

# Game development software engineering: digital educational game promoting algorithmic thinking

Kornchulee Sungkaew, Piyamas Lungban, Sirinya Lamhya

Program in Multimedia Technology and Animation, Faculty of Science and Technology, Suratthani Rajabhat University, Suratthani, Thailand

## Article Info

### Article history:

Received Jul 11, 2021

Revised May 24, 2022

Accepted Jun 20, 2022

### Keywords:

Algorithmic thinking

Digital educational game

Game development software engineering

Human-computer interaction

Software development

## ABSTRACT

The purpose of this study is to create a digital educational game that promotes algorithmic thinking for elementary school students. However, the processes of game development differ from traditional software development which cannot guarantee its effectiveness in terms of human-machine interfaces. In this article, we propose a new game development software engineering (GDSE) as a model for game development. This new model aims to complement and mitigate the shortcomings of traditional software development. The principles of human-computer interaction are now incorporated in the new model. The GDSE includes design, development, usability inspection, game experience evaluation, educational value evaluation and release. It was used as a research method to develop a game that promotes algorithmic thinking for children. The results of this study are not only a digital educational game that promotes algorithmic thinking for children but also a new game development life cycle that guarantees the performance of digital games in terms of usability enhancement, game experience and educational value.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Kornchulee Sungkaew

Program in Multimedia Technology and Animation, Faculty of Science and Technology, Suratthani

Rajabhat University

222 Moo 9, Surat-Nasan Road, Khuntalay, Mueng, Suratthani, 84100, Thailand

Email: kornchulee.sun@sru.ac.th

## 1. INTRODUCTION

Thailand's education system has been modernizing to prepare young students to have 21st-century skills. Thai students are expected to be innovators rather than innovative users. They should be trained to think more systematically. One of the key skills in being an innovator is programming or coding skill [1]. Although some children may not desire to be programmers, programming skill can build a foundation for systematic thinking which children can apply to solve problems in daily life.

The Institute for the Promotion of Teaching Science and Technology (IPST), the autonomous agency under the Ministry of Education, has therefore improved the basic education core curriculum [2]. The subject of information and communication technology has been transferred to computing science. Content and methods of teaching programming skill are included in the new curriculum and tailored to each child level. For elementary school students, unplugged activity is applied to teach programming concept. The unplugged activity has been implemented in Thai classrooms since May 2018 in the computing science subject.

Unplugged activity is a way to teach programming concept to children from kindergarten to elementary school. Instead of teaching a programming language, fun activities [3]; for example, use arrow cards and maps to plan a trip to a friend's house. It is used to create an understanding of the fundamental principles of computational thinking. However, it can be argued that there are some limitations of unplugged

activity. This includes using wasteful resources such as printed cards and the need for a conductor. Children are unable to practice skills on their own. The activity should be under the control of the teacher. Some parents cannot support their child's proper guidance when doing an unplugged activity at home. Significantly, in the coronavirus disease (COVID-19) situation to maintain social distance, the activity should not be played in the classroom.

One computational thinking skill that can be practiced in elementary school students through unplugged activities is algorithmic thinking. Practicing algorithmic thinking skill should be independent of any programming languages, especially for students who do not have a background in information science education. Students should be able to solve problems even if they have no preliminary knowledge of programming languages. Tools for practicing algorithmic thinking skill should be in visualization and allow students to enter information freely [4].

The idea of applying digital games has arisen. It can reduce the use of papers. It can be used to practice algorithmic thinking skills outside classrooms without the need for teachers or parents. Digital games are ideal tools for young learners. They are familiar with the gameplay. Digital games have graphics that stimulate the learner's interest [5]. Feedbacks in digital games allow learners to know their learning outcomes immediately. It is easy to access. Many digital games have been discovered to develop the brain or some skills for players such as [6]. As mentioned above, researchers have come up with the idea of transforming the unplugged activity into a digital game to make it easier for students to understand the topic of computational thinking. In particular, algorithmic thinking skill can be enhanced. Children can relate the algorithms to programming in the future.

However, the game development process is different from traditional software development. Some studies that have adopted traditional software development as a framework for game development [7] have not been able to certify the performance of games in terms of human-machine interfaces [8]. The most effective model for the game development process could not be identified [9]. The gaps of these models were found that they excluded processes in human-computer interaction [8], [10]. It could be claimed that unexpected problems could arise from the unsatisfactorily designed interfaces. Therefore, this study involves the principles of human-computer interaction (HCI) in the game development process: usability inspection and user experience (UX) evaluation, as these are different activities to expand the quality of interactive technologies [11]. In this article, we will discuss the digital game development process, which includes the HCI principles to the traditional software development. Eventually, based on our experience, a new framework for game development software engineering (GDSE) has been proposed. In conclusion, to solve the limitations of the unplugged activity to train children's algorithmic thinking skill, this study aimed to create a digital educational game (DEG) that promotes algorithmic thinking for elementary school students (aged 7-9) based on the design of the unplugged activity. In addition, to solve the restrictions of existing game development models, this study aims to propose a new GDSE that can certify the performance of digital games in terms of human-machine interface.

## 2. PROPOSED MODEL

The process of game development differs from traditional software development due to multidisciplinary factors such as aesthetic, education, and humanity. Therefore, game development is a complicated process [8], [9]. Some research has applied traditional software development as a framework for the game development process life cycle. It is then presented as a game development model. A specific model, known as the game development life cycle (GDLC), is proposed to carry out game development [7]. It is a six-phase process life cycle consisting of initiation (create a rough concept), pre-production (create and revise game designs, then create a prototype), production (create assets and source codes, combine the two elements and customize the prototype), testing (test the game's functionality and conduct internally the game usability and playability), beta (implement test externally to get feedback for bugs), and release (release to the public). Trista and Rusli [12] developed a game for learning Indonesian history by following the software development processes as a game-based learning method. The method consists of problem identification (indicate the need to improve the study of Indonesian history), literature review (review methods to solve similar problems), system design (design the application), construction (define the game structure, app flows, user interfaces, and game assets), testing (conduct functional tests), and evaluation (assess the perceived ease of use of the app).

