2023 5th International Conference on Computer Science and Technologies in Education (CSTE) | 979-8-3503-2572-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSTE59648.2023.00037

Research on Developing Computational Thinking of Middle School Students Based on Gamified Text Programming Teaching Activities

Junjie Liu
Key Laboratory of Child Development and Learning Science, Ministry of Education
Southeast University
Nanjing, China
liujunjieny@foxmail.com

Yi Bai
Key Laboratory of Child Development and Learning Science, Ministry of Education
Southeast University
Nanjing, China
by.rcls@seu.edu.cn

Xiaojun Xia*
Key Laboratory of Child Development and Learning Science, Ministry of Education
Southeast University
Nanjing, China
xxj.rcls@seu.edu.cn

***Abstract*—With the coming of the digital intelligence era, contemporary middle school students should have the key literacy and ability to quickly adapt to the current social development. Computational thinking, as a critical problem-solving skill in the era of artificial intelligence, has also become a basic literacy for students to face the future. However, the traditional "duck and filler" teaching mode is ineffective in developing computational thinking skills, while the gamified learning approach allows learners to interact with games in an immersive way. Therefore, this study combines games and programming teaching, builds a teaching model of gamified text programming, and develops a series of programming courses. It can develop not only computational thinking skills and improve students' programming ability but also effectively avoid the limitations of traditional programming teaching.**

Keywords—computational thinking, gamified teaching, text programming, teaching model

I. INTRODUCTION

With the rapid development of artificial intelligence technology, countries all over the world have developed corresponding countermeasures to occupy the high ground of technology and practical application. At the same time, the sudden outbreak of the epidemic has led to radical changes in people's lifestyles. The "Classes suspended but learning continues" policy has also made schools and parents reexamine the capabilities and technology needs behind "online teaching" [1]. Computational thinking, as a comprehensive ability, has received much attention for its commitment to solving real-life problems and has even been hailed as a "key core literacy for everyone in the 21st century" [2]. However, it has become particularly urgent to develop students' computational thinking skills efficiently in the current educational environment. This has led to the creation of many ways to develop computational thinking skills [3]. For example, the common graphical programming software Scratch, Mind+, and Blockly can complete the corresponding programming tasks by simply dragging and dropping or building the relevant building blocks. By incorporating computational thinking-related concepts in the form of cards or puzzles, students can complete the learning of computational thinking skills in the process of solving the puzzles [4].

However, these approaches require a high level of abstraction and take the computational thinking concepts out of the activity, focusing more on the development of students' "thinking" and weakening the development of programming skills. Python is a lightweight, text-based programming language that matches the world trend in computer languages. Python's ability to perform programming tasks with simple text-based instructions is easy for students to pick up and is an effective tool for developing computational thinking. Traditional Python programming teaching is based on lectures and exercises, and the whole process is detached from the actual situation. This way of learning is too boring and students easily lose interest in learning. At the same time, it has been proven that gamified programming can improve students' motivation and develop computational thinking skills. Therefore, this study chose the "OZARIA" gamified programming platform, which combines Python programming syntax with gamified-level content. The game levels are used as a vehicle to integrate programming knowledge, which is closer to the real programming learning environment. However, there is no systematic curriculum for teaching Python, and the teaching method of gamified programming needs to be enriched and improved.

II. LITERATURE REVIEW

A. Computational Thinking

The term computational thinking was first coined back in the 1980s by Seymour Papert in his book Mindstorms: Children, computers, and powerful ideas. However, probably due to the limitations of the computing education environment at that time, no clear definition of the concept was given [5]. In contrast, the term computational thinking was conceptualized and made known in 2006 by Jeannette M Wing with a unique academic perspective of computer science. Jeannette M Wing proposes that computational thinking is the application of relevant concepts from computer science to understand human behavior and to design solutions to real-world problems [6]. Jeannette M Wing then even deepened the connotation and meaning of

979-8-3503-2572-0/23/$31.00 ©2023 IEEE
DOI 10.1109/CSTE59648.2023.00037

computational thinking in the context of the evolving times [7-9]. At the same time, it has also provoked more extensive discussions among domestic and foreign scholars. The International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) have developed an operational definition of computational thinking at the K-12 level, which argues that the process of computational thinking to solve problems includes: using computers to solve problems, analyzing data, representing data, and automating problem-solving [10]. Chinese scholars also point out that the essential feature of computational thinking is the process of problem-solving within the constraints of computational models and problems, involving science, technology, engineering, and other subject areas [11].

