學生。教學對象自國小三年級開始學習資訊議題課程，熟悉資訊工具基本操作，了解基本資訊知能。並曾於資訊議題課程中學習視覺化程式語言（Scratch），已具基礎的程式邏輯概念。

3.2. 教學活動設計

本課程設計以運算思維教學為核心，引領學生將廢電池回收問題進行問題拆解，接著探究回收電池電力多寡的共通模式，透過 BlocklyDuino 圖控開源軟體進行電力回收盒的程式設計，接著將回收電池的電力狀況抽象化之後將模式規律透過程式語言寫入到電力回收盒內，最後透過演算法推演提高電力回收的效能。以下為運算思維與監督式學習教學模式之課程架構與 BlocklyDuino 軟體介面：

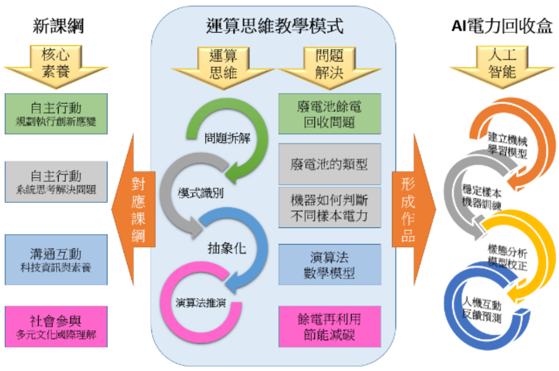


圖 1 運算思維教學模式

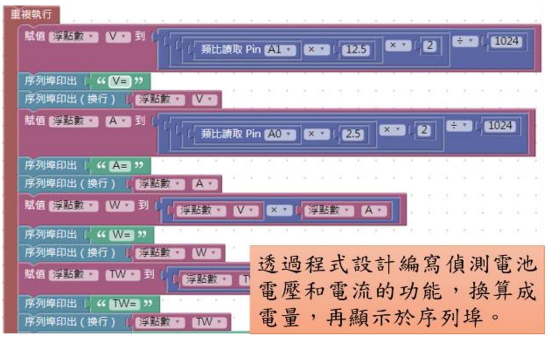


圖 2 BlocklyDuino 圖控開源軟體

在電力回收盒的設計上導入人工智能的監督式學習，透過獲取回收電池的數據後進行人工分析，透過人工訓練機器來建立機器的特徵模型，讓電池接著放入電力回收盒後都能被妥善的回收電力，機器也能針對目前的電力狀況進行預測與回收電力。本研究之教學活動設計共設計九大單元主題課程，分別是風力發電、手動發電、太陽能與化學能發電、發電電流與電壓感測、發電效能比較與探究、自製 AI 電力回收盒、物聯網監測與通知、專家系統與大數據分析、善用能源、非核家園。

3.3. 教學評量

本課程透過素養導向評量檢核表與學生學習日誌來進行質量兼顧的教學評量，評量內容設計如下：

表 2 本課程之教學評量表

素養導向評量檢核表　　　學生學習日誌

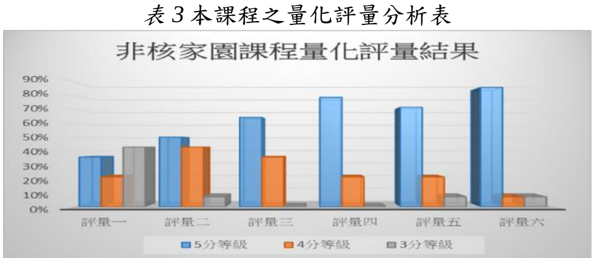
4. 結果與討論

本課程透過教師教學與學生的學習反思討論後，歸納結果如下。

4.1. 學生學習評量結果

本課程實施對象共 14 位高年級學生，在課程實施後針對學生對於非核家園的量化評量結果如下。大多數的學生在五點量表中，自我評估都有超過 4 分以上，只有在第二項、第五項、和第六項有極少數的 3 分，以及第一項有大約 40%的 3 分。各評量的平均分數除了第一項為 3.93 分，其餘也都在 4 分以上。

表 3 本課程之量化評量分析表



項次	評量題目	平均
評量一	1. 我能知道核能發電對環境影響甚鉅。	3.93
評量二	2. 我知道邁向非核家園能改善人類生活。	4.43
評量三	3. 我知道全世界許多國家正努力研發各種邁向非核家園的可能。	4.64
評量四	4. 我知道依目前的科技，台灣要完全進入非核家園的可能性太低。	4.79
評量五	5. 我能與他人團隊合作，透過創意思考來歸納出提升節能減碳效能的策略。	4.64
評量六	6. 我能設計、動手實作創作出能蒐集發電資訊反饋並結合AI的儲電盒。	4.79

4.2. 學生學習單回饋結果

整合知識能力與態度	遇到困難解決方案
(what)上課學到手搖發電 (when)停電的時候會用到 (where)每個家庭都會有 (why)利用手動產生動能帶動馬達產生電能	老師給我們看腳踏車前輪旁邊有一個發電器，可以把我們騎腳踏車的動力轉成電力供電給車燈。
發電的同時，也可能產生污染。	核能發電我還不太懂，但我知道有火力、水力與風力發電了。

4.3. 教師反思

運算思維教學：運算思維結構教學設計能有效幫助教師解構並重新建構自己的教學活動，用更結構化的視角來審視活動流程與安排，大幅改善與提升教師教學成效。

真實動手做：課程安排多真實動手機會，透過學生操作，呈現數據提供學生思考與回應。在找答案的過程裡，讓學生學會學習的方法。教學提供學生探究及成功學習的經驗，立即給予學生回饋。

符合學生學習發展：與自然領域搭配，注意學生學習先備知識，學生能夠在原有的學習能力上繼續深度探究。科技議題強調真實情境知生活應用，學生在操作中能夠體會新興科技目前在生活中的角色。

5. 結論與建議

本課程透過運算思維課程設計整合人工智慧監督式學習概念，引領學生認識演算法與開發監督式學習機器模型（Modeling），並再透過物聯網蒐集樣本對機器進行訓練（Training），最後進行大數據分析出各種樣態後對機器校正（Correction），以進行回收電池效能預測（Predict），達到良好人機協作，提升節能減碳之效。課後針對學生學習成效分別對六個評量表現進行五點量表分析，學生表現平均有 64%達到五分等級，有 25%達到四分等級，有 11%達到三分等級。而學生學習成果日誌中發現，學生能透過教師有效教學在過程中「做用想」以探究與解決真實生活問題，期能提供相關教學活動設計之參考。

致謝

本課程榮獲臺灣教育部國教署 2018 年國小組新興科技認知教案設計競賽獎勵並出版教案供全國小學教學參考。

6. 參考文獻

林育慈和吳正已（2016）。運算思維與中小學資訊科技課程。**教育脈動，6**，5-20。

曲建仲（2018）。機器是如何學習與進步？人工智慧的核心技術與未來。**科學月刊，580**，282-291。

Google (2019). *Exploring Computational Thinking*. Retrieved January 18, 2019, from https://edu.google.com/resources/programs/exploring-computational-thinking/

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

A Pedagogical Practice in Teaching MIT App Inventor Programming in Grade 5 for Computational Thinking Development: Developing a Winning Number Guessing App

Wing-kai LI [1*], Man-hay LAM[2*], Shing-chun YEUNG[3]
[1,2,3] Ling To Catholic Primary School, Hong Kong
kainova@lingto.edu.hk, lamk@lingto.edu.hk, yeungshingchun@lingto.edu.hk

ABSTRACT

This is an example illustrating a pedagogical practice in teaching MIT App Inventor programming. A total of 26 students in a Grade 5 class were taught to develop a Winning Number Guessing app. The students were first engaged by playing the completed app, which arouses their interests and familiarise themselves with the programming tasks. With the guidance of teachers, students were able to think about the programming flow and the functions of different components they used through in-class activities. Their computational thinking skills were therefore gradually developed. Teachers also guided students to find related blocks in the programming environment in order to facilitate their coding processes. Students learnt blocks of "Button" components and the "Logic" involved in building the Winning Number Guessing app. This classroom practice supported that "to play, to think and to code" (Kong, Lai & Sun, in press) is a promising pedagogy for computational thinking development. High order thinking questions and extension tasks were designed for catering students' diversity and creativity. We shall share with the forum participants a short video on classroom activities.