It could be argued that the limitation of [7]'s GDLC model is the insufficient integration of HCI principles. Although HCI principles such as game usability and playability are included in the testing phase, they focus solely on functional testing and internal operations. The result of the test is a bug report, while the results of nonfunctional tests such as user experience reports were not taken into account in the study. To create a game, human psychology is required. A good user interface (UI), user experience (UX), and

gameplay experience (GX) are all essential to keep players engaged [10], [13]. For the game-based learning method of [12], it can be claimed that the inclusion of HCI principles is insufficient in terms of the educational value of games. This application was measured the perceived ease of use, though it was not measured the ability to improve a player's learning experience. In [14]–[16] found that the utilization of virtual worlds in game-based learning as well as rich UIs with a plethora of controls, such as wireless sensors, might confuse novice users from their educational goals and even lead them to succumb to accomplish desired learning objectives, thus student feedbacks should be considered to evaluate their experience. Abarkan *et al.* [17] states that to create possible modifications to the learning game, it must be tested with a sample of learners. We, therefore, consider HCI principles including usability inspection, game experience evaluation, and educational value evaluation (if DEGs) as essential phases in game development. Based on the pieces of literature and our experience in the field of HCI, we propose game development software engineering (GDSE), a blend of software engineering processes and HCI principles. The GDSE life cycle is divided into six main phases: design, development, usability inspection, game experience evaluation, educational value evaluation (if DEGs), and release as shown in Figure 1.

Our new GDSE model is introduced for game developers as a framework to create better games in terms of usability enhancement, game experience and educational value. This model applies the pre-production and production phases of traditional software engineering. The GDSE model differs from traditional software engineering in respect of post-production phases, especially the testing phase. It is not just the functional testing procedures of traditional software engineering, but also the testing techniques used in the HCI field; usability inspection, game experience evaluation, and educational value evaluation; are included in the post-production phases.

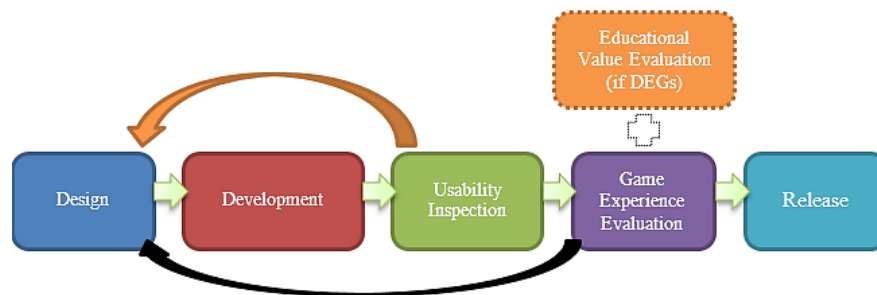


Figure 1. The proposed game development software engineering

### 3. RESEARCH METHOD

#### 3.1. Design: identifying the design of the unplugged activity

Twenty-two second-year students studying the multimedia and animation laboratory 2 course were assigned to identify features of Hunger Solver. Hunger Solver is the name of an unplugged activity used in computing science subject for elementary students [18]. It is a card game designed as a classroom activity to encourage children on algorithmic thinking. It allows students to break down a long journey into a sequence of short steps. Algorithmic thinking can be practiced if students are able to plan sequences.

- Content: students have a simple task; they need to take the two children to the ice cream shop.
- Map card: the map card has a predefined starting point for the two children and an ending point (the ice cream shop) in Figure 2(a). The teacher or the class conductor needs to print a large number of maps with different predefined starting and ending points. One map is for one level. Once children can pass a level, the teacher changes the map.
- Arrow cards: students create a sequence of arrows to create a path from the starting point to the ending point as shown in Figure 2(b).
- Gameplay: students are allowed to view a map showing the locations of the two children and the ice cream shop. They are given arrow cards and assigned to plan a step-by-step itinerary for the two children to travel to the ice cream shop. Once the time set by the teacher expires, the planned route must be done. The teacher asks if anyone else came up with a different path that would work. The teacher informs feedbacks whether they could route the characters to get the ice cream. The teacher changes the map with the shuffle of the two children's location and an ice cream shop. Then the students have a new experiment for thinking. The cake shop exemplifies the obstacles two children must avoid during their journey. This means that students need to plan a safer route to the destination.

Some limitations of the unplugged activity were found. It is unmanageable. The unplugged activity was created as a classroom activity for students to practice algorithmic thinking. Thus, a teacher or instructor or moderator who comprehends the gameplay is required for the session. The teacher has to alter different maps and shuffle the location of the characters on the map once students are able to complete the level. Also, feedbacks or comments on the decisions on the designated path are given to students. Therefore, playing an unplugged activity on their own or outside of the classroom may seem difficult for children because of these requirements. In addition, the unplugged activity is unapproachable. Children who are not in the classroom cannot participate in the activity. They lost the opportunity to practice their algorithmic thinking skill. In addition, the unplugged activity is wasted resources. It is a paper activity. Papers are required to create event materials such as different maps with different predefined starting points and ending points, a number of the same arrow cards. Especially, the activity is social playing. The activity is designed to be played in the classroom. Students are required to gather on-site. However, in the COVID-19 situation which needs to maintain social distance, the activity should not be played in the classroom.

