Education Tech Research Dev
https://doi.org/10.1007/s11423-023-10240-1

DEVELOPMENT ARTICLE

"If it's sunny, don't take an umbrella": a systematic evaluation of design principles for CT teaching games

Xina Jiang[1] · Casper Harteveld[2] · Yuqin Yang[3] · Anthony Fung[4] · Xinyuan Huang[5] · Shihong Chen[6]

Accepted: 29 April 2023

Abstract

Computational thinking (CT) has become an important skill for the new generation, and CT teaching games have been introduced to lower the barriers that novices face in learning programming. However, despite the prevalence of and demand for these games, little is known about the effectiveness of their design or about the principles that are conducive to learning using these games. In this article, we present a project in which we triangulated design research with controlled experiments to evaluate the effectiveness of a computational puzzle design (CPD) framework in guiding the development of CT teaching games for children. Using CT tests, game logs, and a survey, we evaluated the learning outcomes of and engagement with various versions of the game *LittleWorld*, designed based on various principles of the CPD framework and implemented in Chinese elementary education ($n = 202$). The results validate the CPD framework and demonstrate that it is a practical and systematic tool for designing CT teaching games. The findings of this study provide design implications concerning (1) the application of meta-gaming and (2) how serious game design research should be conducted.

Keywords Games · Early years education · Computational thinking · Serious game design · Evaluation methodology

Introduction

Computational thinking (CT), also known as algorithmic thinking, is increasingly acknowledged as the fourth "R" ability (after Reading, wRiting and aRithmetic) (Henderson et al., 2007). Starting CT education early is considered pivotal for children to persist in this domain (Chen et al., 2017). It has also been recommended that CT concepts be learned before coding skills, because the latter are extremely challenging for novices (Kazimoglu et al., 2011). CT teaching games allow players to program without coding statements and help them focus on computational problem-solving patterns and logic; thus, they are highly appropriate for teaching CT and related concepts to children (Denner et al., 2019; Harteveld et al., 2014; Kafai, 2006; Maloney et al., 2008; Petri & Wangenheim, 2017).

Extended author information available on the last page of the article

However, only a limited amount of work has been done on how to guide the design of such CT teaching games (Spieler et al., 2020; Troiano et al., 2020).

The aim in this study was to fill this gap by developing a computational puzzle design (CPD) framework to guide the process of designing CT teaching games for children. Adopting design-based research, we first proposed a CPD framework based on a prototype in our previous research (Jiang et al., 2019). We then designed the CT teaching game *LittleWorld* to validate and refine the CPD framework with learners around the age of 6. This study makes a theoretical contribution by developing a CPD framework for the design of CT teaching games for children and a practical contribution by developing *LittleWorld*, a CT teaching game that supports CT for children.

Computational thinking and computational problem-solving

Although the definition of CT has evolved, the term is widely understood to not be limited to training in coding skills but to include cultivating computational problem-solving (CPS) skills and analytical patterns. CT has been widely recognized as associated with CPS (Jiang et al., 2019; Kaleliöğlu et al., 2016; Lye & Koh, 2014; Weintrop et al., 2016). In recent years, many researchers have begun to focus on the relationship between CT and CPS. Studies have found that solving problems with computers can improve the ability of students to solve problems involving mathematics and other STEM subjects (Jona et al., 2014; Lockwood & Mooney, 2017; Voskoglou & Buckley, 2012; Yadav et al., 2011). Subsequent studies have introduced problem-solving steps and features from the field of art into CPS, and a creative CPS model has been developed, based on the premise that problem solving can further enhance CT capabilities (Chevalier et al., 2020; Febrian et al., 2018; Knochel & Patton, 2015). These findings highlight the point that, in terms of problem solving, CT is relevant to many disciplines and that the integration of customized problems into curricula can enhance competency in both CT and other disciplines (Yadav et al., 2016a, 2016b). In addition, CT and CPS are mutually reinforcing and CPS is the core of CT content and promotion (Maharani et al., 2019). Many studies have found that decomposition and abstraction in CT are used when defining problems, in the generalization stage that occurs in the formation of solutions, and when solutions are tested and evaluated in the debugging and algorithmic stages (Angeli & Giannakos, 2020; Lee et al., 2011; Lockwood & Mooney, 2017; Selby & Woollard, 2013). We thus focus on CPS as the core of CT learning.

CPS refers to the process of defining a problem, figuring out a solution, and assessing the effects of the solution in a computational manner (Liu et al., 2011). It includes three main steps: formulating a problem (i.e., abstracting and representing the problem in computationally meaningful ways, such as constructing symbol-based models of the problem that can be solved based on CS knowledge); building an algorithmic solution; and testing and optimizing the solution (i.e., troubleshooting and iterating to develop a more effective solution) (Jiang et al., 2019). To solve a problem, students must comprehend other analytical concepts. For instance, students may need to utilize the concept of looping to solve a problem that requires several repetitive manipulations. Therefore, we developed a CPS-analytic classification of CT concepts for 6-year-old children that includes two categories: CPS-related concepts and other analytical concepts. The former refers to the main steps used to solve problems, while the latter includes concepts used to analyze and solve specific problems. This classification guided our development of the assessment items and the CPD framework.

CPS learning for children

The target learners in this study were children around the age of 6, who are in the transitional stage from preschool to elementary school. Typically, children in this age group are beginning to develop skills in logic and reasoning, are being introduced to alphabet and math concepts, and are being encouraged to explore the world and their environment (UIS Information Paper, 2013). An overall improvement in these skills and qualities will lay the foundation for the development of CPS (Bers, 2018; Fessakis et al., 2013; Leonard et al., 2016). Children in this age group are in various stages of education in different countries and regions. The U.S. system classifies children in this age group as kindergarten students (Irwin et al., 2021). The United Kingdom classifies them as Key Stage 1 (British Council, 2013). In Scandinavia, taking Sweden as an example, children in this age group are receiving preschool class education (OECD,2017). In China, children around the age of 6 begin the first grade of elementary school (Yu & He, 2011). In our empirical study, we selected five classes of first graders from a school in Jiangsu province, China to participate in the experiment.

Studies have shown that learning CPS in a proper manner is appropriate given the level of cognitive development of 6-year-old children (Papadakis et al., 2016; Portelance & Bers, 2015). Children of that age can sequence instructions to create algorithms, understand and use grids by counting character movements, run programs and check for errors, and control structures with conditional and loop concepts (Brennan & Resnick, 2012; Papadakis et al., 2016; Rodríguez-Martínez et al., 2020). CPS learning for 6-year-old children includes three main steps. The first is formulating problems, which can be challenging for young children. Studies have shown that for young children decomposing problems can be more difficult than abstracting and identifying problems (Lavigne et al., 2019; Wang et al., 2021), unless they are implementing the decomposition process by connecting the problem with information with which they are already familiar (Bers, 2018; Sullivan et al., 2017). The second step is building algorithmic solutions. Novice program learners tend to prefer using simple trial-and-error approaches to construct problem-solving strategies (Shute et al., 2017). Learners need to be guided to create solution strategies, as a pure trial-and-error approach may limit CPS development (Chevalier et al., 2020; Olgun, 2017). The third step is testing and optimizing the solutions. Most CPS studies for children have focused on testing and debugging—that is, identifying and correcting errors (Fessakis et al., 2013; Grover & Pea, 2013; Lin et al., 2020; Maharani et al., 2019). This study focuses on developing the ability of learners to optimize solutions, because researchers have found that reviewing a solution and rethinking a more efficient (requiring fewer instruction cards) solution could encourage players to adopt an analytical reasoning strategy, with which they can achieve a deeper understanding of CPS (Guzdial, 2008; Jiang et al., 2019; Wing, 2018).

CPS learning instruments for children

Problem solving plays a role in CT learning. This constructivist process, which is a key process included in CT learning, can be traced back to algorithmic thinking. Algorithmic thinking was proposed by Papert (1980), a pioneer in children's programming education, based on Piaget's constructivist learning theory for children. All CPS learning approaches for children, including activities (Rodriguez et al., 2017), games (Leonard

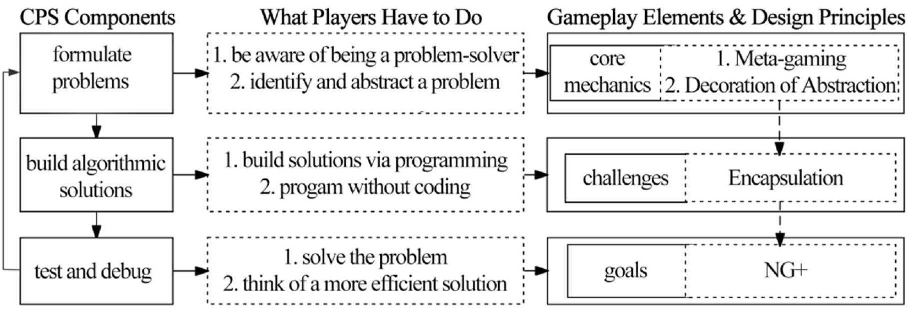

Fig. 1 The original computational puzzle design (CPD) framework

et al., 2016), software (Marcelino et al., 2018), toys (Wang et al., 2021), and robots (Bers, 2018; Chevalier et al., 2020), are designed to involve problem solving.

Papert also developed Logo, the well-known programming language for children (Papert, 1980), which provided a model for subsequent developers who programmed learning environments for children. Such environments for children aged around 6 include CHERP (Kazakoff et al., 2013), ScratchJr (Portelance & Bers, 2015; ScratchJr, 2018), TangibleK (Bers, 2018), and Lightbot (Yallihep & Kutlu, 2020). In these environments, (1) children can program by dragging and dropping command symbols and immediately execute the program to test it; (2) the program runs with the real-time actions of the character, which are consistent with the currently running commands to help the player locate errors; and (3) the programming paradigms adapt to metaphors (Fessakis et al., 2013) (e.g., children are "talking to" or "helping" a game/animated character instead of a processor), which is highly appropriate for 6-year-old children's learning. When designing the game, all of the above points related to the CPS cognitive characteristics of this age group were taken into account in this study.

CPD framework and *LittleWorld*

CPD framework

The key to CPS is problem-solving skills. These are particularly important for the design of problems in learning materials, which corresponds to the design of game puzzles in a gamified learning environment. Therefore, in our previous work (Jiang et al., 2019) we proposed a computational puzzle design (CPD) framework. However, the original CPD framework presented only preliminary design principles based on a literature review and was not supported by interventional experimental data. As shown in Fig. 1, based on the three main steps in CPS (i.e., formulating problems, building algorithmic solutions, and testing and optimizing the solutions), the original CPD framework consisted of three aspects. These aspects are discussed in the following sections and illustrated with reference to our game *LittleWorld*. *LittleWorld* enables players to explore the mysteries of the insect world, subtly cultivating their CPS skills and introducing them to analytical concepts as they help insects solve problems.

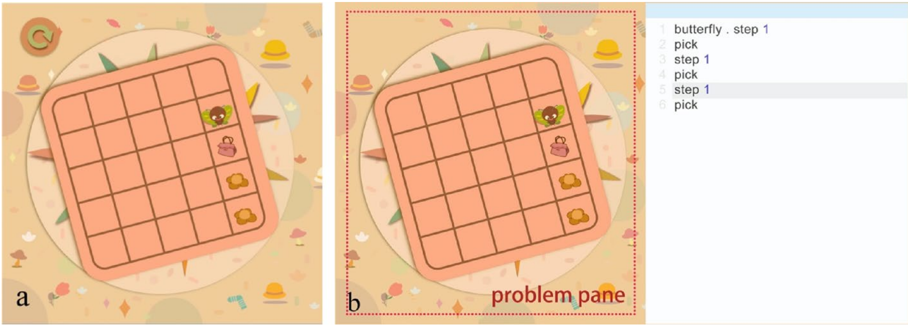

Fig. 2 Comparison of the basic game and the optimized game. **a** The basic game. **b** The optimized game based on meta-gaming

Formulating problems: meta-gaming and decoration of abstraction

We aimed to support CPS learners in formulating a problem in the core mechanics of our game. To improve the ability of players to formulate problems, a CPS game must (1) provide players with a problem that inspires them to become a problem-solver and (2) ensure that players can define the problem by representing the problem abstractly as they formulate it. To achieve these two objectives, we adopted two design principles: meta-gaming and decoration of abstraction.

Meta-gaming In most game designs, players identify themselves as a character in the game. For example, in the game *LittleWorld* (Fig. 2a), players press keys on the keyboard to play the role of a butterfly by moving and picking up various items. However, CPS game designers must also remind the players that they have an identity that is distinct from their character in the game, so that they are made to feel that they are a problem solver rather than a character who is part of the problem. Thus, we optimized the basic game using the meta-gaming principle by putting the basic game into a larger one, thereby framing the basic game as a subset problem that needed to be solved within a broader problem. Specifically, we put the basic game into the "problem pane" of the new larger game, as shown in Fig. 2b. In the new larger game, players are unable to control the game's character step by step; instead, they must observe the entire problem environment and propose a solution. Hence, the players do not identify as the butterfly but are inspired to become problem-solvers who help the butterfly. By implementing these design changes, we successfully transferred the players' identity from that of being a character in the game to being a problem-solver. The role of problem-solver is hidden in the game, unlike the role of the insect in the game's story, but players use this "observer's perspective" to help the insects solve problems.