KEYWORDS

computational thinking, logic, MIT App Inventor, pedagogy, programming

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

運算思維教育的教學反思:
教授小五學生利用 MIT App Inventor 編寫估計神秘數字應用程式的教學實踐

李永佳 [1*]，林汶熹 [2*]，楊承峻 [3]
[1,2,3] 天主教領島學校，香港
kainova@lingto.edu.hk，lamk@lingto.edu.hk，yeungshingchun@lingto.edu.hk

摘要

本文闡述一個以 MIT App Inventor 教授編程的實踐例子。26 名小五學生學習如何編寫一個估計神秘數字（Winning Number Guessing）的應用程式。他們首先試用已完成的程式，藉此加深對程式的理解和引發學習編程的興趣。老師在課堂活動中引導思考流程和所需組件，並發展運算思維。此外，老師亦引導學生在編程環境尋找組件及相關流程結構，協助學生編寫應用程式，學生在編寫程式的過程會學習到不同「按鈕（Button）」的程式碼。本教學實踐引證「玩-思考-編程」（Kong, Lai & Sun, in press）是有利運算思維教育發展的課堂學習模式。

關鍵字

MIT App Inventor；教學法；運算思維；編程；邏輯

1. 前言

在數碼化世界中，運算思維是每個人都應具備的技能（Wing, 2006）。Brennan 和 Resnick（2012）提出了運算思維課程的教學框架，當中以運算思維概念（Concepts）、實踐（Practices）和視野（Perspectives）作為編程學習的三個重要範疇指標，以回應 Jeannette Wing 在 2006 年所提倡的觀點。Kong（2016）認為運算思維課程應按照「趣創者理論（Interest-driven Creator Theory）」設計，把學生培養成爲解決數碼問題的創造者。

2. 編寫估計神秘數字應用程式

26 名小五學生在是次教學實踐中會學習利用 MIT App Inventor 建立手機應用程式，以促進數學邏輯思維學習。透過建立估計神秘數字程式，學生能夠學習到關於「按鈕（Button）」的程式碼，並且加強對「事件」、「等於、不等於、和」和「如果━則」等條件程式碼的理解。 估計神秘數字應用程式的設計如圖 1 所示。

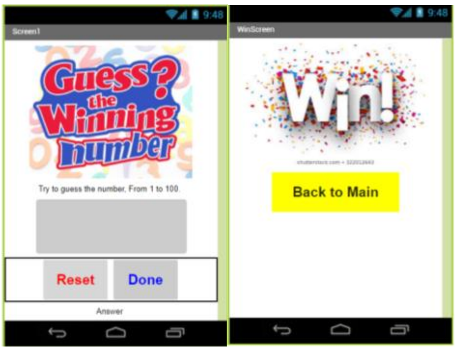


圖 1 估計神秘數字應用程式的界面設計

3. 編寫估計神秘數字應用程式的教學實踐

根據「趣創者理論」（Chan et al., 2018），課程設計和教學法應以引起學習興趣為目標，從而激發學生的數碼創意。學生在教學實踐中會先試玩已完成的生字應用程式。接著，學生需要回答老師關於程式運作的問題（詳情見圖 2）和完成工作紙（詳情見圖 3），旨在加深他們對程式的理解及培養他們的演算思維。

主持隨意寫出一個數字，參加者輪流在 1 至 100 中說一個數字；
因應參加者所說的數字，而收窄範圍，說中了數字者便勝出；
如抽出的數字為 72，第一位參加者說 50，主持宣布把範圍收窄至 50 至 100；
第二位參加者說 80，主持宣布把範圍收窄至 50 至 80，如此類推。

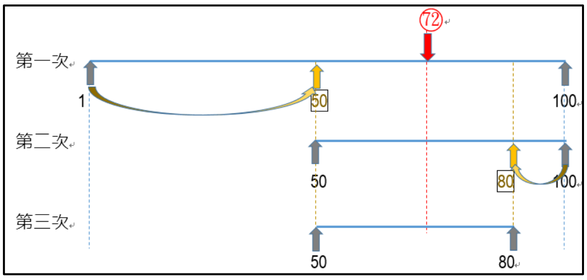


圖 2 估計神秘數字的課堂討論念展示圖

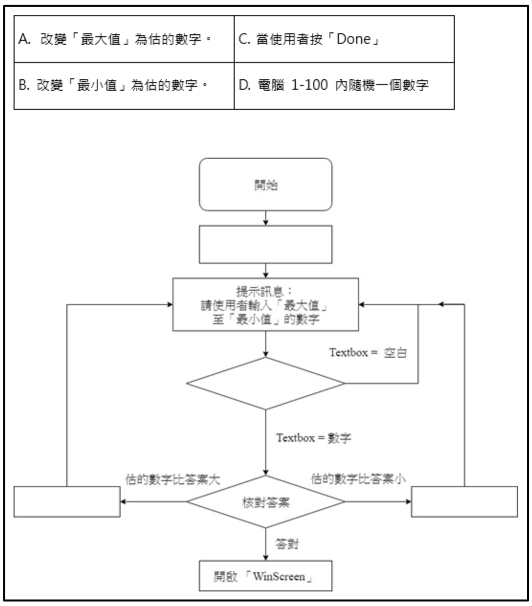


圖 3 估計神秘數字應用程式課堂工作紙

完成工作紙後，我們會鼓勵學生思考編程時所需要的組件，以及了解應用程式的界面設計。編寫估計神秘數字應用程式所需之組件如圖 4 所示。

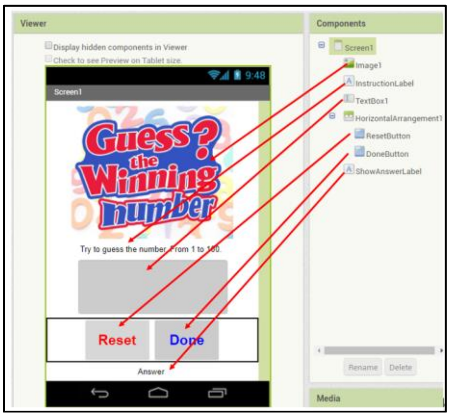


圖 4 估計神秘數字應用程式的組件及界面設計

了解程式的組件及界面設計後，學生便可以開始製作所需的變數（Variable）、命名和定義，如圖 5 所示。

程式內的變數 (Variables) :

RandomNumber:	記錄遊戲中需要估的數字 (由電腦隨機抽出)
LowerHints:	記錄範圍中的「最小值」
UpperHints	記錄範圍中的「最大值」

圖 5 變數定義及其程式碼

開始時需要由電腦隨抽出一個「神秘數字（RandomNumber）」，並建立相應設定的程序（Procedure）。由於方便測試和驗證，該「神秘數字」會顯示在 ShowAnswerLabel，直至程式完成後才隱藏。程序的程式碼如圖 6 所示。

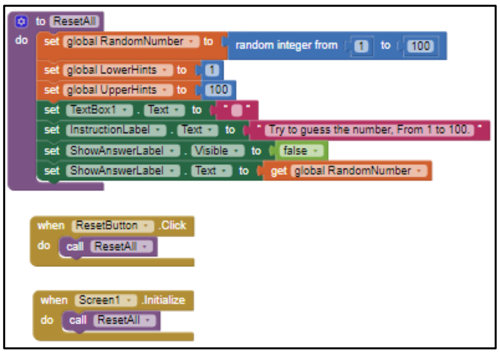


圖 6 程序的程式碼

學生在逐步建構估計神秘數字應用程式前需要預先測試應用程式，以確認神秘數字是隨機產生的。這個過程可以深化他們對運算思維實踐的理解，包括「測試及除錯」及「反覆構思及漸進編程」。

當學生設置按下按鈕「事件（Event）」後的條件和結果並找出除錯方法後，他們便能夠理解及應用「條件（Condition）」和「運算子（Operator）」的運算思維概念。按下按鈕事件及條件的程式碼如圖 7 所示。

圖 7 按下按鈕事件及條件的程式碼

學生的多樣性是必然存在的，延伸學習也是課堂設計的重要一環，課堂工作紙加入思考問題和延伸任務能讓學習拓展至課堂以外，訓練學生高階思維，啟發學生無限創意。思考問題和延伸任務如圖 8 所示。

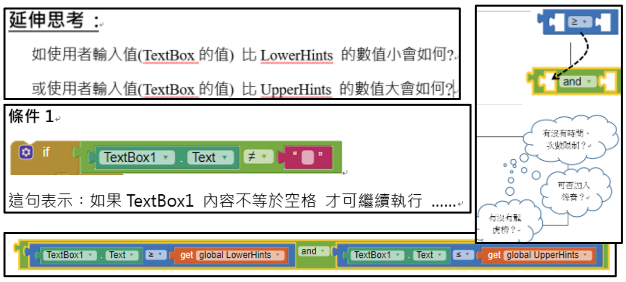


圖 8 思考問題和延伸任務的程式碼

4. 反思：以教授編寫估計神秘數字應用程式看運算思維教育發展

學校在教授運算思維時焦點應放在讓學生透過編程建立運算思維概念、實踐和視野，培養和提升學生運用數碼科技解難的能力，而不是重於學生編寫應用程式。

5. 結論

運用「玩-思考-編程」教學法能增強學生對學習編程的動機，提升數碼能力（Digital Competence）及創意能力，培養學生利用數碼科技去解決問題。

6. 參考文獻

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association.* Vancouver, Canada: American Educational Research Association, 1-25.

Chan, T. W., Looi, C. K., Chen, W., Wong, L. H., Chang, B., Liao, C. C., ... & Jeong, 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."

DK (2018). *Help Your Kids with Computer Science*. United States: DK Publishing.

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.

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

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

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Research on Personalized Teaching Strategy Driven by Artificial Intelligence in K12 Stage

Qiang LIU
Shude High School, China
22805727@qq.com

ABSTRACT

It has always been the vision of educators to implement personalized teaching in K12 education stage. Traditional teaching strategies such as hierarchical and classified teaching and small-class teaching have been applied, but the effect is relatively limited. This paper discusses the pain points in the process of traditional personalized teaching, studies the technical guarantee based on artificial intelligence and big data, accurately collects and analyzes the multi-dimensional data such as classroom record data, practice data and evaluation data of middle school students in the process of education, and makes a data-based portrait of students. The strategy of teaching students in accordance with their individual characteristics is studied, thus really realizing personalized teaching based on data.