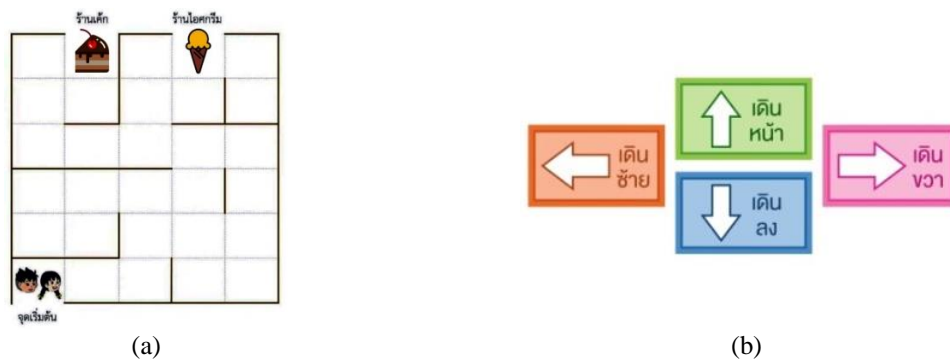


Figure 2. The unplugged activity named Hunger Solver, (a) a map card of the unplugged activity and (b) arrow cards of the unplugged activity. Source: from [19]

### 3.2. Development: transforming unplugged activity to digital game prototypes

Those twenty-two students individually transformed features of the unplugged activity into features of the digital game prototypes. Based on personal experience in gaming; they have not taken an introductory game design course which will be enrolled in their third-year; they designed and developed their games in distinct contents. However, all games have the same objective: finding the way to a certain destination, and then those were called “Find the Way”. Finally, there were twenty-two Find the Way games. The individual game is a single-player game for promoting children's algorithmic thinking skill, developed using Scratch. Figure 3 displays some examples of digital game prototypes.



Figure 3. Some examples of digital game prototypes called "Find the Way"

### 3.3. Usability inspection: inspecting usability problems of game prototypes

Usability inspection is the common term for a group of methods that purpose to find usability problems in user interfaces. Many inspection methods, such as heuristic evaluation, cognitive walkthrough, feature inspection, and standard inspection, are normally performed to find technical problems by a set of independent evaluators. It can be conducted early in the software development lifecycle, typically after the design stage or when user interfaces have not yet been built [20].

Instead of focusing on technical problems which normally performed by evaluators, developers, or quality assurance testers, we focused on playability or usability which is performed by playtesters. After the production of a game, playtesters are responsible for playing all levels to detect troubles or problems. Playtester feedbacks aid the development team in improving the game before releasing it to the general audience. Thus, we propose usability inspection as a process performed after the development stage rather than the design stage.

Thus, after twenty-two single-player games have been fully developed, the usability of the game prototypes was inspected by children to ensure that they can play the game on their own without the assistance of an adult. Feature inspection; a traditional usability inspection method to identify the tasks operated by a user with an application [20]; was applied to verify the interface design of those game prototypes. Video clips recorded during gameplay obtained from computer screen recording were monitored for usability problems. A total of eighty-three elementary school students (aged 7-9) in the academic service network schools of the Faculty of Science and Technology: Sueksa Songkhro Surat Thani School and Nikhom School, were obtained randomly to test the game prototypes.

### 3.4. Revision: designing and developing a beta version of the digital game

The usability problems were then applied to guide the design of a new game prototype, although core content and gameplay are still derived from the unplugged activity. Two fourth-year students studying in Multimedia Technology and Animation Project course jointly designed a new prototype and developed a beta version; the next phase of the game is ready to be tested by external team members [7]. The students completed the Basic Computer Game Design course when they were in their third-year. Their knowledge and personal experience in gaming may benefit game development. A beta version of the digital game was eventually created and renamed "Hunger Program". The landing page of the digital game is shown in Figure 4(a). The digital game's content is consistent with the content of the unplugged activity. The goal is to find a path to get the characters (the boy and the girl) from the starting point (the ice cream) to the ending point (the ice cream). The game components are shown in Figure 4(b).

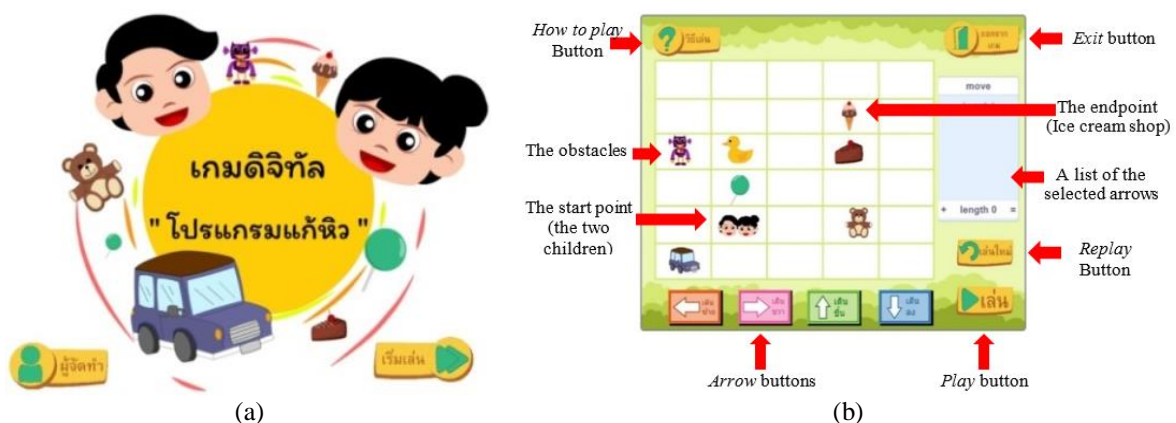


Figure 4. A beta version of the digital game named "Hunger Program", (a) the landing page of the digital game and (b) the digital game components