On the foundation of the above studies, theoretical and practical investigations on computational thinking have gradually flourished. Among them, Brennan moreover built a 3D framework of computational thinking that contains computational concepts, computational practices, and computational perspectives based on the Scratch programming environment [12]. Yu Ying dissects the constructivist perspective and combines the trigonometric structure of thinking with a six-group thematic view of thinking to provide a reference for the IT curriculum with computational thinking-oriented teaching objectives [13]. As shown above, computational thinking is an integrated and comprehensive thinking skill and problem-solving skill that provides a landing point for adapting to the digital intelligence era.

B. *Gamified Teaching*

Gamified teaching is the application of game elements and game concepts to specific teaching scenarios [14]. Gamified teaching not only increases the interest in the game tasks, but more importantly, it can match the cognitive characteristics and cognitive styles of middle school students. The design of gamified programming was not a one-step process, which usually required students to design and modify the basic code modules to complete the given programming tasks. The whole process is the embodiment and expression of various sub-skills of computational thinking. Therefore, gamified programming can solve the problem of traditional teaching of pure programming knowledge that hardly enhances students' skills such as abstraction and deconstruction. More importantly, the gamified programming environment provides more realistic programming situations and students can better internalize computational thinking by making the transition from logical language to programming language. It even stimulates students' intrinsic motivation to learn and increases the effectiveness of teaching objectives. The influence of gamified instruction is gradually expanding, and more and more scholars have conducted a lot of research on its teaching effects. Sun et al. used Scratch software to develop a computational thinking instructional model adapted to elementary school students, based on its gamified levels. This provides a more diverse approach to teaching programming to children [15]. Rania et al. conducted an empirical study to investigate the problems of students in programming through gamification. The results proved that gamified teaching can increase students' motivation and obtain very effective teaching incentives [16]. In summary, the gamified learning approach can reduce the cognitive load of the learners and allow for full engagement in the learning process, thus continuously improving the development of students' abilities in all areas.

III. Instructional Design

This study aims to develop the computational thinking skills of middle school students and design gamified text programming teaching activities based on the OZARIA game platform, which teaches students programming through the easy-to-understand and easy-to-use Python text programming language. On the one hand, it allows students to write code in a more realistic programming environment to accomplish their teaching objectives. On the other hand, gameplay can reduce students' resistance to Python programming and make them more interested in the course content.

A. *Gamified Text Programming Teaching Model Construction*

With the updating and iteration of teaching philosophy and teaching technology, the teaching mode is also changing continuously. In the traditional "duck and fill" teaching mode, students usually receive knowledge passively, which undoubtedly reduces students' interest in learning. In the current development of the epidemic, the construction of teaching mode is even more important for the student's learning effectiveness, so it should be considered more carefully. The "4P" learning theory used in this study was proposed by Resnick, which is important for the development of programming and computational thinking, including Project, Passion, Peers, and Play [17]. Among them, Project means that students are personally involved in the process of learning to create some kind of product to maximize its value, which in this study is the whole game level. Passion is to stimulate students' interest in learning, and their enthusiasm and persistence are the keys to development, while Peers is to promote students' collaboration in the process of project creation, replacing communication with teachers with communication among students, and Play is to give students ownership in the project, allowing them to explore as themselves.

The "5E" approach was developed based on constructivist learning theory and was initially applied to biology-related courses, but as the curriculum model itself was expanded, it was also applied to other subjects. The 5E includes Engage, Explore, Explain, Elaborate, and Evaluate, in which teachers and students have different teaching behaviors [18]. In the engagement stage, teachers need to set up life situations according to the course content to stimulate students' interest in learning and actively participate in the learning process. In the exploration stage, the teacher sets the corresponding learning tasks, and the students are free to explore and build their knowledge framework based on the knowledge they have acquired in the process of investigation and complete the internalization and absorption of knowledge. In the explanation stage, students need to deeply analyze the conclusion of the stage, explain the reasons for it, and better promote the understanding and mastery of knowledge. In the elaboration stage, students need to transfer their knowledge, relate it to phenomena or practical problems in their lives, and master the paradigm of solving similar problems. In the evaluation stage, the combination of teacher evaluation and

student evaluation allows students to have a deeper understanding of their knowledge mastery.