KEYWORDS

artificial intelligence; personalized teaching; educational data; educational data analysis

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

K12 阶段人工智能驱动个性化教学策略研究

刘强
四川省成都市树德中学，中国
22805727@qq.com

摘要

在 K12 教育阶段实施个性化教学一直是教育人的愿景，运用了如：分层分类教学、小班化等传统教学策略但效果相对有限。本文讨论了传统个性化教学过程中面临的痛点，研究了以人工智能、大数据为主要技术保障，对教育过程中学生的课堂实录数据、练习数据、测评数据等多维度数据进行精准收集与分析，对学生进行数据化画像。根据不同学生的个性化特征进行因材施教的策略研究。真正实现基于数据的个性化教学。

关键词

人工智能；个性化教学；教育数据化；教育数据分析

1. 研究背景

1.1. 研究背景与技术保障基础

从 1956 年开始，人工智能的发展历经 60 年共二次起伏，现在已经进入第三次热潮！再结合大数据技术近年得到长足发展，在教育领域我们希望能依靠人工智能技术驱动个性化教学策略得以落地，真正让我们的教育能够做得到因材施教。

1.2. 数据收集手段已经成熟

教师面对几十位同学的时候，如何让教师对每位同学进行更准确的个性化关注，并且发现每位同学的优势与短板？随着大数据技术的发展，收集数据的手段与工具已经成熟，能够给教师提供更专业的数据支撑，用技术手段辅助老师更好地关注每个学生。是目前人工智能大数据技术能够解决的，技术基础已经成熟。

1.3. 数据分析模型日趋成熟

教育过程数据会产生维度众多的非结构化数据，而且在传统信息化工作中无法形成可处理、可分析的数据。目前人工智能大数据技术的发展已经能够把多维度的教育数据转换成结构化的可用数据信息。

人工智能技术将各种教育场景数据连接起来，起到穿针引线的作用，利用数据进行辅助性或决策性的分析。所以数据收集与分析模型已经日趋成熟，为人工智能深入教育提供了技术可能。

1.4. 个性化教学的痛点分析与问题提出

在 K12 教育阶段实施个性化教学提出了一些教育教学手段，如分层分类教学，小班化等策略。但是效果都是相对有限的，有如下的痛点：

痛点 1：课堂的传统教学模式在教师意识层面存在巨大教学惯性，传统教师对于信息化工具仅停留在 PPT 辅助展示。

痛点 2：传统信息化手段还未达到解决老师提升教学效率的核心层面，对于学生的学习数据无法精准反馈，对于学生的判断还简单停留在教师的印象和测评结果上，支撑数据不够精细。

痛点 3：传统教学班级，同时进行的教学时间、相同的授课教师、一样的课表、无差别的考试，很难做到个性化教学，分层教学缺乏依据与手段。

痛点 4：所有同学练习相同的作业，训练内容与标准完全统一，K12 阶段学生课业负担过重，无效训练太多，无法实现精准性补益。

1.5. 问题提出

在以上教学痛点中，结合人工智能、大数据技术能否精准获取学生学习过程性数据，并将其进行分析，将每位同学精准的学习情况进行有效反馈给教师，并且请结合教师的专业判断和智慧学习平台给学生准确推送个性化的训练内容，达到精准训练，精准教学？

2. 人工智能驱动个性化教学的策略研究

随着人工智能、大数据技术的发展，教育教学系统的作用不仅仅是简单的技术支持，而是能够深入教学课堂教学实现个性化教学因材施教的技术保证。实现个性化教学的策略研究如下：

2.1. 策略一：建立校园智慧教育系统

图 1 智慧校园教育系统逻辑

建立人工智能驱动的智能教学系统，将会衔接教师的教和学生的学，教师的知识点的精讲与点评、学情分析、教学效果评价与任务的发布、精准推送的学习资源等教学动作都通过教学系统完成。学生对知识进行探索与探究、展示与分享、提交接受任务、获取评价报告等都在系统支持下完成。

2.2. *策略二：建立校园智慧云全流程数据采集，对学生进行数据画像，为个性化教学做数据支撑；*

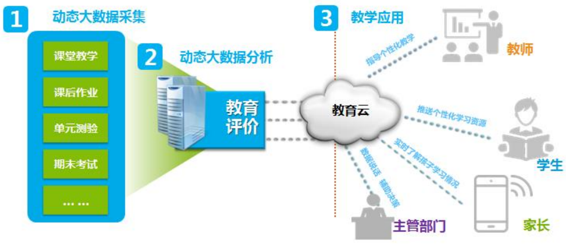


图 2 多维度的数据采集与数据服务

校园智慧云是基于大数据个性化学习系统。将对教育过程中所有学生学习活动进行全维度数据采集，包括：课堂教学数据、日常的作业与测评、考试相关数据、学生阅读行为采集与分析等全维度数据记录。

2.3. *策略三：校园智慧云课堂—以学定教、分层与定制化个性化的教学设计 ；*

在智慧云大数据支持下，教师依据各类学生大数据分析结果，可以看到学生通过预习资源数据的反馈，包括了学生对教学内容各知识点的数据反馈，教师对学生不同知识点需求，分层、定制化个性化进行教学目标设计。最后是利用移动终端技术，将学案推送给学生，让每个学生获得自己的专属学案，为后续的教学提供个性化的学习材料。

2.4. *策略四：校园智慧云课堂—实施学生个性化学习方案；*

对于学生来说，通过对他日常过程性的数据采集分析，包括他的考试数据，练习数据和作业数据等，能够有针对性的通过系统分析出他的薄弱环节，这个薄弱环节是基于知识点图谱的分析和诊断。

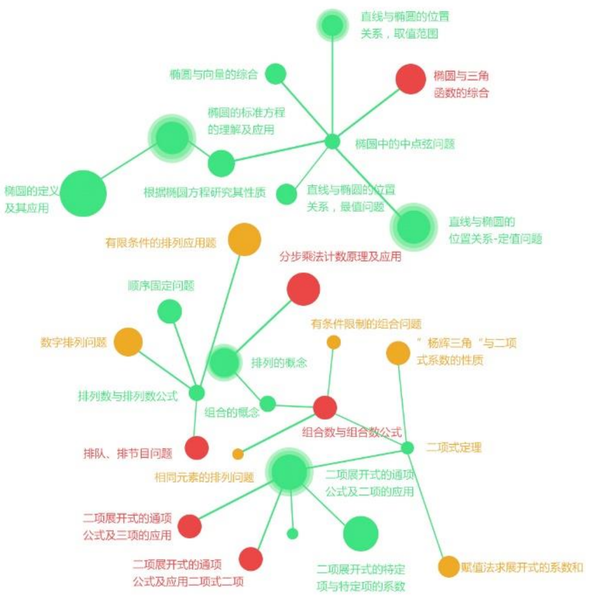


图 3 知识图谱体系

该学生可能二项式展开这个知识点出了问题，系统会构建出二项式展开这个知识点的知识图谱，前趋知识点是什么，后趋知识点是什么呢，通过过程性数据的分析，他能找到学生的前趋知识点是哪些环节可能出现了掌握不牢靠的情况。那么这样就能够有针对性的进行资源的推送，帮助他去改进和提升。从而能够进一步去优化他的学习过程。

2.5. *策略五：校园智慧云课堂—以学定练、实施个性化练习；*

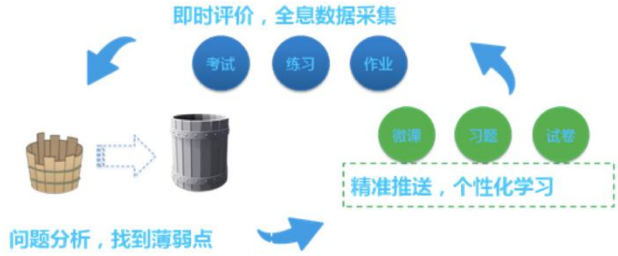


图 4 精准推送个性化学习任务流程

让每个孩子回家后有不同的家庭作业，校园智慧云具有自适应学习功能，可以根据学生对知识点掌握的不同情况，自动的给学生推送适合他学习情况的个性化学习任务。真正的实现智慧云支撑下的个性化学习。实现真正的因材施教。

3. 结语

K12 阶段教育教学过程中用人工智能、大数据技术把大规模的数字化信息转换成为可以分析的结构化数据信息。通过互联网将各种场景连接起来，对学生进行智慧精准反馈，为学生提供个性化学习方案，针对个体的不一样的教案与课程，精准的立足于解决知识弱点与盲区的测试，真正做到准确分层教学、个性化教学，真正做到因材施教的教育愿景。

4. 参考文献

张进宝、姬凌岩（2018）。是"智能化教育"还是"促进智能发展的教育"AI 时代智能教育的内涵分析与目标定位。**现代远程教育研究**，14-23。

梁迎丽、梁英豪（2019）。人工智能时代的智慧学习:原理、进展与趋势。**中国电化教育**， 16-21。

闫志明、唐夏夏、秦旋（2017）。教育人工智能（EAI）的内涵、关键技术与应用趋势——美国《为人工智能的未来做好准备》和《国家人工智能研发战略规划》报告解析。**远程教育杂志**，26-35。

Bernard, J. , Chang, T-W., & Popescu, E. (2017). Learning style identifier: Improving the recision of learning style identification through computational intelligence algorithms. *Expert Systems with Applications*, (75), 94-108.

Kim, Y., Behnagh, R. F., & Soyata, T. (2017). Towards e-motionally aware ai smart classroom current issues and directions for engi-neering and education. IEEE Access, (6), 5308-5331.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

A Pedagogical Practice in Teaching MIT App Inventor Programming in Grade 6 for Computational Thinking Development: Developing a Factor App

Hon Wai MOK
Islamic Primary School, Hong Kong
mohw@islamps.edu.hk

ABSTRACT

This is an example illustrating a pedagogical practice of how "to play, to think and to code" (Kong, Lai & Sun, in press) in teaching MIT App Inventor programming. A class of 25 Non-Chinese students in Grade 6 were taught to develop a factor app (Kong, 2019; CoolThink@JC, 2019). Students learned how to use the programming language, "FOR LOOP" to list the factors of a number. In order to arouse students' interests and enhance their understandings on the programming tasks, students were first engaged in playing with the completed Factor app. The app helped students to practice their math skills by testing their knowledge of prime and composite numbers as well as their factors. Students applied the CT concepts such as "procedure", "conditionals" and "algorithmic thinking" in the programming decision process. Learning by doing teacher-student practice could be applied during the lesson. The sequences of the code were at first changed with students needing to later arrange into them correct order. Students' computational thinking practices, such as "testing", "debugging", and "being incremental and iterative" could be consolidated with teacher guided lessons.

