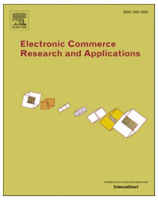
Electronic Commerce Research and Applications 44 (2020) 101011

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/elerap



Enhancing Computational Thinking Capability of Preschool Children by Game-based Smart Toys

Szu-Yin Lin[a], Shih-Yi Chien[b,*], Chia-Lin Hsiao[c], Chih-Hsien Hsia[a], Kuo-Ming Chao[d]

[a] *Department of Computer Science and Information Engineering, National Ilan University, Yilan County, Taiwan*
[b] *Department of Management Information Systems, National Chengchi University, Taipei, Taiwan*
[c] *Department of Information Management, Chung Yuan Christian University, Taoyuan City, Taiwan*
[d] *School of Computing, Electronics and Mathematics, Coventry University, Coventry, UK*

ARTICLE INFO

Keywords:
Cognitive development of children
Computational thinking
Game-based learning
Tangible user interface
Smart toys

ABSTRACT

Computational thinking has become an important issue in the field of education. Because preschool and kindergarten learners are capable of exercising their cognitive abilities to resolve basic computational logic, this demographic has raised significant interest in studying their learning intentions and behaviors. However, prior research fails to examine the effects of teaching computational logic to kindergarten children. Therefore, this study aims to investigate the influences of teaching approaches in guiding preschool children to learn computational logic and programming concepts to enhance their problem-solving skills as well as computational thinking abilities. A novel teaching framework is designed to develop the learner's cognitive abilities, which adopts the smart toy game-based learning approach along with a tangible user interface (TUI) to enhance children's learning performance and interests. The proposed teaching approach integrates the game-based learning concepts into the TUI system, where the learning processes allow the learners to effectively practice the conceptual knowledge and efficiently advance their problem-solving skills. The results suggest using the developed game-based TUI system can increase preschool children's learning behaviors as well as enhance their learning interests and computational thinking abilities.

1. Introduction

The advanced achievements in computer science have altered the manner in how humans use and learn information technology. Nowadays, science and technology have become omnipresent. People not only browse science and technology knowledge on the World Wide Web, but also access the relevant information through a variety of mobile electronics and smart toy applications. The next generation is also expected to advance from browsing users to inventors and creators who will intensify the competitiveness of global powers. There has been a worldwide initiative to teach programming logic to children for enhancing their computational thinking abilities and problem solving skills. The concept of computational thinking proposed by Wing (2006) has been extensively discussed across various fields and has gradually become an important part of education. Computational thinking is a process of comprehending and solving problems, which is not limited to a single discipline or field. It has become indispensable in analyzing numerous types of real-world computational problems, as well as developing innovative knowledge in different domains (Furber, 2012). For example, the STEM (science, technology, engineering, and mathematics) field aims to enhance learners' logical thinking abilities and problem-solving skills through the practice of programming logic.

Computational thinking has been suggested as a required skill that every individual should acquire (Wing, 2006). Brennan and Resnick (2012) concluded learning programming languages can improve learners' computational thinking abilities. Computational thinking processes require learners to approach problems by developing logical solutions and practicing concise communication to explain their reasoning activities (Lye and Koh, 2014; Shafto, 1986). Lindberg et al. (2019) also point out an increased global trend in learning computational thinking in education, which reveals the importance of studying programming lessons for preschool children. Three types of computational thinking are involved in developing a program: computational concepts, computational practices, and computational perspectives (Brennan and Resnick, 2012), where each component addresses a different learning objective and can vary in degree of difficulty.

* Corresponding author.
E-mail address: sychien@nccu.edu.tw (S.-Y. Chien).

https://doi.org/10.1016/j.elerap.2020.101011
Received 8 March 2019; Received in revised form 21 August 2020; Accepted 23 September 2020
Available online 28 September 2020

Additionally, the cognitive development theory from Piaget (1976) and the scaffolding theory from Vygotsky which summarized by Berk and Winsler (1995) suggest that preschool children require more assistances for converting conceptual ideas into tangible entities. To decrease learning barriers, prior studies have developed visual programming language platforms to cater different age learners and skill levels, as well as provide diverse teaching methods (Smith et al., 2019) and provide an opportunity for developing computational thinking to preschool students (Ching et al., 2018). With the graphical design and interface, visual programming language (VPL), compared with traditional text-based programs, is substantially easier to learn programming skills and can enhance the students' learning interests (Grover and Pea, 2013). For example, the Lego EV3 system uses the VPL platform with programmable bricks to provide a tangible user interface (TUI) to the learners in developing the (robotic) system applications. In addition, through the use of a VPL system, learners can experience the rewarding process of problem solving while encountering program bugs, and solving the task at hand. This method has been known as game-based learning, which combines entertainment and problem solving elements into the learning processes. Prior research conclude game-based learning can effectively enhance students' learning interests and improve overall learning performance in high school education (Papastergiou, 2009).

Although great efforts have been devoted to developing hardware and software tools for students to learn programming skills and improve their computational thinking abilities, most current developments focus primarily on junior and senior high school students rather than preschool students (Lye and Koh, 2014). In other words, most resources are currently designed for experienced users and examine how the developed tools can further advance these users' computational knowledge. As preschool children have limited cognitive abilities and may require more assistance to transform abstract concepts into real-world entities, the game-based learning approach with TUI applications provide a better way to motivate preschoolers to actively participate in the learning materials as well as increase their computational thinking abilities. However, there lacks sufficient resources providing basic concepts for beginner learners, such as the research gaps identified in Zhang and Nouri's review study (2019). As the preschool children's learning capability is limited by their cognitive level (Koslowski, 1980), the conventional syntax-based programming languages can inject significant difficulties (Chien et al., 2018) and is therefore inappropriate for these leaners. This implies that the traditional teaching tools may be insufficient for preschool children. Due to children's limited cognitive ability, teaching program logic and determining a suitable curricular for preschool students remain a challenge.

The objective of this study aims to create an appropriate teaching framework for enhancing the preschoolers' computational thinking abilities, learning interests, and learning achievements. The game-based learning method along with a TUI is created to guide the preschool learners to study computational logic and programming concepts. This goal has the following sub-goals:

(1) applying a game-based learning method to strengthen a learner's computational concepts, computational practices, and computational perspectives
(2) creating a TUI to engage a preschoolers' learning interests as well as encourage them to participate in the learning activities

Two rounds of user studies were conducted to collect the empirical data to examine the research questions. The results show that the developed teaching approach (game-based learning with the TUI system) can effectively increase preschool learners' cognitive thinking performance, as well as improve their learning behaviors. In addition, the results also reveal that the participated preschoolers' prior learning experience can not only influence their study performance but also their learning behaviors and interests. Our study presents the following contributions: first, to the best of our knowledge, this is one of the most limited research that integrates game-based learning concepts into TUI systems to enhance preschoolers' computational thinking abilities. Second, the proposed teaching framework has been empirically validated and shown promising results in improving students' learning abilities, which can benefit the preschool educators and instructors in delivering the conceptual course materials.

The article is organized as follows: the Section 2 reviews the factors affecting computational thinking; Section 3 introduces the theoretical model used in the study; Section 4 includes the details of methods; Section 5 discusses the experimental results; Section 6 concludes the study findings.

2. Literature reviews

This study aims to examine the effects of applying game-based learning methods to design course materials, as well as improve preschool children's computational thinking skills. To provide better understandings of the research questions, the following literature reviews include computational thinking, tangible user interface, smart toys, learning cycle teaching strategies, and digital game learning methods.

2.1. Computational thinking

Computational thinking is a process in which a person develops a series of thinking strategies to approach a problem (Wing, 2006). The idea of educating children's computational thinking first appeared in early 1960 (Rees et al., 2016). Due to the lack of evidence to support its effectiveness, little attention was paid to construct the relevant educational tools to enhance learners' computational thinking abilities and problem solving skills (Lye and Koh, 2014). However, with the advancement of technology, computational thinking education has become an important topic in recent years (Lindberg et al., 2019). For example, the development of VPL (such as Scratch) allows students to learn the programming skills and required knowledge in an effective and efficient manner. In other words, the reduced learning cost helps students reduce their cognitive loads during the learning processes and makes it easier to develop computational thinking abilities through the learning procedures (Shafto, 1986).

Wing (2006) concluded "*computational thinking is for everyone and everyone must have the skills*." Computational thinking ability has been suggested to significantly correlate with STEM (science, technology, engineering, and mathematics) education (Khine, 2018). Since STEM is highly related to the information technology fields, this suggests that computational thinking abilities should be considered as a critical subject and take root in information science education. Various countries have devoted considerable efforts to promote computational thinking education in academia, industry and government units. For instance, in the United Kingdom, the programming-related courses are regulated as a compulsory course in the secondary school's syllabus. Germany, Netherlands and Japan have also developed information technology capability indicators to strengthen the course materials and ensure the students have sufficient computational thinking abilities.

The computer science teachers association (CSTA) in the United States established a core competency standard for computer science education, which develops a framework for K-12 CS-related education (Seehorn et al., 2011). The framework divides the CS courses into three levels (Fig. 1). To better examine the preschool students' computational thinking abilities, our study focuses on the fundamental level (i.e., level-1). The first level is for students from kindergarten to sixth grade, where the education guideline concentrates on the students' understanding of the basic CS concepts. This level aims to develop students' creativity, active learning abilities, and explored capabilities to encourage students to apply computational thinking ability into basic or daily science and technology matters.

Computational thinking ability can be categorized into three

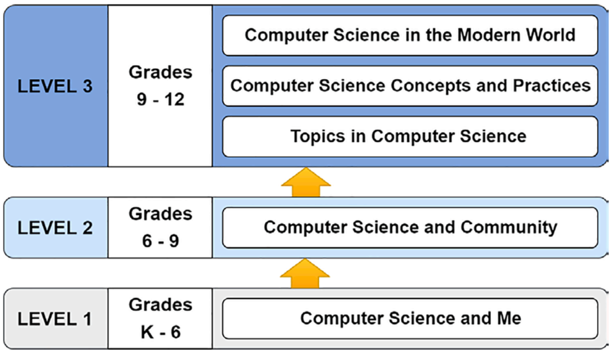

LEVEL 3
Grades 9 - 12
Computer Science in the Modern World
Computer Science Concepts and Practices
Topics in Computer Science
LEVEL 2
Grades 6 - 9
Computer Science and Community
LEVEL 1
Grades K - 6
Computer Science and Me

Fig. 1. K-12 computer science core indicators (Seehorn et al., 2011).

dimensions (Brennan and Resnick, 2012; Lye and Koh, 2014), including computational concepts, computational practices, and computational perspectives. This study adopted these three dimensions as the basis to design the course materials and learning outcomes, which allowed the learners to practice logical thinking as well as comprehend the computational knowledges. Therefore, through the learning processes, a learner first studied the fundamental computational concepts and then resolved the advanced programming questions in the computational practices phase. Once accomplished, the user should be able to fully understand the relationships in a given programming context. Detailed descriptions of the three dimensions are summarized in Table 1.

A considerable amount of effort has been devoted to examine learners' computational thinking abilities. For instance, Yadav et al.'s (2017) and Denning's (2017) studies developed the teaching guidelines to introduce the computational thinking related materials to the learners. In addition, Shute et al. (2017) proposed a model to assess students' computational learning outcomes; whereas Sullivan and Bers (2018) and Pérez-Marín et al. (2020) focused mainly on the preschool children's computational thinking learning performance. These studies suggested that computational thinking has become the fundamental and critical skills in this digital age.

