engineering proceedings

MDPI

Proceeding Paper

Digital Game Approaches for Cultivating Computational Thinking Skills in College Students †

Li-Xian Chen [1,*], Shih-Wen Su [2], Chia-Hung Liao [2], Mei-Jin Hsiao [3] and Shyan-Ming Yuan [2]

[1] School of Big Data, Fuzhou University of International Studies and Trade, Fuzhou 350202, China
[2] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu City 30010, Taiwan; alvin.cs00@nycu.edu.tw (S.-W.S.); aiallen.cs07g@nctu.edu.tw (C.-H.L.); smyuan@nycu.edu.tw (S.-M.Y.)
[3] School of Art & Design, Fuzhou University of International Studies and Trade, Fuzhou 350202, China; meijins@ms69.hinet.net
* Correspondence: lixian.cs98g@g2.nctu.edu.tw
† Presented at the IEEE 5th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability, Tainan, Taiwan, 2–4 June 2023.

Abstract: Computational thinking (CT) has become one of the critical goals of teaching CS programming courses. Computational skills consist of skills taken in a computational form in learning programming and dealing with daily life. More research adopted games to teach CT skills. This paper investigated two games, Little Alchemy 2 and Dr. Sudoku, to promote CS students' CT skills and applied international Bebras tests to measure their CT skills. The results showed that CT skills in problem decomposition and pattern recognition could be improved via digital games. Thus, this study contributes to computing education using available digital games to promote CS college students' CT abilities.

Keywords: computational thinking; digital game-based learning; CS education; problem decomposition; pattern recognition

check for updates

Citation: Chen, L.-X.; Su, S.-W.; Liao, C.-H.; Hsiao, M.-J.; Yuan, S.-M. Digital Game Approaches for Cultivating Computational Thinking Skills in College Students. *Eng. Proc.* **2023**, *55*, 62. https://doi.org/10.3390/engproc2023055062

Academic Editors: Teen-Hang Meen, Kuei-Shu Hsu and Cheng-Fu Yang

Published: 6 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Computational thinking is an algorithmic skill used by computer and mathematical scientists and is one of the basic daily life skills that everyone needs [1,2]. Computational thinking applications are ubiquitous in life, and anyone can use computational thinking skills to solve problems [2]. CT skills include problem decomposition, pattern recognition, abstraction, and algorithm design as the concept of computational thinking [1,2]. For example, computer programmers who are programming games often break the code into small chunks to control a character's movement and sound effects. In a game-based learning environment, like Little Alchemy 2, students learn that mist can be decomposed into air and water (Figure 1). Problem decomposition is a useful problem-solving strategy. It can help us write a complex computer program, plan a holiday, or make a model plane. Using computational thinking skills can help people simplify complex things in different subject areas [3].

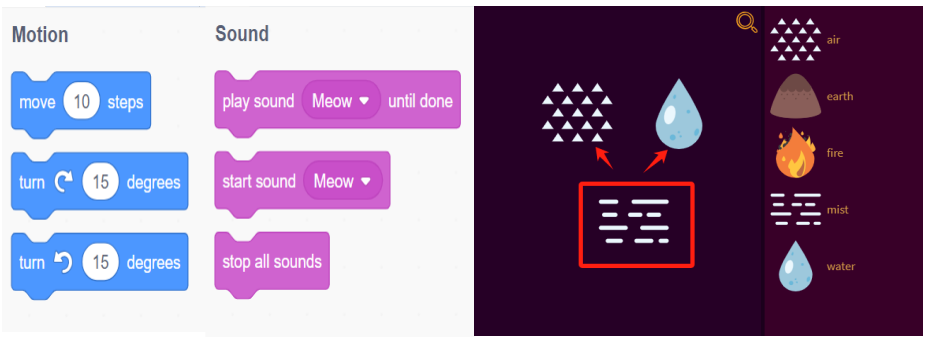


Figure 1. Problem decomposition in the programming and game environment.

The skills of CT enable students to decompose problems, find the rules, and reconstruct them into familiar problems when they encounter new difficulties [4]. In programming education, students imitate codes and learn algorithms to strengthen their mathematical modeling and calculation abilities. However, nearly half of college students have no experience with programming, and it is not accessible to "code" [5]. Teaching rule discovery and pattern recognition in programming courses is challenging [6]. As a result, there are considerable problems with CT training in the curriculum.