### 3.5. Game experience evaluation: evaluating user experience of the game

To ensure user satisfaction with the software, user experience should be assessed as a cornerstone of the HCI field [21]. In the game design discipline, measuring user experience or player feelings is known as game experience or gameplay experience [22], [23]. Among tools for evaluating game experience, the game experience questionnaire (GEQ) was widely used [23]. In this study, THGEQ; GEQ for Thai children [24]; was employed. The statements of THGEQ were extracted from KidsGEQ [25], translated into Thai language,

refined to be understandable for Thai children. It was reported to be suitable for Thai children aged 7-9 to measure their perceptions of game experience. Another group of eighty-seven elementary school students at Nikhom School were randomly assigned to play the game. Then, they expressed their feelings about the game using a visual analogue scale to rate 7 statements covering the 7 dimensions of THGEQ as shown in Table 1.

Table 1. Statements of THGEQ [24] extracted from KidsGEQ [25]

	Dimensions	Statements
#1	Challenge	1. It was exciting.
#2	Competence	2. I was good at it.
#3	Flow	3. I felt like I was inside the game.
#4	Immersion	4. The game was beautiful.
#5	Positive Affect	5. I thought it was fun to play the game.
#6	Negative Affect	6. I found it tiresome/exhausting.
#7	Tension	7. I have grumbled/complained while playing the game.

### 3.6. Educational value evaluation: evaluating the educational value of the game

PISA framework; the Programme for International Student Assessment on reading, mathematics and science knowledge; was applied to game designing. The game named Plethora was designed to cover concepts of PISA to support children for that knowledge [26]. In this study, we have applied Bebras tasks to the game testing. Bebras is an international initiative aiming to promote Informatics (Computer Science, or Computing) and computational thinking among school students of all ages [27]. The contest tasks covering the concept of algorithmic thinking used in the UK in 2017 and 2018 were administered to assess the progression of the algorithmic thinking skill after playing the game as shown in Figure 5. The group of eighty-seven elementary school students at Nikhom School played the game. Their algorithmic thinking was tested before and after playing the game.

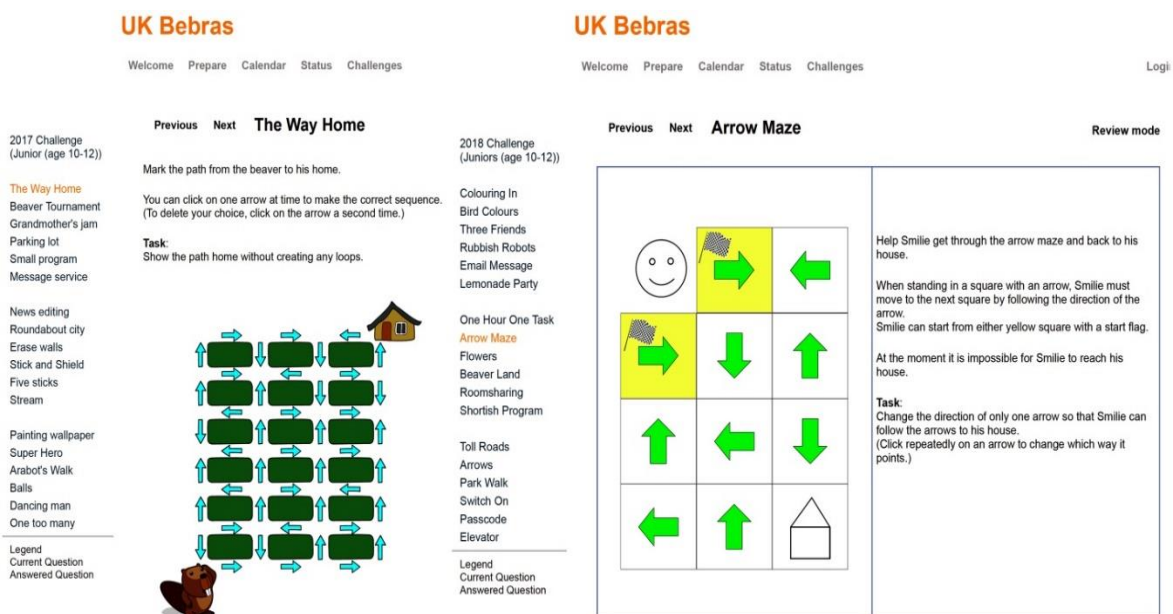


Figure 5. Bebras tests used in the UK in 2017 and 2018 Source: From [27]

### 3.7. Release: releasing the game

The game was released at Nikhom School, an elementary school in Suratthani province, Thailand. In addition to being one of the schools in the academic service network of the Faculty of Science and Technology, the school has been designated due to the shortage of computer science teachers. This game may be a supplement to the course. This game can be played either in the classroom or in the activity period. Children can play the game by themselves, at school or home.

## 4. RESULTS AND DISCUSSION

### 4.1. Software engineering processes and HCI principles in game development

Based on insufficient HCI principles in existing game development models, the new GDSE was proposed. The new GDSE model consists of six phases, including the design phase, which is usually the pre-production phase of traditional software engineering, and the development phase, which is typically the production phase of the traditional model. The GDSE model differs from traditional software engineering in respect of post-production phases. This is not only the functional testing phase but also usability inspection, game experience evaluation, and educational value evaluation which are the testing techniques used in the HCI field.

