Integrating Computational Thinking with Digital Storytelling to Enhancing Expression Ability

Chih-Kai CHANG[a*] & Xin-Yu Dai[b]
[a]*Department of Information and Learning Technology, National University of Tainan, Taiwan*
[b]*Department of Information and Learning Technology, National University of Tainan, Taiwan*
*chihkai@mail.nutn.edu.tw

Abstract: Cultivating computational thinking has become an important key to the curriculum of science and technology learning area. When students use interactive media, they can no longer be just receiver, but use interactive media to self-expression. We will let students in grades 3-4 learn ScratchJr, because most students in grades 5-6 learn Scratch at Tainan city. The student use ScratchJr to create the animations, stories and games which interest them. We use creation as a motivation, and children can learn how to solve problems by creating their projects over and over. Children can cultivate their own computational thinking by this way. Digital storytelling can enhance cognitive ability, and make things specific and clear. This is a good way to make everyone to understand the abstract knowledge. Let students use ScratchJr to make digital storytelling on the iPad mini in order to cultivate students' computing thinking. The main goal of this thesis is to improve students' expressing ability to perspectives in the three-dimensional framework of computational thinking. Twenty-seven students in third grade participated in this study. They used ScratchJr to make digital storytelling on the iPad mini. The researcher designed a series of training curriculum. All training curriculum were taught by researcher. Students completed three projects in the basic, intermediate and advanced three-phase training curriculum. Students present and save their projects by the way of YouTube in this thesis. Teaching students to use the iPad mini for video recording, and providing students the interview questions. Arrange students to work in groups of two, use iPad mini to interviews with each other, and then record the interview process. In the end, the qualitative analysis method was used to analyze the students' projects after the three-phase training curriculum and the video of the interview. It will explore the objective of this study: Enhance students' expression ability by the integration of computational thinking into digital storytelling teaching.

Keywords: Science and technology learning area, computational thinking, ScratchJr, digital storytelling, expression ability.

1. Introduction

1.1 Learning by making games

Digital game-making activities can engage students to enhance computational thinking skills. Many findings of relevant studies are very positive (Clark, Tanner-Smith & Killingsworth, 2016; Hayes & Games, 2008; Kafai, 2012; Kafai & Burke, 2015). However, the process of creating a digital game is an ill-structured problem. Many students seem to face difficulties in higher-order thinking skills such as problem-solving, computational thinking, communication, and cooperation. During the last decade, a lot of tools have been developed to scaffold students rapidly build their games such as Scratch, Kodu, GameMaker, and AppInventor. Those tools or programming language did help students and teacher overcome initial development issues. Unfortunately, students tend to spend too much time and energy in becoming competent in building games using specific authoring tools. As a result, students do not have enough opportunity to develop a systematic perspective about software development life cycle and practice other higher-order skills, such as communication and collaboration.

In this paper, a pair programming curriculum is presented to promote the acquisition of multiple higher-order thinking skills at the same time via digital storytelling activities. In other words, students in a pair can systematically design digital storytelling projects with the goals of increasing their communication, collaboration, motivation and improving their computational thinking skills. The pair programming approach of game making not only can enhance students' social skills (communication and collaboration), but also can match the requirements of digital storytelling. This pair programming approach seems not only promote computational thinking, but also very appropriate for digital storytelling activities (Chang, Tsai & Chin, 2017).

1.2 Digital Storytelling

Digital storytelling is a short form of digital media that allows people to share aspects of their stories. Individuals can use it to tell a story or an idea. The most important characteristic of digital stories is that they no longer conform to the traditional narrative mode, because they can combine static images, moving images, sound and text, as well as non-linear and interactive features. The expressive power of technology provides a broad base for integration. It enhances the experience of the author and the audience and allows for greater interaction. Education workers generally believe that the advantage of digital storytelling is that students can choose their creative expression through a range of technical tools. Learners begin to use these tools to create meaningful content. One form of digital storytelling is microfilm, which is "a very brief presentation that lasts from a few seconds to no more than five minutes. It allows the cashier to combine personal writing, photographic images or video, narrative, sound and music. Many people, regardless of their skill level, can tell their stories through images and sounds and share them with others.