2.2. Tangible user interface

Tangible user interface (TUI) is a user interface in which people can interact with digital information through the physical components. The purpose of TUI is to enhance collaboration, learning and design ability by providing a physical form for digital information, thereby enhancing human ability to learn and manipulate physical objects and digital information (Ishii, 2008). In addition, Ishii's group, one of the pioneers of the TUI, develops a TUI application called Tangible bit (Ishii, 2008), which provides a physical form of digital information that allows people to operate digital signals and connect the physical objects with the digital data. A simple example of a TUI is a computer mouse, an interface that allows people to interact with digital information through physical objects. One of the notable developments, Augmented Urban Planning Workbench (Ishii et al., 2002), simulates real world phenomena (e.g., airflow, shadows, and reflections), constructs landscapes with real world materials (e.g., clay or sand), and integrates these properties into a three-dimensional space. Additionally, a variety of smart toys can be seen as an extension of TUI computational thinking education. For example, BRICKO, an educational computing thinking robot, supports tangible interface, social and entertainment interactions while educating children (Pedersen et al., 2018).

Table 1
Dimensions of computational thinking ability (Lye and Koh, 2014).

Dimension	Description	Examples
Computational concepts	The concepts used in programming processes	Variables The loops
Computational practices	The methods of solving programming problems	Being incremental and iterative Testing and debugging Reusing and remixing Abstracting and modularizing
Computational perspectives	The student's understanding of the relationships between oneself and others in a technological context	Expressing and questioning the use of technology

Because most tangible computing environments are too complex for young children, prior research proposed Tangicons (the non-electronic physical programming cubes for beginner learners) to incorporate tangible bricks into physics games, allowing children to learn programming in a fun way (Scharf et al., 2008). Researchers who have compared the ease of understanding between graphics and tangible user interfaces, and found that tangible and graphical systems are equally easy to understand. However, through a tangible interface, users are more likely to try and actively participate (Horn et al., 2009). Examples of TUI that are actually available for children to work with include puzzle pieces; their inherent physical syntax of connectable elements provide a powerful and expressive metaphor for building tangible systems (Oh et al., 2013). Another example is TanProRobot 2.0, which consists of three parts: a tangible programming block, a robotic car, and several manipulators. The child can program the robot car by performing programming blocks to perform certain operations and interact with the car through manipulation. It can help children learn programming concepts and get a glimpse of event handling concepts (Wang et al., 2016). Other examples of the latest tangible systems include MakerWear, a tangible interface for wearable design that allows children to apply computational thinking (Kazemitabaar et al., 2017). MakerWear is a great example of how new technology settings can fundamentally change the interaction of electronic media and behavior, as well as perception, physical manipulation and overall social activities (Bergsmark and Fernaeus, 2016).

This study will let children learn the concept of computational thinking ability through TUI interface and smart toys, such as color answer cards and Arduino toy car robots. The review of smart toys will be introduced in the next section.

2.3. Smart toys

A smart toy is a toy that can respond to user feedback and change its behavior according to environmental stimuli. It can act according to the design model, and usually adapts to the player's abilities. Smart toys typically have electronics consisting of microprocessors or microcontrollers, memory storage devices, and numerous forms of input–output devices (Boss et al., 2001). While computers can represent children in the medium of social and intellectual development, some researchers believe that using computers before the age of 7 reduces important developmental tasks and other types of learning (Healy, 2000). Therefore, some scholars have proposed an interactive interface with smart toy that children can use alone or in combination with a computer, combining current popular mobile device learning, tangible interfaces, and a variety of home technologies (Plowman and Luckin, 2004). If a toy contains only a unilateral action or a single message-transmitting display but has little ability to adapt user's intentions, it should not be classified as a smart toy. In other words, the distinguishing factor a smart toy has is the ability to integrate machine/system applications into gaming experience to create human-like intelligence.

Additionally, most of today's smart toys have a networked mechanism, requiring a demand for people to pay higher attention to privacy rights. Because the definition of privacy may not be fully understood by children in the early childhood stage, children are more likely to unknowingly reveal private information (Rafferty et al., 2017). Therefore, related studies should refer to and comply with the privacy protection requirements of Hung's study (2016), including the concept that children may not understand privacy and children will disclose as much information to smart toys as they can trust. For example, Delprino et al. (2018) develop ABBOT that combines intelligent tangible objects with outdoor sports. ABBOT's tangible objects help children capture images of the elements they find interesting in the physical environment, inspiring a greater interest to explore the outdoor environment. Evidently, several smart toy robots have recently been invented for the purpose of educating and engaging. Additionally, Relkin and Umaschi Bers (2019) designed TACTIC-KIBO, a tangible learning tool, to encourage preschool children in activity learning programming concepts and skills. Sullivan et al. (2017) concluded the use of KIBO robot kit can effectively advance learners' computational thinking abilities as well as increasing their learning interests. Evidently, several smart toy robots have recently been invented for the purpose of educating and engaging. Table 2 summarizes the smart toys with its educational functions.

In this study, the mBot Arduino robot is selected as an intermediary TUI interface that can be used in conjunction with teaching. Additionally, this study also required self-designed answer color cards, a series of teaching courses, and our own image recognition program written on the mBot robot.

2.4. Learning cycle teaching strategies

The learning cycle is a concept in which people can learn different knowledge or skills through experiences. Usually learning cycles have multiple phases, and the last phase can follow the first phase to be a complete learning cycle. The 5E learning cycle was developed by Biological Sciences Curriculum Study, and specifically for purpose of teaching science. This model describes a teaching sequence that can be used for entire programs, specific units, and individual lessons (Duran and Duran, 2004; Bybee et al., 2006). Table 3 shows each phase of the 5E learning cycle.

In this study, we apply the 5E learning cycle as the basis for the teaching methodology and experiments. In addition, we also combine the methods used by Garris et al.'s (2002) Input-Process-Outcome Game Model to design the teaching concept. The detailed game-based TUI for computational thinking methodology will be introduced in Section 3.

Table 2
Smart toy robots with educational function.

Name	Educational Functions
KIBO	KIBO is the screen-free robot kit for kids that lets 4–7 year-olds create, design, decorate and bring their own robot to life. KIBO is an easy and fun way to bring robotics and coding to young learners and spark their interest in STEAM. (KinderLab Robotics, 2014)
Dash & Dot	A robot that can walk and identify obstacles. It can also respond, sing or dance. This robot allows children aged 5–12 to learn and practice programming skills. (Wonder Workshop, 2015)
mBot	mBot is an educational robot for beginners that makes teaching and learning robot programming simple and fun. mBot also aids in the development of logical thinking and design skills. (Makeblock, 2015)
Cubetto	The Cubetto Playset is a Montessori inspired coding toy that allows children aged 3–6 to program a friendly wooden robot. The toy is powered by tangible programming language made of colorful wooden blocks (Primo Toys, 2016)
Codey Rocky	A robot that includes a combination of software and hardware that allows children to learn programming concepts through play and creation. This robot uses mBlock as its programming language. (Makeblock, 2017)
KUBO	KUBO aims to teach younger children the basics of programming and computational thinking in a simple and intuitive way. This game also guides children through a series of challenges using small robots, square cards and maps. (KUBO Education, 2017)
ROBOPAL	A programmable learning robot that uses magnetic coding blocks as its programming language. It also encourages children to have fun while learning computational thinking. (Kickstarter, 2017)

Table 3
5E Learning cycle.

Stage	Phase	Introduction
1	Engagement	The teaching model of this period completely imitate the learning task. Activities should be able to connect past and present learning experiences and focus on the thinking process of students' learning outcomes at present activities. This phase should engage students to explore the concepts, processes, and techniques of the mind.
2	Exploration	This phase of the teaching model provides a common empirical basis for students to identify and develop current concepts, processes and techniques. During this period, students actively explored their environment and manipulated teaching materials and teaching aids.
3	Explanation	In this period, students learn to communicate their observations. Students also learn to interpret their observations and draw a meaningful conclusion. Teachers can simultaneously introduce formal definitions explaining concepts, processes or behaviors.
4	Elaboration	Elaboration teaching mode challenges students to gain a deeper understanding of the material through application. Through additional investigation or creating presentations, students ensure they have a firm understanding on the content.
5	Evaluation	This period allows students to self-assess and reflect, ensuring they fully comprehend the material. This also provides teachers with an opportunity to assess the progress of students in achieving educational goals.

2.5. Digital Game-based learning

Digital game-based learning is learning through a technological gaming platform. In the game, learners achieve a sense of accomplishment through solving and overcoming challenges. Digital game-based learning should take into account both gameplay and education, and achieve the goal of entertaining and learning. Prensky (2003) pointed out that the features of digital game-based learning include the following 12 features. The various features mentioned below will be applied in this study.

(1) Entertaining: the game is fun and engaging for the learner.
(2) Gameplay: provides a form of play. Motivates learners in a fun and appealing way.
(3) Regularity: make the content of the game structured. It will make it easier for learners to organize gameplay, and interact in the game.
(4) Goals: the specific tasks in the game can clearly guide the users to learning through play.
(5) Interaction: the game interface is user-friendly and intuitive.
(6) Adaptability: the game design can vary in degree of difficulty according to the learner's level.
(7) Outcomes and feedback: provide opportunities for users to learn.
(8) Sense of victory: learners achieve a sense of accomplishment through overcoming barriers in the game.
(9) Conflict: competing and challenging: Challenges users with barriers and tasks for them to face and overcome.
(10) Problem Solving: design questions in the context of the game to inspire learner creativity.
(11) Social interaction: learners build a relationship with other game players, creating a sense of community.
(12) Representation and story: the learner is interested in the storyline and game tasks, and is emotionally invested in the game.

3. Theoretical guidelines and research hypotheses

This study aims to improve how program logic is taught to preschool children, bettering the students' computational thinking capabilities. Game-based learning is used as the strategy, teaching through hands-on practice and tangible interaction. The Input–Process–Outcome game model is also adopted to assist preschool children in acquiring computational thinking.

3.1. Design concepts

Based on constructivist learning, this study plans to let preschoolers learn through more specific means such as interactive interface and hands-on operation of real objects. Programming is taught through games to improve computational thinking. A review of the literature on digital game-based learning and children game-based teaching reveals that games could improve the study interest and outcome of preschool children (Hogle, 1996). The main design concept includes three aspects: (1) curricula designed on the three dimensions of computational thinking proposed by Brennan and Resnick (2012), i.e., computational concepts, computational practices, and computational perspectives; (2) framework constructed on Input–Process–Outcome game model; (3) appropriate tangible user interface (TUI) adopted or designed for game-based teaching to aid preschool children in their acquisition of computational thinking and to function as their instructional scaffold.

3.2. System planning

This section discusses the curricular design and game design. The curricula primarily follow the three dimensions of computational thinking by Brennan and Resnick (2012): computational concepts, computational practices, and computational perspectives, to enable preschool children to grasp the logic of programming. Additionally, contents for different study-levels are designed based on the 5E learning cycle. The concept of a learning cycle came from Piaget's theory of cognitive development, which are teaching strategies (Llewellyn, 2005). The proposed curricular design scheme is as shown in Fig. 2:

The curricula are framed around computational concepts, where several fundamental concepts are applied to the TUI designs to match the participants' knowledge levels, including sequences, loops, parallelism, events, conditionals, operators, and data concepts. As the participant's abilities of cognitive perceptions and motor movements greatly contribute to the computational thinking performance in this study, we therefore used the ***perception*** and ***behavior*** mapping approaches in the TUI designs (Table 4). The perception mapping includes visual perception and tactile perception. The visual perception is based on the properties of appearance, position, location, shape, size, distance, and color; on the other hand, the tactile perception includes sliding, clicking, and touching senses. The behavior mapping contains gross-motor (e.g., jumping, stepping, turning, and flicking) and fine-motor movements (e.g., pinching, grabbing, gripping, and twisting). For example, to complete a *sequence* concept, the participant has to first use color and click/slide as the visual and tactile senses in TUI perception, and then integrate these perception concepts to jump/step/turn/flick in the gross-motor movements and twist in the fine-motor movements.