The new GDSE model was used as a research method to develop a game that fosters algorithmic thinking for children aged 7-9. Game development started with design. The initial game design was created. In this study, unplugged activity was the basic design of the game. The unplugged activity design illustrated the basic concept of the game. After the concept and the design were initiated, the development of the game prototype was begun. In this case, the unplugged activity features were converted to the digital game prototypes features. To develop a usable game, a usability inspection was performed. Children who are actual users were included in the study. The game has been evaluated to identify usability problems while children were playing. If problems were found, the game was redesigned and redeveloped. It has been reported that a usable game could improve the user experience of the games [28]. Therefore, game experience evaluation should be the next step in capturing how players feel about the game. THGEQ, a questionnaire for Thai children was used in this study. In the case of DEGs, educational value evaluation should be assessed to determine the effectiveness of the game in supporting certain knowledge. For this research, the effectiveness of the digital game in promoting children's algorithmic skill was studied by Bebras tasks. The digital game was finally available for release.

### 4.2. A software architectural view of the digital game

#### 4.2.1. System architecture

The system architecture of the game is similar to that of the software architectural concepts to develop gaming applications for educational purposes used by [29], [30]. The system's component level architecture diagram is depicted in Figure 6 and its explanations are followed.

- User interface: Players interact with the PC game via a mouse which is the main input device. A click is the principal user action. Buttons will be clicked to corresponding activities: play, replay, how to play, next, exit. A player can pick several arrows to make a route by clicking. The player can click the scroll button on the "move" panel to view a longer arrows sequence. The PC screen is an output device to display the player's activities and game responses, and a speaker makes audio to inform the player of interactions happening. Sound effects are played when characters collide with one another. Narrative voices are loud when how to play button is clicked.
- Game runtime environment: Device application programming interfaces (APIs) allow players to interact with the system using their plugged-in hardware. The input device, such as a mouse, is supported by the game engine. The way to handle the input is by an event handler. The computer listens to the input event (such as mouse click) then the custom code is executed in response to the input. Game logic including main loop, event listeners, and game objects are all found in the core module. When a player clicks the play button, the loop for checking the sequence of arrows is executed then the scoring mechanisms have updated the score. If the player achieves that level, he is then transferred to the next levels, where the game objects are repositioned randomly. This continues until the player fails the game or exits it. A script engine is made up tied to each game object. The game's data is cached in the application.

Considering the design constraints of the system, a backend database should be developed. The game's data, such as score, is temporarily stored in cache memory. Score and other information such as user accounts and demographics data should be able to keep in a remote database and could be able to retrieve for beneficial analysis.

#### 4.2.2. Process view of the game

Gameplay and game mechanics depict the process of the system. The game encourages players to practice sequential thinking by planning characters' journey. Players plot a path for the characters to catch the ice cream by clicking the arrows in sequence: left, right, up, or/and down. There is no time or score pressure. Players can reroute if they are unsure. The Replay button is clicked to clear the previously defined arrows. Players will then be able to create a new sequence of arrows. Once the itinerary is complete, the Play button will be clicked. Then the characters will move along the given path. If the characters are able to catch the ice cream successfully, players will win a certain level. They earn one score and will be forwarded to the next levels. For a new level, the map will be changed where the character's position and the ice cream will be switched randomly. If the player has planned the route wrong and the characters cannot travel to catch the ice cream, the player will

not be able to score and will not be forwarded to the next levels. The player needs to rethink; they cannot pass a level unless they can think in the right order. Every two levels, the game gets increasingly difficult. Players are provided with more complicated maps, that is, the location of the ice cream is far from the starting point and there are obstacles on the way to the destination. In each level of the same difficulty (there are two sub-levels for the same difficulty), the location of the ice cream shop is randomly placed in different locations to encourage players to think, analyze and plan different routes. There are ten levels of difficulty in total. After level 20, the game will have continuous difficulty. Players can play indefinitely without ending levels.

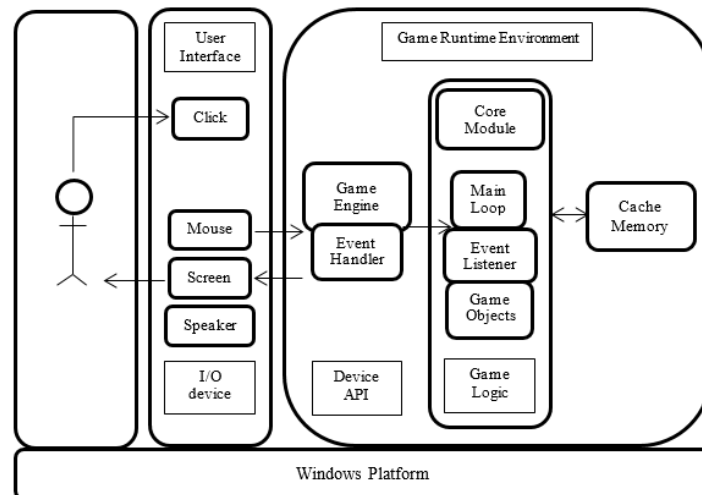


Figure 6. The component level architecture of the digital educational game promoting algorithmic thinking

### 4.3. Usability problems of digital games for children

Through video clips recorded while children were playing games during a usability testing session, we aimed at inspecting some of the usability problems encountered by children aged 7 to 9. Even though most children aged 7 to 9 understood initially how to play all 22 Find the Way games, some children could not play the game alone without the help of an adult. It could be said that while converting from unplugged features to digital features, some features were not designed for children aged 7 to 9 resulting in usability problems as shown in Table 2. Usability problems are important suggestions for developers to optimize their software applications [31], [32]. It is useful for game designers as a set of design guidelines for better game design that allows children to play by themselves either at school or home.