Therefore, this study built a systematic curriculum for middle school students to learn Python based on the "OZARIA" platform and conducted a total of 16 weeks of teaching activities. In the process of teaching the curriculum, a teaching model of gamified text programming was built by combining the "4P" learning theory and the "5E" teaching method. The specific teaching model is shown in Figure 1.

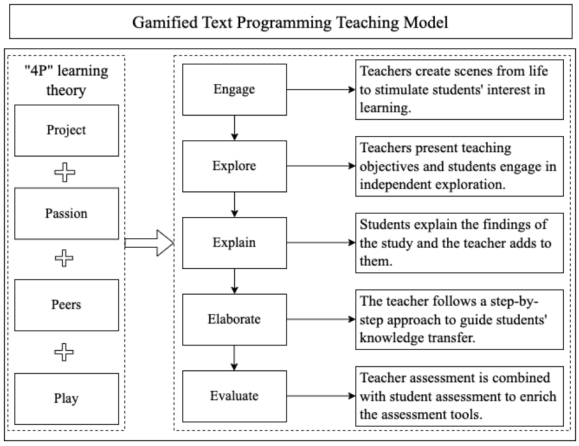

Fig. 1. Gamified text programming teaching model.

B. Gamified Text Programming Instructional Design

1) Teaching Contents

In the "Sequential Structures" course, students will first understand the concept of "sequence" and translate the character's movement path at a game level into a series of sequential structures. They will also use Python code to implement the game character's movement and acquire debugging skills when facing code errors. Only after continuous debugging can the game task be completed successfully. The specific teaching contents are shown in Table 1.

TABLE I. TEACHING CONTENT DESIGN

Knowledge of game levels	***Knowledge of python programming***
Through interactive teaching, understand the concept of "sequential structure" in the game. Deepen the understanding and mastery of the concept of " sequence ", and be able to use python movement function to plan Move the character in the game.	Use hero.moveRight(),hero.moveLeft(), hero.moveUp(),hero.moveDown() methods in python code to implement the "sequence" function in the game and complete the game. Students will acquire debugging skills and build their program instructions.

2) Teaching Objectives

The setting of teaching objectives needs to follow certain principles and be well thought out. Teaching objectives can improve the curriculum design, and make the curriculum content arrangement reasonable and more in line with the cognitive level of middle school students. If the objectives are designed too low, it is easy for students to reach them and will weaken their learning motivation. Conversely, if the objectives are designed too high, students will easily become bored and resistant to the course, thus making the teaching effect much less effective. Therefore, to better guide students to think and efficiently master the skills related to computational thinking and knowledge of Python. The teaching objectives of this lesson are set according to Bloom's taxonomy of teaching objectives, which are defined at the levels of knowledge and skills, process and method objectives, and emotional and attitudinal values [19]. The specific teaching objectives are shown in Table 2.

TABLE II. DESIGN OF TEACHING OBJECTIVES

Knowledge of game levels	***Knowledge of python programming***	***Computational thinking Objectives***
Knowledge and skills objectives	Understand the meaning of "sequential structure", and use python to move functions to complete programming tasks. Implement the "sequential" functions and learn how to debug.	Paragraph Ability to grasp the concept of sequence in computational thinking.
Process and method objectives	Explore to understand and master the "sequence", and be able to use the correct move instruction "sequence" to complete the programming tasks in the game; at the same time, through continuous adjustment of the code to master debugging skills.	Ability to use sequential concepts to complete tasks and to acquire debugging skills in the process of modifying code.
Emotional attitude and value objectives	It will enhance students' interest in programming, strengthen their problem-solving ability, and develop their computational thinking skills in the teaching process.	Ability to transfer the concept of sequence to life scenarios and improve students' computational thinking skills and interest in programming.