The TUI is embedded into game-based learning by designing games that suit the tangible interface. The actual teaching content is developed based on the 5E learning cycle according to the cognitive development theory for children; moreover, the content divides the curricula into several steps: engage, explore, explain, elaborate/extend, and evaluate.

Table 4
TUI Perception and behavior mapping for computational thinking.

TUI mapping concepts	Corresponding abilities	Corresponding skills
Perception mapping	*Visual*	appearance/position/location/shape/size/distance/color
	Tactile	click/slide/touch
Behavior mapping	*Gross motor movement*	jump/step/turn/flick
	Fine motor movement	grab/grip/twist/slide
Computational thinking fundamental concepts	TUI *perception* mapping	TUI *behavior* mapping
Sequences	*vision:* color *tactile sense*: click/slide	*gross motor movement:* jump/step/turn/flick *fine motor movement:* twist
Parallelism	*vision:* color/location *tactile sense*: click	
Events	*vision:* color/location/appearance *tactile sense*: click	
Loops	*vision:* color/shape *tactile sense:* click/slide	*gross motor movement:* jump/step/turn/flick *fine motor movement:* grab/grip
Conditionals	*vision:* color/location/appearance *tactile sense:* click	*gross motor movement:* jump/step/turn/flick *fine motor movement:* twist/slide
Operators	*vision:* location *tactile sense:* click	
Data	*vision:* color/position *tactile sense:* click	

Computational practices and computational perspectives refer to the problems encountered and solved by the learners during game-based learning. Their elements could be used for game design, such as increasing the difficulty of the game, adding passing criteria and restrictions, or breaking a problem down into several smaller ones as game checkpoints, i.e., creating individual checkpoints or a series of related checkpoints. The process includes tasks to be completed, and preschool children are guided to identify the problems and identify solutions. Teachers, students, and teams should communicate and ask/answer questions to accomplish the goal of the game, encouraging each person to speak out his/her ideas. The game is designed with the Input–Process–Outcome game model and according to the game features put forward by Prensky (2003). The game design scheme is shown in Fig. 3.

The three stages of input, process, and output are discussed as follows:

1. Input:
   This stage includes the curricula and game features. The curricula are developed around the computational thinking framework in curricular design according to the steps of the 5E learning cycle and

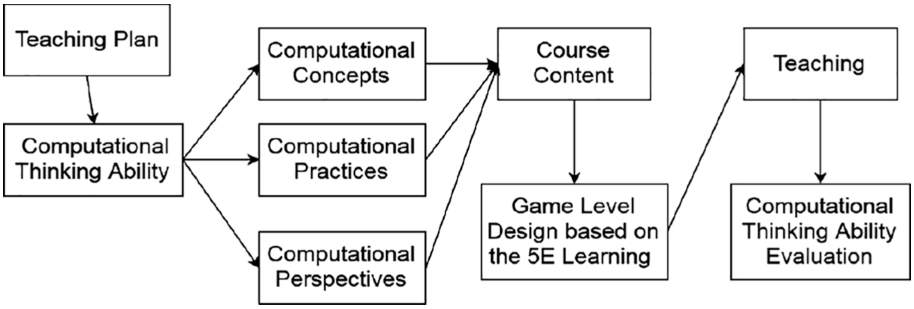

Teaching Plan
Computational Thinking Ability
Computational Concepts
Computational Practices
Computational Perspectives
Course Content
Game Level Design based on the 5E Learning
Teaching
Computational Thinking Ability Evaluation

Fig. 2. Schematic illustration of the curricular design.

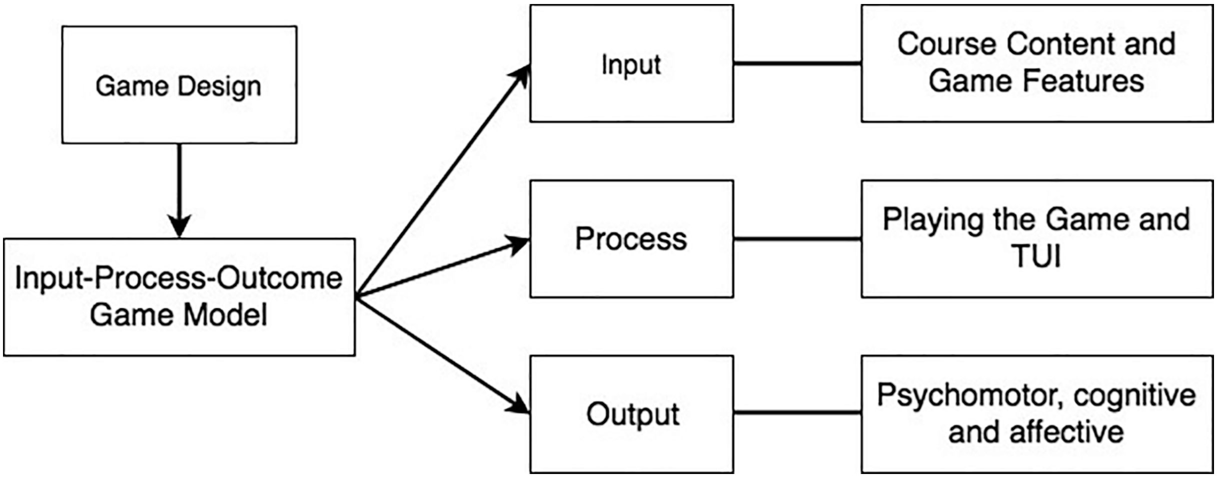

Game Design
Input-Process-Outcome Game Model
Input
Process
Output
Course Content and Game Features
Playing the Game and TUI
Psychomotor, cognitive and affective

Fig. 3. Schematic illustration of the game design.

with tangible user interface as the instructional scaffold for preschool children, as shown in Fig. 4.

2. Game cycle:

In this step, a game cycle is formed by user judgment, user action, and system feedback. The user's judgment, as indicated in the Input–Process–Outcome game model of Garris et al. (2002), constitutes interest, enjoyment, task involvement, and confidence. The preferences and tastes of preschool children are demonstrated through games; therefore, the game-based learning of this study takes into account the learning characteristics of these children and adopts the experiential learning by John Dewey to develop an instructional scaffold, used with an appropriately interactive tangible user-interface, such that the learning interest of preschool children is kindled during the course of the game. As mentioned by Garris et al. (2002), user action is realized in the design of course content and game features, which include fantasy, rules, targets, sensory stimulation, challenge, mystery, and controllability. The system feedback affects the actions of the users through "learning by doing." The feedback received from students is used to influence user judgments in the subsequent rounds. As a result, students exhibit higher motivation to learn, and the memory and training of preschool children are reinforced in this iterative process to achieve the learning goal.

3. Outcome:

The expected results at the end of learning program-logic can be observed in three aspects: psychomotor, cognitive, and affective. These results should be achieved through the teaching method proposed in this study. Psychomotor learning output pertains to computational practices, which is the application of skills learnt during programming to the solving of problems. Cognitive learning output is related to computational concepts. Affective learning output relates to computational perspectives, which is the expression of thoughts by students on their work. Fundamental concepts of programming, which include declarative, procedural, and strategic, are acquired through game-based learning. The first refers to the information and facts required to perform tasks. The second is the programming knowledge on the procedure to perform the tasks. The third is the application of programming concepts learnt, to different circumstances in order to arrive at new programming rules for general or new scenarios. The correspondence between the psychomotor, cognitive, and affective learning results acquired in the game cycle and the three dimensions of computational thinking is summarized in Table 5.

Table 5
The three dimensions of computational thinking.

Computational thinking	Ability	Learning outcome	Outcome performance
Computational thinking concepts (CT-C)	Fundamental programming concepts such as sequences and looping.	Cognitive learning outcome	Declarative knowledge, procedural knowledge, strategic knowledge
Computational thinking practices (CT-P)	Problem solving skills such as testing and debugging.	Psychomotor learning outcome	Skill-based learning results
Computational thinking perspectives (CT-V)	Expressing/ connecting/ questioning	Affective learning outcome	Confidence, self-efficacy, attitude, and preferences

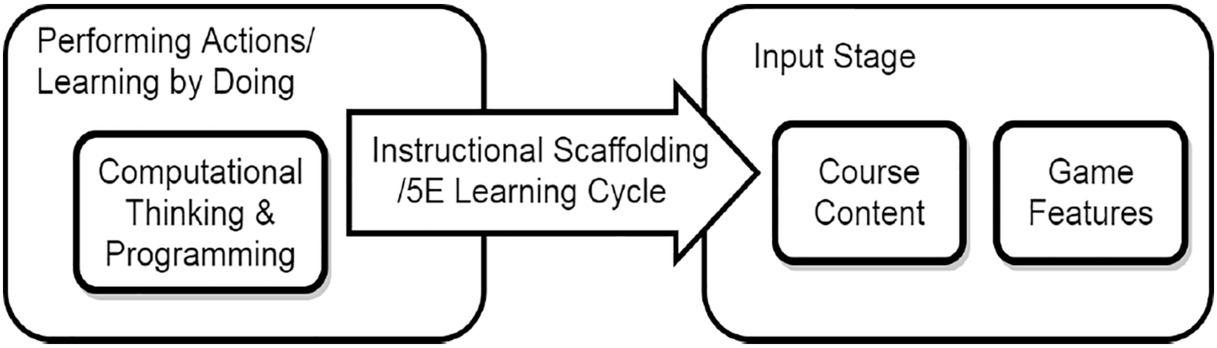

Performing Actions/ Learning by Doing
Computational Thinking & Programming
Instructional Scaffolding /5E Learning Cycle
Input Stage
Course Content
Game Features

Fig. 4. Incorporating computational thinking into the input stage through 5E learning cycle.

To improve the computational thinking capability of preschool children, curricula for the teaching of programming are developed on the basis of the Input–Process–Outcome game model by Garris et al. (2002). Computational concepts, computational practices, and computational perspectives are taught in the course, with "learning by doing" as the learning strategy that fits into the cognitive features of preschool children and a suitable TUI as the instructional scaffold to aid study. The comprehensive teaching and game designs are shown in the conceptualized game-based learning system exhibited in Fig. 5.

In the cognitive process during preschool children's learning journey, teaching is dictated by cognitive level and requires the support of instructional scaffold. Based on cognitive theory, this study uses tangible user interface (TUI) as the instructional scaffold for preschool children to enhance computational thinking through the teaching of program logic. According to Hiroshi Ishii and Ullmer (1997), TUI has the following characteristics: physical representations being coupled with digital information, tangibles embodying operational information and feedback, physical representations embodying mechanisms for interactive control, and physical representations perceptually coupled with actively mediated digital representations (images, sounds, etc.). As discussed by Antle (2007) on children, the relations between TUI physical and digital representations are classified into perceptual mapping, behavior mapping, and semantic mapping. In this experiment, we design an interactive interface for the learning of program logic in preschool children based on the above principles and characteristics of TUI. The course work targets computational thinking and teaches preschool children program-logic in a fun, game-based method.

3.3. Design of interactive tangible user interface

To examine the interactive behaviors between the TUI and learners (Fig. 6), Antle and Wise (2013) suggest five interconnected components:

1. Physical objects
   Instead of a virtual GUI, the learners are able to interact with the physical objects via the TUI, including visual (e.g., colors), haptics (e.g., textures), audio (e.g., tones), and spatial properties (e.g., locations).
2. Digital objects
   Digital objects are also known as virtual entities, which include visual, audio and spatial effects. The hybrid TUI environment provides multi-touch functions that allows the learners to directly interact with the digital objects.
3. Actions
   The learner's perceptual behaviors (such as manipulating a physical object) can be transferred to a TUI system.
4. Informational relations
   The informational relations include the mapping relation between physical objects and virtual objects, as well as the resulting behaviors between the real world components and simulated entities. For example, if a file folder is represented by a water bottle, twisting the bottle cap corresponds to opening the file folder.
5. Learning activities
   The assigned learning activities greatly influence the learners' behaviors in interacting with the TUI.