Table 2. Usability problems of digital games for children

Usability problems
The instructions are unclear, not specific or too specific, and not in an attractive form.
Goals and methods of achieving them are unclear and not explained early.
There are feedbacks in games that children do not understand.
Non-native words are uncommon to children.
Interfaces that increase a child's cognitive load can be frustrating.
The lack of experience with the mouse makes the child feel inconvenient to drag.

### 4.4. Game experience of children

End-user engagement is a legacy feature in HCI, whether in design, development or evaluation. Their engagement will eventually personalize a product created for themselves or others. In the case of children, it is challenging to involve them in the participation process. There is a study that has presented a set of tools: CHECK, ActiveInfo and PICO-Art for working with children [33]. However, these tools are in the form of booklets which children need to inform activities, checklists of 'why' questions which children need to answer and discuss. For this study, we involved children in the evaluation process. The performance of the digital game in terms of game experience was assessed using the game experience questionnaire (GEQ). After playing the game, the eighty-seven elementary school students of Nikhom School have provided their demographic data including gender, age, and level/class number as shown in Table 3 and then



expressed their feelings about the game by using THGEQ. Table 4 shows that students strongly agreed that the game was exciting, it was beautiful, it was fun to play the game, and they were good at playing it. This indicates that the game can motivate students to play. It could be assumed that the more they play the game, the more they have a chance to practice algorithmic thinking skill. On the other hand, some students agreed that the game is tiresome. It could be considered that since the game is not an entertainment game, it encourages students to think systematically, so it might be difficult for young children especially for the youngest (aged 7). The children had to put in a lot of effort to complete each level resulting in the feeling of fatigue. Therefore, it could be suggested that DEGs should be designed to be easy to play [34] so that children can eventually come up with solutions, they can adapt and reuse their skill to solve a wider variety of challenges [35].

Table 3. The demographic data of the students

	Gender		Age		Level/Class number			
	Boy	Girl	7	8	P1/1	P1/2	P2/1	P2/2
Number (N=87)	46	41	51	36	26	25	19	17
Percent	52.9	47.1	58.6	41.4	29.89	28.74	21.84	19.54

Table 4. Students' feelings on the digital game

Statements	Students' feelings	Agreement levels		
		$\bar{x}$	S.D.	Meaning
1.	It was exciting.	3.68	0.72	Strongly agree
2.	I was good at it.	3.24	0.88	Strongly agree
3.	I felt like I was inside the game.	2.89	1.31	Agree
4.	The game was beautiful.	3.52	0.80	Strongly agree
5.	I thought it was fun to play the game.	3.40	0.89	Strongly agree
6.	I found it tiresome/exhausting.	2.53	1.35	Agree
7.	I have grumbled/complained while playing the game.	2.32	1.46	Neither agree nor disagree

**4.5. Educational value of the game**

The performance of the digital game in terms of educational value was assessed using the Bebras tasks used to test algorithmic thinking skill in 2017 and 2018. The improvement of algorithmic thinking skill of the eighty-seven elementary school students of Nikhom School in Table 3 has been assessed. Before playing the game, the students were assigned to solve the competition tasks individually. If they were able to solve the tasks, they got 1, if not, they got 0. Table 5 displays a sample of each student's Bebras test scores before and after playing the digital game. After playing the game, they were assigned to solve the tasks individually again. The results show that 49 out of 87 students were able to solve the Bebras task named "The Way Home" and 38 of them failed to score. There were 54 out of 87 students were able to solve the task named "Arrow Maze" and 33 students could not solve the task. In conclusion, 59.2% of students, on average, were able to solve both tasks after playing the digital game. In other words, most students (more than half of the target audiences) have "improved" algorithmic thinking skill after playing the digital game. It could be supported by the study of [4] reporting the benefits of maze games to support learning. A maze game is used as a tool to help players practice their path-finding skill. It has been found that the game can encourage players to practice their algorithmic thinking skill or use algorithms to find alternative routes. Moreover, puzzles help students focus on solving problems. The more problems children can solve, the more they can develop their algorithmic thinking skill. Importantly, puzzles can motivate students to have fun solving problems [36].

Table 5. A sample of each student's Bebras test scores before and after playing the digital game

Student No.	"The Way Home"			"Arrow Maze"		
	Pretest score	Posttest score	Result	Pretest score	Posttest score	Result
1	0	1	higher score	0	1	higher score
2	1	1	same score	1	1	same score
3	0	1	higher score	0	1	higher score
4	1	1	same score	1	1	same score
5	0	1	higher score	0	1	higher score
6	0	1	higher score	0	1	higher score
7	1	1	same score	0	1	higher score
8	0	1	higher score	1	1	same score
9	1	1	same score	0	1	higher score
10	0	1	higher score	0	1	higher score

## 5. CONCLUSION

IPST designed and developed an unplugged activity called “Hunger Solver” to develop the logical thinking and problem-solving skills of Thai elementary school students. Due to the activity’s restrictions, a digital game called the “Hunger Program” was created. The design of the unplugged activity is the basic idea behind the game design. In game development, the new GDSE is used as a game development life cycle instead of other models that rely only on traditional software development. Based on the proposed GDSE compliance for game development, usability problems of the digital game were revealed in the phase of usability inspection. The usability problems have been used as important guides for game developers to optimize the game. The process of engaging children in the game experience evaluation phase and the educational value evaluation phase could indicate outstanding advice on how to personalize the game to gain acceptance. These are the principles of HCI incorporated into the new GDSE model resulting in improvements of the digital game in terms of usability enhancement, game experience and educational value that other models cannot identify.