KEYWORDS

computational thinking, pedagogy, primary school Non-Chinese students, programming, learning by doing

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

運算思維教育的教學反思：
教授小六學生利用 MIT App Inventor 編寫尋「因」遊戲的教學實踐

莫漢威
伊斯蘭學校，香港
mohw@islamps.edu.hk

摘要

本文闡述一個透過「先遊戲、後思考、再編程」（to play, to think and to code）（Kong, Lai & Sun, in press）的課堂實踐模式來教授編程的教學例子。25 名六年級非華語學生會在本單元建立一個數學遊戲程式 — 尋「因」遊戲（Kong, 2019; CoolThink@JC, 2019）。學生透過學習使用循環（for loop）列出一個數字的因數。這個程式會測試學生對質數、合成數和因數的知識，從而協助他們提升數學能力。學生要運用「程序」（procedure）及「條件」（conditionals）的編程概念，並學習如何運用「算法思維」（algorithmic thinking）來確定一個數字是質數或合成數。他們首先會試玩已完成的尋「因」遊戲程式，從而加深對程式的理解和引發學習編程的興趣。老師在課堂前運用表列法（圖 1）教授除數（divisor）、被除數（divident）、商數（quotient）及餘數（reminider）的關係，即除數除以被除數，若餘數等於零，則該除數便是其中一個因數。籍此基礎概念再引導學生思考整個設計流程。為配合非華語學生學習的獨特性，課堂會經常以「邊幹邊學」（learning by doing） 的模式進行，增加師生、生生互動。學生對程式製作有初步了解後，老師會先將部份程式碼調亂，再讓學生在編程環境中進行測試、重組、修改組件及相關流程結構，透過這樣的實踐、測試及除錯來幫助學生更明白程式的運作。

Equation	Divisor	Quotient	Reminder	Is Factor
6÷ 1	1	6	0	✓
6÷ 2	2	3	0	✓
6÷ 3	3	2	0	✓
6÷ 4	4	1	2	×
6÷ 5				
6÷ 6				

圖 1 表列法

關鍵字

運算思維；教學法；非華語學生；編程；邊幹邊學

1. 前言

美國哲學家杜威（1897）提出「邊幹邊學」（learning by doing）的教育理論。他認為學習應該是相關的和實用的，而不僅僅是被動的和理論的。而 Kong（2016）表示，運算思維課程根據「趣創者理論」（Interest-driven Creator Theory）設計，培養學生成爲數碼創造者，以數碼科技解決問題。運算思維課程與一般課程不同之處是講求學生的學習經歷，從實踐中找出錯處並作出修正，再延續發展及優化。

2. 編寫尋「因」遊戲

25 名六年級非華語學生會在本單元建立一個數學遊戲程式 — 尋「因」遊戲，透過學習使用循環（for loop）列出一個數字的因數。普遍學生對數學課題中因數的概念比較弱，希望學生藉建構程式時有機會再次思考除數（divisor）、被除數（divident）及餘數（reminider）與因數之間的關係。學生並不單單設計程式，更要運用「算法思維」（algorithmic thinking）透過編程來判斷一個數字是質數或是合成數。尋「因」遊戲的界面設計如圖 2 所示。

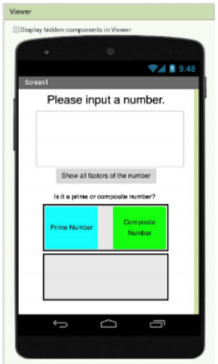


圖 2 尋「因」遊戲的界面設計

3. 編寫應用程式的教學實踐

根據「邊幹邊學」理論，教師於課堂設計和教學法上需提倡“多動手、多思考”的教學活動來刺激學生的學習興趣。於施教上運用核心三步曲：先遊戲、後思考、再編程（to play, to think and to code），以助學生完成課堂的學習目標。學生會先試玩完整版的遊戲，接著，學生需要思考程式的運作，為免學生盲目跟從筆記中的指示堆砌出程式碼，老師使用試誤法（trial and error method）（Starch, 1910）的教學方法，先為學生預備部份程式碼（見圖 3），學生需要將餘下的程式碼放回正確的位置，過程中使用平版電腦進行實時測試，與組員討論、找出錯處並作出修正，學生在過程中能加深認識對運算思維中序列的概念。

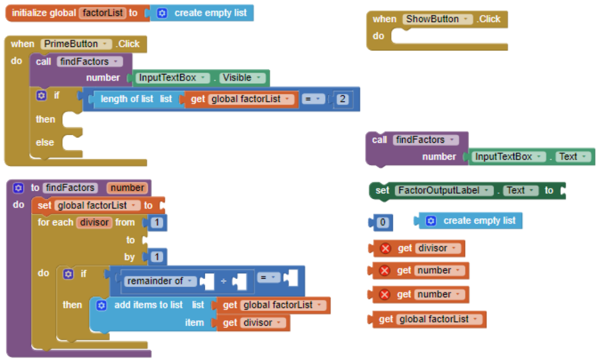


圖 3

完成後，老師鼓勵學生將程式中某些步驟上下對調，再進行測試，學生從探究中再加深了解程式的運作及程序，這個過程可以深化他們對運算思維實踐中「測試及除錯」和「反覆構思及漸進編程」的理解。

課堂中老師鼓勵學生將程式優化，從運算思維實踐中經歷反覆構思及漸進編程。有學生發現測試數字是質數或合成數的時候，正如電腦程式中最基本的“0”和“1”關係，所以可以將“prime number”、“composite number”按鈕合併。學生將以下用來控制“prime number button”、“composite number button”的介面（見圖 4）及程式碼作出優化（見圖 5.1、5.2）。

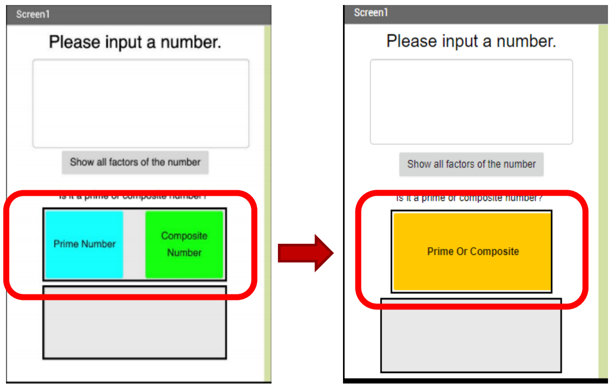


圖 4 學生將原先按鈕由兩個改為一個

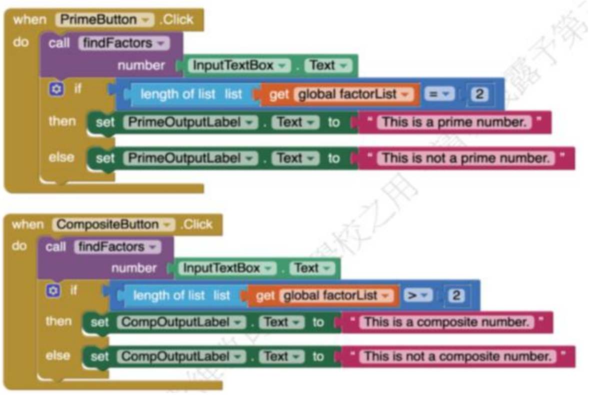


圖 5.1 學生將原先的程式碼由兩組合拚成一組

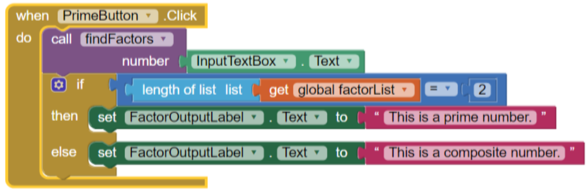


圖 5.2 學生將原先的程式碼由兩組合拚成一組

學生認為這樣的優化，能讓使用者有多一重思考的機會，從顯示出來的因數數量，再連繫到質數與合成數的關係，這樣更能適切地照顧學習這數學課題的初學者及不同學生的學習需要。

4. 反思：運算思維對學生的發展需要

運算思維教育可幫助學生由單純的科技消費者培育為一群科技應用的創造者。學生創意是無限的，只欠一個合適的平台，賽馬會運算思維教育就正正提供一個平台，讓學生從日常生活中發揮數碼創意，並為他們對創新科技的發展和應對未來的挑戰作好準備。教授運算思維不單只讓學生學習編寫程式，更重要是課堂上容讓學生出錯，這點正正與一般學科科目的課堂剛剛相反，一般學科都是由老師指出錯處，但教授運算思維卻是鼓勵學生嘗試、出錯，讓他們自行找出錯處並作出修正，這正正是有助發展學生自主學習，培養他們運用數碼科技解決問題之高階思維的能力。

5. 結論

運算思維中「測試及除錯」和「反覆構思及漸進編程」對小學階段極為重要，學習旅程難免會跌跌碰碰，學生從失敗中學習，作出修正，繼而再進一步構思及優化，我相信這個學習的態度不單能應用到學科上，甚至對於學生的成長都能帶來莫大的益處。

6. 參考文獻

CoolThink@JC (2019). *Factor Game*. Retrieved October 10, 2019, from https://portal.coolthink.hk/

Dewey, J. (1897). *My Pedagogic Creed*. United States: Franklin Classics Trade Press.

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.

Kong, S. C. (2019). Learning Composite and Prime Numbers Through Developing an App: An Example of Computational Thinking Development Through Primary Mathematics Learning. In *Computational Thinking Education*, 145-166. Springer, Singapore.