These five components are developed by the cognitive and learning theories, which can serve as the guidelines for the designs of learning activities, as well as the architecture of a TUI. In addition, the guidelines evaluate a TUI based on its relevance and usefulness. Table 6 summarizes the theoretical perspectives and provides 13 design guidelines in TUI elements (Antle and Wise, 2013). For example, the guideline-1 (distributing information across modalities can increase effective working memory capacity) involves physical and digital objects, where using the physical and digital components allow the learners to decrease

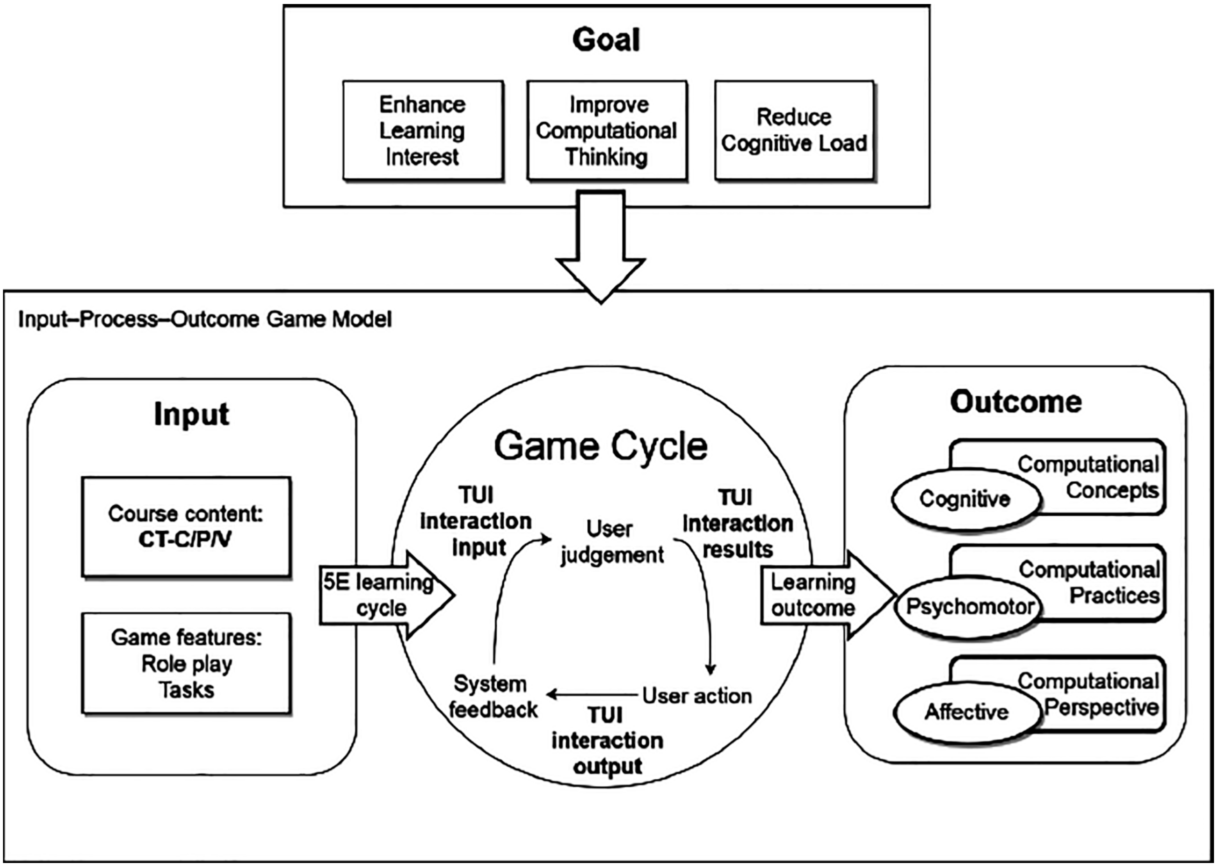


Fig. 5. Conceptualization of game-based learning system.

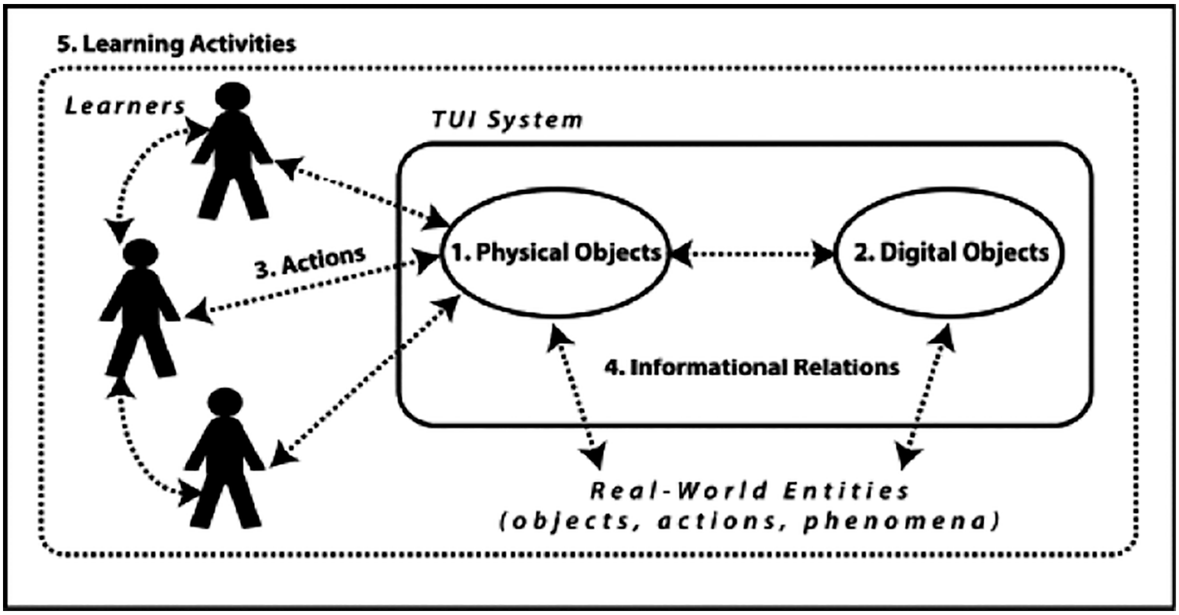


Fig. 6. Interaction between TUI and learners (Antle and Wise, 2013).

Table 6
Design guidelines in TUI for learning activities (1: physical objects, 2: digital objects, 3: actions on objects, 4: informational relations, 5: learning activities).

Guidelines, cited from (Antle and Wise, 2013)	1	2	3	4	5
1. *Distributing information across modalities can increase effective working memory capacity.*	X	X			
2. *Integrating spatial sources of information across and within modalities can minimize the extraneous cognitive load imposed to synthesize inputs.*	X	X			
3. *Using concrete representations can support interpretation of symbolic representations of abstract concepts.*	X	X		X	
4. *Making mappings between the form and behavior of physical and/or digital objects and real-world entities coherent can reduce extraneous cognitive load.*				X	
5. *Creating contextualized tasks or personal objects can support learners in forming individually meaningful goals for interacting with the TUI.*	X				X
6. *Using spatial, physical, temporal or relational properties can slow down interaction and trigger reflection.*	X	X	X	X	X
7. *Distributing parts of mental operations to actions on physical and/or digital objects can simplify and support mental skills.*			X	X	
8. *Leveraging image schemas in input actions can improve usability and system learnability.*			X		
9. *Using conceptual metaphor(s) based on image schemas to structure interaction mappings may bootstrap learning of abstract concepts*				X	
10. *Designing objects that allow for spatial reconfiguration can enable mutual adaptation of ideas.*	X	X			
11. *Creating configurations in which participants can monitor each other's activity and gaze can support the development of shared understandings.*	X	X			
12. *Distributing roles, information and controls across the TUI learning environment can promote negotiation and collaboration*	X				X
13. Creating constrained or co-dependent access points schemes can compel learners to negotiate with each other.			X		

the cognitive load and extend the working memory capacity.

Studying program logic through TUI without increasing the cognitive load for learning and also aids preschoolers in acquiring computational thinking (Lin, 2015). This study uses modified Arduino robots to design interactions with TUI suitable for preschool children based on the thirteen TUI design guidelines (mentioned in the teaching methods) and the cognitive limitation of children. The overall interactions are shown in Fig. 7.

Physical and digital objects are designed according to the first guideline of using multiple perceptions to enhance learning and memory. Visually perceived color is used in this study to define program instructions, and the children place tactile colored-cards and answer-cards in response. Based on the second guideline of reducing excessive cognitive load, multiple inputs are summarized and simplified into four types of robot motions, i.e., forward, backward, turn left, and turn right. These have their corresponding components in the colors and program and can be applied with guideline 10, which permits the arbitrary matching of objects to support the creative thinking of preschool children. Guideline 3 is the use of specific representations for abstract ideas, where Arduino robots are employed as specific representations to showcase the results contributed by the children. The five elements for the interactions with the tangible user interface depicted in Fig. 7 are summarized as follows:

1. Physical objects

   The design of the input interface adopts the visual attribute of color for preschool children, who are visually sensitive and whose hand muscles are under development. Colored cards of A4-size and 10 × 10 answer cards are arranged and combined. The output interface uses the Arduino robots to display the program instructions.
2. Digital objects

   The program instructions are split into forward, backward, left turn, and right turn, based on the movement of Arduino robots. Preschool children interact in their study through game-based learning and the game levels are set based on the Input–Process–Outcome game model. Guideline 8 outlines the use of graphics to enhance the usability of inputs and their compatibility with learning. In this study, the graphics are simplified to a single color to suit the study subjects. Guideline 13 lets the students discuss among themselves in a restricted or dependent manner. To promote interaction between preschool children, we design game levels so that they can discuss and encourage one another.
3. Actions

   The students arrange and place colored cards into different combinations to accomplish the tasks of the game, and observe the outputs of Arduino robots to amend the execution results. Guideline 4 and 7 show the information mapping between physical and digital