Decoration of abstraction Two factors are important in cultivating the ability of children to identify a problem and then transform it into an abstract representation. First, the problem should be computationally solvable; second, the problem environment should require the problem-solver to conduct a process of abstraction to solve the problem. Therefore, an abstract problem based on a CPS learning objective was designed first and then the problem was contextualized in a vivid setting by decorating it with "embodied coats (i.e., insect characters, props, and scenes)." We call this design process "decoration of abstraction." As

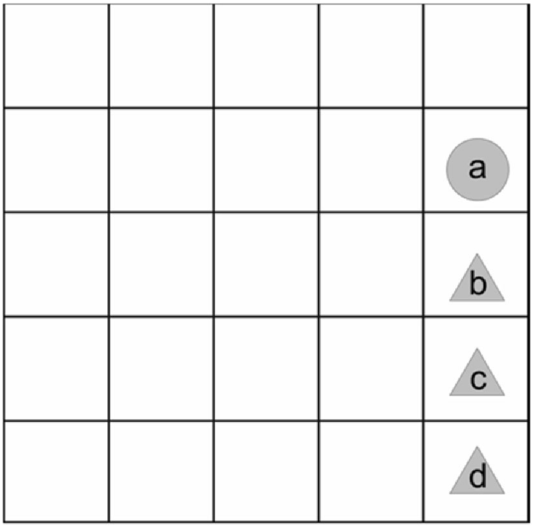

Fig. 3 A symbol-based problem prototype

shown in Fig. 3, the game *LittleWorld* demonstrates this principle in a symbol-based looping problem. The problem consists of the actions of moving and picking up three times: circle "a" should go forward and pick up the triangles "b," "c," and "d." Hence, by solving the concrete problem illustrated in Fig. 2b, children can gradually become aware of the abstract concept of "loops" through this concrete experience of repetitive operations to solve problems. In other words, by associating abstraction with concrete objects within the experience of the children, we are anchoring abstract reasoning in the children's world. In this problem-solving process, players can learn many CPS concepts, such as pattern recognition (i.e., recognizing that the essence of the problem is a loop), abstraction (i.e., abstracting the problem prototype from the concrete game), data representation (i.e., representing the problem as a symbol-based prototype as shown in Fig. 3), and generalization (i.e., transforming other problems that need to be solved by repeated actions into the problem of the loop).

Building algorithmic solutions: encapsulation

The second step in CPS is building algorithmic solutions, which occurs in the act of coding. However, this act is too advanced for young children, even when it is done in natural language rather than computer language. Moreover, research has indicated that violating syntax could represent good CPS, especially when developing creative and efficient solutions (Chen et al., 2017). Thus, CPS skills may not be relevant to coding skills. To reduce this potential barrier, we proposed the design principle of encapsulation. Encapsulation, meaning to encapsulate statements of programming code into instruction cards, was inspired by Kelleher's definition of programming as the act of assembling a set of symbols representing computational actions (Kelleher & Pausch, 2005). This design principle has been demonstrated in many programming games (e.g., *Robot Turtles*, *The Foos*) designed for young children.

Figure 4b shows how players could program by dragging instruction cards from the library and arranging them in a programming area. In Fig. 2b, the player has to code like a real programmer and spell words. For example, to make the butterfly step forward, the

Fig. 4 The optimized game based on encapsulation

player is required to spell a word and type "s," "t," "e," "p," "1." This study follows many studies that have analyzed how to reduce the cognitive load of children who are learning to spell. For example, the Picture Word Inductive Model (PWIM) was designed to attenuate the cognitive loads of K-6 children when learning spelling (Jiang & Perkins, 2013). One study investigated the effect of a digital dictionary format on the incidental acquisition by students of spelling knowledge and on their cognitive load (Liu et al., 2014). Spelling imposes a high cognitive load, and spelling work is quite difficult for preschool students. Effective means have been proposed to manage cognitive loads in spelling as a second grade school writing skill (Saada-Robert, 1999). Compared with the situation shown in Fig. 2b, in Fig. 4b, the sentence is encapsulated in the card, and the players just need to drag and drop the "↑" instruction card. Thus, the game design shown in Fig. 4b reduces children's cognitive load and allows them to focus on the logic and structure rather than on the syntax of programming. This CPS approach is more appropriate for young children (Kelleher & Pausch, 2005). The CPS skills that players can acquire in this situation include functions (e.g., using a loop to build algorithm solutions), variables (e.g., changing the number to control the number of iterations of the loop), parameters (e.g., influencing the numbers of the loops), and making and remixing (e.g., arranging different instruction cards to build a loop).

Testing: NG+

The last step in CPS is testing, which is also the goal of *LittleWorld*. Testing consists of (1) finding errors and (2) debugging to correct the errors found in the testing phase. To develop these two abilities, players implement and repair their program visually. After the players click the RUN button (Fig. 4b), the butterfly's actions are indicated as various instruction cards light up one by one. This step helps the players understand how their algorithm works. It also helps them to easily locate their mistakes because the lighting on the instruction cards pauses when the butterfly stops moving.

In complex software, debugging also refers to the development of more efficient programs, i.e., optimization. The design principle *NG*+is used to develop this kind of debugging ability. *NG*+is short for New Game Plus, a popular mode that enables players to replay the game after they complete it. In *LittleWorld*, *NG*+encourages players to rethink solutions and redesign more efficient ones based on a reward system where they gain more stars when they use fewer instruction cards. As shown in Fig. 4a, the player solves the problem with six cards, gaining one star. Figure 4b shows that by nesting "moving forward" and "picking up" into the loop, the player gains three stars by using only three cards. The *NG*+mode thus encourages players to review their solutions and adopt an analytical reasoning strategy, by which they can achieve a deeper understanding of CPS. In addition, as the players experiment with the program in this mode, they develop their CPS skills.

Above all, the CPD framework establishes a correspondence between CPS learning objectives and game components, and shows how to develop the CPS competencies of players through game play. Thus, the proposed design principles can serve both teachers and game designers in developing games to teach CPS. However, the above arguments are based on theoretical derivations, and their effectiveness and scientific validity in developing the CPS of players have yet to be confirmed.

The game

We applied design research and research-through-design methods in designing *LittleWorld* (Dixon, 2019; Gaver, 2012; Zimmerman et al., 2007). We first used the three identified CPS skills to develop the core gameplay of *LittleWorld*: (1) presenting a visual problem, (2) allowing players to design a solution, and (3) testing whether the solution works. We then integrated the analytical concepts into the game, which were chosen from the Code.Org (2010) syllabus of a CT curriculum designed for children. These analytical concepts include both declarative and procedural categories. Declarative analytical concepts, including instruction, sequence, and algorithm, belong to declarative knowledge, which refers to "what," such as specific facts consisting of concepts, theories, models, and ideas. Procedural analytical concepts, which include debugging, looping, and conditionals, belong to procedural knowledge, which refers to "how," such as rules and steps for solving problems (Berge & Hezewijk, 1999). According to the adaptive control of thought theory (Anderson, 1992), the acquisition of descriptive knowledge is a prerequisite for learning procedural knowledge. Therefore, the descriptive analytical concepts are arranged in the first few levels and the procedural analytical concepts are arranged in the latter levels, as explained below.

LittleWorld consists of five worlds, each with six levels (i.e., 30 levels in total) that gradually increase in difficulty. Each world features a different insect and focuses on different analytical concepts. In World 1, players learn the concepts of instruction, sequence, and algorithm (declarative knowledge) by helping a fly to return home. In World 2, players learn the concepts of instruction, sequence, and algorithm (declarative knowledge) by helping a dung beetle push its ball into a hole. In World 3, players learn the concept of debugging (procedural knowledge) by helping an ant move a house. In World 4, players learn the concept of looping (procedural knowledge) by helping a butterfly collect items. In World 5, players learn the concept of conditionals (procedural knowledge) by helping a bee collect nectar and make honey. The game is playable on PC and iPad. For our study, we used the PC version exclusively.

Levels introducing declarative analytical concepts

The declarative analytical concepts for the 6-year-old learners included instruction, sequence, and algorithm, as shown in Table 1. Figure 5a–d illustrates the graphic interface used to introduce the three concepts to the learners. As shown in Fig. 5a, we guided the players to manually drag the fly by one tile to complete the task. Next, as shown in Fig. 5b, the popup displayed the "↑" instruction card, which means "step forward." The players were reminded by an audio and textual explanation that, "the operation you did just now is an instruction." As shown in Fig. 5c, in the next level, the players were required to control the fly using instruction cards rather than doing so directly. The game then issued more instructions to solve more complex problems, such as the level shown in Fig. 5d. By playing a few levels, the players gradually realized that they were giving instructions to the fly. The system reminded the players, "When you give multiple instructions at a time, you're providing an 'algorithm'; when you give multiple instructions in order, you're providing a 'sequence.'".

Levels introducing procedural analytical concepts

The procedural analytical concepts include debugging, looping, and conditionals. These concepts should be integrated into a specific problem environment so that players can acquire analytical skills and concepts through solving specific problems. The concept of conditionals is used as an example. First, "conditional" is presented as "making decisions based on different conditions" (Duncan & Bell, 2015) so that children aged around 6 can understand it. Then, a problem is designed, as illustrated in Fig. 6a. The players are required to create a program that instructs the bee to collect nectar from flowers and make honey in the honeycomb. However, a yellow dot covers the target tile, preventing players from determining whether a flower or a honeycomb is beneath, and thus from deciding whether to collect nectar or make honey. At this time, the popup shown in Fig. 6b introduces a game tool and explains, "If you want to make a decision between two or more things, try 'conditionals,'" and shows players how to use the tool. After players solve the problem using the "conditional" card shown in Fig. 6c, the game reminds them, "This is a conditional that makes decisions based on different conditions."

It is worth mentioning that all of the system buttons and their corresponding operations are explained in the game teaching levels so that preschool children can understand the meaning and operation of each button. As mentioned above, all of the system buttons and their corresponding operations are explained in terms of arrows and gestures via popups with audio and textual explanations. During the trial, a research assistant was present to explain the game operation steps to the students, and to answer questions and provide assistance as needed.

Literature on CPS assessment

For the experimental part of this research, it is necessary to determine how to assess CPS. This section presents a review of relevant studies on (1) CPS assessment approaches, (2) CPS assessment content and criteria, and (3) engagement.

Table 1 Framework of the content of a CPS assessment

Category	Content	Meaning for children
CPS skills	Formulate problems	Identify the problem pattern and abstract the symbol-based prototype of the problem
	Design solutions	Develop a solution
	Test and optimize	Experiment and review the solution
Analytical concepts	Instruction	A direction or order that tells a computer to perform a specific operation
	Sequence	The order in which events, actions, etc. happen or should happen
	Algorithm	A series of instructions in order
	Debug	Identify and correct errors
	Loop	Do something repeatedly
	Conditional	Decide between two or more things

Fig. 5 Levels introducing declarative analytical concepts

Fig. 6 The level introducing conditionals

CPS assessment approaches

Several assessment approaches have been proposed in the past decade, such as the functionality-based rubric (Metcalf et al., 2021), the computational thinking test (Román-González et al., 2018), motion charts (Tang et al., 2020), the climate change model (Weintrop et al., 2014), and CPS metrics (Moreno-León & Robles, 2015; Troiano et al., 2019, 2020). However, these approaches assume that learners have developed digital objects with, for example, a platform such as Scratch (Resnick et al., 2009) and use these objects to assess their learning. In contrast to this learning-by-making process (i.e., constructionist learning), we focused on learning-by-playing (Kafai and Resnick, 2009), which is more suitable for 6-year-old learners.

Our assessment approach, inspired by Brennan and Resnick (2012), is intended to comprehensively assess the acquisition of CPS by analyzing the presence of coding blocks, collecting data on authoring details and in-process learning information through artifact-based interviews, and then assessing specific CPS concepts by asking the participants to explain, debug, and remix a project to obtain further process-in-action learning data. In our context, we analyzed how students played using metrics from the game logs, assessed specific CPS concepts via a test, and then conducted interviews with the students to explain the test data.

CPS assessment content and criteria

Because there is no standard definition of CPS and there are different learning goals for those of different age groups and disciplines, we had no normative framework for assessing CPS to serve as a reference. We used the SDARE framework by Liu et al. (2011), which consists of five components adapted from the standards of the Computer Science Teacher Association: (1) using machine-recognizable syntax, (2) analyzing data, (3) generating solutions with algorithms (a series of ordered steps), (4) abstracting and presenting problems, and (5) achieving the most efficient and effective solution. Based on these five components, we refined the aforementioned three main steps of CPS as (1) formulating problems with abstract representation, (2) building algorithmic solutions using symbol-based coding language, and (3) testing and optimizing the most effective solutions using minimal resources.