1.3 Pair Porgramming

Pair programming, often used in computer programming courses, is a collaborative concept in writing programs. The effects of pair programming are obvious in the way it enhances students' performance, reduces programming errors, and produces high-quality programs. Meanwhile, students' ability to write a specified program is also improved, while their confidence is enhanced (Gorla & Lam, 2004; Han, Lee & Lee, 2009; McDowell, Werner, Bullock & Fernald,2006). Pair programming is defined as the activity that two programmers complete one specific programming task together. Each pair of programmers play two roles: one programmer writes the program, and the other examines whether it is correct or not. Different from individual programming, pair programming can produce a shorter program and a better quality, more logical thinking process. During the programming process, team members share their knowledge by discussing their views on the topics, including problem-solving skills and the programming concepts. Based on the same activity, pair programming produces better quality coding faster than individual programming. Very often, pair programming brings several benefits: a better understanding of a logical argument, a stronger motivation to learn, and more joyful, better learning experiences that come from sharing in a classroom (Forte & Guzdial, 2005). Thus, in this study, pair programming is adopted for grouping in digital storytelling activities.

2. Curriculum Development

ScratchJr is a derivative of Scratch, used by more than 10 million people around the world (Faber, Wierdsma, Doornbos, van der Ven & de Vette, 2017; Kalogiannakis & Papadakis, 2017). However, write code to the basic reading skills, therefore, developers need to another language, it will provide a simplified way, when they were younger learning code, without any reading. ScratchJr is a visual programming language designed to introduce programming skills to children between the ages of 5 and 7. By creating projects in the canoe, children can learn to think creatively and rationally, even though they can't read. It's free for iOS, android and chromebooks. The code is created by dragging blocks to a coding region and combining them. All blocks are based on ICONS (no text), the language children can use before reading. Blocks are connected from left to right, just like words. The user interface is much simpler than Scratch. The number of categories of blocks and the number of blocks in each category are reduced, so only the most basic categories remain.

Researchers design and teach ScratchJr's lessons. The course lasts 16 weeks, one session a week for 40 minutes, and the researchers designed an 18-page lecture themselves. Table 1 shows the outlines of the course. After each student has completed the course, after each stage model, they produce an animation story that USES ScratchJr to create their own. The animation stories created by these students themselves will serve as the basis for the research objectives. ScratchJr's lessons allow students to learn from building block programs to think in math. In addition to being effective in science, technology, engineering, and mathematics, these learning algorithms can be applied to many fields, such as social science, writing, and music. ScratchJr is designed with three modes: basic, intermediate, and advanced. ScratchJr's six types and 28 building block programs are divided into three modes of difficulty. Four lessons in the each pattern is, the former three classes are used to the use of the teaching building program, the last section is to allow each student to make a work, each work with LonelyScreen video, finally on YouTube, for later research.

Table 1. Syllabus of course model.

MODE	ACTIVITY	TIME
PREPAPRATION	Code.org course	One session
INTRODUCTION	Introduction to the basic usage of iPad and ScratchJr interface.	One session
BASIC	Choose roles, add or remove roles, change backgrounds, add or delete pages Learn the basic programs by building blocks Be familiar with the following functions: Start on Green Flag (B11), Move Right (B21), Move Left (B22), Move Up (B23), Move Down (B24), Turn Right (B25), Turn Left (B26), Hide (B35), Show (B36), Set speed (B53), End (B61), Go to Page (B63) Produce a multi-page, multi-character animation	Four sessions
INTERMEDIATE	Learn to touch, repeat, modify roles and backgrounds, and add text Learn the intermediate programs by building blocks Be familiar with the following functions: Start on Tap (B12), Hop (B27), Go Home (B28), Say (B31), Grow (B32), Shrink (B33), Reset Size (B34), Wait (B51), Stop (B52), Repeat (B54) Produces a multi-page, multi-character, and text-based animation	Four sessions
ADVANCED	Learn control flow, infinity, add sound Learn advanced programs by building blocks Be familiar with the following functions: Start on Bump (B13), Start on Message (B14), Send Message (B15), Pop (B41), Play Recorded Sound (B42), Repeat Forever (B62) Produce a full story with multiple pages and characters and a written narrative	Four sessions
PEER ASSESSMENT	Watch classmates' work	One session
REFLECTION	Students interview each other and use the iPad mini to record videos	One session