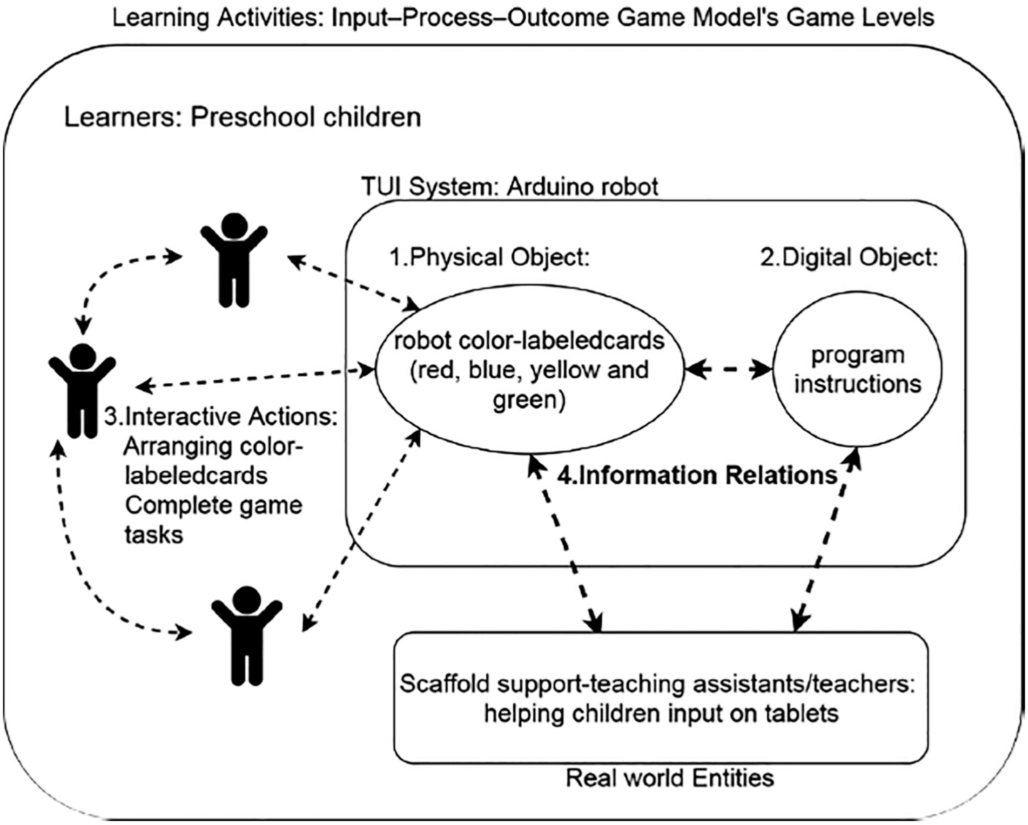

Learning Activities: Input–Process–Outcome Game Model's Game Levels
Learners: Preschool children
TUI System: Arduino robot
1.Physical Object:
2.Digital Object:
robot color-labeledcards (red, blue, yellow and green)
program instructions
3.Interactive Actions: Arranging color-labeledcards Complete game tasks
4.Information Relations
Scaffold support-teaching assistants/teachers: helping children input on tablets
Real world Entities

Fig. 7. Interactions with tangible user-interface.

objects. The program instructions forward, backward, left turn and right turn of Arduino robots are mapped to the colors red, yellow, blue, and green, respectively, and the symbols are limited to a single color to reduce the cognitive load. From guidelines 3 and 9, the mapping is classified into perception and behaviors. Because preschool children have limited vocabulary, the TUI here is constructed based primarily on perceptual and behavior mappings, tailored to the cognitive characteristics of the students.

4. Informational relations

 To accommodate the cognitive level and locomotion capability of preschool children in perceptual mappings, colored cards are adopted to satisfy the visual requirements. The red, yellow, green, and blue cards are assigned as forward, backward, left turn, and right turn movements, respectively. In behavior mappings, preschool children arrange the colored cards to exhibit their thought process on the causal relationship of an event. The combination of these cards represents the mechanism by which preschool children solve the game tasks.

The above physical and digital objects, actions, and informational relations are summarized in Table 7 below.

To comply with guideline 5, i.e., supporting learners to achieve the learning goal with scenarios, students learn in this study through game-based learning, with game levels designed on the Input–Process–Outcome game model. According to guidelines 6 and 12, the performance of tasks in this study enables collaboration among learners.

5. Learning activities

 The design of learning activities affects the students in their use of the tangible user interface, i.e., human–computer interaction. In this study, game levels are set according to the Input–Process–Outcome game model. Details of the game level design are described in the next section.

Table 7
Design of TUI interactions.

	Physical objects	Digital objects	Action	Informational relations
Input	red cards	programming-sequences – forward	Select and place	Perceptual mappings: red – forward
	yellow cards	programming-sequences – backward	Select and place	Perceptual mappings: yellow – backward
	blue cards	programming-sequences – left	Select and place	Perceptual mappings: blue – left
	green cards	programming-sequences – right	Select and place	Perceptual mappings: green – right
Output	Arduino robots	programming-execution	Execution	Behavior mappings: order of colored cards – robots

After the cognitive construction of preschool children, scaffold is adopted to aid their learning, with teaching assistants and teachers supporting the scaffold as per design guideline 2 and 4. When helping with the input interface, the teaching assistants use cell phones or tablets to operate image recognition APP, and put the instructions by students as arranged and combinations of colored cards into Arduino robot.

3.4. Learning Activities-Game levels

The learning activities are implemented with the 5E learning cycle into game-based learning. The programming course teaches program logic to preschool children through games. Three appropriate game levels are planned to satisfy the attention and cognitive levels of preschool children. The first explains the way robots move. Preschool children bring their life experience into the game through role-play, where they understand the operation of robots and are engaged more attentively to learn the basic logic and concepts of programming. The second is challenge game, where tasks, such as finding the shortest distance or being the first to reach the destination, are assigned. Additionally, the accuracy of program instructions are gradually improved in the game cycle, such as moving a few steps forward and walking to the left. The third level focuses on debugging and analyzing problems to identify the errors in logic instructions. The game levels are shown in Table 8 below.

The course content is embedded in the game-based learning model and illustrated from the aspects of teaching and learning. Learning is initiated in preschool children to study program logic through game-based methods. The three game levels are explained with 5E. In the first level, preschool children study program logic, using picture cards to initiate robot movements, and understand the mechanisms of robot movements by simulating robots. Logic of sequences is also taught. In the second level, preschool children execute tasks, such as directing robots to reach the destination and move accurately. The third level is to correct the mistakes and identify the reasons behind them. The above are summarized in Table 9.

At the engagement stage of the 5E learning cycle, picture cards are used with questions to guide the preschool children in their understanding of robot movement mechanisms and their attempt to provide answers. In the exploration stage, they are permitted to investigate freely. In the explanation stage, preschool children state the challenges they encounter and discuss together. However, standard answers are not offered, such that they could be guided to arrive at their own solutions. In the elaboration/extension stage, they are encouraged to apply the program logic learned. In the evaluation stage, open questions are asked to let the children express their feelings, thoughts about what they have learnt, etc. The roles that teachers play at each stage and responsibilities are shown in Table 10 below.

The three game levels and the 5E learning cycle ensure participation, exploration, interpretation, elaboration, and evaluation at the respective stages of learning. Two experimental studies are conducted, which contribute to different learning objectives, leading to different game levels, class content, and computational thinking abilities (Table 11).

In this instructional model, the operation of Arduino robots tangibly displays the input and output information. The teaching focuses on the procedures and methods employed by preschool children during the learning process to achieve the game goals and to overcome the challenges encountered. Their grasp of program logic is evaluated by task-completion results. In this method, a suitable user interface can be selected as required, and the game levels can be designed according to the course content and interface.

Table 8
Game level setups.

Game level	Design goal	Design function	Literature background
Level 1 Robot simulation game	Scenario simulation; role projection; familiarization with interface	Introducing the game Input interface-colored cards	Game feature-Fantasy Children cognitive feature-mimicking TUI-perceptual mappings-vision
Level 2 Challenge game	Strengthen logic; mode of action; team work	Practicing Time limit/getting around barrier/shortest distance/minimum instructions Output interface-colored cards	Game feature-Control Game feature-Challenge Children's cognitive feature-perceptual dominance/limited attention TUI-perceptual mappings- vision
Level 3 Task game	Analyzing problems; finding mistakes	Locating mistakes Output interface-colored cards	Game feature-Goals Game feature-Mystery TUI-perceptual mappings- vision

Table 9
different game levels in 5E learning cycle.

Game level	Game outline	5E strategy	Activity
Level 1 – Robot simulation game	Simulating the robot Initiating the robot	1E engage	Arousing interests through games Explaining program logic and robot construction with picture cards
Level 2 – Challenge game	Reaching the destination on time via the shortest path	2E explore 3E explain	Motion of robots Discussing the ways of robots' moves
Level 3 – Task game	Identifying and correcting the wrong path	4E elaborate/extend 5E evaluate	Task execution Reviewing and supervising the children's problem solving processes

Table 10
5E learning cycle in teaching activity.

5E strategy	Game function/interface	Teaching activity
1E – Engage	Level 1: colored cards + Arduino robots	Guiding problem-solving Warm-up activities
2E – Explore	Level 2: colored cards + Arduino robots	Explaining program sequences with robot movements Let preschool children play freely
3E – Explain	Level 2: colored cards + Arduino robots	When a robot executes a task, let preschool children state the difficulties encountered
4E – Elaborate/extend	Level 3: Arduino robots	When a robot executes a task, let preschool children make flexible use of the fundamental skills learned
5E – Evaluate	Level 3: Arduino robots	Asking open questions to let preschool children express themselves

3.5. *Hypotheses*

The designs of the proposed research as well as the experimental studies were based on the aforementioned computational thinking elements (CT-C, CT-P and CT-V) and 5E learning cycle, in which these conceptual frameworks have been suggested to enhance learners' computational thinking ability as well as their learning attitudes and behaviors. Therefore, based on these theoretical guidelines, we form the following hypotheses:

H1: the developed game-based learning approach along with the TUI applications will enhance the preschool learners' computational thinking ability.

H2: the developed game-based learning approach along with the TUI applications will encourage the preschoolers to actively participate in the course activities, as well as increase their learning interests.

H3: the effects of learning experiences will not only influence the learners' task performance, but also change their learning intentions and behaviors.

The influences of the above hypotheses will be examined via the empirical user studies, where the details can be found in the next section.

4. Methods

This study develops a learning framework (game-based learning along with tangible user interface) to enhance preschool learner's computational thinking abilities as well as improve their learning

Table 11
Game levels and teaching strategy.

5E learning cycle	Game level	Experimental schedule	Teaching content	Computational thinking (CT) ability
1E – Engage	Level 1	1st study	Introduction and explanation Simulating/exploring	Fundamental capability
2E – Explore	Level 2	1st study	Sequences	CT-C
3E – Explain	Level 2	1st study	Expressing/connecting	CT-V
	Level 2	2nd study	Initiation on condition	CT-C
4E – Elaborate/Extend	Level 3	2nd study	Scoring/gaining experience	CT-P
5E – Evaluate	Level 3	2nd study	Testing/debugging	CT-P
	Level 3	2nd study	Expressing/connecting	CT-V

achievements. The teaching methods integrate the course materials (computational thinking concepts, practices, and perspectives) with the TUI components to enhance participants' cognitive, psychomotor, and affective abilities. The overall experimental design follows the Input–-Process–Outcome game model (Fig. 8), along with the 5E learning cycle and TUI interactions. The TUI system is used in the empirical user studies, which adopts Arduino robots as the learning smart toy to enable the participants to study the computational thinking knowledge and to practice the problem-solving skills via the physical device. The cognitive, psychomotor, and affective abilities are developed from perceiving different course materials and the learning outcomes are examined in various levels of questions.

And as the research focuses on examining how the proposed learning scheme affects the learner's cognitive knowledge and behaviors, the independent variables (IV) include the use of TUI application and the participants' knowledge levels, whereas the dependent variables contain learners' computational thinking ability and learning interest (Table 12).

Table 12
Independent and dependent variables.

Independent variable	Description
TUI applications	Applying the programming logic in the TUI system allows the participants to develop their cognitive abilities and facilitate their learning activities, resulting in different computational thinking abilities and learning interests.
Knowledge levels	The 1st and 2nd studies are conducted one week apart, where the knowledge differences may contribute to different learning strategies and/or behaviors.
Dependent variable	**Description**
Computational thinking abilities	Three levels of logical questions (easy/normal/difficult) are used to examine the participants' cognitive abilities.
Learning interests	Three types of learning behaviors (bystander/alone/collaborate) are identified to indicate the participants' learning interests.