Furthermore, the digital game could solve the limitations of the unplugged activity: social playing, unmanageable, unapproachable, and wasted resources. Children could play the game on their own either at school or home with no requirement for a conductor and no demand for wasteful resources. Importantly, the game could encourage children to practice algorithmic thinking skill. Algorithmic thinking skill not only effectively delivers children with programming concept but also develops children's logical thinking to solve complicated problems in daily life. For limitations of this article, “Hunger Program” is only one digital educational game used to validate the efficiency of GDSE, we consider having more validations with games in various categories. Also, this article focuses on the outcome of the game on humans, thus the discussion of trade-offs between the non-functional properties and functional properties of the game are not pointed, for example, whether it requires more computational power when the usability and interactivity of the game increased. The properties such as response time, mean time to failure, mean time to repair, would be presented in the next articles.

## ACKNOWLEDGEMENTS

The authors would like to express gratitude to the directors of Sueksa Songkhro Surat Thani School and Nikhom School. We also highly appreciate the great support of the teachers of the four classes. We would also like to extend our thanks to the students for their cooperation. Finally, we wish to thank Suratthani Rajabhat University for the research funding.





## REFERENCES

- [1] K. Howland, J. Good, J. Robertson, and A. Manches, “Every child a coder? research challenges for A 5--18 programming curriculum,” in *Proceedings of the 14th International Conference on Interaction Design and Children*, Jun. 2015, pp. 470–473, doi: 10.1145/2771839.2771954.
- [2] IPST, “The basic education core curriculum,” *The Institute for the Promotion of Teaching Science and Technology (IPST)*, 2017, Accessed: Mar. 17, 2020. [Online]. Available: <http://164.115.23.205/web/index.php/news-and-announcements/49-2010-10-27-03-54-42/887-the-basic-education-core-curriculum-be-2551>
- [3] B. Rodriguez, S. Kennicutt, C. Rader, and T. Camp, “Assessing computational thinking in CS unplugged activities,” in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Mar. 2017, pp. 501–506, doi: 10.1145/3017680.3017779.
- [4] G. Futschek, “Algorithmic thinking: The key for understanding computer science,” in *Proceedings of the 2006 international conference on Informatics in Secondary Schools - Evolution and Perspectives: the Bridge between Using and Understanding Computers*, 2006, pp. 159–168.
- [5] W. Yang, S. K. K. Chow, and X. J. Yuan, “The structural design computer model-based games for assisting mathematics classroom learning,” in *Proceedings of the 2018 International Conference on Distance Education and Learning*, May 2018, pp. 54–58, doi: 10.1145/3231848.3231853.
- [6] Y. Liu, F. Abdullah, P. Paliyawan, R. Thawonmas, and T. Harada, “Improving brain memory through gaming using hand clenching and spreading,” in *Motion, Interaction and Games*, Oct. 2019, pp. 1–3, doi: 10.1145/3359566.3364689.
- [7] R. Ramadan and Y. Widayani, “Game development life cycle guidelines,” in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Sep. 2013, pp. 95–100, doi: 10.1109/ICACSIS.2013.6761558.
- [8] S. Aleem, L. F. Capretz, and F. Ahmed, “Game development software engineering process life cycle: a systematic review,” *Journal of Software Engineering Research and Development*, vol. 4, no. 1, pp. 6–36, Dec. 2016, doi: 10.1186/s40411-016-0032-7.
- [9] A. Osborne O’Hagan, G. Coleman, and R. V. O’Connor, “Software development processes for games: a systematic literature review,” in *European Conference on Software Process Improvement*, 2014, pp. 182–193, doi: 10.1007/978-3-662-43896-1\_16.
- [10] J. Kasurinen, A. Maglyas, and K. Smolander, “Is requirements engineering useless in game development?,” in *REFSQ 2014. Lecture Notes in Computer Science*, 2014, pp. 1–16.
- [11] E. L.-C. Law, “The measurability and predictability of user experience,” *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '11*, 2011, doi: 10.1145/1996461.1996485.
- [12] S. Trista and A. Rusli, “HistoriAR: Experience Indonesian history through interactive game and augmented reality,” *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 4, pp. 1518–1524, Aug. 2020, doi: 10.11591/eei.v9i4.1979.
- [13] Y. Udjaja, V. S. Guizot, and N. Chandra, “Gamification for elementary mathematics learning in Indonesia,” *International Journal*




