engineering proceedings

MDPI

Proceeding Paper

Effect of Learning Style on Non-Programmed Computational Thinking Activities †

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

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

Abstract: As information explosion and algorithms are proliferated in the digital age, computational thinking becomes more critical. Nevertheless, most activities that promote computational thinking are always presented in a programming or block language, which tends to horrify students. This study aimed to develop a series of non-programmed, computational-thinking training activities to demonstrate if they improve students' computational thinking skills without writing code. In the meantime, it is checked whether students' learning styles influence their outcomes. After an investigation consisting of a 16 h gamified learning session, we observed that students' computational thinking skills improved overall. Furthermore, the initial learning style of the students influenced the effect of different dimensions of computational thinking in the learning procedures.

Keywords: computational thinking; learning style; game-based learning; digital learning

check for updates

Citation: Chen, Y.-Y.; Su, S.-W.; Chen, L.-X.; Liao, C.-H.; Yuan, S.-M. Effect of Learning Style on Non-Programmed Computational Thinking Activities. *Eng. Proc.* **2023**, *55*, 40. https://doi.org/10.3390/engproc2023055040

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

Published: 1 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

In the digital age, encouraging students to use computational thinking to solve, analyze, criticize, judge, and summarize is the focus of AI and Big Data generation education. As the community increasingly depends on computer technology, developing computing thinking becomes essential in long-term teaching principles, and different countries have developed separate national education strategies to meet these challenges [1]. For example, the UK has incorporated computational thinking education into the regular curriculum, meaning primary and secondary school students must learn how to program. Moreover, states and provinces in the United States, such as Massachusetts, have developed exclusive curricula and standards to evaluate skills for computational thinking [2].

With the development of computational thinking training, approaches to training computational thinking have expanded into non-programming (e.g., digital games) and unplugged activities in addition to general programming training [3,4]. Asbell-Clarke et al. exploited Zoombinis, a popular Computational Thinking (CT) learning game for age eight to adults, to train students in grades three to eight [5] and proved that well-crafted learning games could be effective in promoting student learning. Hooshya et al. also obtained similar results when they developed an adaptive educational game, AutoThinking, to train computational thinking skills and conceptual knowledge in fifth graders [6]. In gamified and non-programmed environments, students have demonstrated higher levels of interest, satisfaction, and technology acceptance in learning computational thinking [7,8].

The non-programmed activities' goal is to enable students become accustomed to thinking in the mode of computer scientists, including through programming and related

classroom activities, learning to discover problems, disassemble problems, conceptualize and model, and find feasible computing solutions. However, students' acceptance and management of the learning materials vary according to each student's thinking ability. Students' habits of absorbing and managing material affect their learning styles [9], which is a preferred way of learning that affects how an individual receives stimuli, remembers, thinks, and solves problems. Students who have different learning styles have different problem-solving abilities [10]. To cultivate students' computational thinking literacy, we must consider their different learning styles.

In this study, we aim to prove that the latest training improves CT literacy and to investigate the relationship between learning style and computational thinking ability.

2. Literature Review

2.1. Learning Style

A learning style is a preferred way of learning that affects how an individual receives stimuli, remembers, thinks, and resolves problems. Since students with different learning styles solve problems differently, how they solve them influences how well they absorb information [11]. Many study results have shown how to advance the teaching of various subjects, including mathematics and science [9]. In this study, we applied the Felder–Silverman learning style model from North Carolina State University to group our students into four dimensions of learning style (Figure 1) [12].

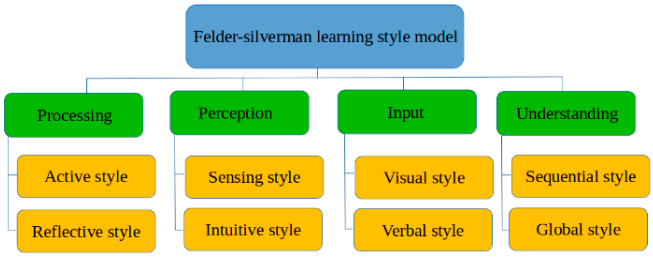


Figure 1. Felder–Silverman learning style model.