Brennan and Resnick (2012) suggested that there can be three perspectives in a CPS assessment: (1) computational concepts (i.e., key concepts in programming languages, such as loops, conditionals, events, etc.), (2) computational practices (i.e., practices engaging in the processes of artifact construction, such as abstracting, testing, reusing, etc.), and (3) computational perspectives (i.e., computational ways of understanding and describing themselves, inter-object relationships, or even the world, such as expressing and questioning). Informed by this model, and based on the CPS-analytic classification of learning concepts for young children mentioned above, we defined computational concepts in our work as the analytical concepts to solve specific problems and computational practices as the three (refined) CPS skills. Although we considered the computational perspective to be outside the scope of this study, we included it in the first CPS skill of formulating problems, such as to teach children to understand and describe the problem in a computational manner (i.e., identify and group problems that need to be solved by repetitive operations into loops) and to present problems in a symbol-based prototype, including inter-object relationships (i.e., the relationships between the relevant factors that make up the problem).

On this basis, we developed a two-component CPS framework (Table 1) with (1) CPS-related concepts to formulate problems from computational perspectives and guide a problem-solving practice in a computational manner (categorized by the three CPS skills) and (2) analytical concepts such as loops and conditionals to solve specific problems.

Engagement

The term "engagement" originated in the field of distance education research, and numerous studies have demonstrated that learning through gameplay can greatly enhance the engagement of learners (Hamari et al., 2016; Ke et al., 2016; Moon & Ke, 2020). The literature in the field of game-based learning (GBL) defines engagement in two categories: some researchers have considered GBL engagement to be about fun, playfulness, and enjoyment (Hainey et al., 2011), while others have proposed that GBL engagement involves a motivation to learn (Farrell & Moffat, 2014; Hainey et al., 2011) and have suggested that engagement is critical for student motivation and learning, both during the educational activity and afterwards (Hamari et al., 2016; Harteveld, 2011; Iliya et al., 2015). This study refers to these two types of engagement as "playing engagement" and "learning engagement," respectively.

Castell and Jenson (2003) argued that "without play, education becomes a force of compliance." Based on the GBL engagement studies described previously, we believe that game-based learning for 6-year-old learners is an aesthetic and non-utilitarian process that involves both learning and experiencing/playing. Thus, we assert that by feeling engaged in playing CPS teaching games, students (1) have an enjoyable experience; that is, playing engagement, and (2) increase their motivation to learn more (i.e., they may continue learning CPS beyond playing the game); that is, learning engagement. Given the importance of engagement in educational games and for learning in general (Huizenga et al., 2009; Sabourin & Lester, 2014), in this study we were interested in both direct learning gains (as measured by the CPS assessment and criteria in Table 1) and engagement (which may indirectly affect learning). As such, we included engagement as an assessment criterion.

The present study

The goal of this study was to determine whether games designed based on the CPD framework can enhance young children's engagement with and learning of CPS. Specifically, we attempted to answer the following question. To what extent does each CPD principle or the interactions of the principles affect children's engagement with and acquisition of CPS? Because decoration of abstraction and encapsulation are necessary in the design of CPS teaching games for 6-year-old learners, there was no need to assess these two design principles. Regarding the principle of decoration of abstraction, what players achieve in the game is what was designed into the game, and it is necessary to decorate a symbol-based question prototype based on a certain CPS concept; otherwise, players cannot learn the concept in the game. Therefore, we used the decoration-of-abstraction principle to design a CPS teaching game for every version of the game *LittleWorld*. Regarding the principle of encapsulation, because coding is too advanced for 6-year-old learners, encapsulation is fundamental for the design of non-coding programming games for 6-year-old learners. We encapsulated the coding sentences into instruction cards and then assessed the other two design principles: meta-gaming and NG+. We formulated the following hypotheses:

H1 A game that includes meta-gaming increases (a) CPS acquisition and (b) engagement more than games without meta-gaming.
H2 A game that includes NG+ increases (a) CPS acquisition and (b) engagement more than games without NG+.
H3 A game that includes meta-gaming *and* NG+ increases (a) CPS acquisition and (b) engagement more than games without meta-gaming and/or NG+.

Methods

Materials: four versions of the game

To assess the design principles of meta-gaming and NG+, we developed four versions of *LittleWorld*. The differences between the four game versions are as follows.

G1 = original *LittleWorld*, including encapsulation and decoration of abstraction
G2 = G1 with NG+
G3 = G1 with meta-gaming

G4 = G1 with NG+ and meta-gaming

In G1 (Fig. 2a), players control a character to complete a task using a keyboard. G2 is the same as G1 except for the inclusion of NG+, which is a reward system: players gain more stars when they finish a task using fewer key clicks. G1 and G3 (Fig. 4) differ in their manner of controlling the character. In the meta-game in G3, the problem is nested in the problem window of the game, which requires players to control the character by building and running a program using instruction cards. G4 is the same as G3 but includes NG+. G1 and G3 also record the numbers of stars in the game log files, which are used to evaluate students' learning results; however, the players do not see the popup window of the stars.

With these versions clarified, we explain how we evaluate our hypotheses. In line with H1, which concerns meta-gaming, we expect G3 to perform better than G1; for H2, which concerns NG+, we expect G2 to perform better than G1; and for H3, which concerns the combined effect of the principles of meta-gaming and NG+, we expect G4 to perform better than G1, G2, and G3.

6 Research contexts and participants.

The experiment was conducted in a public elementary school in Jiangsu[1] Province in China. First-grade students were chosen to be our participants because Chinese first graders are about 6 years old and therefore fit the target population of this study both physiologically and cognitively. Our participants were in five different classes of approximately 36–45 each. Each of the four classes played a different version of the game (i.e., G1–G4). To control for the positive effects of our game-based educational intervention, we asked the students in the fifth class to participate in both the pre-test and post-test without having played any version of the game. We refer to this condition as "no game." Our study was quasi-experimental in nature, with the students not randomly assigned to our conditions and the classes taught by different instructors. This grouping was adopted because the experiment was more operationally feasible with natural classroom groupings. This was because the subjects were young and needed the help of classroom teachers to provide discipline, and because it was difficult to achieve time uniformity because of the different arrangements in different classes. However, the demographics of the students were similar across the classes and the results of the pre-test showed that there were no significant differences between the classes.

The analysis included data on 202 of the 228 children who participated in the study. We excluded 24 students who missed important class sessions and 2 additional participants with cognitive learning disabilities. Regarding demographics, 40.82% of the children were female and 59.18% were male. They ranged in age from 5 to 7 years, with an average of 5.82 years.

Research design and procedure

We conducted a design-based experiment to investigate the effects of different versions of the game *LittleWorld*, which were designed based on the principles of the CPD framework. We provided a three-hour training workshop before the experiment, in which the four

[1] Jiangsu, located on the east coast of China, is among the most economically developed provinces in the country and is an important industrial base. The proportion of the provincial population with a primary level of education or higher is much higher than the national average.

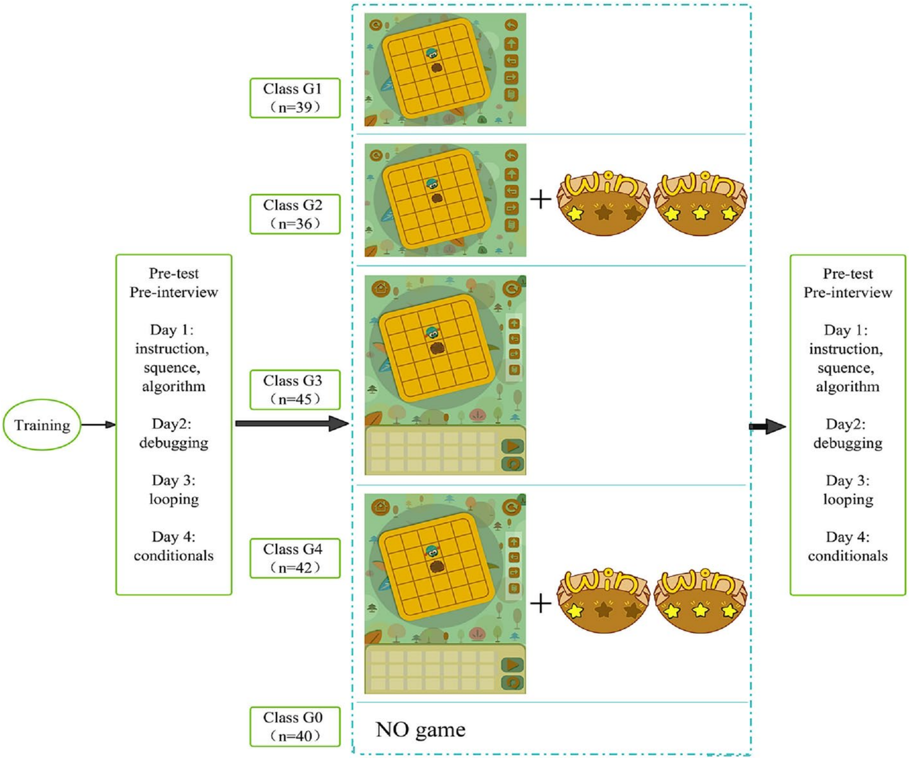

Fig. 7 Game testing protocol for all of the participants included in our study ($n=202$)

versions of the game and the learning objectives (CPS and analytical concepts) were introduced to the instructors. The instructors knew each student well both inside and outside the classroom, and were skilled at communicating with them. Believing that the research team would not be very good at guiding the emotions of the first graders and in imposing discipline when they were excited to see the game in the pilot experiment, we asked the classroom teachers to help us in these aspects. To minimize the role of the instructor, we chose people who had the same undergraduate degree to serve as instructors and asked them to follow the same instructional language in the training. After piloting the experiment, we conducted a formal experiment in the form of a curriculum. The curriculum was unveiled over four classroom sessions for each of the five classes of students (i.e., Day 1: Worlds 1–2 for instruction, sequence, and algorithm; Day 2: World 3 for debugging; Day 3: World 4 for looping; and Day 4: World 5 for conditionals). Each session lasted approximately 90 min and took place in the school's computer classroom.

Every session included six steps, as shown in Fig. 7: (1) training (because half of the students had never used computers, we taught them how to use a mouse and how to open the game), (2) pre-interview, (3) pre-test, (4) gaming, (5) post-test, and (6) post-interview. We used the pre-interviews to investigate the students' pre-understanding of the analytical concepts. We interviewed ten students chosen randomly from five classes, which was sufficient because most of the time, the students had no understanding of analytical concepts. In the pre-test, the children had to solve a problem by applying analytical concepts. During

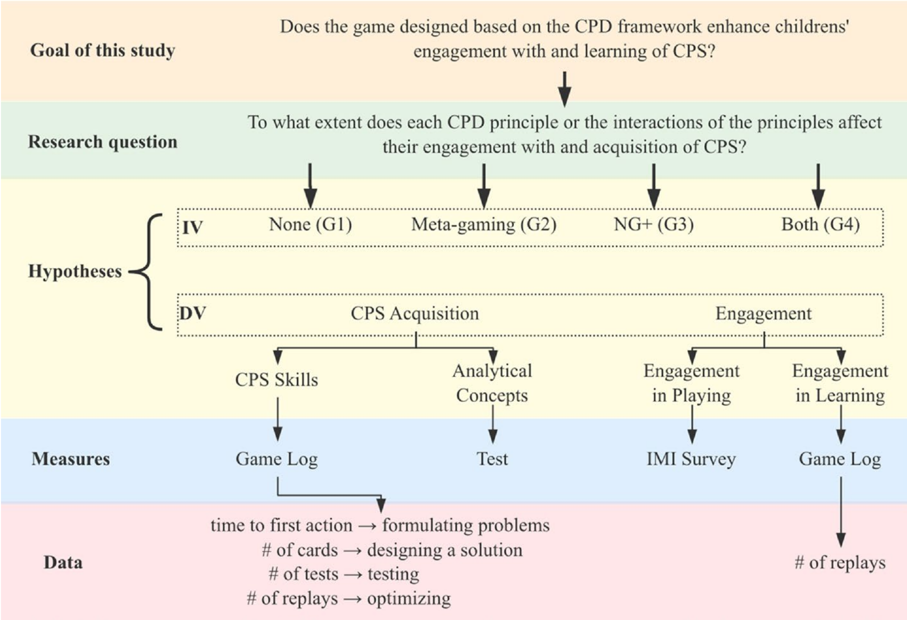

Fig. 8 Generating and evaluating the hypothesis

the gaming step, each student used a PC to play *LittleWorld*. After they played the game, the students were retested and interviewed again. In the post-interviews, we interviewed approximately 15–20 students randomly selected from each of Classes G1, G2, G3, and G4 (60–80 students in total per session). To help the children remember what they had just played, the research assistants showed the interviewees the game interface.