Many researchers move from programming to non-programming environments to cultivate computational thinking. Recent programming education has emphasized how computational thinking can be fostered using more game-based learning in programming education [5,7,8]. The digital game environment provides students with repetitive actions to find rules, plan resource arrangements, and optimal paths, and solve encountered problems at the current level [7–9]. Planning is like computational thinking in a game environment, and following directions is like programming. The playing process reflects students' logic and ability to find rules and solve problems. Therefore, this study applied two digital games (Little Alchemy 2 and Dr. Sudoku) to explore the possibility of enhancing students' computational thinking skills.

2. Related Works

Computational thinking is valued in science, technology, engineering, and mathematics (STEM) education [10]. In the 21st century, people have more opportunities to apply information technology to integrate knowledge, skills, values, and attitudes. Computational thinking and programming are problem-solving methods used at different learning stages and disciplines. Whether programming education or courses that cultivate computational thinking skills, it enables students to discover helpful information, find rules, and use existing skills to solve problems in their daily lives [11,12]. Computational thinking and digital literacy are components of programming skills [5]. An attractive computer science education, such as building block programming and game-based learning environments, keeps students engaged in learning [1]. People have the opportunities to explore and make mistakes in a digital game-based learning environment, synthesize information, discover rules, and formulate strategies [11]. Game-based learning environments also stimulate students' imagination, creativity, and motivation to learn, thereby helping to improve CT ability. Therefore, digital games have been promoted as effective systems for teaching rules and knowledge [13,14]. Block-based game environments, like Scratch and Webduino Blockly, have been adopted as CT development tools [15]. Commercial or online web games enable students to engage in more natural playing behaviors that are not constrained by specific learning subjects. Sudoku can be used to teach collaboration and problem-solving abilities, which can be transferred to other learning and living conditions [16]. Little Alchemy 2 lets players combine icons representing concepts, objects, substances, and elements to discover more items. Effective combinations form a new project waiting to be combined with other things [17]. Sudoku has been identified as a tool for helping players to teach reasoning skills and logical thinking [11]. Using Dr. Sudoku and Little Alchemy 2, we explored whether games can improve students' problem decomposition and pattern recognition skills.

A proper evaluation of CT competencies can review the effectiveness of training activities. Bebras is an international information science challenge designed to increase learners' interest in informatics and information and communication technology as a basic science by solving problems on computer science topics. Questions about computers and various logical thinking skills (e.g., pattern recognition, problem decomposition, algorithm, abstraction) are used to increase interest and creativity in the computer science field. This study applied the Bebras challenge to investigate whether the game activity was effective in improving students' CT skills.

3. Methods

3.1. Participants and Procedures

In June 2021, two weeks before the end of the semester, we conducted a quasi-experimental study with 76 juniors studying at an applied university in Fuzhou, China. These students have rudimentary C language and network programming concepts. Two classes were divided into playing Little Alchemy 2 (32 students) and Dr. Sudoku (44 students). To reduce potential distractions, we did not tell the students the research purpose of playing the game, nor did we give them any other information about group conditions. Bebras' CT tests were taken before and after 20-minute game training individually for 30 min. Finally, the students had an open discussion about the gameplay process and filled out a gaming questionnaire. The experimental process is shown in Figure 2.

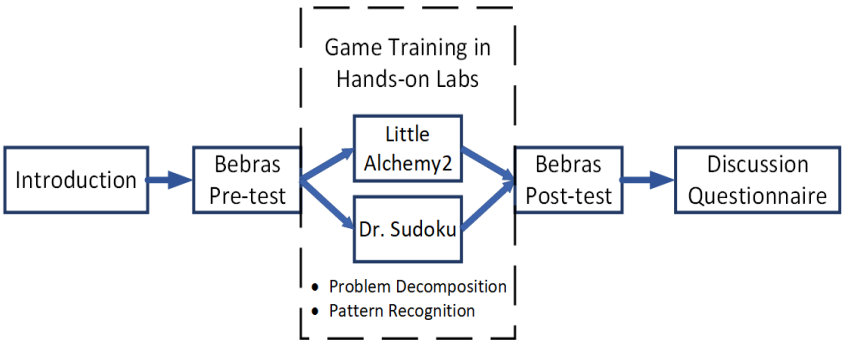


Figure 2. Game education process for training CT.