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

Starch, D. (1910). A Demonstration of the Trial and Error Method of Learning. *Psychological Bulletin, 7*(1), 20.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Cross-learning Stage Technology Implementation Activities Probe-Obstacle Avoidance Car

Shih-wen SU[1*], I-fan TSAI[2*]
[1]Keelung Municipal ShenMei Elementary School, Taiwan
[2]Keelung Municipal MinChwan Junior High School, Taiwan
alvinmalego@gmail.com, evonne722@gmail.com

ABSTRACT

This course focuses on the convergence of science and technology courses in elementary and middle schools. At the elementary school stage, the Motoduino board and Scratch program are used to control the obstacle avoidance car. Students are guided to use problem solving, pattern recognition, abstraction, and algorithms of computational thinking to learn the control logic of obstacle avoidance. At the stage of middle school, students work with electricity control and also the computational thinking to achieve the same learning goal as the elementary school students. It is hoped that the key factors such as obstacle sensing and motion control will trigger the learning transfer of students. Whether facing the obstacle avoidance car task of the elementary school program-controlled version or the middle school electric-controlled version, they can follow the same logic to solve similar problems.

KEYWORDS

computational thinking, STEM, Scratch, arduino, learning transfer

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

跨學習階段科技實作活動初探-以避障車為例

蘇仕文 [1*]，蔡依帆 [2*]
[1] 基隆市立深美國小，臺灣
[2] 基隆市立銘傳國中，臺灣
alvinmalego@gmail.com，evonne722@gmail.com

摘要
本課程聚焦在國中小科技課程的延續，在 STEM 課程架構下，以避障車為教學主題，設計符應國中小學生起點行為的課程內容。在國小以 Motoduino 板搭配 Scratch 程式控制車體避障動作，引導學生利用運算思維的問題拆解、模式識別、抽象化及演算法，學習避障轉向的機制邏輯。在國中則配合科技領域電與控制學習內容，使用微動開關電路原理，同樣透過運算思維的拆解，使車體達到避障轉向效果。希望藉由障礙感測與動作控制等關鍵因素引發學生的學習遷移（transfer of learning）效果，無論在面對國小程控版或國中電控版的避障車任務，皆能依循相同邏輯以解決問題。

關鍵字
運算思維；STEM；Scratch；Arduino；學習遷移

1. 前言
STEM 課程的設計著重於科技統整，希望學習者除了透過課程習得知識與技能之外，也能對未來的課程內容產生學習遷移（Schwartz & Martin,2004）。基於此目的，本課程的設計概念為一跨學習階段的科技實作活動，在國小階段以資訊課程融入運算思維，幫助學生充分理解避障車的感測與轉向控制機制，在國中階段即以此基礎，進一步思考如何透過電子電路進行感測與控制（圖 1），希望藉由學習遷移產生知識的類化、降低學生認知負荷，並提升學習者的高層次思考與問題解決能力。

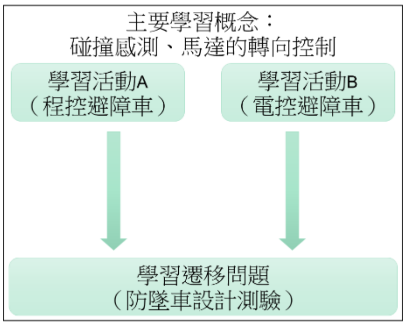


圖 1 避障車學習遷移模式（修改自 Schwartz & Martin,2004）

2. 文獻探討

2.1. 運算思維
運算思維（computationla thinking, CT）被視為適應未來的一項重要能力，在教學上，程式設計是最常用於詮釋運算思維教學的教學主題，而視覺化的程式語言則是運算思維教學中最常使用的工具（Hsu, Chang,& Hung, 2018）。另外，運算思維教學活動主要是藉由實作過程學習問題拆解、模式識別、抽象化及演算法步驟的思維模式（林育慈、吳正己，2016）。

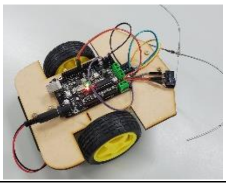
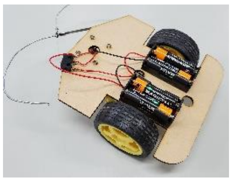
2.2. STEM
美國麻州省於 1998 年開始實施科學、科技、工程與數學（Science, Technology, Engineering, Mathematics; STEM）的整合式教學法，使學生能自行建構知識理論，主動探索問題的解決方法，從而培養學生對教材內容的實際應用能力（STEMTEC，2000）。Becker & Park（2011）之後設分析研究發現，STEM 科際整合教學在小學、中學以至於大學教育階段的學科之間確實具備整合功能，對於各學科知識內容的內化也有顯著助益。

2.3. 學習遷移
學習遷移（transfer of learning）指的是將某一個情境所習得的知識或技能應用到新的情境上。根據 Bransford 及 Schwartz（1999）的觀點，學習遷移是「為未來學習準備（preparation for future learning, PFL）」，意即學習者當下所習得的知識，是為了未來面對新情境的學習而作準備（蔡福興、游光昭、蕭顯勝，2008）。本課程活動中的學習遷移，聚焦於利用運算思維來解析避障車的問題任務，以學習車輛感測與轉向機制的原理概念，並透過後續防墜車的問題情境，觀察學習遷移的現象與成效。

3. 教學設計
本跨學習階段科技實作活動「避障車」，分別在國小高年級與國中八年級實施，課程主要目的在於讓學生使用運算思維四步驟解析問題情境，從而學習碰撞偵測的處理邏輯與轉向的控制機制等兩大概念原則，以期未來在面對類似問題情境時可類化推理。國小及國中兩個版本避障車的課程架構與教學簡述如表 1。

表 1 跨學習階段避障車課程架構

	程控避障車	電控避障車
學習階段	國小高年級	國中八年級
學習目標	利用Scratch以程式控制的方式，從微動開關感測器，控制車輪馬達轉向，製作避障車	以正反向供電電路，配合微動開關三個腳位，控制車輪馬達開閉路與轉向，製作避障車
課程節數	3節課，120分鐘	3節課，120分鐘
控制硬體	Motoduino控制板	電路控制
控制軟體	Scratch2、Transformer2	無

控制邏輯	左側撞擊則右輪反轉，車體向右避 右側撞擊則左輪反轉，車體向左避 前方撞擊則兩輪同時反轉，車體向後避
車體照片	 

活動實施過程中，學生首先針對任務進行問題拆解，拆解出「無碰撞」及「有碰撞」兩種狀況，再進一步辨識出「無碰撞-前進、有碰撞-左右轉或後退」等四種模式，接著歸納出「某側碰撞則另一側車輪反轉」的抽象概念，最後再據此歸納相應的演算法步驟，並發展為程式架構。國小程控避障車的程式積木，以及國中電控避障車的電路圖如圖 2、圖 3 所示。

圖 2 程控避障車程式積木

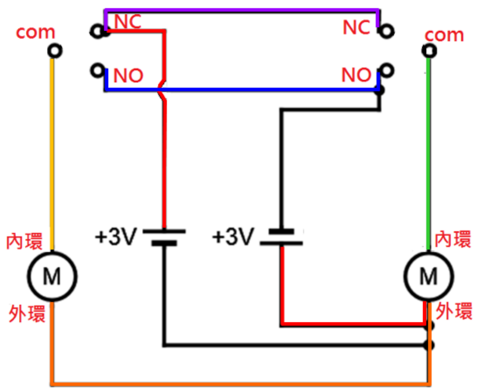


圖 3 電控避障車電路圖

4. 結果與討論

4.1. 程控避障車

國小程控避障車的教學過程中，學生在車體組裝上並無太大困難，但由於杜邦線材顏色並未統一，在接線跟除錯花費稍多時間，教師可透過投影片畫筆註記、實物投影機等科技工具輔助教學。在完成接線後，程式積木的學習相對順暢，這是由於國小高年級學生已具備 Scratch 遊戲設計的基礎，故基本感測器的數據讀取與條件判斷，對學生並無太大難度，皆能順利完成程式控制車，一半以上學生甚至可以自行微調程式積木的各項參數，讓程控車遇到障礙時能有更多元、靈活的反應動作。

4.2. 電控避障車

國中電控避障車的教學過程中，學生對於碰撞與馬達轉向之間的關係皆能快速理解，但如何透過微動開關腳位控制供電方向，使馬達產生反轉避障動作，是最讓學生感到困難的部分，教師可準備與電路圖顏色相應的線材，幫助學生自行除錯。此外，電控避障車需要進行電路的銲接，教師使用實物投影機統一示範電烙鐵的操作，讓具有銲接經驗的學生可以先開始，再針對沒有銲接經驗的學生以小群組方式面對面示範。

5. 結論與建議

本次科技實作活動的目標對象為跨學習階段的學生，但由於教學時間安排因素，國小與國中課程並非針對同一批學生執行，未來若要進行教學實驗，建議可拉長教學實施及資料蒐集時間，針對同一批學生實施本跨學習階段科技課程，方能進行更深入的教學成效分析。但本課程仍提供一種運用科技實作活動產生學習遷移之課程詮釋方式，可做為科技教師實施連貫性課程之參考。另外，在學習遷移成效的評量部分，建議以情境式問題測驗進行（蔡福興、游光昭、蕭顯勝，2008），例如：利用運算思維分析並設計一臺能自動偵測懸空並轉向的防墜車，如此應更可觀察到關於學習遷移的具體表現與證據。

6. 參考文獻

林育慈和吳正已（2016）。運算思維與中小學資訊科技課程。**教育脈動，6**，5-20。

蔡福興、游光昭和蕭顯勝（2008）。從新學習遷移觀點發掘數位遊戲式學習之價值。**課程與教學，11**(4), 237-277。

Becker, K. H., & Park, K. (2011). Integrative Approaches Among Science, Technology, Engineering, and Mathematics (STEM) Subjects on Students' Learning: A Meta-analysis. *Journal of STEM Education: Innovations and Research*, *12*, 23-37.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to Learn and How to Teach Computational Thinking: Suggestions Based on a Review of the Literature. *Computers & Education*, *126*, 296-310.