Overview

Figure 8 provides an overview of the study. The research questions were derived from the research objectives, and then the independent variables (different design principles and their associated effects) and dependent variables (acquisition and engagement and their four variables after splitting) were identified, after which the measures and data corresponding to the four dependent variables were listed.

Measures

We measured whether the students learned not only analytical CPS concepts (e.g., specific commands or functions) but also CPS skills. Analytical CPS concepts were observed and assessed via tests and interviews, and CPS skills were evaluated via game log data. We further measured student engagement using a survey (learning engagement) and the game logs (playing engagement).

CPS test

The CPS test consisted of six subtests, one for each analytical concept: instruction (first day), sequence (first day), algorithm (first day), debugging (second day), looping (third day), and conditionals (fourth day). Each subtest consisted of three questions. To narrow the knowledge transfer gap, the questions were first designed in the form of a game and then transformed into an abstract form. As shown in Fig. 9, there were two questions about conditionals. The problem in Fig. 9a is "help the bee to collect nectar or make honey according to the conditions under the orange dots," which is similar to the problem to be solved in *LittleWorld*. Figure 9b shows a mathematical problem involving an abstract quantitative comparison. The left panel of Fig. 9c shows students solving problems. We put the problems on the screen, the research assistant read aloud the problems on the screen, then the students solved them on the worksheet. Each class was assigned a special research assistant, namely, the instructor (see the right panel of Fig. 9c), who knew the students well and was skilled in communicating with them. The instructor could further explain the problems if a student felt confused, although during the course of the experiment no help from the instructor was needed.

Inspired by Bers' work on grading differences in performance (Dagienė & Stupurienė, 2016) and based on the pilot research, we designed the following 6-point scale, where higher scores represent more efficient solutions:

0 points: Did not attempt/other
1 point: Did not provide the expected solution
2 points: Mostly did not provide the expected solution
3 points: Partially provided the expected solution
4 points: Mostly provided the expected solution
5 points: Provided the expected solution

Figure 10 shows sample answers from the pilot study to the problem shown in Fig. 9, which correspond to different scores on the 6-point scale:

Figure 10a: 0; the student did not try to solve the problem, and just wrote "I can't" in Chinese.
Figure 10b: 1; the student gave a solution in Chinese characters but did not solve the problem.
Figure 10c: 2; the student gave a solution with instructions that learned in *LittleWorld* but did not think about different conditionals.
Figure 10d: 3; the student recognized the concept of conditionals but failed to make decisions based on different conditionals.
Figure 10e: 4; the student gave a correct outline of the solution and drew three instructions in the required order: "step forward," "loop two times," and "conditional."
Figure 10f: 4.5; the student's answer was almost correct.
Figure 10g: 5; the student gave the complete answer: "move one step forward," then nest "conditional" and "move one step" in the "loop," and then run the loop two times.

Thus, for each subtest, the students could obtain a maximum score of 15 points (i.e., three questions with a maximum score of five points each). Because scoring requires

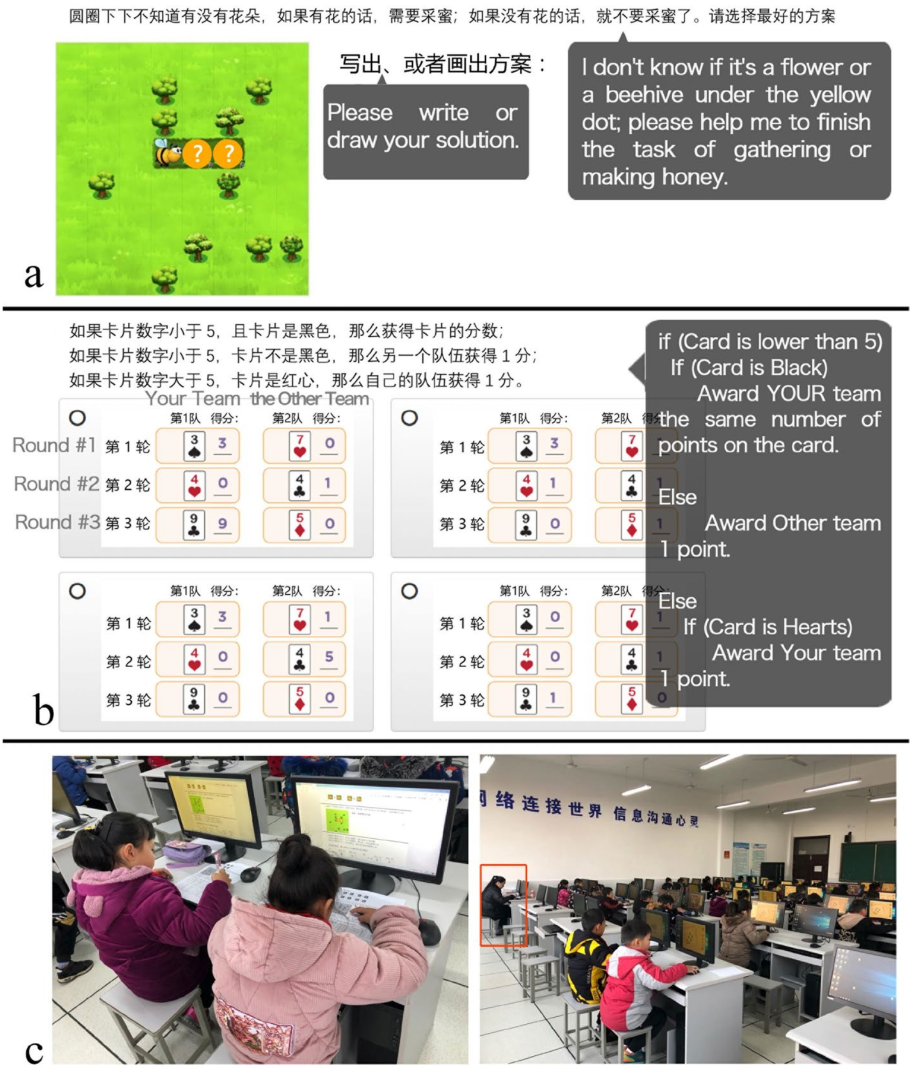

Fig. 9 Problems on the conditional test. **a** A route-planning problem that is concrete and similar to the game. **b** A quantitative comparison problem that is more abstract than the game-like question. **c** Students solving the problem in **a**

interpretation, two researchers discussed and assigned all of the scores.To obtain insights into the students' understanding of the CPS concepts, we conducted informal individual interviews with them before and after their gameplay. In each interview, we first asked them to explain a particular CPS concept, such as instruction. If they did not understand the concept, we followed up with a term that helped to explain the concept (e.g., "order" for instruction). Our interview data were used to supplement our quantitative results. Here, therefore, the interview data are not analyzed separately in the results section but discussed in the context of the CPS test results.

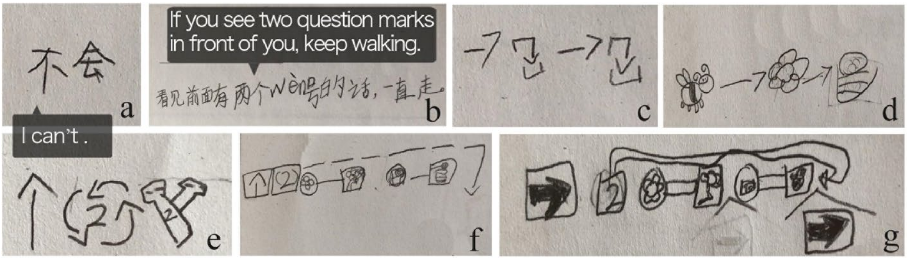

Fig. 10 Sample of the students' answers in the pilot study

Game log data

We used the game log data to measure the CPS skills. The students could perform many types of in-game actions, so it was necessary to analyze which actions were related to CPS. We identified four key actions related to CPS:

1. Formulating a problem: the process between the initial view of the problem and taking the initial action to address the problem, measured as the total time before the first action is taken. More time taken suggested that the players put more effort into understanding the problems.
2. Designing a solution: dropping and arranging the instruction cards in the programming area. This action was only possible in the meta-gaming versions (G3, G4) and was measured in terms of the total number of cards used. The use of more cards suggested that the players put more effort into trying to determine the best solution.
3. Testing and Optimizing

   (a) Testing: clicking the "Run" button and modifying the program. This action was only possible in the meta-gaming versions (G3, G4) and was measured as the total number of tests. Running more tests indicated that the players worked harder to solve the problems.
   (b) Optimizing: optimizing the solution after completing the level, measured as the total number of replays. Making more effort to improve the solutions suggested that the players aimed to better understand how to solve the problems.

In *LittleWorld*, the logs recorded the number of actions taken on every level and the timestamp of each action. Based on the above mapping, we extracted what the students were doing on each level, whether formulating a problem or designing a solution. We extracted these gameplay patterns from the raw game log files across all levels for each player.

Engagement measures

We also explored the students' engagement with serious games in terms of engagement with playing (enjoyment) and engagement with learning. For playing, we used the Children IMI Interest/Enjoyment Scale (MacLeod et al., 2017), which consists of seven statements scored on a 5-point Likert scale, where 1 represents "Totally disagree" and 5 represents

“Totally agree.” We translated this scale into Mandarin Chinese and included smiley faces to help the students fill out the survey (see the appendix), a strategy that other researchers have used with young audiences (Goudas et al., 1994).

Engagement in learning is typically viewed as occurring when students display the desire to be successful in the learning process (Saeed & Zyngier, 2012). In addition, there is another concept that is similar to, but different from, it—”persistence”—which measures the ability to overcome potential difficulties (Denis et al., 2010; Schunk et al., 2013). For this aspect of engagement, we used the same metric as is used for optimizing: the number of plays or the total number of replays after completing a level. If students optimized their performance, they showed persistence and a desire to do better. As such, this metric also showed whether the students were persistent in their learning.

Results

We present the results in this section regarding CPS acquisition of analytical concepts in terms of the mastery of analytical concepts, as measured through the tests and interviews, and the mastery of CPS skills, as measured through the game data logs. We then describe the results for engagement, as measured using the survey and game data logs.

CPS acquisition of analytical concepts

We analyzed the analytical concepts according to *declarative knowledge* (“what”), which includes instruction, sequence, and algorithm; and *procedural knowledge* (“how”), which includes debugging, looping, and conditionals (see Table 1).

Declarative analytical concepts

We verified that the pre-test scores and the treatment were independent ($p = .82$), and we found that the pre-test scores had a significant effect on the post-test scores ($p < .001$). Therefore, analysis of covariance (i.e., one-way ANCOVA) was used to evaluate the effects of different principles as the independent variable with the total pre-test scores of declarative concepts (i.e., instruction, sequence, and algorithm for a total maximum score of 45) as the covariate and the corresponding total post-test scores as the dependent variables. The results showed that there was a significant difference between the groups in the post-test scores when we controlled for the pre-test scores, $F(4, 202) = 21.87$, $p < .001$.

To examine our hypotheses, we conducted planned contrasts using a Bonferroni correction for multiple comparisons. The planned comparisons revealed that none of the principles performed significantly differently, indicating that the significant overall effect presented was because of the game itself. All of the game conditions performed significantly differently from the “no game” condition.

To measure improvements from the pre-test to the post-test for each concept, we performed paired t-tests on the average scores for instruction, sequence, and algorithm for the combined treatment groups (i.e., G1–G4). Each declarative knowledge concept improved significantly. The effect was much stronger for algorithm, as evidenced by a larger Cohen’s d (*Cohen’s d for Instructi*on = 0.54, *Sequence* = 0.38, *Algorithm* = 1.24) and there were greater differences in the pre-/post-test scores. Furthermore, the pre-test scores for algorithm ($M = 3.48$, $SD = 2.13$) were much lower than those for instruction ($M = 11.31$, $SD = 2.49$) and sequence ($M = 10.93$, $SD = 3.44$), suggesting that the students had already mastered the concepts of instruction and sequence but did not understand the concept of algorithm.

The interview data revealed the same difference in the students' understanding of instruction, sequence, and algorithm. In the pre-interviews, the students understood "direct someone to do something" but not instruction; however, in the post-interviews, the students made statements such as "instruction is 'directing someone to do something'" and "instruction means to make others do something as you asked." These responses provide evidence that the students had already acquired ideas about what instruction means and that they understood it to be equivalent to "directing someone to do something." For sequence, we observed a similar implicit understanding (Reber, 1996). In contrast, the students lacked such an understanding of the concept of algorithm. This is not surprising, as it is one of the most difficult concepts to understand (Charoula et al., 2016; Choi et al., 2017).

Procedural analytical concepts

The procedural analytical concepts tested in our study were looping, debugging, and conditionals. We analyzed the data using an ANCOVA (i.e., with the game version as the treatment variable, the total pre-test scores as the covariate, and the corresponding total post-test scores as the response variable), and the assumption concerning the independence of covariates was met ($p = .43$). The results revealed a significant overall effect ($p < .001$).