(1) Active/reflective continuum: First, there is the active/reflective continuum, which determines the method of processing information preferred by learners. Learners with an active learning style show action-oriented learning characteristics and are accustomed to understanding how things work through trying. Reversely, learners with a reflective learning style are likely to be introverted but good at thinking and prefer to work alone.

(2) Sensing/Intuitive continuum: The second dimension determines how learners prefer to take in the information, which is divided into the sensing and intuitive continuums. A learner with the sensing style is good at memorizing concrete facts rather than abstractive theories and generally performs with patience and caution. Conversely, a learner with the intuitive style is excellent at figuring out innovative concepts. Most of them find it hard to endure to learn with many memorable and tedious calculations.

(3) Visual/verbal continuum: The visual/verbal continuum is the third dimension determining how learners prefer information to be presented. Students who prefer the visual style perform better on graphics or video scaffolds. On the other hand, students who prefer the verbal style benefit from the written or spoken narration. However, there is no noticeable difference when studying because the information is generally present in visual and written narrations simultaneously.

(4) Sequential/global continuum: As the last dimension, the sequential/global continuum indicates learners' preferences for organizing information. A learner with a sequential-style preference is good at linear inference, which means learning step by step. A learner who prefers the global learning method is used to mastering the whole cycle of the learning content first. Once the global-style learners understand the gen-

eral direction of the whole, they can quickly solve complex problems or summarize things innovatively.

Even though the model presents learning styles in four pairs of opposing styles, the learning styles possessed by learners are not strictly dichotomous but rather entail a number of heavily weighted items. It means students may have both active and reflective learning styles, but one's learning attitude may be more active. Therefore, each learner obtains four tendencies in four dimensions after completing the measurement.

2.2. Computational Thinking

The concept of computational thinking (CT) first appeared in Jeanette Wing's study in 2006. In Wing's definition, CT is the concept of representing the formulation and solution of a problem in a way that computers can understand and implement [13]. Over the past decades, increasing attention has been paid to applied CT in education, and various definitions of CT have emerged from different disciplines [14].

We applied the classification of CT from the Google educational resource website to design our curriculum [15]. The four indicators in the definition are decomposition, pattern recognition, abstraction, and algorithm design separately. First, decomposition relates to breaking down problems into a series of minor ones. Pattern recognition refers to the ability to generate predictions and test models by observing each small problem and considering how similar problems have been solved. Another indicator is an abstraction, which focuses on the essential details while ignoring redundant information. In the process of abstraction, laws or principles will be discovered that underlie the patterns. Lastly, algorithm design relates to the ability to develop the instructions to solve similar problems and repeat the process.

3. Method

3.1. Experimental Design

The research designed a 16 h course separated into four days to examine whether non-programmed gamified activities could cultivate students' computational thinking. The Google educational resource website [15] (Computational Thinking for Educators) suggests four concrete elements in computational thinking: problem decomposition, pattern recognition, abstraction, and algorithm design. In the courses, we applied different types of games to cultivate these computational literacies, e.g., Little Alchemy, Rummikub, Gartic.io, and Robozzle.

In the experiment, we designed three parts of the process in each of the four competence courses. The first part focused on game teaching for an hour. The second part was a self-exploring period that allowed for students to play the game and find winning tips in two hours. The last part was the game competition and post-test the next day.

3.2. Self-Exploring Design

After game teaching, a self-exploring period was arranged for students to explore the tips of the games taught. During the self-exploring period, the experiment requested students to finish their study sheet as a scaffold when learning the games. There were also two trainers for 40 students to help if students encountered any problems that were difficult to resolve, ensuring that self-exploration was smooth.

3.3. Competition Design

The courses designed the game competition to increase students' motivation when studying alone. After the self-exploring, students played the game in the single-mode in all the games to battle with the computer. Meanwhile, the trainers recorded their game ranking and rewarded the class's highest-ranking students.

3.4. Assessment Design

In order to evaluate students' original degree of CT competence and its promotion after the courses, the study used Bebras Challenge [16], an international challenge that tests informatics and computational thinking, to examine the results. The researchers selected questions from the Bebras Australia Computational Thinking Challenges of 2017, 2020, and 2021 and translated them into the junior high school level. There were 15 questions in total as a formal contest with 5 questions for each level (easy, medium, hard).