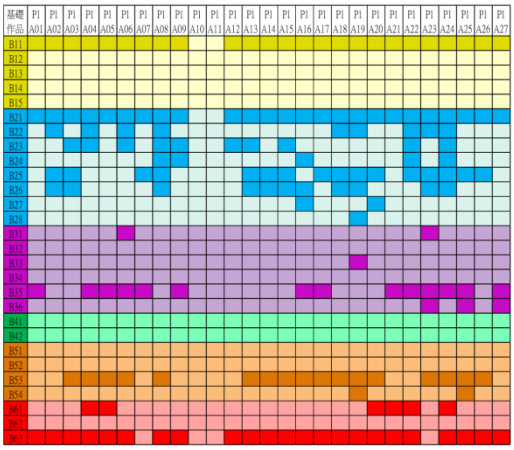
3. Data Collection And Evaluation

This study collects three ScratchJr projects completed by each student after three stages: basic, intermediate and advanced. Therefore, each student should have three works, one for each of the basic, intermediate and advanced stages. However, due to the absence, only 24 students have three complete works, two students have only one work and one student has only two works. Finally, the students interviewed with each other. The two groups used the iPad mini as a tool to interview each other's questions, and then recorded the interview results. When students interview with each other, the researcher designed the following questions to provide as interview guidelines:

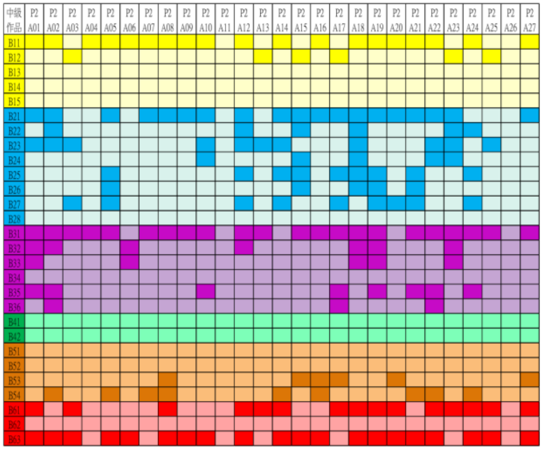
What's your story about?
What are the blocks your character uses? Why do you choose those bricks?
Which part do you think you are doing best?
Do you think there are any other ways to play the role's building block arrangement?
Does your story go according to your meaning?
Do you think it's difficult to use blocks to make character movements?
If you have more time, what would you like to add?
Do you think it's interesting to tell a story with ScratchJr? What's more interesting?

Because of the absence, only 25 interviews were recorded. This means that the total number of students who have comprehensively analyzed the three works is 24, and the video record of the analysis and discussion is 25.

In 28 basic ScratchJr projects, the blocks selected by students can be shown as Fig. 1. At first it was the basic stage mode, so the students only used the building block program B11 in the yellow startup building block program. Blue moving blocks program taught B21 to B26 these six blocks, but there are students from beginning to end in B21 a building program, and it is the students have been using B27 and B28 hasn't taught the building blocks of the program, because blue mobile building blocks, program is relatively simple building blocks. Only B35 and B36 were taught in the stage of purple building blocks, but some students have already used B31 and other building block programs. The building block program for orange control only teaches B53, so some students use it. As for B54, which has not been taught by students, it is wrong to examine its usage, so it will not be included in the project of computational thinking. The final red end of the block program, because to use the screen to switch to the next page, so most students are useful to B63. In fact, if the program sequence is complete, B61 will be used, but because it does not affect the results, many students are useless.