Planned contrasts showed that the post-test performance for G3 and G4 (i.e., the meta-gaming versions) was better than for G1 and G2 (i.e., the non-meta-gaming versions). There was no difference between G3 and G4 or between G1 and G2. These results suggest that meta-gaming was conducive to the mastery of procedural concepts (e.g., loop) and that NG+ had no effect. Indeed, G1 and G2 did not even perform better than the "no game" condition.

The individual results for looping, debugging, and conditionals were similar. Here, we use the results for looping as an example. The differences between the pre- and post-test scores for each group are presented in Fig. 11. Consistent with the planned contrast analysis, the percentage of students whose scores improved by more than 4 points was higher for G3 (25.81%) and G4 (36.36%) than for the other conditions. Moreover, as shown in Fig. 12, only the players in G3 and G4 had scores in the 12–15 range, indicating near-perfect scores for every question.

The interview data confirmed these results. During the pre-interviews, the players had no idea about the meanings of looping, debugging, and conditionals, which is understandable given that these concepts are difficult to learn from daily life, unlike the concept of instruction. Take conditionals as an example. During the post-interviews, the players in G1 and G2 thought of conditionals in terms of "guess what is in there," "guess right or wrong," and "identify true or false." As illustrated in Fig. 13, the players had to guess whether there was a flower or a honeycomb under the yellow dot. If they guessed correctly, the bee completed the task, but did not do so otherwise. Thus, they seemed to translate the gameplay of conditionals into a guessing game.

In contrast, players in G3 and G4 thought of conditionals in the following statement: "if it's sunny, don't take an umbrella; if it's rainy, take an umbrella." Although none spoke in terms of "if–else," they clearly comprehended conditionals as "making decisions according to different conditions" (Duncan & Bell, 2015) and were able to apply this concept to solve

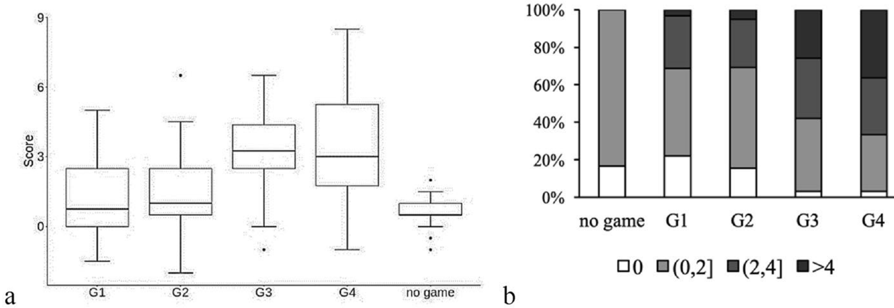

Fig. 11 Mean score differences between the pre- and post-tests. **a** Boxplots. **b** Bar charts

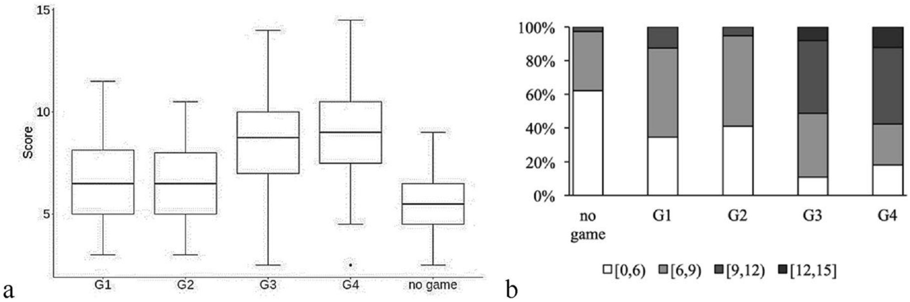

Fig. 12 Post-test scores for looping. **a** Boxplots. **b** Bar charts

problems outside of the game. As illustrated in Fig. 6, in G3 and G4, guessing is not possible. Players must instead tell the bee, "if it is a flower, collect nectar; if it is a honeycomb, make honey." As such, the gameplay enforces the use of conditionals.

Acquisition of CPS skills

Table 2 shows descriptive statistics on the measured CPS skills with the metrics derived from the game log data. Overall, we see that players in G3 and G4 (i.e., the meta-gaming versions) took more time to formulate problems. Players in G4 (i.e., meta-gaming with the NG + version) tried more solutions, based on the number of cards used, and tested more solutions than G3. G2 players played the most after completing the game, followed by G4 players. Indeed, G2 players ($M = 55.67$, $SD = 27.14$) continued to play approximately 10 times more than G1 players ($M = 5.56$, $SD = 6.05$). These results indicate the effects of meta-gaming on the CPS skill "formulate problems" and of NG + on "optimizing," and the combined effects of meta-gaming and NG + on "designing solutions" and "testing."

To test these effects, we conducted ANOVAs for each metric, and we found significant differences between the versions (see Table 3). Planned contrasts with a Bonferroni correction largely confirmed the aforementioned observations and indicated that G4 players took longer than G3 players before their first action ($p = .005$). For "number of cards used," there was a significant difference between G4 and G3 ($p = .003$), showing the positive effect of NG +. Unlike the analytical concepts results, these game log data results demonstrate that

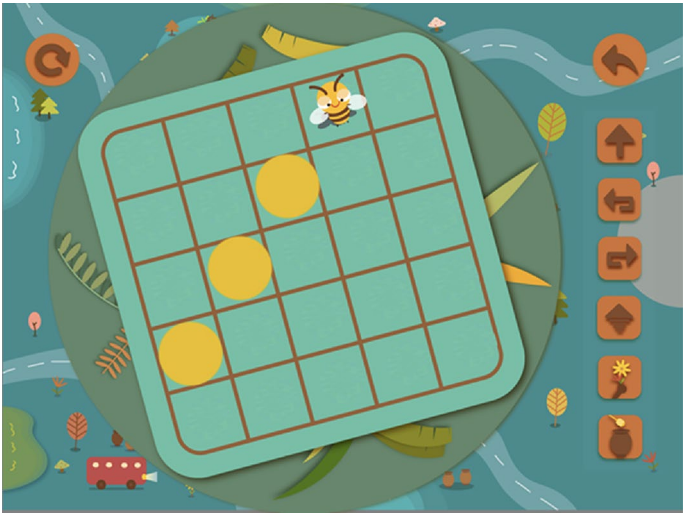

Fig. 13 Conditional level in G1 and G2. The player must guess whether a flower or a honeycomb is under each yellow dot

Table 2 Descriptive statistics of CPS skills measured with metrics from the game log data in M (SD) over all levels played

Game version	Time to first action	# of cards	# of tests	# of replays
G1	319 (480)	–	–	5.56 (6.05)
G2	276 (189)	–	–	55.67 (27.14)
G3	760 (360)	353 (79.9)	143 (34.2)	7.47 (10.79)
G4	1069 (480)	405 (126.4)	171 (44.7)	13.76 (13.42)

G1 and G2 did not allow for testing, and thus have no statistics for the "number of tests" metric

Table 3 ANOVA results for the four metrics of CPS skills

Metrics	Df1	Df2	F	*p*
Time to first action	3	146	34.48	<.001
# of cards	3	146	343.5	.001
# of tests	1	70	8.75	.004
# of replays	3	146	77.36	<.001

For consistency, we also conducted ANOVA for the metric "times of testing"

NG+ affected outcomes and suggest that the combination of meta-gaming with NG+ in G4 also had an effect.

To further investigate these results, we examined the role of NG+ (i.e., the star reward system) in G3 and G4. We conducted independent-sample t-tests on the rewards obtained for the levels focused on teaching declarative analytical concepts (i.e., Worlds 1 and 2) and those focused on teaching procedural analytical concepts (i.e., Worlds 3–5). In both cases, the G4 players earned more stars than the G3 players (see Table 4). We further found

Table 4 Independent-sample *t*-tests results for rewards in analytical concepts

Concept	M (SD)		*t*	*p*	Cohen's *d*
	G3	G4			
Declarative	19.33 (8.55)	24.69 (7.01)	−3.04	.003	0.347
Procedural	21.62 (7.76)	28.23 (6.11)	−4.03	< .001	0.485

Table 5 Results of the linear regression for analytical concepts for the G3 and G4 players

Variable	Declarative			Procedural		
	Parameter	Std. error	*p*	Parameter	Std. error	*p*
Constant	15.43	2.16	< .001	6.34	1.41	< .001
Pre-test scores	0.88	0.24	< .001	0.19	0.10	.07
Number of stars	0.49	0.05	< .001	0.62	0.04	< .001

that the rewards for G3 and G4 were positively related to the total post-score for declarative analytical concepts ($r = .76$, $p < .001$) and the total post-score for procedural analytical concepts ($r = .89$, $p < .001$). These results imply that the number of stars is a predictor of students' test performance.

To confirm this implication, we conducted linear regressions to predict the post-scores on the declarative and procedural analytical concepts using the pre-test scores and the number of stars as predictors (see Table 5). Both models were statistically significant and suitable (adjusted $R^2 = .63$ and .80, respectively). As expected, we found that the G4 players gained more stars than the G3 players and that the number of stars obtained predicted a student's test performance. However, although the effect was strong, as demonstrated, it was not sufficient to indicate a significant difference between G4 and G3 on the analytical concepts (Fig. 14).

Engagement

Regarding engagement, we first assessed the students' enjoyment of the four versions using the intrinsic motivation inventory (IMI) (MacLeod et al., 2017). The results are presented in the boxplot shown in Fig. 15. The solid points and thick lines inside the box represent the average and median, respectively, of each group's enjoyment scores. An ANOVA did not show a statistically significant difference between the game groups on the IMI scores, $F(3, 140) = 0.676$, $p = .568$. This result indicates that while the distributions differed somewhat, notably for G4, all of the game types were well received by the students. These findings challenge the widely held notion that reward systems lead to a better gameplay experience (Chen et al., 2010; Klasen et al., 2012; Walsh & Anderson, 2012).

Regarding engagement in terms of learning, we have already presented the results for the number of replays, which suggest that NG+can effectively promote student engagement. Thus, although NG+does not have significant effects on the learning of analytical concepts and enjoyment, it is a simple way to motivate students to try to do better (Chen et al., 2010), which is an important CPS skill.

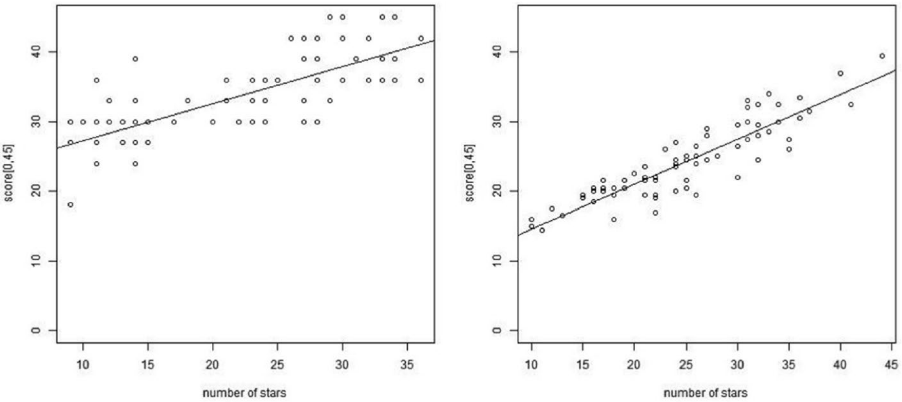

Fig. 14 Scatter diagram of the stars and total scores (max. 45) for declarative (left) and procedural (right) for the G3 and G4 players

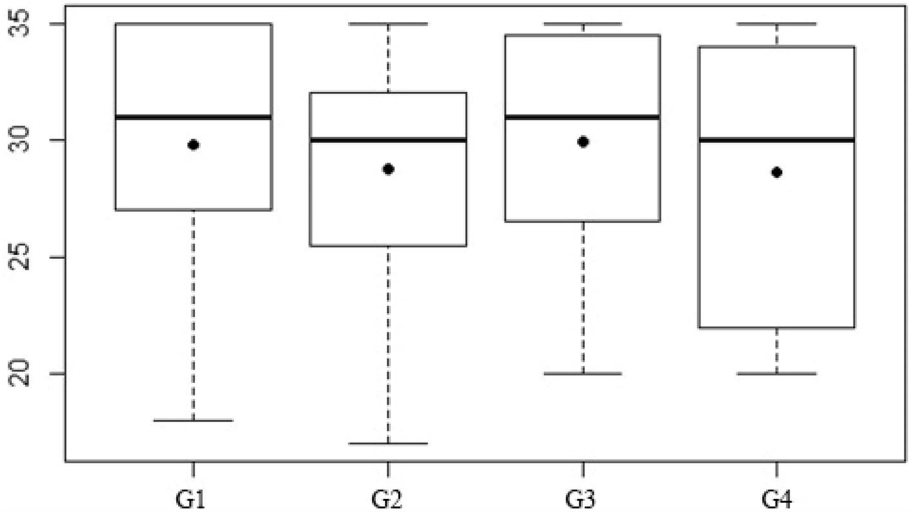

Fig. 15 Boxplots of the Interest/Enjoyment scale for each game version

Discussion

Helping children develop computational thinking (CT) is an important goal of education in the artificial intelligence era, and CT teaching games hold great potential for helping children develop CPS. It is likely that CT teaching games will continue to proliferate (Harteveld et al., 2014). In addition, with advances in game development technology, teachers have been encouraged to customize programming games to suit their students' level of development in computational thinking (Finkelstein et al., 2009; Galeos et al., 2020; Jung et al., 2019; Lode et al., 2013). However, the vast majority of teachers understand computational thinking and learning science but not games and design; thus, many programming games are based only on experience and lack the guidance of scientific design theories (Hutchison, 2012; Leake & Lewis, 2017; Qian et al., 2018; Yadav et al., 2016a, 2016b), which can lead to unsatisfactory game instruction or hinder teachers from developing their own gamified teaching resources. However, most of the research in the field has focused on the feasibility of using games to develop computational thinking and on the learning effects of games; there is little research on the design of programming games, perhaps because of the difficulty of achieving interdisciplinary collaborations between computer education and game design (Grover & Pea, 2013; Hsu et al., 2018; Lindberg et al., 2019; Miljanovic & Bradbury, 2018; Zhang & Nouri, 2019). To address this issue, this study developed a CPD framework to guide the design of CPS teaching games and *LittleWorld*, a CT teaching game based on the CPD framework, to demonstrate the feasibility of the CPD framework in guiding design practice. In addition, the game was used as experimental material for an intervention experiment to ensure that the experimental data accurately reflected the scientific validity and effectiveness of the design framework. In this section, we first discuss the empirical findings of our quasi-experimental design study and then discuss the implications for the design of CT teaching games.