4. Results and Discussion

4.1. Dataset

The sample for this study consisted of 352 seventh-grade students from a secondary school in Taichung. In the experiment, all the students participated in the gamified courses corresponding to problem decomposition, pattern recognition, abstraction, and algorithm design, which are four kinds of core competencies designed by experts. Since the course lasted four days, several subjects could not participate in the entire process, so 307 samples were obtained after deducting the subjects absent in the middle.

The pretest assessment consisted of fifteen questions worth 8 points, while the post-test consisted of six questions on each dimension, totaling twenty-four. The score range was 0 to 120.

4.2. Data Results

Among the 175 males (56.8%) and 133 females (43.2%) in the course, the mean post-test score was higher in both genders than the pretest score (Table 1). In this study, gender was used as an independent variable, and an independent-sample one-way analysis of covariance (ANCOVA) was performed on the post-test scores, controlling for pretest to eliminate student differences. According to the analysis, CT learning outcomes were not different between genders, F (1, 304) = 0.902, $p = 0.343$ (Table 2).

Table 1. Descriptive statistics of pretest and posttest.

	Gender	N	Min	Max	Mean	Standard Deviation
Pretest	Overall	307	0	112	59.01	23.34
	Male	174	8	112	59.84	23.69
	Female	133	0	112	57.92	22.91
Posttest	Overall	307	20	120	61.09	18.47
	Male	174	20	120	62.14	19.48
	Female	133	20	105	59.70	17.03

Table 2. ANCOVA of gender on post-test.

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Between groups	227.50	1	227.5	0.902	0.343
Within groups	76,710.906	304	252.239		
Total	1,248,500	307			

Previously, this course has been tested for its learning effect in a short-term version, so we are confident that this experiment improves students' computational thinking skills, and the research results support this hypothesis (Table 3).

Table 3. Paired *t*-test of pretest and post-test of overall.

	N	df	t stat	t critical	*p*-Value
Overall	307	306	1.724	0.089	0.043 *

* $p < 0.05$.

Since we want to examine whether students' original learning styles influence their computational thinking literacy improvement, we first observed the frequency distribution table of their learning styles (Table 4). According to the results, the student's preference is average in the first, second, and fourth dimensions. Nevertheless, more students prefer the visual learning style to the verbal one.

Table 4. Frequency distribution table of Soloman's learning style.

	Preference	N	Percentage
1st dimension	Active	190	61.9%
	Reflective	117	38.1%
2nd dimension	Sensing	160	52.1%
	Intuitive	147	47.9%
3rd dimension	Visual	261	85%
	Verbal	46	15%
4th dimension	Sequential	138	45%
	Global	169	55%

We implemented four UNIANOVA tests to detect the difference between four learning style dimensions with four competence improvements. In each ANOVA test, the interaction effect between dimensions was examined to guarantee independence, and all interaction effects had no significance.

The competencies' advances in composition and abstraction were not affected differently depending on the type of learning style. In contrast, pattern recognition and algorithm design differed significantly in how they were taught in terms of the second and first dimensions (Tables 5 and 6), in which pattern recognition was more advanced among intuitive learners than sensing learners, and algorithm design displayed more remarkable improvements among reflective learners than active learners.

Table 5. UNIANOVA of pattern recognition improvement.

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Between groups (2nd dimension)	227.655	1	227.655	4.179	0.042 *
Within groups	16,616.597	289	54.481		
Total	18,156.052	307			

* $p < 0.05$.

Table 6. UNIANOVA of algorithm design improvement.

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Between groups (1st dimension)	305.671	1	305.671	5.48	0.020 *
Within groups	16,120.267	289	55.779		
Total	16,978.184	307			

* $p < 0.05$.

5. Conclusions

The experimental results showed no significant difference between male and female groups. Gender was not a significant factor that caused apparent differences when conducting the computational thinking training courses. Despite this, students' computational thinking skills improved after the courses, and their original learning styles significantly influenced their computational thinking literacy growth.

We found that intuitive learners showed more significant gains in pattern recognition competence training than sensing learners. Sensing learners preferred to learn concrete and factual information and details, facts, and figures and disliked surprises and complications. Conversely, intuitive learners preferred abstract and original information, preferring to discover relationships and possibilities. To learn pattern recognition, learners must experiment with several combinations to find the best fit. It may not appeal to intuitive learners, which explains why intuitive learners performed better than sensing learners.