3.2. Training Environments

This study explored whether games enhance learners' problem-solving and pattern-recognition CT skills. Little Alchemy 2 was adopted to teach students problem decomposition skills by disassembling and synthesizing elements. According to the characteristics of two or more elements, such as air, fire, water, and earth, the game combines them into 720 items. Observing the crafted items, the player can break them into related constituent elements. Figure 3 shows that steam can be decomposed into water and lava. Then, a worksheet was provided for students to fill out. Another game, Dr. Sudoku, required players to fill in the missing 1 to 9 in the 9×9 cells. In this game, there are three functions (slips of different colors, display of all candidate numbers, and display of some candidate numbers) for students to solve the puzzle. Players use their logical reasoning abilities during the gameplay. For example, red boxes display that two numbers of 2 already exist in the two rows of green arrows, and students use the horizontal line division method to infer that the unfilled number in the middle block is 2 (Figure 4).

Bebras Challenge Test was taken to understand students' problem decomposition and pattern recognition skills before and after the CT game training. Considering that the subjects are college students with a fundamental programming foundation, and the difficulty of the competition is only up to 12-grade levels, eight questions, including four simple questions, three medium questions, and one difficult question, were chosen. The Bebra challenge questions from 2013 to 2016 were selected as pre-test and post-test to evaluate the two CT skills. After adding up the dimensions of each question and normalizing the aggregated scores to one hundred, we obtained the ability scores for the two dimensions (i.e., problem decomposition and pattern recognition) and the total computational thinking score. For example, the pre-test dimension contains six questions for problem decomposition and pattern recognition. The students correctly answered the first and second questions in the pre-test (q1: pattern recognition and problem decomposition, q3: pattern recognition). The total computational thinking skill score was 25, the problem decomposition skill score was 16.7 points, and the pattern recognition skill score was 33.3 points.



Figure 3. Little Alchemy 2 screen shot.

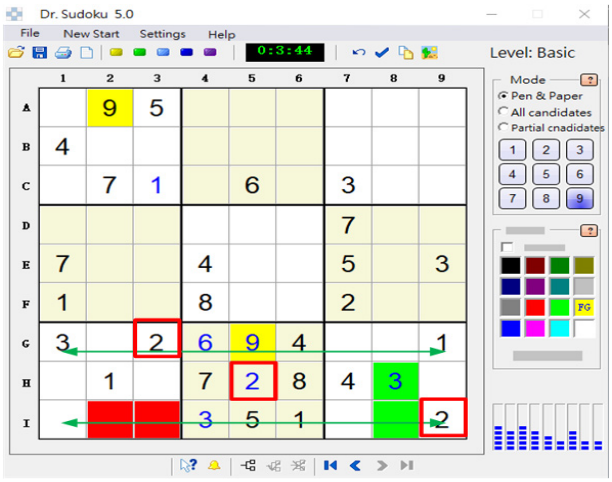


Figure 4. Dr. Sudoku screen shot.

4. Results and Discussion

The independent sample *t*-test showed no significant difference in pre-test scores between the two groups (t(56.92) = 1.64, p = 0.106, d = 0.396). The students' starting CT skills of the two groups showed no difference (Table 1).

Table 1. Independent sample *t*-test of pre-test in two groups.

	Mean (S.D.)		df	t	Effect Size (d)
	Little Alchemy 2 (N = 44)	Dr. Sudoku (N = 32)			
score	257.27 (76.72)	223.23 (97.43)	56.92	1.64	0.9

The paired sample *t*-test for paired data showed that the post-test total score of Little Alchemy 2 was significantly larger than the total pre-test score (t(31) = −4.12, p = 0.000, d = 0.858), and the total post-test score of Dr. Sudoku was significantly larger than total pre-test score (t(43) = −4.48, p = 0.000, d = 0.832). The average post-test scores of Little Alchemy 2 and Dr. Sudoku had improved, and the average improvement score (i.e., the post-test score minus the pre-test score) of Little Alchemy 2 was greater than that of Dr. Sudoku (78.85 > 57.43) (Table 2).

Table 2. Paired sample *t*-test for paired data of Little Alchemy 2 and Dr. Sudoku.