Schwartz, D. L., & Martin, T. (2004). Inventing to Prepare for Learning: The Hiddenefficiency of Original Student Production in Statistics Instruction. *Cognition and Instruction*, *22*, 129-184.

STEMTEC (2000). *Science, Technology, Engineering, and Mathematics Teacher Education Collaborative (STEMTEC) Proposal*. Retrieved Dec 3, 2019, from http://k12s.phast.umass.edu/~stemtec/about/mission/proposal.html

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Artificial Intelligence and Internet of Things Teaching Activity Design: AIOT Resource Recovery

Hsi-Che TSENG
National Hualien Advanced Industrial Vocational School, Taiwan
jackjean@mst.hlis.hlc.edu.tws

ABSTRACT

This article focuses on easy-to-use and easy-to-access Micro:bit and uses emerging technologies as the theme to design courses suitable for high school students. The courses include basic programming, Bluetooth 4.0 remote control, Internet of things and artificial intelligence. Finally, the artificial intelligence and Internet of things technology is used to complete AIoT resource recovery project. This course expects students to experience Bluetooth 4.0, Internet of Things and artificial intelligence technology, and be able to apply the aforementioned technology to solve problems in life.

KEYWORDS

Micro:bit, Bluetooth 4.0, Internet of Things, artificial intelligence, image recognition

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

人工智慧及物聯網教學活動設計：AIoT 資源回收

曾希哲
國立花蓮高級工業職業學校，台灣
jackjean@mst.hlis.hlc.edu.tw

摘要

本文以容易使用及方便取得的 Micro:bit 為核心，以新興科技為主題，設計適合中學學生的課程，其課程包括程式基礎、藍牙 4.0 遙控、物聯網及人工智慧等課程，最後以人工智慧及物聯網的技術完成 AIoT 資源回收專案，本課程期望學生體驗藍牙 4.0、物聯網及人工智慧的技術，並能應用前述技術解決生活的問題。

關鍵字

Micro:bit；藍牙 4.0；物聯網；人工智慧；影像辨識

1. 前言

新興科技中最受矚目的部分為藍牙（Bluetooth） 4.0、物聯網（Internet of Things, IoT）及人工智慧（Artificial Intelligence, AI），而且在產業界已經開始使用相關技術，創造出許多實用的產品，其中扮演關鍵角色的是人工智慧（AI）結合物聯網（IoT）所形成的 AIoT 新趨勢（蔡明朗，2019）。

目前在教育的現場對於 AIoT 的學習都是大學的技術性的課程，非常缺乏大學以下的體驗式課程，本文提出適合中學階段學習的 AIoT 課程，以 Micro:bit 為核心，程式採用積木型語言，內容包括程式基礎、藍芽 4.0 遙控、物聯網、資料分析及人工智慧，最後以人工智慧及物聯網的技術完成 AIoT 資源回收專案。

2. 文獻探討

機器學習包括下列四個步驟，獲取數據、分析數據、建立模型與預測未來（曲建仲，2018），本文使用深度學習作影像辨識，其為機器學習中最常用於影像辨識，深度學習是讓電腦可以自行分析資料找出「特徵值」，而不是由人類來決定特徵值，就好像電腦可以有「深度」的「學習」一樣。

深度學習的類神經網路需要非常大量的運算，這些運算會導致運算時間過程，本文的深度學習使用 MobileNets（Howard，2017）及 SqueezeNet（Iandola，2016）兩種類神經網路，以縮短計算時間。

3. 教學設計

3.1. 教學對象

本課程的教學對象為高職一年級電子科的學生，相當於高中一年級的學生，學生的性別以男生為主，由於國中階段程式設計沒有全面實施，但是多數學生有接觸過程式設計，國中階段程式設計教學以 Scratch 為主，有學習過程式的學生都已熟悉 Scratch，另外 Micro:bit 在花蓮算是新穎的教具，只有少數的學生有接觸過 Micro:bit。

技職體系學生的特質是喜歡動手做，尤其喜歡開發板及感測模組所組成的系統，對理論科目較不感興趣，數理能力普遍不佳，不喜歡閱讀英文，學生學習實況如圖 1 所示。

圖 1 花蓮高工學生學習實況

3.2. 教學活動設計

本課程設定的學生學習起點為對程式設計稍有概念且未曾學習物聯網及人工智慧，故課程首先以 Micro:bit 為基礎，學習運算思維的概念，再利用 Micro:bit 及程式基礎能力逐步學習藍牙 4.0、物聯網及人工智慧的基本概念及應用，物聯網部分的學習目標為瞭解具獨立功能的物體在網路的運作，人工智慧部分的學習目標為瞭解類神經網路的訓練，最後整合物聯網及人工智慧的概念以 AIoT 資源回收專案為應用實例，讓學生瞭解整個 AIoT 系統的運作。

學習活動設計的主要策略有實作中學習，以學生動手操作為主；其次為思考及實作反覆進行，以問題引導學生思考並以實作完成結果；再者以問題解決為主，以教師事先擬定的問題及學生自訂的專案題目，讓學生探索並解決問題。

本課程以 Google Website 作為學習紀錄，讓學生在學習過程中搭配學習紀錄的撰寫，可讓學生留下學習歷程之外，還可以利用學習紀錄的整理，思考新習得的知識與舊知識的連結，學習紀錄的執行結果可利用影片留下動態的展示，並可讓學生練習口語表達。

3.3. Micro:bit 基礎教學

Micro:bit 基礎教學主要是幫助已經學習的學生做複習，也為完全沒有接觸過的學生建立基礎，並讓學生熟悉程式開發環境 - MakeCode 編輯器，如圖 2 所示，程式部分的學習內容有變數、循序結構、選擇結構及重複結構，外部周邊感測器的學習內容有 5X5 LED 顯示幕、按鈕開關、光線感測器、溫度感測器、重力加速度計、伺服馬達、OLED 顯示模組。

為提高學生的學習動機，基礎學習中會穿插高中男生喜愛的機器人控制或自走車的循跡控制，設計挑戰任

務，讓學生在遊戲的氛圍中，應用 Micro:bit 基礎解決挑戰的任務，並訓練學生解決問題的能力。

圖 2 MakeCode 編輯畫面

3.4. Micro:bit 與藍牙

一般使用手機透過藍牙與 Micro:bit 連線，其工作模式為由手機的藍牙發出掃描信號，取得附近藍牙裝置的地址，經由使用者選取藍牙裝置連線，連線完成後，由 Micro:bit 的藍牙提供服務，讓手機藍牙根據該服務與 Micro:bit 藍牙交換資料。

本課程在手機程式設計的部分採用 App Inventor 2，如圖 3 所示，該程式語言也是積木語言，與 Micro:bit 的積木語言類似，學生可輕易完成程式的編輯，並送到手機上安裝。

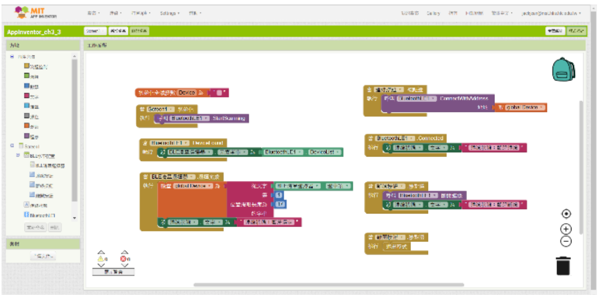


圖 3 App Inventor 2 編輯畫面

藍牙連線部分的學習內容包括手機與 Micro:bit 的連線、手機控制 Micro:bit LED 顯示幕、手機讀取並顯示 Micro:bit 重力加速度計值、手機控制 Micro:bit IO 引腳及手機遙控家電。

3.5. Micro:bit 與物聯網

本課程使用 ELECFREAKS 所生產的 Smart Science IoT Kit 作為物聯網的擴充板，該擴充板內建硬體時鐘（real time clock）及 ESP8266 具有 Wi-Fi 連線功能，可以讓 Micro:bit 紀錄時間，並可連接網際網路，使用物聯網雲。

物聯網部分的學習，希望學生能建立物聯網的基礎概念及網際網路主從式架構的概念，以及物聯網雲的設計及使用，另外更希望學生能學習物聯網所收集資料後，繼續進行資料分析，並從中獲得洞見或預測。

Micro:bit 與物聯網學習的內容為主從式架構的概念、Micro:bit 連線 Wi-Fi 熱點、Micro:bit 連線網路時間伺服器取得正確時間後存入硬體時鐘的應用、Micro:bit 溫度感測值傳送到物聯網雲（ThingSpeak）並將資料下載及物聯網雲收集資料的分析，最後補充介紹 ELECFREAKS 自建物聯網（KidsIoT）的設定與資料上傳、IFTTT 的設定及測試、Micro:bit 透過 IFTTT 傳送 LINE 簡訊。

3.6. 資源回收專案

目前最流行的人工智慧為深度學習，尤其以影像辨識最受到注目，本課程以 App Inventor 2 提供的影像分類網站作為影像辨識的基礎，其過程包括影像收集、影像標籤、訓練模型、驗證模型及下載模型，整個過程可在網站中完成，如圖 4 所示，無須撰寫任何程式，訓練完成的模型可在 App Inventor 2 的編輯器中使用。