3) Teaching process

Engagement stage. Interest in learning is more fundamental than education, and it plays a vital role in improving the quality of learning and enhancing learning outcomes. Students with a positive interest in learning can hardly help but develop an attitude of active exploration and active knowledge-seeking in the process of learning [20]. Usually, interest in learning is the starting point for promoting students to begin to carry out development, as Dewey said, "Without interest, the thinking made is superficial and hasty" [21]. To better promote students' interest in learning, teachers need to use real-life situations as a starting point and let students discover what they want to learn from the phenomena around them. In this lesson, "Sequential Structure", the teacher gives examples of real-life phenomena. For example, the three meals of the day (breakfast - lunch - dinner), or the process of brushing teeth in the morning: open the toothpaste - squeeze the toothpaste onto the toothbrush — brush your teeth--rinse your toothbrush. Lead students to think about the " sequence" involved; so that they can understand that

the sequence is all around us. Then, the teacher introduces the content of this lesson, i.e., "sequential structure in games".

Exploration stage. In the traditional classroom teaching mode, the teacher is usually the main body of teaching activities, obscuring the main position of students. This often leads to a decrease in students' motivation to learn, resulting in poorer learning outcomes. At this stage, the teacher returns the main role of the classroom to the students and allows them to explore on their own. Students first need to draw the movement path of the game character based on their existing knowledge structure and use the code to implement the drawn movement path. In this process, students may have code errors, which require students to debug the code and finally run the code successfully. The movement path construction diagram is shown in Figure 2.

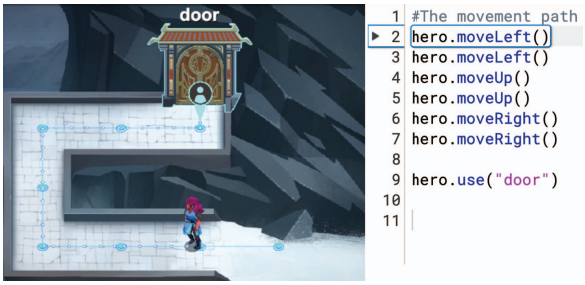

Fig. 2. The movement path construction diagram.

Explanation stage. In the explanation phase, students should start by showing the diagram of the movement path drawn in the game level of the lesson and how it was converted into code. And the teacher complements the students' expressions by guiding them to realize that the algorithm in programming is a step-by-step process of implementing the code. And using standardized programming terminology to describe the concepts of objects and methods that appear in them gives students a deeper understanding of the programming concepts involved. Finally, the teacher throws out questions to guide the students to think about the similarities and differences between the sequential structure in life and the sequential structure in games.

Elaboration stage. During the elaboration stage, the teacher challenged the students with additional levels containing sequential structures. At the same time, the teacher helps students internalize the concept of " sequence" into their cognitive structure by guiding them to think and discuss how to use the programming concept " sequence" in real-life applications. After thinking about it, students understand that programs actually run in a certain order, and so do algorithms. The students were also able to model the "sequential structure" from real-life events, which not only motivated them to learn but also reinforced their knowledge of the new concepts in this section.

Evaluation stage. During the evaluation stage, the teacher should guide students to think about the following question, "What do you think are the key points of the sequential structure? Can you share your understanding of the order of instructions in programming?". Give students time for self-thinking and group discussion, and ask them to comment on the above questions. The teacher evaluates the students' responses and corrects the language used in the presentation. In this session, students can both find out from the evaluation where their understanding of this point is lacking, so that they can better improve it. At the same time, it also enhances students' summarizing and expressing abilities.

IV. Conclusion

In the context of the current digital intelligence era, highly innovative forms are needed to develop students' computational thinking skills. This study creatively uses the Python programming language to develop students' computational thinking and programming skills. It combines gamification with teaching and learning in an interesting way. Combining abstract programming knowledge with gamified scenarios can solve the problem of students feeling a lack of interest in learning programming. The gamified text-based programming learning approach has the following advantages. On the one hand, students first need to participate in the game breakthrough, these actions are implemented by Python code, and the positive feedback from the game drives students to actively participate in learning activities [22]. On the other hand, the focus of game-based programming teaching can avoid simply explaining boring programming knowledge, because students will deepen their understanding and mastery of programming knowledge through the process of game play. Based on reflecting on the limitations of traditional programming teaching methods, this study combines the "5E" teaching method and the "4P" learning theory to build a gamified text programming teaching model. And a series of programming courses are designed to give corresponding teaching cases. We hope the research in this paper can bring insights to researchers and front-line educators, and suggest suggestions for the development of computational thinking education and programming education.