	Mean (S.D.)		df	t	Effect Size (d)
	Pre-Test	Post-Test			
Little Alchemy 2 (N = 32)					
Total score	223.22 (97.43)	302.08 (85.97)	31	−4.12 ***	0.858
Problem decomposition	50.00 (23.18)	76.04 (27.08)		−4.78 ***	1.033
Pattern recognition	60.93 (27.30)	75.00 (21.55)		−2.85 **	0.572
Dr. Sudoku (N = 44)					
Total score	257.27 (76.72)	314.69 (60.33)	43	−4.48 ***	0.832
Problem decomposition	63.25 (21.73)	81.81 (22.10)		−4.65 ***	0.847
Pattern recognition	69.31 (20.94)	72.27 (22.50)		−0.77	0.136

** p < 0.01, *** p < 0.001.

Next, we try to analyze whether these two games have an improved effect on problem decomposition ability and pattern recognition ability. Paired sample *t*-test results showed that pre-test and post-test scores of problem decomposition (t(31) = −4.78, p = 0.000, d = 1.033) and pre-test and post-test scores of pattern recognition (t(31) = −2.85, p = 0.008, d = 0.572) were all the statistically significant difference. After playing the Little Alchemy 2 game, the problem decomposition score progressed from 50.00 to 76.04, and the pattern recognition score improved from 60.93 to 75.00 (Table 2). We analyzed the participants' pre- and post-test performance and found that more than half of the students progressed their problem decomposition and pattern recognition scores after playing Little Alchemy 2 (Figure 5). Due to the lack of recognizable cues in this game-based learning environment, students were strongly driven to create new objects based on their exploration. This result shows that they use their world knowledge to drive their exploration.

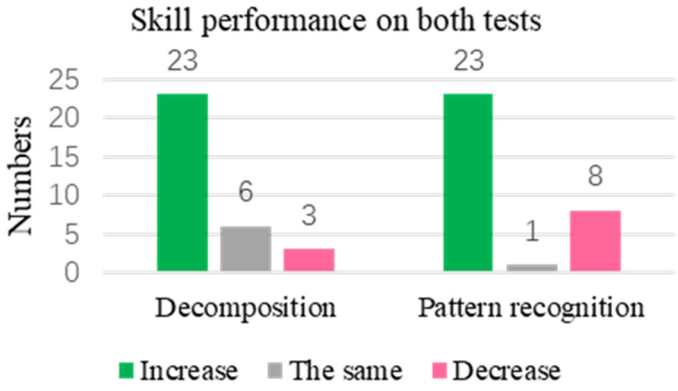


Figure 5. Skill performance after playing the Little Alchemy 2 game.

In the Dr. Sudoku game, problem decomposition pre-test and post-test scores ($t(43) = -4.65$, $p = 0.000$, $d = 0.847$) were statistically significant differences. Problem decomposition scores had progressed from 63.25 to 81.81 (Table 2). However, the pattern recognition score had no statistically significant difference, and more than half of students' pattern recognition scores decreased (Figure 6). This result suggested that filling in numbers from Dr. Sudoku to find the pattern was already familiar to college students, so there was no noticeable improvement.

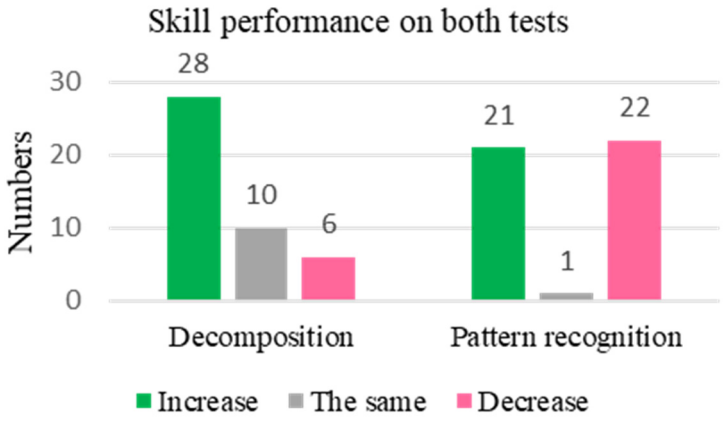


Figure 6. Skill performance after playing Dr. Sudoku game.

Although CS students already have basic programming concepts of C and Java, the problem decomposition and pattern recognition abilities of CT skills can be effectively enhanced via the digital game (Little Alchemy 2) with self-control and self-exploration. Therefore, non-programming digital games should be added to the CS course to teach students CT abilities.

5. Conclusions and Future Work

This paper investigated how the problem decomposition and pattern recognition of CT skills can be trained through specific digital games. This study found that utilizing non-programming digital games can promote computational thinking skills in college CS students. In addition, a digital game (Little Alchemy 2) with the function of free decomposition and integration can effectively improve the problem decomposition and pattern recognition ability. Since the experiment participants are only CS students, data can be collected from the School of Biology and Technology, the School of Foreign Language, and the School of Art and Design to analyze the benefits in the future.