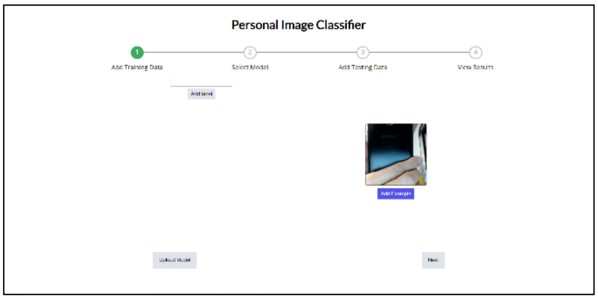


圖 4 App Inventor 2 編輯畫面

以人工智慧及物聯網聯合的 AIoT 資源回收專案學習包括人工智慧的基礎、回收物照片的收集及標籤、影像辨識模型的訓練及驗證、手機實際辨識結果傳送到 Micro:bit 控制外部周邊模組，並將回收物數量傳送至物聯網雲紀錄。

4. 結論

本文使用 Microbit 作為核心，建立適合中學及國小階段的 AIoT 課程，其主要內容為程式基礎、藍牙 4.0 遙控、物聯網、人工智慧及影像辨識，最後完成由人工智慧及物聯網組成的 AIoT 資源回收專案，期望學生透過本課程體驗藍牙 4.0、物聯網及人工智慧的技術，並能應用前述技術解決生活的問題。

5. 參考文獻

蔡明朗（2019）。物聯網在 AI 人工智慧運用。**電信月刊**，**191**，17-18。

曲建仲（2018）。機器是如何學習與進步？人工智慧的核心技術與未來。**科學月刊**，**580**，282-291。

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Retrieved December 1, 2019, from https://arxiv.org/pdf/1704.04861.pdf

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). *SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5MB Model Size*. Retrieved December 1, 2019, from https://arxiv.org/pdf/1602.07360.pdf

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Using Five Steps of Design Thinking to Create "ECO-TAXI"

Mau Fai WONG[1], Yuk Yue Vicky WONG[2*]
[1,2] King's College Old Boys' Association Primary School No.2, Hong Kong
mfwong@kcobaps2.edu.hk, yywong@kcobaps2.edu.hk

ABSTRACT

This paper describes a practical example of how to use the "Five Steps of Design Thinking" to create an application "ECO-TAXI". A group of 3 Primary 5 pupils aged 10-11 wrote a community-based application that solves the traffic problems in Hong Kong. Teachers guided students in the classroom to use the "Five Steps of Design Thinking" to think the steps of the whole process and the components needed by the subsequent program, and then cooperate with MIT App Inventor to write the program, consolidating the concept of computational thinking helps students successfully writing an application. The teaching practice in this paper illustrates the importance of "Five Steps of Design Thinking" in the teaching of computing thinking. The application "ECO-TAXI" will be taken as an example and discussed below.

KEYWORDS

computational thinking, MIT App Inventor, design thinking, ECO-TAXI, coding

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

運算思維教育的教學反思：
運用設計思維五部曲創作程式「ECO-TAXI」

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

摘要

本文闡述一個如何利用以「設計思維五部曲」創作應用程式「ECO-TAXI」的實踐例子，3名年約10-11歲的小學生如何編寫一個以解決社區問題為本的應用程式。老師在課堂中引導學生利用「設計思維五部曲」，思考整個流程的步驟以及其後程式所需之元件，再配合MIT App Inventor編寫程式，鞏固運算思維的概念，幫助學生成功地編寫應用程式。本文的教學實踐闡明「設計思維五部曲」是對運算思維教育的重要性，下文會以應用程式「ECO-TAXI」作為例子，並進行探討。

關鍵字

運算思維；MIT App Inventor；設計思維；ECO-TAXI；編程

1. 前言

教育局課程發展處建議在小學階段引入編程來培養學生的運算思維，希望通過適當設計的學習活動，為學生提供獲取和應用算思維和編程技巧的機會（課程發展議會，2017）。設計思維是一種「以人為本」的方式，以人的需要而出發，尋求各種創新解決問題的方案。當中包括五個漸進的步驟，分別為理解需求、需求定義、創意想像、製作原型及實際測試（Plattner, 2010）。而本文則透過這五個步驟為例子，引導學生創作程式以解決社區問題。

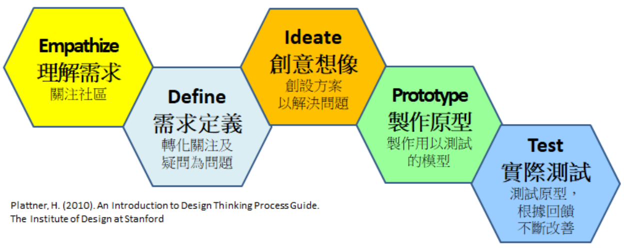


圖 1「設計思維五部曲」模型

2. 「設計思維五部曲」

2.1. 理解需求 Empathize（關注社區）

是次製作程式之目的是以解決社區問題為主，但要解決社區問題首先自然需先知道社區出現甚麼問題。關注社區問題並非一時之間的事情。因此，教師需時常提醒學生關心身邊事物及新聞時事，透過剪報及新聞分享等的方式，加強學生對社會問題的認識。

2.2. 需求定義 Define（轉化關注疑問為問題）

學生發現香港其實有不少的社區問題尚未能解決，而是次關注點則在於交通問題。如本校位於半山區，交通頗為不便，老師及學生或需要乘坐的士上學，因而交通費用較為昂貴。但亦因此引發學生思考是否有方法以減輕交通費用的負擔。減輕交通費用的負擔的方法或有不少，教師引導學生思考或透過蒐集資訊，列出不同問題出現的可能性及其解決方案。最後再重新定義出真正的需求，而是次設計則以如何減輕乘坐的士的費用為主題。

2.3. 創意想像 Ideate（創設方案以解決問題）

當完成需求定義後，學生逐漸了解用戶的需求，並且在需求定義階段分析和合成了所需的觀察結果。教師以腦力激盪法引導學生思考不同創新方法以解決問題，最終提出以人為本的解決方案，以MIT App Inventor創作程式「ECO-TAXI」。學生可隨意將自己的見解提出，亦可延展其他同學的想法，想出無奇不有的結果。

2.4. 製作原型 Prototype（製作以測試的模型）

製作原型能將構想過程轉化為具體的東西（Brown, 2010）。在製作完成品前，學生會在初期設計出較粗糙、簡單的程式原型，他們需要思考程式中會使用的元件，且需要一面測試一面提出解決問題的方案。製作原型在初步階段，需要經不同用戶提出意見，作出修改，調整再調整。目的是為前三個階段找出每個問題的最佳解決方案。基於用戶通過體驗檢測，被接納、改進和重新檢查，或被拒絕而結束。學生們對「ECO-TAXI」內的功能更了解，設法解決內部的局限和目前出現的問題，在互動的過程中，更了解過程中的行為、想法和運用有全面的理解。

圖 2 ECO-TAXI 界面設計原型

2.5. 實際測試 Test（測試原型，回饋不斷改善）

經過製作原型時所得出的不同意見，學生歸納後繼續編寫「ECO-TAXI」的最後階段。在程式中，學生透過運用 GPS 讀取座標，令用戶以位置建立共乘組，其他用戶則能按照上載至 FirebaseDB 的資料，觀看其評分

而選擇共乘組。學生的設計使用了他們認為最佳的解決方案並嚴格測試整個程式。透過這測試階段的結果，往往出現新的或多個問題。學生再次透過運算思維方式、行為和感覺的理解，處理這些需求。例如，用戶提出需要知道車程的距離及其費用，更希望能選擇乘客的性別等等。在這個測試階段不斷反覆產生新問題，為了排除一些問題或錯誤，會運用一些新的解決方案和更改，這些改進亦使學生更能將程式深入地了解用戶所需，調整成更方便使用的應用程式。

圖 3 ECO-TAXI 實際測試後界面設計

3. 反思

在設計整個應用程式收集信息和製作原型，不斷審視每個可見的問題並促成解決方案。測試階段的結果又用程式框架，這個程式更靈活地展現用戶需要。問題與解決方案是可同時發展的（Dorst, 2001）。學生經過了反覆測試，以「設計思維五部曲」作為編寫應能啟發新的見解，過程循環不息。這「設計思維五部曲」往往不是順序進行的，它們也可以平行地發生。它的優點取決於系統化地識別了在設計和任何創新的問題，解決項目中要進行的模式。每個項目都將涉及正在開發的產品的具體性，但每個階段背後的核心仍然存在，在後期階段獲得的知識可以反饋到早期階段。

4. 總結

設計者與用戶之間的理解是創新設計的基礎（Kelley, 2015）。設計思維過程是迭代的，靈活的，側重於設計者和用戶之間的協作，重點是基於真實用戶的想法，感覺和行為重現設計想法。設計者平時需要多留意身邊發生的事物，且要多思考和記錄優秀的意見，積極的去實踐這些想法，在行動中學習，在失敗中受教。

5. 參考文獻

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

Brown, T., & Wyatt, J. (2010). Design Thinking for Social Innovation. *Stanford Social Innovation Review*, *8*, 30-35.

Dorst, K., & Cross, N. (2001). Creativity in the Design Process: Co-evolution of Problem-solution. *Design Studies*, *22*(5), 425-437.

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

Plattner, H. (2010). *An Introduction to Design Thinking Process Guide*. The Institute of Design at Stanford.

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

Put an Unplugged Pedagogy Using an Educational Board Game of Computational Thinking into Practice —Taking the Board Game named Robot City V2 into an Example

Shih Hung YANG[1*], Ting Chia HSU[2]
[1]Taipei Municipal Datong High School, Taiwan
[2]National Taiwan Normal University, Taiwan
yangtwosu@hotmail.com, ckhsu@ntnu.edu.tw

ABSTRACT

This instructional design aims at employing a board game named "Robot City V2" into computational thinking education. The authors designed 8-period courses in the subject of information and technology. The student cultivated their computational thinking literacy in the game-based learning and learned the concept and logics of three basic structures and procedure in structural programming. Based on the feedback after the courses, the students gained high scores after the instruction. Their textual feedback also expresses they were fond of the courses, implying that the instructional achieved the expected objective.