4.1. Course materials

The course materials first introduce the basic computational thinking concepts (CT-C) to engage (1E in Fig. 8) participants' attention to learn the fundamental logic of programming skills (e.g., *sequence* in a program) as well as to strengthen cognitive ability. The middle level focuses on computational thinking practices (CT-P) that applies explore (2E) and explain (3E) strategies and involves in testing and debugging practices to increase psychomotor ability. In this phase, the participants have to understand the course materials and transfer the cognitive knowledge into physical behaviors to control the TUI components. The advanced level emphasizes on computational perspectives (CT-V) and exerts elaborate/extend (4E) and evaluate (5E) strategies to enhance affective ability. The participants have to understand a variety of computational concepts and comprehend the relations among the TUI components in order to correct or revise a robot's path. Through this process of method, it can support the importance of studying programming lessons for preschool children (Lindberg et al., 2019).

4.2. Apparatus- tangible user interface (TUI)

The TUI testbed system contains two major components, a cardboard with answer cards in the planning phase and a robot car with color cards in the testing phase. While learning the computational thinking knowledge, the students have to apply the programming logic to guide a robot car to reach the assigned destination. In the initial planning stage (Fig. 9a), the participants paste the answer cards (Fig. 10) in the grid to develop a sequence of routes, which allow them to effortlessly create or revise the paths. Once finished, the paths are examined in the real environment (Fig. 9b) to test whether the robot car can avoid all the potential obstacles (i.e., bricks) and reach the destination (Fig. 9c).

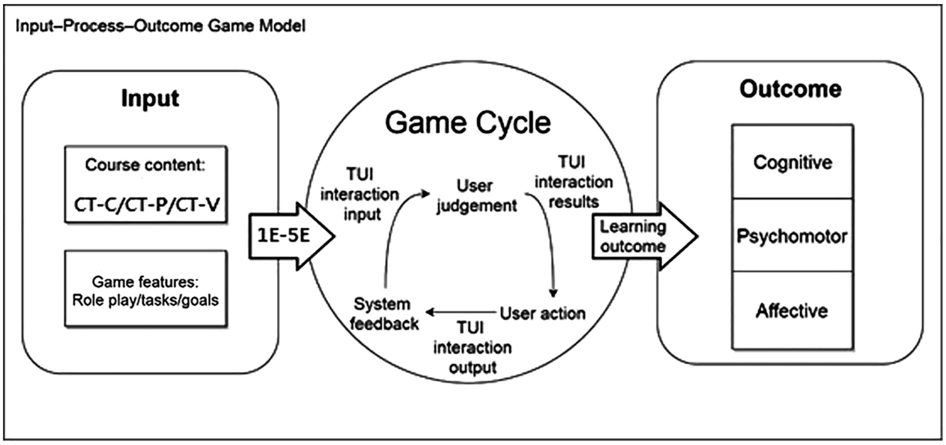


Fig. 8. Input-Process-Outcome Game Model.

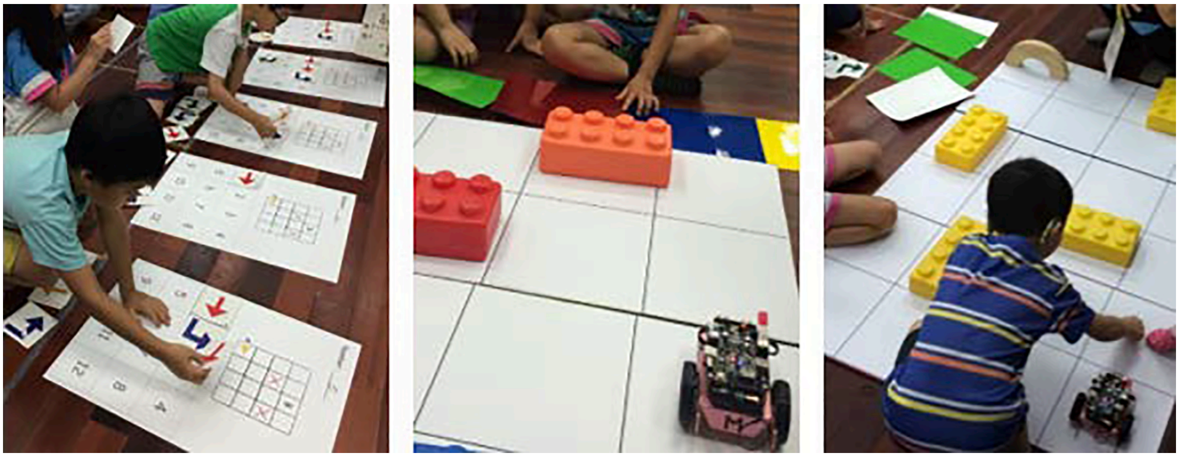


Fig. 9. TUI across all the experimental phases.

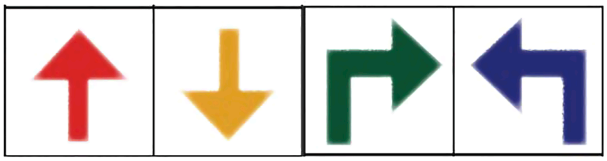
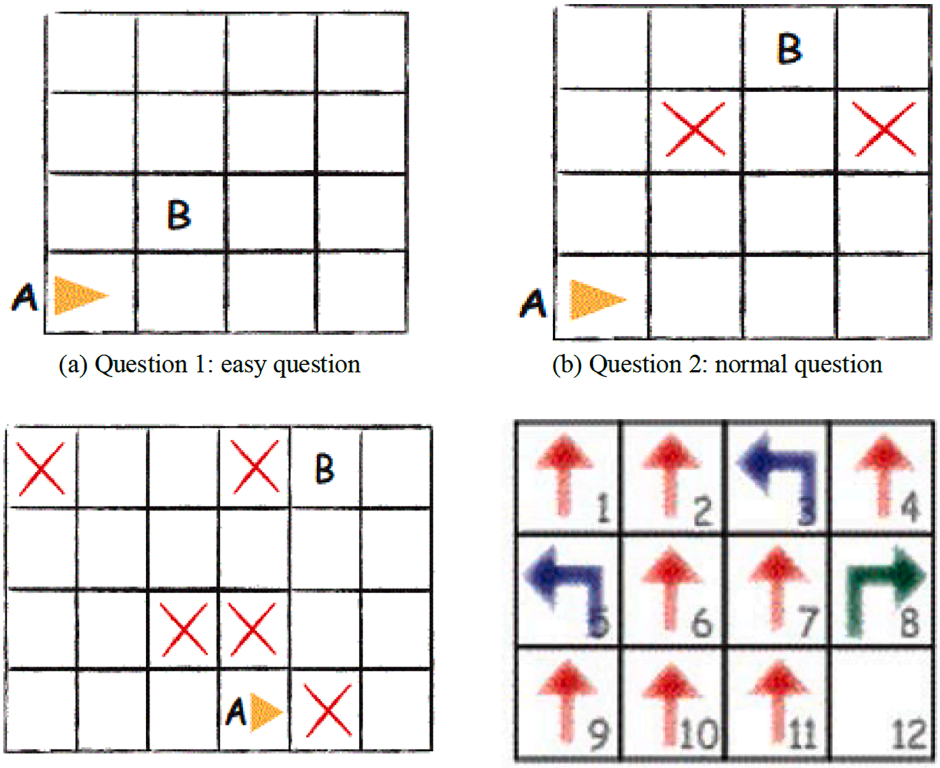


Fig. 10. Color cards for directing a robot's movements.

In the testing phase (Fig. 9b & c), the students need to place the color cards (Fig. 10) on the map to direct the robot car's movements (e.g., initiate or change paths). The robot car utilizes a sensor to measure the color and respond to the direction. For example, a red color card represents going forward, whereas a yellow color card indicates the backward movement. This approach allows the participants to not only practice their computational thinking concepts in the longitudinal fashion but also develop the psychomotor skills by matching the cognitive elements with physical objects.

4.3. Assess learning performance

To evaluate the participants' learning performance, 29 computational thinking related questions are retrieved from Bebras (n.d.). These questions are adopted in our study to assess the students' learning performance after perceiving the course materials and interacting with the



(a) Question 1: easy question

(b) Question 2: normal question

(c) Question 3: difficult question

(d) A sample answer of question 3 from a participant

Fig. 11. Questions applied to evaluate students' learning performance. "A" represents the starting point, "B" indicates the destination, and the red "X" signifies the obstacles that blocks a robot's path. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

TUI components (cardboard, answer cards, and the Arduino robot car). To better match the participants' cognitive capabilities, the questions are categorized into three levels, easy (Fig. 11a), normal (Fig. 11b), and difficult (Fig. 11c). Game level 1 (i.e., easy phase) concentrates on engaging (1E) the participant's attention in solving the task of driving the robot car from point A to point B. These easy questions are used to ensure the participant realizes the basic control mechanisms underlie the TUI system. After familiarizing with the basic concepts, game level 2 provides more challenging tasks to enhance learners' computational thinking abilities, where the normal question requires the participants to use explore (2E) and explain (3E) strategies and apply higher level of computational skills to resolve the task and driving the robot car to the destination without hitting any obstacles (indicated by the red X in the grid on Fig. 11b & c). Game level 3 (difficult question) focuses on improving the psychomotor and affective abilities, where an operator has to elaborate/extend (4E) and evaluate (5E) the robot's paths and avoid the potential flaws. In other words, the difficult question demands the students to identify the obstacles along a robot car's path and develop a path plan to guide the robot to the destination in a more complex condition. A sample answer of question 3 can be found in Fig. 11d.

4.4. Participants

Seven kindergarten students participated in the experiments, with ages ranging from 5 to 6 years old. The selection process of these seven students began with receiving permission from their respective parents. Parents were informed of the research goals, experimental tasks, experimental procedures, types of data collection, methods, and measures before providing consent. Parental consent letters were used to document parents' permission for their child to participate in this study. Parents were also informed of the potential risks associated with the experiment, including potential frustration, boredom or fatigue. Parents were also ensured their child remained anonymous. In order to conceal participants' identities for security purposes, participants were randomly assigned user ID's before conducting the experiment. There is no record kept linking participants with their user ID, ensuring their identity remains anonymous. Because students' participation was voluntary, the students were allowed to refuse participation and the students' parents or kindergarten instructors were allowed to stop the study at any time.

4.5. Experimental conditions

The experiment follows a within-group design. Two studies are conducted, each study includes a 20-minute teaching lesson and a 20-minute question session. The participants first take question 1 and 2 in the pretest to examine their computational thinking abilities before perceiving any teaching materials (Fig. 12). Question 3 aims to evaluate the learning outcome and is therefore tested after taking the teaching session. These questions aimed to validate first hypothesis (i.e., computational thinking ability). In order to avoid participants from potentially experiencing fatigue from 40-minute experimental sessions, the first and second studies are conducted one week apart. This also allows us to examine the learning effects between the two experimental studies. It is also supported by the theory proposed by Lye and Koh (2014) to teach learners explain their reasoning through concise communication to solve their problems.

The students' improvements of computational thinking capabilities are evaluated twice, before and after the experiments. Since conventional assessments (e.g., final scores) provide little information about learners' underlying cognitive strategies and may fail to reflect their true intents, the study is evaluated by multiple assessments including the correctness of answers and the observations of learning behaviors/interests during the classes. In other words, the students' learning activities while using the experimental testbed system are recorded to examine their learning awareness and information processing behaviors. To assess the learning interest of students (i.e., second hypothesis) as well as examine the change of learners' learning intentions and behaviors (i.e., third hypothesis), behavior observations are made at fixed time intervals to continuously record the game-related activities of the students. To evaluate the cognitive load, the event-based behavior observation forms are used to record specific actions, such as idling, giving up, refusing to learn, losing focus, talking about things unrelated to the class. The participants' learning outcomes (including observed learning behaviors and the correctness of answers) are the dependent variables, and the TUI with game-based learning are the independent variables.