Research-through-design findings

Following the preliminary design framework study (Jiang et al., 2019), this study developed game samples based on the theoretical framework and used them as experimental materials for intervention experiments, and then optimized and refined the original framework based on the experimental data. In this section, the findings from the experimental data are first presented. Then, the optimized design framework based on these findings, the CPD 2.0, is discussed. Table 6 provides an overview of the conclusions related to the research hypotheses that we can draw based on the data analysis and results of our design research and research-through-design in the design of CPS teaching games (Dixon, 2019; Gaver, 2012; Zimmerman et al., 2007). We discuss the findings for both CPS acquisition and engagement.

CPS acquisition findings

CPS acquisition involves analytical concepts and CPS skills (see Table 1). Regarding analytical concepts, we found that including meta-gaming, NG+, or both did not enhance learning of declarative concepts; however, a game with meta-gaming was conducive to the mastery of procedural analytical concepts. There may have been a ceiling effect regarding instruction and sequence, as many students already intuitively understood the meanings of these concepts (±11 out of 15 on the pre-test). In contrast, for algorithm, students did not have such a prior understanding, and their understanding increased after they played a game with meta-gaming. Although the rewards (number of stars) were strongly related to the players' test scores, our results highlight that the inclusion of NG+ was not why they obtained such scores. This suggests that high-scoring students tend to pursue (and obtain) more stars, whereas low-scoring students did not. Thus, from the analytical concepts perspective, we only find support for H1 in the context of procedural analytical concepts.

Table 6 Findings on the effects of design principles on CPS acquisition and engagement

Principle	CPS acquisition						Engagement	
	Analytical concepts		CPS skills				Playing	Learning
	Procedural	Declarative	Formulating problem	Designing solution	Testing	Optimizing		
Meta-gaming(G3)	H1	–	H1	H1	H1	–	–	–
NG+(G2)	–	–	–	NA	NA	H2	–	H2
Both(G4)	H1	–	H3	H3	H3	H2	–	H2

The table shows the effects of meta-gaming (G3), NG+(G2), and the combination of meta-gaming and NG+(G4) on CPS acquisition (divided into analytical concepts and CPS skills) and engagement. We tested three hypotheses: H1: G3 > G1; H2: G2 > G1; H3: G4 > G1 and G2 and G3. The table indicates what hypotheses are supported. G1 is the game version without meta-gaming or NG+

We found that meta-gaming was more conducive for promoting CPS skills, with the exception of optimizing, because we found that NG+ was responsible for cultivating such skills. However, the combination of meta-gaming with NG+ promoted CPS skills even more. This suggests that while meta-gaming is capable of promoting certain CPS skills more than others (designing and testing solutions could not be measured for G1 and G2), NG+ plays an important role in the acquisition of these skills. The observation that NG+ fostered optimization behavior suggests that the inclusion of meta-gaming does not motivate players to improve. In sum, for CPS skills, we found support for H1 and H3 on formulating the problem, designing solutions, and testing, and for H2 on optimizing.

In summary, the results demonstrate that meta-gaming plays a key role in students' learning of procedural knowledge. Thus, to apply such knowledge, it is beneficial to position players as problem-solvers. This positioning directly articulates certain aspects of CPS (i.e., designing and testing solutions) and indirectly encourages players to be more reflective on how to formulate problems and to be more thoughtful about what they are doing (i.e., using an algorithm to solve a problem). Such encouragement can be further fueled by NG+, as the results indicated that NG+ can encourage players to create better solutions, especially in combination with meta-gaming. On its own, NG+ seems to be limited to a trial-and-error approach to improve solutions.

Engagement findings

This research explored engagement in playing and learning for CPS teaching games. We found that the game did not lead to better perceived engagement for players regardless of whether it included meta-gaming, NG+, or both. This finding subverts the widely acknowledged view that a star system facilitates the gameplay experience in entertainment games (Huang et al., 2010; Johansen et al., 2018; Lewis et al., 2016; Siu & Riedl, 2016; Wang & Sun, 2012). Our results should be understood in the context of students' never having played games in the classroom; for most such students, any type of game would

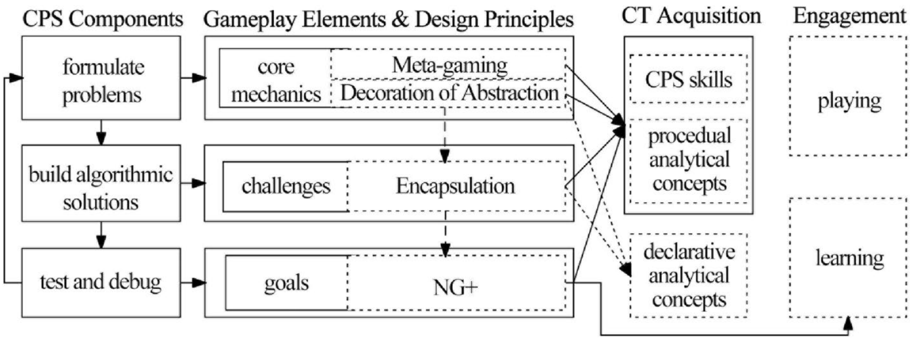

Fig. 16 The revised computational puzzle design framework: CPD 2.0. The solid arrows and boxes are a clearer indication than the dashed arrows and boxes of how the principles affect CPS acquisition and engagement

be engaging. However, the gameplay data indicated that the game versions that included NG+led to a better learning engagement: the star system motivated players to replay the game more times to create a more efficient solution and earn higher rewards. This is exactly what the star system was designed for (Deci et al., 2001; Filsecker & Hickey, 2013; Goh et al., 2017; Huang et al., 2010; Richter et al., 2015). Thus, we can conclude that NG+increases engagement in CPS teaching games from a learning perspective.

Together, the findings regarding CPS acquisition and engagement suggest that (1) meta-gaming is key to a deeper understanding of CPS and (2) NG+can moderate the effect of meta-gaming and increase players' engagement to learn. However, our work does not provide evidence that the increased engagement to learn fostered by NG+translates into actual learning beyond practicing the skills of optimizing solutions.

Revised CPD framework

Based on this empirical work, a more systematic design framework can be developed from the original computational puzzle design (CPD) framework (see Fig. 1). Compared with the original framework, the revised framework, which we refer to as CPD 2.0, refines certain design principles to facilitate the learning of certain CPS knowledge or improve engagement. This revised framework consists of four aspects (see Fig. 16): (1) the CPS components, which are the core of CT learning; (2) the gameplay elements and the design principles (see the section on the CPD Framework); (3) the CPS concepts mentioned in Table 1, classified into two dimensions: procedural knowledge (CPS skills and procedural analytical concepts) and declarative knowledge (declarative analytical concepts); and (4) engagement in playing and learning. As we argued, decoration of abstraction and encapsulation are fundamental principles for the design of every type of CPS teaching game. The results for G1 (the version without meta-gaming and NG+) suggest that even a basic version of the game fosters CPS acquisition and is engaging. As discussed, our results regarding meta-gaming and NG+suggest particular causal relationships, shown in Fig. 15. Designers and researchers can use this revised framework to design and assess CPS teaching games, respectively.

In the future, the relationships in this framework will be further examined. Other principles will also be explored.

Design implications

The findings of this study provide design implications concerning (1) the application of meta-gaming and (2) how research on designing serious games should be conducted.

Applying meta-gaming

Research (Rhodes et al., 2017) has suggested that some serious games are effective in teaching procedural knowledge. Our key finding is that children can master procedural knowledge more effectively in games that include meta-gaming than in those that do not. Although the CPD framework is meant for CPS teaching games, meta-gaming may be applicable to other types of games for children. The constraints of vocabulary and language comprehension make it difficult for children to understand procedural knowledge through verbal expression (LeFevre et al., 2006; Rittle-Johnson & Schneider, 2014). However, children can cultivate the problem-solving ideas and skills contained in procedural knowledge by applying them to solve problems (Surif et al., 2012) beyond language. Take, for example, the function "conditionals" in this study. It is almost impossible for young children to understand the formal definition of "conditionals," which "perform different computations or actions depending on whether a programmer-defined Boolean condition evaluates to true or false" (American National Standards Institute, 2007). However, the concept of conditionals can be conveyed to children by customizing a problem in a game and guiding them to analyze and represent the problem, abstract the problem pattern, and select conditions to solve the problem. Thus, game puzzle design is key to facilitating young learners' acquisition of procedural knowledge. Here, we summarize our method of translating procedural knowledge into game puzzles. This method consisted of four steps:

1. Translate procedural knowledge into actions that the target audience can understand.
2. Build a symbol-based problem prototype that needs to be solved using these actions.
3. Decorate the symbol-based abstract problem as a concrete game.
4. Put the concrete game into the problem pane of a new "big" game and guide players to solve the problem.

Meta-gaming may have additional implications that require further study. For example, games are especially effective for situated learning, putting the learner into authentic learning environments (Gee, 2007). However, although through meta-gaming learners may be better able to reflect on and articulate the problem, this may come at the cost of being (emotionally) distanced from the task. Thus, whether meta-gaming should be applied may depend on the educational objectives and the problem context.

Research insight on serious game design

Through the *LittleWorld* project, we gained the insight that unless games are intentionally designed to achieve a certain learning outcome, their effects are unlikely to emerge clearly (Harteveld, 2011; Lee et al., 2012). In other words, it would be impossible to implement serious games using a “gamification” design strategy (i.e., first create a generic game shell, then inject certain questions or interactions to teach content, which is perceived as a disruption and distraction to the flow of serious games). This insight emphasizes two aspects of designing serious games: (1) a learning objective concerning “what to assess” should be intentionally designed into the gameplay, and (2) how the learning outcomes should be measured, which involves “how to assess,” should be considered when the learning objectives are designed. The latter includes considerations of what data to collect to evaluate the learning outcomes, both from the game and from other sources. In our case, for example, we used tests to measure analytical CPS concepts, and we used game logs to measure CPS skills.

In conclusion, design and assessment should be regarded as interdependent rather than separate. The intentional design (of games and assessments) has been advocated by other researchers (Harteveld & Sutherland, 2015) and is part of game-based assessment approaches that are grounded in evidence-centered design (Kim et al., 2016; Mislevy et al., 2003). Related to this entanglement of design and assessment, our work further demonstrated that integrating the two meaningfully using a “research-through-design” approach is a promising practice for serious game design and research.

Limitations and future directions

Our work was conducted in the context of Chinese elementary education and classroom settings. Because we worked with young children, we have no reason to believe that the effects would be different in other countries; however, children who have had more exposure to educational games may respond differently. Besides, although the physiological development of children is similar across countries, cultural and educational differences may affect the generalizability of this study across countries and regions.

The classroom setting restricted the time available for the study. This forced us to choose certain assessment activities (i.e., the CPS test and survey) that may not have accurately measured CPS acquisition and engagement. We addressed this potential limitation by complementing the CPS test data with interviews and game log data, which confirmed our findings. Nevertheless, future work could involve inviting students to complete “unplugged” activities (Dagiene & Stupurienė, 2016; Rodriguez et al., 2017; Thies & Vahrenhold, 2013) as a form of assessment. Future studies could also derive different assessment metrics from game log data.