KEYWORDS

computational thinking, Robot City, unplugged, board game

Kong, S.C., Hsu, T.C., Huang, R.H., Looi, C.K., Milrad, M., Shih, J.L., So, H.J., & Dagiene, V. (Eds.). (2020). *Proceedings of International Teacher Forum on International Conference on Computational Thinking Education 2020.* Hong Kong: The Education University of Hong Kong.

使用運算思維教育桌遊實踐不插電教學—以「新機器人蓋城市」桌遊為例

楊士弘 [1*]，許庭嘉 [2]
[1] 臺北市立大同高級中學國中部，臺灣
[2] 國立臺灣師範大學，臺灣
yangtwosu@hotmail.com，ckhsu@ntnu.edu.tw

摘要

本次教學活動旨在利用「新機器人蓋城市」桌遊進行運算思維教學，搭配筆者所設計的八節教學課程，讓學生透過遊戲式學習培養運算思維的能力，並學習結構化程式設計的三大結構與副程式的概念與邏輯。根據課後的回饋顯示，學生在該教學之後拿到很高的分數，他們的文字回饋表達了對課程的喜愛，證明本次教學有達到預期的成效。

關鍵字

運算思維；機器人蓋城市；Robot City；不插電；桌遊

1. 前言

「十二年國民義務教育」在台灣掀起了一陣旋風，其中影響最大的其一為資訊科技課，從原本的重大議題，提升成為每學期皆有一堂課，甚至和生活科技科共同獨立出「科技領域」，可見政府對科技的重視非比尋常。在如此快速的轉變下，由於教學內容較以往產生了非常大的變化，導致在面對課程綱要的轉變時，教師可能因為缺乏相關知識或教具，而影響學生的學習成效。

為此，我們希望能以多元化的教具來輔助教學，特別選定主打運算思維與程式設計的「新機器人蓋城市 Robot City V2」桌遊，並設計漸進式的學習課程。期盼教師能在教學的現場有更多教具和教案可以選擇，除了能在沒有電腦能使用的時候也能上電腦課外，學生也可以透過「玩中學」輕鬆地學習運算思維與程式設計的概念。

2. 文獻探討

2.1. 運算思維

運算思維（Computational Thinking）最初是由周以真（Jeannette M. Wing）教授在 2006 年所提出，希望能夠讓人們了解，運算思維涉及以電腦科學的基本概念來解決問題，設計系統並理解人類問題，是一個每個人都有的基本技能，而不僅僅是電腦科學家（Wing, 2006）。

關於運算思維所包含的元素，學者們陸續提出了多種不同的定義，而目前在台灣較常被用到的是林育慈和吳正己（2016）整理多篇文獻後所提出的版本，認為運算思維可分為抽象化（abstraction）、問題分解（problem decomposition）、模式化與模擬（modeling and simulation）及演算法思維（algorithmic thinking）四大項。

2.2. 桌遊教學

在 K-12 的教育中通常會使用多種的教學策略來教運算思維，其中最常見的便是使用電腦，但通常也會使用一些不插電的教學方式，像是桌遊、紙牌遊戲、或一些動覺類的活動，來教授像演算法等概念的運算思維（Kalelioglu et al., 2016）。

陳介宇和王沐嵐（2017）整理了臺灣近 15 年來的桌遊相關研究，發現 2002 年至 2012 年間僅有 21 篇，而 2013 至 2016 年間竟出現了 116 篇。而在遊戲的內容上，教育類又佔了最多數（59.9%），由此更可證明桌遊在教學上的應用逐漸受到重視。

3. 教學工具

本次教學使用「新機器人蓋城市 Robot City V2」運算思維桌遊做為教學工具。該套桌遊為國立台灣師範大學許庭嘉教授率研究團隊，與國民中學數個領域的教師，經過長時間討論開發完成。桌遊的架構以運算思維為核心，透過遊戲的方式培養學生問題解決的能力，並從中習得程式設計的三大結構：循序結構、選擇結構、重複結構，以及基礎的副程式概念。

為方便教育現場教師進行教學，桌遊的開發方向特別加強了遊戲的自由度，不僅總牌數的添加可因教學的內容而改變，地圖的大小、配製及人數的多寡，甚至遊戲的規則皆可因教學現場需要而做調整。另外，遊戲也支援「陣營對戰」，讓玩家不僅在不同組間得以互相競爭，在組內更能培養團隊合作的精神，增加與同儕的互動，進而達到「結隊程式設計」的訓練。

圖 1 學生組合多張控制卡來控制機器人移動

圖 2 玩家必須蒐集資源完成蓋房子的任務以賺取分數

圖 3 小組對抗賽戰況激烈

圖 4 結隊程式設計須仰賴適當討論及換牌

4. 教學設計

4.1. 教學對象

本次教學對象為台北市某國中七年級學生，男生 67 人，女生 63 人，共 130 人。大部分的同學有在小學學過 Scratch 程式設計，但對其精熟度卻不甚理想。除此之外，學生們在國中階段並未接觸到任何程式設計課程。

4.2. 教學流程

本次教學時間為 8 節課，共 400 分鐘，流程的安排如表 1，並於所有課程結束後利用額外的課程時間進行測驗與問卷回饋：

表 1 教學內容與上課結束分配

教學內容	上課節數	課堂分鐘數
規則介紹、循序結構	1 節課	50 分鐘
選擇結構	1 節課	50 分鐘
重複結構	1 節課	50 分鐘
結隊程式設計	1 節課	50 分鐘
副程式	2 節課	100 分鐘
流程圖	1 節課	50 分鐘
總複習	1 節課	50 分鐘

4.3. 教學評量

教學評量使用筆者針對此款桌遊搭配流程圖設計的評量，以並在教學結束後以網路問卷詢問學生對參與課程的感想與啟發。教學評量範例如圖 5。

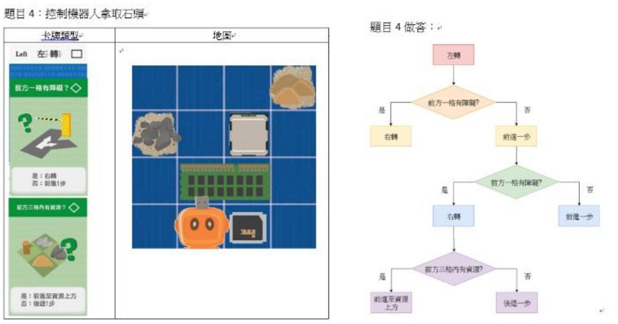


圖 5 教學評量範例

5. 結果與討論

5.1. 學習評量分析

130 位學生中，有 74 位學生在課後的測驗中得到了滿分，低於 60 分的同學則有 9 位。分析學生們在測驗中寫錯的題目，發現最常出現錯誤的是選擇結構綜合應用的流程圖繪製。

5.2. 學習回饋分析

分析學生在表單中所填寫的回饋，得知多數的學生對本次的課程皆抱持著正向肯定的態度，認為本套課程可以幫助他們培養邏輯思考、程式設計以及同儕合作的默契，在玩樂中不失教育性。僅有極少數的同學表示規則太難不太理解，或是太無聊等負面訊息。

經過這次的體驗讓我深刻的知道一個程式碼是需要很久的時間或是許多人組成的。

一樣的排組可以玩出不一樣的路徑，每個牌都超強，而且對程式設計的思考邏輯可以有更多不同的方向去思考他。

我發現程式設計很無聊，但玩桌遊就比較不會，而且透過桌遊也可以學習程式設計，總之我覺得這機器人蓋程式遊戲不錯。

原來用程式設計的理念跟流程圖也可以設計出這麼有趣的遊戲！

訓練程式設計的概念，也可以訓練腦力和思考能力，也和隊友訓練團隊合作、互相討論的精神!

我覺得有點無聊，因為我還搞不懂暫存卡是幹嘛的，我們這組不知道要怎麼補卡，老師重複講了好多次，還是有點搞不清楚。

6. 結論與建議

6.1. 結論

本次教學旨在利用「不插電」的桌遊來改變傳統的程式設計教學，期盼達到良好的學習成效。在課程結束後分析學生的成績和回饋得知，學生對於桌遊學習程式設計概念的模式反應良好，不僅在測驗中拿到很高的分數，在文字回饋中也看出對課程的喜愛，證明本次教學有達到預期的成效。

6.2. 建議

遊戲的卡牌與機制對部分的同學來說略為複雜，因此除了要把規則講清楚外，更需要發下每組一張簡化版的規則書，並將對於遊戲規則較為清楚的同學平均分配到各組，以免發生學生因不理解規則導致無法融入課程，而影響學習成效。

另外針對課程的後續發展，建議再加深與後續電腦實作課程之間的結合，讓「不插電」與「插電」的概念得以相輔相成。

7. 參考文獻

林育慈和吳正已（2016）。運算思維與中小學資訊科技課程。**教育脈動，6**。2019 年 12 月 16 日，取自 https://pulse.naer.edu.tw/Home/Content/02287aac-dc26-4ad4-b87e-2881e942dc16?insId=40977899-d342-4f01-94a7-66d446c9d3bb

陳介宇、王沐嵐（2017）。臺灣桌上遊戲研究與文獻之回顧分析。2019 年 12 月 16 日，取自 https://sites.google.com/site/taiwanbgstudy/home

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, *4*(3), 583-596.

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

CTE2020

Coo/Think @JC
賽馬會運算思維教育
Inspiring digital creativity 啟發數碼創意

URL:
www.eduhk.hk/cte2020

Email:
cte2020@eduhk.hk

Created and Funded by

香港賽馬會慈善信託基金
The Hong Kong Jockey Club Charities Trust
同心同步同進 RIDING HIGH TOGETHER

Co-created by

香港教育大學
The Education University of Hong Kong

Massachusetts Institute of Technology

香港城市大學
City University of Hong Kong