References

[1] Gong, X., & Qiao, A. L, "The impact of game-based experiential learning on computational thinking,"Modern Educational Technology no. 11, pp. 119-126, 2021.

[2] Lye, S. Y., & Koh, J. H. L, "Review on teaching and learning of computational thinking through programming: What is next for K-12?" Computers in Human Behavior no .4, pp. 51-61, 2014.

[3] Kaiqin Yang, Xin Liu, and Guang Chen, "The Influence of Robots on Students' Computational Thinking: A Literature Review," International Journal of Information and Education Technology vol. 10, no. 8, pp. 627-631, 2020.

[4] Sigayret, K., Tricot, A., & Blanc, N, "Unplugged or plugged-in programming learning: A comparative experimental study," Computers & Education, no. 184, 2022.

[5] Papert, S. A, "Mindstorms: Children, computers, and powerful ideas," Basic books, 2020.

[6] Wing, J. M, "Computational thinking," Communications of the ACM vol. 49, no. 3, pp. 33-35, 2006.

[7] Wing, J. M, "Computational thinking and thinking about computing," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 366, no. 1881, pp. 3717-3725, 2008.

[8] Cuny, J., Snyder, L., & Wing, J. M, "Demystifying computational thinking for non-computer scientists," Unpublished manuscript in progress, referenced in http://www. cs. cmu. edu/~ CompThink/resources/TheLinkWing. pdf, 2010.

[9] Nor Hasbiah Ubaidullah, Zulkifley Mohamed, Jamilah Hamid, and Suliana Sulaiman, "Discovering the Role of Problem-Solving and

Discussion Techniques in the Teaching Programming Environment to Improve Students' Computational Thinking Skills," International Journal of Information and Education Technology vol. 11, no. 12, pp. 615-623, 2021.

[10] Barr, D., Harrison, J., & Conery, L, "Computational thinking: A digital age skill for everyone," Learning & Leading with Technology, vol. 38, no. 6, pp. 20-23, 2011.

[11] He, Q. M.,Lu, H. Q., & Feng, B. Q, "The core task of teaching computer fundamentals is the development of computational thinking skills," China University Teaching no. 9, pp. 5-9, 2010.

[12] Brennan, K., & Resnick, M, "New frameworks for studying and assessing the development of computational thinking," In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada vol. 1, p. 25, 2012.

[13] Yu, Y., Zhou, D., & Yu, W, "Implication Analysis and Structural Construction of Computational Thinking," Modern Educational Technology no. 05, pp. 60-66, 2017.

[14] Werbach, K., Hunter, D., & Dixon, W, "For the win: How game thinking can revolutionize your business," vol. 1, Philadelphia: Wharton digital press, 2012.

[15] Sun, L., Hu, L., & Zhou, D, "Single or combined? A study on programming to promote junior high school students' computational thinking skills", Journal of Educational Computing Research, vol. 60, no. 2, pp. 283-321, 2022.

[16] Elshiekh, R., & Butgerit, L, "Using gamification to teach students programming concepts," Open Access Library Journal, vol.4, no. 8, pp. 1-7, 2017.

[17] Resnick, M, "Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play," MIT press, 2017.

[18] Bybee, R. W, "Using the BSCS 5E instructional model to introduce STEM disciplines," Science and Children, vol. 56, no. 6, pp. 8-12, 2019.

[19] Bloom, B. S., & Krathwohl, D. R, "Taxonomy of educational objectives: The classification of educational goals," Book 1, Cognitive domain. Longman, 2020.

[20] Hu, J. M., & Zhao, L. Z, "Developmental stages of learning interest, influencing factors and stimulation paths. Curriculum," Teaching Material and Method no. 11, pp. 78-85, 2021.

[21] Dewey, J, "Interest and effort in education," Forgotten Books, 1913.

[22] Thomas Hvid Spangsberg and Martin Brynskov, "The Nature of Computational Thinking in Computing Education," International Journal of Information and Education Technology vol. 8, no. 10, pp. 742-747, 2018.