Because of the limited time and the request to try to complete all of the levels, the students may have been less inclined to try to improve their scores, which may have affected the results regarding NG+. Additionally, because of the more complicated interface and actions needed, the players using the meta-gaming versions (G3, G4) generally took longer to complete the levels, which likely deterred them from replaying. It would be interesting to observe the results when students are not confined to classroom times and can play elsewhere, such as at home.

Our study was quasi-experimental, with students not randomly assigned to our conditions and the classes taught by different instructors. However, the demographics of the students were similar across the classes and we were able to minimize the role of the instructor because the activity was led and orchestrated by the research team and all of the instructors were trained. There is a potential issue with validity, in that those students who did not solve the problem in the quiz, or only partially solved it, did so because they did not know the principle or did not know how to phrase it as desired by the question. To truly address this issue, a separate study to measure validity is needed, which will be part of the next step of the study.

Finally, there is much more to CT than learning related concepts and skills (Jiang et al., 2019; Kazimoglu et al., 2011). It is also important to change students' attitudes regarding CS and to instill in them behaviors and mindsets similar to those of a computer scientist when solving problems. In this way, research on the design of CT teaching games will not only develop learners' CPS abilities at the conceptual level but also establish computer scientist-like behaviors and mindsets from the operational and conceptual perspectives.

Conclusions

This study contributes to the literature by (1) demonstrating the feasibility of the CPD framework in guiding practice by designing the CPS teaching game *LittleWorld* and (2) validating the effectiveness of the CPD framework through an intervention experiment. This study thus makes both theoretical and practical contributions. First, the optimized and more systematic CPD framework fills the gap in research on designing programming games, which was mentioned in the literature review. Second, the detailed analysis of the design of game cases combined with design principles can provide a reference paradigm for teachers who lack experience in games and design. Third, this study was an interdisciplinary effort involving researchers in the fields of computer education and game design. It seeks to break down barriers between science and art disciplines and to innovate and expand research on CPS games, including on the designing of serious games, in terms of perspective and methodology.

Appendix 1: survey questions used to measure engagement

(a) I enjoyed doing this activity very much.
(b) This activity was fun to do.
(c) I thought that this was an exciting activity.
(d) This activity held my attention very well.
(e) I would describe this activity as very interesting.
(f) I thought that this activity was quite enjoyable.
(g) While I was doing this activity, I was thinking about how much I enjoyed it.

Chinese version:

标准		完全 不赞同	一般 不赞同	中立 态度	一般 赞同	完全 赞同
						
1	我非常喜欢这个游戏					
2	这个游戏很有趣					
3	我认为这是一个令人兴奋的游戏					
4	这个游戏很吸引我的注意力					
5	我认为这个活动非常有趣					
6	我觉得这个活动很有趣					
7	当我玩这个游戏时，我在想我是多么喜欢它。					

Funding This study is supported by the Humanities and Social Sciences Youth Foundation, Ministry of Education of the People's Republic of China, 22YJCZH068, to Xina Jiang, Province-Level Educational Reformation Research, 2021091 and National Natural Science Foundation of China, 62107020, to Yuqin Yang

Declarations

Conflict of interest The authors do not have any conflicts of interest.

Ethical approval All applicable international, national, and/or institutional guidelines for the care and use of animals were followed.

Informed consent All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards. Informed consent was obtained from all individual participants involved in the study.

References

American National Standards Institute. (2007). American National Standard – Programming Language FORTRAN. https://www.fortain.com

Anderson, J. R. (1992). Automaticity and the ACT theory. *The American Journal of Psychology, 105*(2), 165–180. https://doi.org/10.2307/1423026

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Bers, M. U. (2018, December 15). *The TangibleK robotics program: Applied computational thinking for young children*. Retrieved from http://ecrp.uiuc.edu/v12n2/bers.html/

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. 25.

British Council. (2013). The education systems of england & wales, scotland and northern ireland. https://www.britishcouncil.org

Castell, S. D., & Jenson, J. (2003). OP-ED serious play. *Journal of Curriculum Studies, 35*(6), 649–665. https://doi.org/10.1080/0022027032000145552
Charoula, A., Joke, V., Andrew, F., Mary, W., Margaret, C., Joyce, M.-S., & Jason, Z. (2016). *A K-6 computational thinking curriculum framework: implications for teacher knowledge on JSTOR*. Retrieved from https://www.jstor.org/stable/jeductechsoci.19.3.47?seq=1#metadata_info_tab_contents
Chen, P., Kuo, R., Chang, M., & Heh, J. (2010). Designing a trading card game as educational reward system to improve students' learning outcome—A Christian University.
Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001
Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education, 7*(1), 39. https://doi.org/10.1186/s40594-020-00238-z
Choi, J., Lee, Y., & Lee, E. (2017). Puzzle based algorithm learning for cultivating computational thinking. *Wireless Personal Communications, 93*(1), 131–145. https://doi.org/10.1007/s11277-016-3679-9
Code.org—Teach computer science. (n.d.). Retrieved December 25, 2018, from https://studio.code.org/courses?view=teacher
Dagienė, V., & Stupurienė, G. (2016). Bebras—A sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education, 15*(1), 25–44. https://doi.org/10.15388/infedu.2016.02
Deci, E. L., Koestner, R., & Ryan, R. M. (2001). Extrinsic rewards and intrinsic motivation in education: Reconsidered once again. *Review of Educational Research, 71*(1), 1–27. https://doi.org/10.3102/00346543071001001
Denis Bedardchristelle Lisondaniel Dallenoeel & Boutin. (2010). Predictors of student's engagement and persistence in an innovative PBL curriculum: Applications for engineering education. *The International Journal of Engineering Education, 26*(3), 511–522.
Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A meta-synthesis of research. *ACM Transactions on Computing Education, 19*(3), 1–35. https://doi.org/10.1145/3277565
Dixon, B. (2019). Experiments in experience: Towards an alignment of research through design and John Dewey's pragmatism. *Design Issues, 35*(2), 5–16. https://doi.org/10.1162/desi_a_00531
Duncan, C., & Bell, T. (2015). A pilot computer science and programming course for primary school students. *Proceedings of the Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/2818314.2818328
Farrell, D., & Moffat, D. (2014). Applying the self determination theory of motivation in games based learning. *Proceedings of the European Conference on Games-Based Learning, 1*, 118–127.
Febrian, A., Lawanto, O., Peterson-Rucker, K., Melvin, A., & Guymon, S. E. (2018, June 23). Does everyone use computational thinking? A case study of art and computer science majors. In *2018 ASEE Annual Conference & Exposition*. Retrieved from https://peer.asee.org/does-everyone-use-computational-thinking-a-case-study-of-art-and-computer-science-majors
Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016
Filsecker, M., & Hickey, D. (2013). Incentives in educational games: A multilevel analysis of their impact on elementary students' engagement and learning. *Computer-Supported Collaborative Learning Conference, CSCL, 1*, 208–215.
Finkelstein, S. L., Nickel, A., Harrison, L., Suma, E. A., & Barnes, T. (2009). cMotion: A new game design to teach emotion recognition and programming logic to children using virtual humans. *IEEE Virtual Reality Conference, 2009*, 249–250. https://doi.org/10.1109/VR.2009.4811039
Galeos, C., Karpouzis, K., & Tsatiris, G. (2020). Developing an educational programming game for children with ADHD. *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA)*. https://doi.org/10.1109/SMAP49528.2020.9248458
Gaver, W. (2012). What should we expect from research through design? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2207676.2208538
Gee, J. P. (2007). *Good video games and good learning: Collected essays on video games, learning and literacy* (1st ed.). Peter Lang Inc.
Goh, D.H.-L., Pe-Than, E. P. P., & Lee, C. S. (2017). Perceptions of virtual reward systems in crowdsourcing games. *Computers in Human Behavior, 70*(C), 365–374. https://doi.org/10.1016/j.chb.2017.01.006

Goudas, M., Biddle, S., & Fox, K. (1994). Achievement goal orientations and intrinsic motivation in physical fitness testing with children. *Pediatric Exercise Science, 6*(2), 159–167. https://doi.org/10.1123/pes.6.2.159

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27.

Hainey, T., Connolly, T. M., Stansfield, M., & Boyle, E. A. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. *Computers & Education, 56*(1), 21–35. https://doi.org/10.1016/j.compedu.2010.09.008

Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., & Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior, 54*, 170–179. https://doi.org/10.1016/j.chb.2015.07.045

Harteveld, C., Smith, G., Carmichael, G., Gee, E., & Stewart-Gardiner, C. (2014). A design-focused analysis of games teaching computer science. *Proceedings of Games+ Learning+ Society*, *10*.

Harteveld, C. (2011). *Triadic game design: Balancing reality, meaning and play*. Springer Science & Business Media.

Harteveld, C., & Sutherland, S. C. (2015). The goal of scoring: Exploring the role of game performance in educational games. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2702123.2702606

Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007). Computational thinking. *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education—SIGCSE '07*. https://doi.org/10.1145/1227310.1227378

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Huang, W.-H., Huang, W., & Tschopp, J. (2010). Sustaining iterative game playing processes in DGBL: The relationship between motivational processing and outcome processing. *Computers & Education, 55*, 789–797. https://doi.org/10.1016/j.compedu.2010.03.011

Huizenga, J., Admiraal, W., Akkerman, S., & ten Dam, G. (2009). Mobile game-based learning in secondary education: Engagement, motivation and learning in a mobile city game. *Journal of Computer Assisted Learning, 25*(4), 332–344. https://doi.org/10.1111/j.1365-2729.2009.00316.x

Hutchison, L. F. (2012). Addressing the STEM teacher shortage in American schools: Ways to recruit and retain effective STEM teachers. *Action in Teacher Education, 34*(5–6), 541–550. https://doi.org/10.1080/01626620.2012.729483

Iliya, A., Jabbar, A., & Felicia, P. (n.d.). *1 gameplay engagement and learning in game- based learning: A systematic review*.

Irwin, V., Zhang, J., Wang, X., Hein, S., Wang, K., Roberts, A., York, C., Barmer, A., Mann, F. B., Dilig, R., Parker, S., Nachazel, T., Barnett, M., & Purcell, S. (n.d.). *Report on the condition of education 2021*. 43.

Jiang, X., Harteveld, C., Huang, X., & Fung, A. Y. H. (2019). The computational puzzle design framework: A design guide for games teaching computational thinking. *Proceedings of the 14th International Conference on the Foundations of Digital Games*. https://doi.org/10.1145/3337722.3337768

Jiang, X., & Perkins, K. (2013). *A conceptual paper on the application of the picture word inductive model using Bruner's constructivist view of learning and the cognitive load theory., 3*(1), 10.

Johansen, S.-G., Årsand, E., & Hartvigsen, G. (2018). *Making computer games that can teach children with Type 1 diabetes in rural areas how to manage their condition*. Retrieved from https://ep.liu.se/konferensartikel.aspx?series=ecp&issue=151&Article_No=2

Jona, K., Wilensky, U., Trouille, L., Horn, M., Orton, K., Weintrop, D., & Beheshti, E. (n.d.). *Embedding computational thinking in science, technology, engineering, and math (CT-STEM)*. 5.

Jung, H., Kim, H. J., So, S., Kim, J., & Oh, C. (2019). TurtleTalk: An educational programming game for children with voice user interface. *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3290607.3312773

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture, 1*(1), 36–40. https://doi.org/10.1177/1555412005281767

KaleliÏOğlu, F., Gülbahar, Y., & Kukul, V. (2016). *A framework for computational thinking based on a systematic research review*. 14.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245–255.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2011). Understanding computational thinking before programming: developing guidelines for the design of games to learn introductory programming through game-play. *International Journal of Game-Based Learning, 1*(3), 30–52. https://doi.org/10.4018/ijgbl.2011070103

Ke, F., Xie, K., & Xie, Y. (2016). Game-based learning engagement: A theory- and data-driven exploration: Game-based learning engagement. *British Journal of Educational Technology, 47*(6), 1183–1201. https://doi.org/10.1111/bjet.12314

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys, 37*(2), 83–137. https://doi.org/10.1145/1089733.1089734

Kim, Y., Almond, R., & Shute, V. (2016). Applying evidence-centered design for the development of game-based assessments in physics playground. *International Journal of Testing, 16*, 1–22. https://doi.org/10.1080/15305058.2015.1108322

Klasen, M., Weber, R., Kircher, T. T. J., Mathiak, K. A., & Mathiak, K. (2012). Neural contributions to flow experience during video game playing. *Social Cognitive and Affective Neuroscience, 7*(4), 485–495. https://doi.org/10.1093/scan/nsr021

Knochel, A. D., & Patton, R. M. (2015). If art education then critical digital making: Computational thinking and creative code. *Studies in Art Education, 57*(1), 21–38. https://doi.org/10.1080/00393541.2015.11666280

Lavigne, H., Lewis-Presser, A., Cuellar, L., Vidiksis, R., & Ferguson, C. (n.d.). *AHA! Island was developed by WGBHs children's media and education group*. 52.