5. Results and discussions

To examine the proposed teaching method, students' learning behaviors/interests and learning performances are evaluated using qualitative and quantitative assessments. Seven students are recruited for this experiment, with two rounds of empirical studies conducted a week apart from each other. The students' computational thinking abilities are examined by three question assessments and their learning activities are observed and recorded by two coders.

5.1. Computational thinking ability

The learning outcomes are measured by three levels of questions, in which the easy question includes simple questions to examine the fundamental logic knowledge, the normal question assesses the psychomotor learning outcomes by asking the users to match the conceptual concepts with physical system components, and the difficult question requires the students to locate and correct potential mistakes to evaluate whether the participants can comprehend the relevant computational concepts. The first two questions (east and normal) are tested before the teaching sessions and the difficult question is examined after perceiving the teaching materials and interacting with the TUI system.

The results (Fig. 13) show while students first perceived the learning materials (1st study), only 1 out of 7 students completes the easy

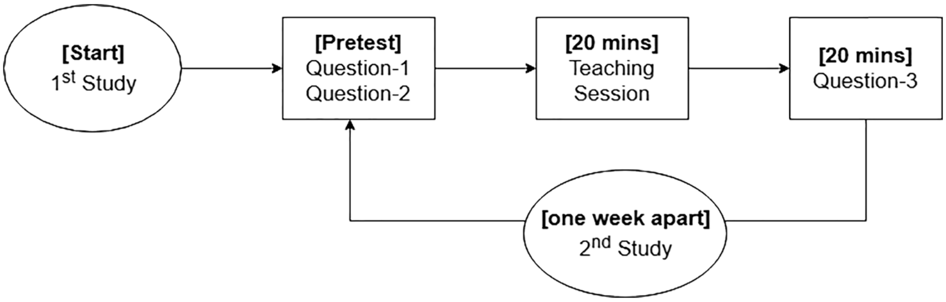


Fig. 12. Experiment flow chart.

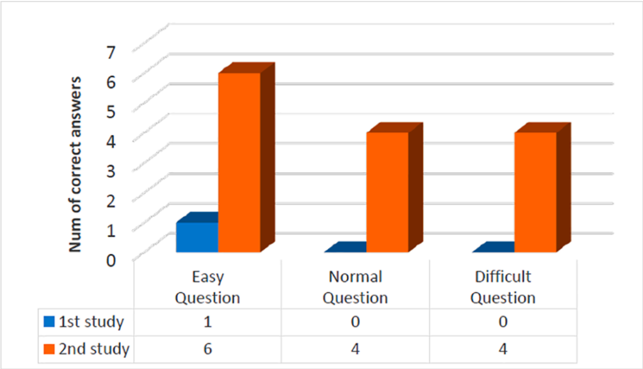


Fig. 13. Overall test results.

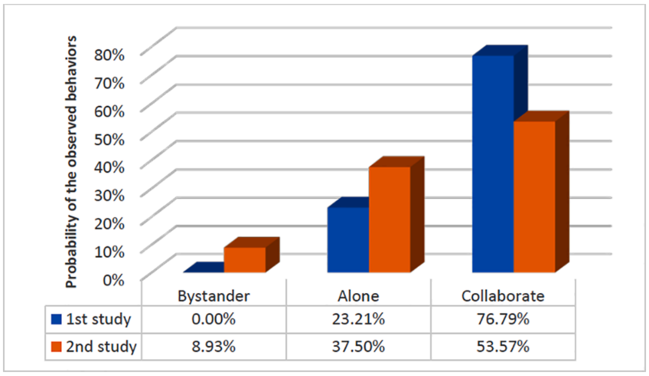


Fig. 14. Probability of the observed learning behaviors.

question correctly and none of them respond to the normal and difficult questions successfully. In the 2nd study, conducting one week after the 1st study, the results reveal most of the participants can answer the questions correctly across all different levels of questions. These results suggest that the TUI system can effectively improve the learning performance as well as enhance students' computational thinking ability.

5.2. *Affective results – learning interest*

The participants' learning interests can be observed through their behaviors, which are identified into three major categories, bystander, alone and collaborate (Table 13). The *bystander* type indicates a student who shows little interest to participate in the experiment or pay no attention to the course activities. The *alone* behaviors represent a user who tends to work on the course materials by herself rather than discuss with other students. The *collaborate* type shows the participants prefer to collaborate with each other and may share strategies to complete the course activities.

The participants learning activities are observed and encoded by two coders every 3 min during the lecture. These variances can reveal the participants' strategies and behavioral differences while perceiving the learning materials as well as interacting with the TUI system. To better represent the distribution of the observed activities, the results are computed and showed based on its probability (Fig. 14).

The results of the first study show 77% of the observed behaviors are identified in the collaborate type and 23% belongs to the alone group. When the students first attend to the lecture, due to the insufficient computational thinking ability, they tend to work together and share their thoughts with each other. After the first lecture, in the 2nd study, the learning patterns reveal that more students prefer to work on the course content by themselves, instead of discussing their ideas with others. The increased probability of alone type could be resulted from higher confidence in the course materials as the students have already experienced the similar lecture a week ago in the 1st study. However, the result again shows more than half of the participants' behaviors are falling into the collaborate type, which suggests the provided TUI system can effectively encourage the students to collaborate with each other.

Table 13
Behavior classification.

Behavior code	Behavior type
Bystander	Uninvolved in teaching activities: idle, look around, aimlessly walk around, observe other students' behaviors and provide no assistance
Alone	A student participates in the teaching events and work on the course activities independently. The participate prefers to perceive the teaching content by herself rather than discussing with others
Collaborate	The students tend to collaborate with each other to share their thoughts and discuss the materials when taking the lecture. The participants may act different roles or tackle different issues while working together

Engaging a learner's interest during the learning phases and directing their attentions to the teaching materials are critical to facilitating the overall learning performance. The bystander effect can be seen as a type of distracting behaviors, in which the students pay little interests to the course contents and show inappropriate activities instead, such as idling, chatting with classmates, talking about irrelevant topics. The observed results suggest the proposed TUI applications can effectively attract learners' attention and encourage them to devote their cognitive resources to the appropriate affairs, leading to great improvements of learning performances and low distracting behaviors. Additionally, while experiencing the similar learning activities, the TUI approach can successfully secure user attention to the suitable events and satisfy various learners' behavioral preferences to match their needs. In fact, based on our observations, most of the distracting behaviors are occurred while the students are waiting in line for testing their answers via the robot car on the map (Fig. 9b & c). In other words, these inactive learning behaviors are not resulted from the TUI system itself.

6. Conclusions

This study introduces game-based learning and tangible user interface (TUI) to enhance preschool children's computational thinking abilities. The results show the proposed instructional methods are suitable for this purpose. The provided entertaining scenarios and user-friendly interfaces are adopted to lower the learning difficulties and to increase the learning interests and behaviors of students such that their computational thinking abilities could be improved. The learners can transfer the cognitive abstracts into physical behaviors through the interactions with the TUI applications, thus enhancing their computational thinking. Although smart toys can help the development for a TUI system, there are still many types of smart toys and various options for designing learning materials to adapt learners' age or knowledge differences. In this study, we used Arduino-based robot cars and color-labeled cards to develop the TUI applications. The developed framework (scenarios and systems) can benefit the research community in improving preschoolers' computational thinking without increasing their learning difficulties.

The learning outcomes are measured by three different levels of questions, where the results suggested the game-based learning approach along with the TUI system can effectively improve the students' learning performance as well as enhance their computational thinking abilities. In other words, the participated preschool children's computational thinking was improved by learning program logic through the game-based learning materials and interactive interface. Our first hypothesis is therefore supported. Additionally, the students' learning interests were evaluated by their engagement activities, where both an individual's behaviors and the group participations are examined. The results again revealed the developed teaching approach can help the preschoolers to actively participate in the course activities as well as increase their learning interests, which support the second hypothesis. Since the 1st and 2nd studies are conducted one week apart, this allows us to examine the influences of participants' learning experiences affecting their performance and behaviors. As expected, students' experience differences did affect their task outcomes, where better performances were observed in the more experienced users. In addition, an experienced individual tends to work alone to solving the tasks rather than collaborating with others. This may suggest once the preschool learners believe they have sufficient knowledge regarding the task, they will have less incentive to share knowledge or cooperating with others. These results also support our third hypothesis. In addition, the transition between class activities and its effect on study-results could be further investigated. In terms of the cognitive load, as the game-based interactive interface is straightforward to operate, preschool children familiarize themselves with it rapidly and are able to play while engaging in thinking; distraction usually occurs during the wait time. Furthermore, other arrangements can be made in future to permit better transition between activities.

In terms of curricular planning, two experimental studies are conducted in this study; here, two computational thinking elements are implemented, i.e., sequences and events. More classes can be conducted in future, and their duration could increase such that more aspects of program logic can be covered. In terms of the instructional method, the teaching and testing are performed with the assistance of the game cycle, in which advancement in game levels is achieved. In this way, the learners will have a sense of accomplishment. For the future, the game cycle can be used as a secondary assistance in passing the levels, such that students could have an opportunity to make amendments if they commit mistakes. Cognitive features of preschool children in different growth and development stages should also be analyzed while designing teaching methods. The framework proposed here could be used to set up game levels to enable students to learn while playing on game-based TUI. The present study adopts the input-process-outcome game model to construct the research framework. The overall research model is shown in Fig. 5. To use this proposed method, a researcher first needs to identify the course materials and expected outcomes, in which each course content requires different TUI interaction input and output (as the perception and behavior mapping shown in Table 4) and can lead to different outcomes to enhance the learner's ability. For example, the sequence concept is one of the CT-C contents, which uses colors/clicks as the input and flick/twist as the output to train a participant's cognitive ability. The proposed teaching approach has been proved that can effectively enhance students' learning outcomes; however, there are some limitations in this study. One of the major issues is the limited sample size. As the preschool students are the target users in this study, this increases the difficulties in recruiting greater number of participants. In addition, since the preschool students can focus their attention in a short amount of time, researchers may need to conduct several separated studies to collect sufficient responses. Therefore, in this study, two around of studies are organized, which inject extra challenges to ensure the participants are available as well as willing to attend both experimental sessions. The limited experimental duration also restricts the number of measures applying to examine the learning activities and outcomes.

As the results demonstrate the proposed method can encourage learners to actively participate in the course activities and enhances their learning performance, the developed teaching framework can benefit the research community in enhancing preschool learners' computation thinking abilities. Future work will include further examinations on various learning contents across different degree levels and evaluate the improvements on learners' cognitive thinking ability. To verify how participants consume their cognitive resources, we plan to apply an EEG device to examine the brain activities and eye-tracker to detect the attention allocation.

CRediT authorship contribution statement

Szu-Yin Lin: Conceptualization, Methodology, Data curation, Writing - original draft, Writing - review & editing. **Shih-Yi Chien:** Methodology, Visualization, , Writing - review & editing. **Chia-Lin Hsiao:** Software, Investigation, Validation, Writing - original draft. **Chih-Hsien Hsia:** Writing - review & editing. **Kuo-Ming Chao:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2410-H-197-002-MY3 and MOST 109-2410-H-004-067-MY2.

References

Antle, A.N., Wise, A.F., 2013. Getting down to details: using theories of cognition and learning to inform tangible user interface design. Interact. Comput. 25, 1–20. https://doi.org/10.1093/iwc/iws007.

Antle, A.N., 2007. Tangibles : Five Properties to Consider for Children. In: Workshop on Tangibles at Conference on Human Factors in Computing Systems (CHI 2007). pp. 1–10.

Bebras - International Challenge on Informatics and Computational Thinking [WWW Document], n.d. URL <https://www.bebras.org/> (accessed 7.10.19).