Aside from that, the analysis also found that reflective learners performed better in algorithm design training than active learners. Students in algorithm design training solved puzzles independently without a need to discuss or compete with others, which required more patience to brainstorm and try. In contrast to reflective learners, active learners processed information by attempting it rather than negotiating and explaining it. It may be a crucial difference that separates reflective learners from active learners.

As discussed above, this study proved that the gamification training method is effective for secondary school students. However, students' learning styles influence the course improvement. In order to ensure that students with distinct learning preferences can participate in the curriculum to their strengths, learning styles and a variety of teaching activities (e.g., teamwork) need to be considered for each core competency in the future.

Author Contributions: Conceptualization, Y.-Y.C. and S.-M.Y.; methodology, Y.-Y.C. and S.-M.Y.; validation, Y.-Y.C.; S.-W.S. and C.-H.L.; formal analysis, Y.-Y.C. and S.-W.S.; data curation, Y.-Y.C. and S.-W.S.; writing—original draft preparation, Y.-Y.C. and S.-W.S.; writing—review and editing, Y.-Y.C. and S.-M.Y.; visualization, L.-X.C. and C.-H.L.; 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).

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. Angeli, C.; Voogt, J.; Fluck, A.; Webb, M.; Cox, M.; Malyn-Smith, J.; Zagami, J. A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *J. Educ. Technol. Soc.* **2016**, *19*, 47–57.
2. Masami, I.; Roberto, A.; Maitree, N. Develop Computational Thinking on AI and Big Data Era for Digital Society-Recommendations from APEC InMside I Project. 2018. Available online: https://www.criced.tsukuba.ac.jp/math/apec/2021/pdf/InMsideFinal.pdf (accessed on 10 December 2022).
3. 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, Taitung, Taiwan, 5–7 February 2021.
4. Chen, L.X.; Su, S.W.; Chen YYLiao, C.H.; Yuan, S.M. Cultivating The Computational Thinking Literacy Through Online and Offline Game-based Activities. In Proceedings of the 5th IEEE Eurasian Conference on Educational Innovation, Taipei, Taiwan, 10–12 February 2022.

5. Asbell-Clarke, J.; Rowe, E.; Almeda, V.; Edwards, T.; Bardar, E.; Gasca, S.; Baker, R.S.; Scruggs, R. The development of students' computational thinking practices in elementary-and middle-school classes using the learning game, Zoombinis. *Comput. Hum. Behav.* **2021**, *115*, 106587. [CrossRef]
6. Hooshyar, D.; Pedaste, M.; Yang, Y.; Malva, L.; Hwang, G.J.; Wang, M.; Lim, H.; Delev, D. From gaming to computational thinking: An adaptive educational computer game-based learning approach. *J. Educ. Comput. Res.* **2021**, *59*, 383–409. [CrossRef]
7. 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]
8. Soflano, M.; Connolly, T.M.; Hainey, T. Learning style analysis in adaptive GBL application to teach SQL. *Comput. Educ.* **2015**, *86*, 105–119. [CrossRef]
9. Veronica, A.R.; Siswono TY, E.; Wiryanto, W. Primary School Students' Computational Thinking in Solving Mathematics Problems Based on Learning Style. *Eduma Math. Educ. Learn. Teach.* **2022**, *11*, 84–96. [CrossRef]
10. Hung, Y.H.; Chang, R.I.; Lin, C.F. Hybrid learning style identification and developing adaptive problem-solving learning activities. *Comput. Hum. Behav.* **2016**, *55*, 552–561. [CrossRef]
11. Fadly, W. Profile of students analytical thinking skills in learning style for completing substance pressure problems. *Jambura Phys. J.* **2021**, *3*, 1–15. [CrossRef]
12. North Carolina State University. Teaching and Learning STEM. Available online: https://www.engr.ncsu.edu/stem-resources/ (accessed on 16 July 2022).
13. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33–35. [CrossRef]
14. Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [CrossRef]
15. Exploring Computational Thinking. Available online: https://ai.googleblog.com/2010/10/exploring-computational-thinking.html (accessed on 15 July 2022).
16. Bebras Offical. What is Bebras. Available online: https://www.bebras.org/?q=goodtask (accessed on 15 July 2022).

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.