- of *Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3860–3865, Oct. 2018, doi: 10.11591/ijece.v8i5.pp3860-3865.
- [14] I. Perera, D. Meedeniya, C. Allison, and A. H. D. Mille, “User support for managed immersive education: An evaluation of in-world training for OpenSim,” *Journal of Universal Computer Science*, vol. 20, no. 12, pp. 1690–1707, 2014.
- [15] I. Perera, D. Meedeniya, I. Banerjee, and J. Choudhury, “Educating users for disaster management: An exploratory study on using immersive training for disaster management,” in *2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE)*, Dec. 2013, pp. 245–250, doi: 10.1109/MITE.2013.6756343.
- [16] S. Hewawalpita, S. Herath, I. Perera, and D. Meedeniya, “Effective learning content offering in moocs with virtual reality – an exploratory study on learner experience,” *Journal of Universal Computer Science*, vol. 24, no. 2, pp. 129–148, 2018.
- [17] A. Abarkan, A. Saaidi, and M. Ben Yakhlef, “Learning games creation: IMIE model,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4373–4380, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4373-4380.
- [18] The Institute for the Promotion of Teaching Science and Technology (IPST), *Teacher’s manual for fundamentals of Science and Technology (Computing Science) Primary 1*. Bangkok: Office of the Welfare Promotion Commission for Teachers and Educational Personnel, 2018, pp. 40–57.
- [19] Watratsattha School, “Hunger Solver, the Unplugged Activity,” Krupuii, 2018. <https://sites.google.com/a/kn.ac.th/krupuii/sx-withyakar-khanwn> (Accessed: Apr. 21, 2021).
- [20] J. Nielsen, “Usability inspection methods,” in *Proceedings of the Conference Companion on Human Factors in Computing Systems*, 1995, pp. 377–378.
- [21] D. P.-O. Bos et al., “Human-computer interaction for BCI games: Usability and user experience,” in *2010 International Conference on Cyberworlds*, Oct. 2010, pp. 277–281, doi: 10.1109/CW.2010.22.
- [22] W. Yang, M. Rifqi, C. Marsala, and A. Pinna, “Towards better understanding of player’s game experience,” in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, Jun. 2018, pp. 442–449, doi: 10.1145/3206025.3206072.
- [23] E. L.-C. Law, F. Brühlmann, and E. D. Mekler, “Systematic review and validation of the game experience questionnaire (GEQ) - implications for citation and reporting practice,” in *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play*, Oct. 2018, pp. 257–270, doi: 10.1145/3242671.3242683.
- [24] K. Sungkaew, “Evaluation of DEG7-9 design heuristics by applying game experience questionnaire,” *KMUTT Research & Development Journal*, vol. 43, no. 2, pp. 117–132, 2020.
- [25] W. A. Jsselsteijn, Y. A. W. de Kort, and K. Poels, “The game experience questionnaire,” Technische Universiteit Eindhoven: Eindhoven, 2013.
- [26] J. Gal-Ezer, S. Szekely, and R. Marelly, “Plethora of skills: Play-learn-practice-invent-share,” in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, Jun. 2020, pp. 541–542, doi: 10.1145/3341525.3393984.
- [27] “Bebras-International challenge on informatics and computational thinking,” bebras.org, 2019. <https://www.bebas.org/?q=history> (accessed Jun. 13, 2020).
- [28] J. L. González Sánchez, N. Padilla Zea, and F. L. Gutiérrez, “From usability to playability: introduction to player-centred video game development process,” in *An Interactive Gameplay to Crowdsorce Multiple Sequence Alignment of Genome Sequences: Genenigma*, 2009, pp. 65–74, doi: 10.1007/978-3-642-02806-9\_9.
- [29] D. A. Meedeniya, S. A. P. A. Rukshan, and A. Welivita, “An interactive gameplay to crowdsorce multiple sequence alignment of genome sequences,” in *Proceedings of the 2019 9th International Conference on Bioscience, Biochemistry and Bioinformatics - ICBBB '19*, 2019, pp. 28–35, doi: 10.1145/3314367.3314374.
- [30] G. Gamage, I. Perera, D. Meedeniya, and A. Welivita, “A crowdsourced gameplay for whole-genome assembly via short reads,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 08, pp. 68–84, Jul. 2020, doi: 10.3991/ijoe.v16i08.14821.
- [31] R. Molich, R. Jeffries, and J. S. Dumas, “Making usability recommendations useful and usable,” *Journal of Usability Studies*, vol. 2, no. 4, pp. 162–179, 2007.
- [32] L. Carvajal, A. M. Moreno, M.-I. Sanchez-Segura, and A. Seffah, “Usability through software design,” *IEEE Transactions on Software Engineering*, vol. 39, no. 11, pp. 1582–1596, Nov. 2013, doi: 10.1109/TSE.2013.29.
- [33] J. C. Read, M. Horton, D. Fitton, and G. Sim, “Empowered and informed: participation of children in HCI,” in *Proceedings of the INTERACT*, 2017, pp. 431–446.
- [34] K. Sungkaew and T. W. K. Kaewsuks, “Validating the effectiveness of the DEG7-9 design heuristics by evaluating the game prototype,” *Journal of Research Unit on Science, Technology and Environment for Learning (JSTEL)*, vol. 10, no. 2, pp. 166–182, 2019, doi: 10.14456/jstel.2019.12.
- [35] L. A. Gouws, K. Bradshaw, and P. Wentworth, “Computational thinking in educational activities,” in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education-ITiCSE '13*, 2013, doi: 10.1145/2462476.2466518.
- [36] E. A. Lamagna, “Algorithmic thinking unplugged,” *Journal of Computing Sciences in Colleges*, vol. 30, no. 6, pp. 45–52, 2015.

## BIOGRAPHIES OF AUTHORS






**Kornchulee Sungkaew**     is a lecturer at the Faculty of Science and Technology, Suratthani Rajabhat University. She received the B.B.A. in Business Computer from Prince of Songkla University, Thailand, in 1995 and the M.S. in Information Technology from King Mongkut’s University of Technology Thonburi, Thailand, in 2001 and completed her Ph.D. in Computer Science at the University of Leicester, the UK in 2016. Her research interests are human-computer interaction (HCI): usability, user experience (UX) and digital educational games. She can be contacted by email: [kornchulee.sun@sru.ac.th](mailto:kornchulee.sun@sru.ac.th).



**Piyamas Lungban**    is a student at Multimedia Technology and Animation Program, Faculty of Science and Technology, Suratthani Rajabhat University. She is interested in the design and development of digital educational games. She can be contacted by email: [piyamasfern@gmail.com](mailto:piyamasfern@gmail.com).



**Sirinya Lamhya**    is a student at Multimedia Technology and Animation Program, Faculty of Science and Technology, Suratthani Rajabhat University. She is interested in the design and development of digital educational games. She can be contacted by email: [feasirinya118@gmail.com](mailto:feasirinya118@gmail.com).