Bergsmark, M., Fernaeus, Y., 2016. From patchwork to AppliquÉ: reflections from an interaction design remake. In: Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '16. ACM, New York, NY, USA, pp. 236–244. https://doi.org/10.1145/2839462.2839473.

Berk, L.E., Winsler, A., 1995. Scaffolding Children's Learning: Vygotsky and Early Childhood Education. NAEYC Research into Practice Series. Volume 7. National Association for the Education of Young Children, 1509 16th Street, N.

Boss, S., Bruce, H., Case, C., Miller, K., 2001. Developing smart toys— from idea to product. Intel Technol. J.

Brennan, K., Resnick, M., 2012. In: New Frameworks for Studying and Assessing the Development of Computational THINKING. American Educational Research Association, Vancouver. Canada, pp. 1–25.

Bybee, W.R., Taylor, J., Gardner, A., Van Scotter, P., Carlson, J., Westbrook, A., Landes, N., 2006. The BSCS 5E instructional model: origins, effectiveness, and applications. BSCS 5, 88–98.

Chien, S.-Y., Hsiao, S., Kang, Y., 2018. In: Understanding Students' Intentions in Learning Visual Robotic Programming: An Integration of IS Success Model, Utilitarian Value and Hedonic Value. PACIS, p. 85.

Ching, Y.-H., Hsu, Y.-C., Baldwin, S., 2018. Developing computational thinking with educational technologies for young learners. TechTrends 62, 563–573. https://doi.org/10.1007/s11528-018-0292-7.

Delprino, F., Piva, C., Tommasi, G., Gelsomini, M., Izzo, N., Matera, M., 2018. ABBOT: a smart toy motivating children to become outdoor explorers. In: Proceedings of the 2018 International Conference on Advanced Visual Interfaces - AVI '18. Presented at the 2018 International Conference. ACM Press, Castiglione della Pescaia, Grosseto, Italy, pp. 1–9. https://doi.org/10.1145/3206505.3206512.

Denning, P.J., 2017. Remaining trouble spots with computational thinking. Commun. ACM 60, 33–39. https://doi.org/10.1145/2998438.

Duran, L.B., Duran, E., 2004. The 5E instructional model: a learning cycle approach for inquiry-based science teaching. Sci. Educ. Rev. 3, 49–58.

Furber, S., 2012. Shut Down or Restart? The Way Forward for Computing in UK Schools. The Royal Society, London.

Garris, R., Ahlers, R., Driskell, J.E., 2002. Games, motivation, and learning: a research and practice model. Simul. Gaming 33, 441–467. https://doi.org/10.1177/1046878102238607.

Grover, S., Pea, R., 2013. Computational thinking in K–12: a review of the state of the field. Educ. Res. 42, 38–43. https://doi.org/10.3102/0013189X12463051.

Healy, J.M., 2000. Failure to Connect: How Computers Affect our Children's Minds–for BETTER and Worse. Powells, Portland, Or.

Hogle, J.G., 1996. Considering Games as Cognitive Tools: In Search of Effective "Edutainment".

Horn, M.S., Solovey, E.T., Crouser, R.J., Jacob, R.J.K., 2009. In: Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education. ACM Press, Boston, MA, USA, p. 975. https://doi.org/10.1145/1518701.1518851.

Hung, P.C.K., Iqbal, F., Huang, S.-C., Melaisi, M., Pang, K., 2016. A glance of child's play privacy in smart toys. In: Sun, X., Liu, A., Chao, H.-.-C., Bertino, E. (Eds.), Cloud Computing and Security, Lecture Notes in Computer Science. Springer International Publishing, pp. 217–231.

Ishii, H., 2008. Tangible bits: beyond pixels. In: Proceedings of the 2nd International Conference on Tangible and Embedded Interaction - TEI '08. Presented at the 2nd international conference. ACM Press, Bonn, Germany. https://doi.org/10.1145/1347390.1347392.

Ishii, H., Ullmer, B., 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '97. Presented at the SIGCHI conference. ACM Press, Atlanta, Georgia, United States, pp. 234–241. https://doi.org/10.1145/258549.258715.

Kazemitabaar, M., McPeak, J., Jiao, A., He, L., Outing, T., Froehlich, J.E., 2017. MakerWear: a tangible approach to interactive wearable creation for children. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17. ACM, New York, NY, USA, pp. 133–145. https://doi.org/10.1145/3025453.3025887.

Khine, M.S., 2018. Computational Thinking in the STEM Disciplines: Foundations and Research Highlights. Springer.

Kickstarter, 2017. ROBOPAL: Enrich your coding experience with stories [WWW Document]. Kickstarter. URL <https://www.kickstarter.com/projects/519422151/robopal-build-stem-skills-build-robots-have-fun> (accessed 11.15.19).

KinderLab Robotics, 2014. Robot Kits For Kids | KIBO. KinderLab Robotics. URL <https://kinderlabrobotics.com/kibo/> (accessed 11.15.19).

Koslowski, B., 1980. Quantitative and qualitative changes in the development of seriation. Merrill-Palmer Q. Behav. Dev. 26, 391–405.

KUBO Education, 2017. KUBO Robotics - Discover how easy it can be to teach computer science [WWW Document]. KUBO Education. URL <https://kubo.education/> (accessed 11.15.19).

V. Lin, V., 2015. Computational Thinking and Technology Toys. Honors Thesis Collection 307.

Lindberg, R.S.N., Laine, T.H., Haaranen, L., 2019. Gamifying programming education in K-12: a review of programming curricula in seven countries and programming games. Br. J.Educ. Technol. 50, 1979–1995. https://doi.org/10.1111/bjet.12685.

Llewellyn, D., 2005. Teaching high school science through inquiry: a case study approach. Corwin Press, Thousand Oaks (Arlington, Va.).

Lye, S.Y., Koh, J.H.L., 2014. Review on teaching and learning of computational thinking through programming: what is next for K-12? Comput. Hum. Behav. 41, 51–61. https://doi.org/10.1016/j.chb.2014.09.012.

Makeblock, 2015. Robot Kits for Kids : mBot | Makeblock - Global STEAM Education Solution Provider. Makeblock. URL <https://www.makeblock.com/steam-kits/mbot> (accessed 11.15.19).

Makeblock, 2017. Codey Rocky | Makeblock - Global STEAM Education Solution Provider. Makeblock. URL <https://www.makeblock.com/steam-kits/codey-rocky> (accessed 11.15.19).

Oh, H., Deshmane, A., Li, F., Han, J.Y., Stewart, M., Tsai, M.Y., Xu, X., Oakley, I., 2013. The digital dream lab: tabletop puzzle blocks for exploring programmatic concepts. In: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction. ACM, pp. 51–56.

Papastergiou, M., 2009. Digital Game-Based Learning in high school Computer Science education: impact on educational effectiveness and student motivation. Comput. Educ. 52, 1–12. https://doi.org/10.1016/j.compedu.2008.06.004.

Pedersen, B.K.M.K., Andersen, K.E., Jorgensen, A., Koslich, S., Sherzai, F., Nielsen, J., 2018. Towards playful learning and computational thinking — developing the educational robot BRICKO. In: 2018 IEEE Integrated STEM Education Conference (ISEC). Presented at the 8th IEEE Integrated STEM Education Conference (ISEC) 2018. IEEE, Princeton, NJ, pp. 37–44. https://doi.org/10.1109/ISECon.2018.8340502.

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., Pizarro, C., 2020. Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? Comput. Hum. Behav. 105, 105849 https://doi.org/10.1016/j.chb.2018.12.027.

Piaget, J., 1976. Piaget's theory. In: Piaget and His School. Springer, pp. 11–23.

Plowman, L., Luckin, R., 2004. Interactivity, interfaces, and smart toys. Computer 37, 98–100. https://doi.org/10.1109/MC.2004.1266302.

Prensky, M., 2003. Digital game-based learning. Comput. Entertain. 1, 21. https://doi.org/10.1145/950566.950596.

Primo Toys, 2016. Cubetto: A toy robot teaching kids code & computer programming [WWW Document]. Primo Toys. URL <https://www.primotoys.com/> (accessed 11.15.19).

Rafferty, L., Hung, P.C.K., Fantinato, M., Peres, S.M., Iqbal, F., Kuo, S.-Y., Huang, S.-C., 2017. Towards a privacy rule conceptual model for smart toys. In: Tang, J.K.T., Hung, P.C.K. (Eds.), Computing in Smart Toys, International Series on Computer Entertainment and Media Technology. Springer International Publishing, Cham, pp. 85–102. https://doi.org/10.1007/978-3-319-62072-5_6.

Rees, A., García-Peñalvo, F.J., Jormanainen, I., Tuul, M., Reimann, D., 2016. An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers.

Relkin, E., Umaschi Bers, M., 2019. Designing an assessment of computational thinking abilities for young children. In: Cohen, L.E., Waite-Stupiansky, S. (Eds.), STEM in Early Childhood Education. Routledge, pp. 83–98. https://doi.org/10.4324/9780429453755-5.

Scharf, F., Winkler, T., Herczeg, M., 2008. In: Tangicons: Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class. ACM Press, Chicago, Illinois, p. 242. https://doi.org/10.1145/1463689.1463762.

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., Owens, B.B., Stephenson, C., Verno, A., 2011. K–12 Computer Science Standards [WWW Document]. Computer Science Teachers Association. URL <http://scratch.ttu.ee/failid/CSTA_K-12_CSS.pdf>.

Shafto, S.A., 1986. Programming for learning in mathematics and science. In: Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education - SIGCSE '86. Presented at the seventeenth SIGCSE technical symposium. ACM Press, Cincinnati, Ohio, United States, pp. 296–302. https://doi.org/10.1145/5600.5635.

Shute, V.J., Sun, C., Asbell-Clarke, J., 2017. Demystifying computational thinking. Educ. Res. Rev. 22, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003.

Smith, D.H., Hao, Q., Jagodzinski, F., Liu, Y., Gupta, V., 2019. Quantifying the effects of prior knowledge in entry-level programming courses. In: Proceedings of the ACM Conference on Global Computing Education - CompEd '19. Presented at the ACM Conference. ACM Press, Chengdu,Sichuan, China, pp. 30–36. https://doi.org/10.1145/3300115.3309503.

Sullivan, A., Bers, M.U., 2018. Computational thinking and young children: Understanding the potential of tangible and graphical interfaces, in: Teaching Computational Thinking in Primary Education. IGI Global, pp. 123–137.

Sullivan, A.A., Bers, M.U., Mihm, C., 2017. In: Imagining, Playing, and Coding with KIBO: Using Robotics to Foster Computational Thinking in Young Children. Siu-cheung KONG The Education University of Hong Kong, Hong Kong, p. 110.

Wang, D., Zhang, L., Xu, C., Hu, H., Qi, Y., 2016. A tangible embedded programming system to convey event-handling concept. In: Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '16. ACM, New York, NY, USA, pp. 133–140. https://doi.org/10.1145/2839462.2839491.

Wing, J.M., 2006. Computational thinking. Commun. ACM 49, 33. https://doi.org/10.1145/1118178.1118215.

Wonder Workshop | Home of Dash, Cue, and Dot – award-winning robots that help kids learn to code [WWW Document], 2015. Wonder Workshop - US. URL <https://www.makewonder.com/robots/dash/> (accessed 11.15.19).

Yadav, A., Gretter, S., Good, J., McLean, T., 2017. Computational thinking in teacher education. In: Rich, P.J., Hodges, C.B. (Eds.), Emerging Research, Practice, and Policy on Computational Thinking. Springer International Publishing, Cham, pp. 205–220. https://doi.org/10.1007/978-3-319-52691-1_13.

Zhang, L., Nouri, J., 2019. A systematic review of learning computational thinking through Scratch in K-9. Comput. Educ. 141, 103607 https://doi.org/10.1016/j.compedu.2019.103607.