Author Contributions: Conceptualization, L.-X.C. and C.-H.L.; methodology, L.-X.C. and S.-M.Y.; validation, L.-X.C.; S.-W.S. and M.-J.H.; formal analysis, L.-X.C. and S.-W.S.; writing—original draft preparation, L.-X.C. and S.-W.S.; writing—review and editing, L.-X.C. and C.-H.L.; visualization, L.-X.C. and M.-J.H.; project administration, S.-M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by National Science and Technology Council Taiwan (grant number: 108-2511-H-009-009-MY3) and the High-level Talent Research Project at Fuzhou University of International Studies and Trade (grant no. FWKQJ201909).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors wish to thank the blind reviewers for their insightful and constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hsu, T.-C.; Chang, S.-C.; Hung, Y.-T. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Comput. Educ.* **2018**, *126*, 296–310. [CrossRef]
2. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33–35. [CrossRef]
3. BBC. Introduction to Computational Thinking. Available online: https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1 (accessed on 11 January 2022).
4. Rijke, W.J.; Bollen, L.; Eysink, T.H.; Tolboom, J.L. Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Inform. Educ.* **2018**, *17*, 77–92. [CrossRef]
5. Su, S.-W.; Jung, S.-Y.; Yu, X.; Yuan, S.-M.; Sun, C.-T. Modify, Decompose and Reassemble: Learner-Centered Constructive Teaching Strategy for Introductory Programming Course in College. In Proceedings of the 2022 IEEE 5th Eurasian Conference on Educational Innovation (ECEI), Taipei, Taiwan, 10–12 February 2022; pp. 197–200. [CrossRef]
6. Stachel, J.; Marghitu, D.; Brahim, T.B.; Sims, R.; Reynolds, L.; Czelusniak, V. Managing cognitive load in introductory programming courses: A cognitive aware scaffolding tool. *J. Integr. Des. Process Sci.* **2013**, *17*, 37–54. [CrossRef]
7. Liao, C.H.; Yan, J.Y.; Yuan, S.M.; Chen, L.X. Research on cultivating the computational thinking literacy of secondary school students through non-programming activities. In Proceedings of the 4th Eurasian Conference on Educational Innovation 2021, Singapore, 5–7 February 2021.
8. Shell, D.F.; Hazley, M.P.; Soh, L.-K.; Miller, L.D.; Chiriacescu, V.; Ingraham, E. Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers. In Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, Madrid, Spain, 22–25 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–7.
9. del Olmo-Muñoz, J.; Cózar-Gutiérrez, R.; González-Calero, J.A. Computational thinking through unplugged activities in early years of Primary Education. *Comput. Educ.* **2020**, *150*, 103832. [CrossRef]
10. Robins, A.; Rountree, J.; Rountree, N. Learning and teaching programming: A review and discussion. *Comput. Sci. Educ.* **2003**, *13*, 137–172. [CrossRef]
11. Sun, C.-T.; Chen, L.-X.; Chu, H.-M. Associations among scaffold presentation, reward mechanisms and problem-solving behaviors in game play. *Comput. Educ.* **2018**, *119*, 95–111. [CrossRef]
12. Deci, E.L.; Ryan, R.M. *Intrinsic Motivation and Self-Determination in Human Behavior*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
13. Ch'ng, S.I.; Low, Y.C.; Lee, Y.L.; Chia, W.C.; Yeong, L.S. Video games: A potential vehicle for teaching computational thinking. In *Computational Thinking Education*; Springer: Singapore, 2019; pp. 247–260.
14. Gee, J.P. *Good Video Games+ Good Learning: Collected Essays on Video Games, Learning, and Literacy*; Peter Lang: Lausanne, Switzerland, 2007.
15. Zhao, W.; Shute, V.J. Can playing a video game foster computational thinking skills? *Comput. Educ.* **2019**, *141*, 103633. [CrossRef]
16. Emily, L. In Sudoku Club, Scotten School Students Love Thinking Logically. Available online: https://www.theunion.com/news/local-news/in-sudoku-club-scotten-school-students-love-thinking-logically/ (accessed on 12 January 2022).
17. Leino, O.T. STONE + LIFE = EGG—Little Alchemy as a limit-idea for thinking about knowledge and discovery in computer games. In Proceedings of the Philosophy of Computer Games Conference, Msida, Malta, 1–4 November 2016.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.