Leake, M., & Lewis, C. M. (2017). Recommendations for designing CS resource sharing sites for all teachers. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3017680.3017780

Lee, T. Y., Mauriello, M. L., Ingraham, J., Sopan, A., Ahn, J., & Bederson, B. B. (2012). CTArcade: Learning computational thinking while training virtual characters through game play. *Proceedings of the 2012 ACM Annual Conference Extended Abstracts on Human Factors in Computing Systems Extended Abstracts—CHI EA '12*, 2309. https://doi.org/10.1145/2212776.2223794

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

LeFevre, J.-A., Smith-Chant, B. L., Fast, L., Skwarchuk, S.-L., Sargla, E., Arnup, J. S., Penner-Wilger, M., Bisanz, J., & Kamawar, D. (2006). What counts as knowing? The development of conceptual and procedural knowledge of counting from kindergarten through grade 2. *Journal of Experimental Child Psychology, 93*(4), 285–303. https://doi.org/10.1016/j.jecp.2005.11.002

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology, 25*(6), 860–876. https://doi.org/10.1007/s10956-016-9628-2

Lewis, Z. H., Swartz, M., & Lyons, E. (2016). What's the point? A review of reward systems implemented in gamification interventions. *Games for Health Journal*. https://doi.org/10.1089/g4h.2015.0078

Lin, S.-Y., Chien, S.-Y., Hsiao, C.-L., Hsia, C.-H., & Chao, K.-M. (2020). Enhancing computational thinking capability of preschool children by game-based smart toys. *Electronic Commerce Research and Applications, 44*, 101011. https://doi.org/10.1016/j.elerap.2020.101011

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. *British Journal of Educational Technology, 50*(4), 1979–1995. https://doi.org/10.1111/bjet.12685

Liu, C.-C., Cheng, Y.-B., & Huang, C.-W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education, 57*(3), 1907–1918. https://doi.org/10.1016/j.compedu.2011.04.002

Liu, T.-C., Fan, M.H.-M., & Paas, F. (2014). Effects of digital dictionary format on incidental acquisition of spelling knowledge and cognitive load during second language learning: Click-on vs. key-in dictionaries. *Computers & Education, 70*, 9–20. https://doi.org/10.1016/j.compedu.2013.08.001

Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? A systematic literary review. [Physics]. http://arxiv.org/abs/1703.07659

Lode, H., Franchi, G. E., & Frederiksen, N. G. (2013). Machineers: Playfully introducing programming to children. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. https://doi.org/10.1145/2468356.2479483

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

MacLeod, J., Yang, H. H., & Xiang, Z. (2017). Understanding college students' intrinsic motivation and social interdependence in intercultural computer-supported collaborative learning between USA and China. *The Asia-Pacific Education Researcher, 26*(3), 205–217. https://doi.org/10.1007/s40299-017-0341-6

Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem solving in the context of computational thinking. *Infinity Journal, 8*(2), 109. https://doi.org/10.22460/infinity.v8i2.p109-116

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*. https://doi.org/10.1145/1352135.1352260

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning computational thinking and scratch at distance. *Computers in Human Behavior, 80*, 470–477. https://doi.org/10.1016/j.chb.2017.09.025

Metcalf, S. J., Reilly, J. M., Jeon, S., Wang, A., Pyers, A., Brennan, K., & Dede, C. (2021). Assessing computational thinking through the lenses of functionality and computational fluency. *Computer Science Education*. https://doi.org/10.1080/08993408.2020.1866932

Miljanovic, M. A., & Bradbury, J. S. (2018). A review of serious games for programming. In S. Göbel, A. Garcia-Agundez, T. Tregel, M. Ma, J. BaalsrudHauge, M. Oliveira, T. Marsh, & P. Caserman (Eds.), *Serious games* (pp. 204–216). Springer International Publishing.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *ETS Research Report Series, 2003*(1), i–29. https://doi.org/10.1002/j.2333-8504.2003.tb01908.x

Moon, J., & Ke, F. (2020). In-game actions to promote game-based math learning engagement. *Journal of Educational Computing Research, 58*(4), 863–885. https://doi.org/10.1177/0735633119878611

Moreno-León, J., & Robles, G. (2015). Dr. Scratch: A web tool to automatically evaluate scratch projects. *Proceedings of the Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/2818314.2818338

OECD, et al. (2017). Digital Economy Outlook. https://www.oecd.org

Olgun, Sadik. (2017, April 25). *Computational thinking conceptions and misconceptions: Progression of preservice teacher thinking during computer science lesson planning | SpringerLink*. Retrieved from https://link.springer.com/chapter/https://doi.org/10.1007/978-3-319-52691-1_14

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation, 10*(3), 187. https://doi.org/10.1504/IJMLO.2016.077867

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Petri, G., & Gresse von Wangenheim, C. (2017). How games for computing education are evaluated? A systematic literature review. *Computers & Education, 107*, 68–90. https://doi.org/10.1016/j.compedu.2017.01.004

Portelance, D. J., & Bers, M. U. (2015). Code and tell: Assessing young children's learning of computational thinking using peer video interviews with ScratchJr. *Proceedings of the 14th International Conference on Interaction Design and Children IDC '15*, 271–274. Doi: https://doi.org/10.1145/2771839.2771894

Qian, Y., Hambrusch, S., Yadav, A., & Gretter, S. (2018). Who needs what: Recommendations for designing effective online professional development for computer science teachers. *Journal of Research on Technology in Education, 50*(2), 164–181. https://doi.org/10.1080/15391523.2018.1433565

Reber, A. S. (1996). Implicit learning and tacit knowledge: An essay on the cognitive unconscious. In *Implicit learning and tacit knowledge*. Oxford University Press. Retrieved from https://oxford.universitypressscholarship.com/view/https://doi.org/10.1093/acprof:oso/9780195106589.001.0001/acprof-9780195106589

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Rhodes, R. E., Kopecky, J., Bos, N., McKneely, J., Gertner, A., Zaromb, F., Perrone, A., & Spitaletta, J. (2017). Teaching decision making with serious games: An independent evaluation. *Games and Culture, 12*(3), 233–251. https://doi.org/10.1177/1555412016686642

Richter, G., Raban, D., & Rafaeli, S. (2015). *Studying gamification: The effect of rewards and incentives on motivation* (pp. 21–46). https://doi.org/10.1007/978-3-319-10208-5_2

Rittle-Johnson, B., & Schneider, M. (2014). *Developing conceptual and procedural knowledge of mathematics* (R. Cohen Kadosh & A. Dowker, Eds., Vol. 1). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199642342.013.014

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing computational thinking in CS unplugged activities. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3017680.3017779

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments, 28*(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction, 18*, 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004

Saada-Robert, M. (1999). Effective means for learning to manage cognitive load in second grade school writing: A case study. *Learning and Instruction, 9*(2), 189–208. https://doi.org/10.1016/S0959-4752(98)00044-9

Sabourin, J. L., & Lester, J. C. (2014). Affect and engagement in game-based learning environments. *IEEE Transactions on Affective Computing, 5*(1), 45–56. https://doi.org/10.1109/T-AFFC.2013.27

Saeed, S., & Zyngier, D. (2012). How motivation influences student engagement: A qualitative case study. *Journal of Education and Learning, 1*(2), 252. https://doi.org/10.5539/jel.v1n2p252

Schunk, D. H., Meece, J. R., & Pintrich, P. R. (2013). *Motivation in education: Pearson new international edition course smart eTextbook: Theory, research, and applications, 4/E*. Retrieved from http://catalogue.pearsoned.co.uk/educator/product/Motivation-in-Education-Pearson-New-International-Edition-CourseSmart-eTextbook-Theory-Research-and-Applications-4E/9781292049236.page

ScratchJr—Home. (2018, Dec 15). Retrieved from http://www.scratchjr.org/

Selby, C., & Woollard, J. (2013). *Computational thinking: The developing definition* [Monograph]. University of Southampton (E-prints). Retrieved from https://eprints.soton.ac.uk/356481/

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Siu, K., & Riedl, M. O. (2016). Reward systems in human computation games. *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*. https://doi.org/10.1145/2967934.2968083

Spieler, B., Kemény, F., Landerl, K., Binder, B., & Slany, W. (2020). The learning value of game design activities: Association between computational thinking and cognitive skills. *Proceedings of the Workshop on Primary and Secondary Computing Education*. https://doi.org/10.1145/3421590.3421607

Sullivan, A., Bers, M., & Pugnali, A. (2017). The impact of user interface on Young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice, 16*(1), 171–193.

Summary Report of the 2012 UIS Innovation Metadata Collection. UIS Information paper. (2013). UNESCO Institute for Statistics.

Surif, J., Ibrahim, N. H., & Mokhtar, M. (2012). Conceptual and procedural knowledge in problem solving. *Procedia—Social and Behavioral Sciences, 56*, 416–425. https://doi.org/10.1016/j.sbspro.2012.09.671

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education, 148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

ten Berge, T, & van Hezewijk, R. (1999). *Procedural and declarative knowledge—An evolutionary perspective*.

Thies, R., & Vahrenhold, J. (2013). On plugging "unplugged" into CS classes. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/2445196.2445303

Troiano, G. M., Chen, Q., Alba, Á. V., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G., & Harteveld, C. (2020). Exploring how game genre in student-designed games influences computational thinking development. *Proceedings of the CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376755

Troiano, G. M., Snodgrass, S., Argımak, E., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G., & Harteveld, C. (2019). Is my game ok Dr. Scratch? Exploring programming and computational thinking development via metrics in student-designed serious games for STEM. *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. https://doi.org/10.1145/3311927.3323152

Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computational thinking in a learning environment. [Cs]. http://arxiv.org/abs/1212.0750

Walsh, M. M., & Anderson, J. R. (2012). Learning from experience: Event-related potential correlates of reward processing, neural adaptation, and behavioral choice. *Neuroscience and Biobehavioral Reviews, 36*(8), 1870–1884. https://doi.org/10.1016/j.neubiorev.2012.05.008

Wang, H., & Sun, C.-T. (2012). *Game reward systems: Gaming experiences and social meanings*.

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2021). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education and Development, 32*, 1–23. https://doi.org/10.1080/10409289.2020.1759012

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. *Intelligent Technologies for Interactive Entertainment*. https://doi.org/10.1007/978-3-319-08189-2_3

Wing, J. M. (2018, Dec 15). *Computational thinking*. ResearchGate. Retrieved from https://www.researchgate.net/publication/274309848_Computational_Thinking

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016a). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Yadav, A., Hong, H., & Stephenson, C. (2016b). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends, 60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/1953163.1953297

Yallihep, M., Kutlu, B. (2020). Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and Information Technologies*, *25*, 1237–1254. https://doi.org/10.1007/s10639-019-10008-2

Yu, M., & He, X. (2011). Design And Development Of Chinese Primary And Secondary New Type School Uniform. *Advanced Materials Research, 331*, 679–682. https://doi.org/10.4028/www.scientific.net/AMR.331.679

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/1240624.1240704

Xina Jiang is Assistant Professor in the Faculty of Digital Media at Beijing Normal University, Bei Jing. Her research and teaching focuses on serious game, interactive design, and game user research.

Casper Harteveld is Associate Professor of Game Design at Northeastern University. His research focuses on using games to study and improve decision-making, and through these efforts both to advance our knowledge and to engage a broad cross-section of people globally about societal issues.

Yuqin Yang is an associate professor of the learning sciences at the Faculty of Artificial Intelligence in Education in Central China Normal University. Her research interests include pedagogy and assessment of knowledge building, data science education, learning analytics, and collaborative learning.

Anthony Fung is Professor in the School of Journalism and Communication at the Chinese University of Hong Kong and Professor in the School of Arts and Communication at Beijing Normal University. His research interests include game studies, popular culture, and digital media studies.

Xinyuan Huang is Professor of Digital Media at Communication University of China. His research interests include virtual environments, human–computer interaction, and interactive simulation.

Shihong Chen is Professor of Network and New Media in the School of Applied Arts and Sciences at Beijing Union University. Her current research interests are computer applications, computer education and digital media technology.

Authors and Affiliations

Xina Jiang[1] · Casper Harteveld[2] · Yuqin Yang[3] · Anthony Fung[4] · Xinyuan Huang[5] · Shihong Chen[6]

✉ Xina Jiang
xenajiang@bnu.edu.cn

✉ Yuqin Yang
yangyuqin@ccnu.edu.cn

Casper Harteveld
c.harteveld@northeastern.edu

Anthony Fung
anthonyfung@cuhk.edu.hk

Xinyuan Huang
hxy@cuc.edu.cn

Shihong Chen
Csh398@126.com

[1] Beijing Normal University, 19 Xinjiekouwai St., Beijing 100875, China

[2] Northeastern University, 360 Huntington Ave, Boston, MA 02115, USA

[3] Central China Normal University, No.152, Luoyu Road, Wuhan 430079, Hubei, China

[4] The Chinese University of Hong Kong, Central Ave, Hong Kong, SAR, China

[5] Communication University of China, 1 Dingfuzhuang St., Beijing 100107, China

[6] Beijing Union University, 197 Beitucheng West St, Beijing 100083, China