基礎 作品	P1 A01	P1 A02	P1 A03	P1 A04	P1 A05	P1 A06	P1 A07	P1 A08	P1 A09	P1 A10	P1 A11	P1 A12	P1 A13	P1 A14	P1 A15	P1 A16	P1 A17	P1 A18	P1 A19	P1 A20	P1 A21	P1 A22	P1 A23	P1 A24	P1 A25	P1 A26	P1 A27
B11	■	■	■	■	■	■	■	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
B12																											
B13																											
B14																											
B15																											
B21	■	■	■	■	■	■	■	■	■			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
B22		■		■		■		■										■	■			■	■	■			
B23			■	■		■		■	■			■	■		■							■		■			
B24								■	■							■						■		■			
B25		■	■				■	■					■	■	■		■	■	■	■		■	■	■	■	■	
B26		■	■					■					■	■	■	■		■	■				■	■			
B27																■				■							
B28																			■								
B31						■																	■				
B32																											
B33																			■								
B34																											
B35	■			■	■	■	■		■							■	■				■	■	■	■	■		■
B36																							■		■		■
B41																											
B42																											
B51																											
B52																											
B53			■	■	■	■		■					■	■	■	■	■	■	■	■			■	■	■	■	
B54																			■						■		
B61				■	■															■	■	■		■			
B62																											
B63	■	■	■	■	■	■		■	■			■	■	■	■	■	■	■	■	■	■	■		■	■	■	■



中級 作品	P2 A01	P2 A02	P2 A03	P2 A04	P2 A05	P2 A06	P2 A07	P2 A08	P2 A09	P2 A10	P2 A11	P2 A12	P2 A13	P2 A14	P2 A15	P2 A16	P2 A17	P2 A18	P2 A19	P2 A20	P2 A21	P2 A22	P2 A23	P2 A24	P2 A25	P2 A26	P2 A27
B11	■	■		■	■	■	■	■	■	■		■		■		■		■	■	■	■	■		■			■
B12			■										■		■		■						■		■		
B13																											
B14																											
B15																											
B21	■	■			■		■	■	■	■		■		■	■	■	■	■	■	■	■	■	■				■
B22		■										■			■			■					■	■			
B23	■	■	■							■		■	■	■				■				■	■		■		
B24										■					■			■				■	■				
B25					■							■		■	■		■	■	■		■			■			
B26					■										■			■	■		■						
B27			■		■							■		■			■		■	■	■			■			
B28																											
B31	■	■	■	■	■		■	■	■	■		■	■		■	■	■	■	■		■	■	■	■	■		■
B32	■	■				■						■						■	■				■				
B33	■					■												■	■				■				
B34																											
B35	■	■								■							■		■		■	■		■			
B36		■															■					■					
B41																											
B42																											
B51																											
B52																											
B53								■							■	■	■			■							■
B54		■			■		■	■						■		■					■	■		■			
B61	■		■					■				■	■	■			■	■	■	■		■	■	■	■		■
B62																											
B63	■	■	■		■	■		■	■	■		■		■	■	■		■	■	■	■	■		■	■		■

Figure 1 & 2. Distribution table of blocks in basic projects (left) and intermediate stage (right).

What ScratchJr building block programs do students use in intermediate works, as shown in Fig. 2. In this stage, we taught B12 in the yellow startup building block program, so the student's work no longer begins with B11. Blue mobile blocks program taught B27 and B28 this two blocks program, but students don't use B28, because in the operating interface is B28 button, so students don't have to add it to the program in the sequence. The purple building blocks taught B31, B32, B33, and B34 at this stage. Students prefer to use B31, which is useful for most students. The orange-controlled block program teaches B51, B52, and B54, and because B54 has examples to learn, many students know how to use this important block program. The red end of the building blocks, program and no protestant blocks, but because is the next page so B63 still a lot of people in a row, though students know B61 not doesn't matter, but in order to the integrity of the program sequences, there are more students use the building blocks of the program.

The advanced mode teaches B13, B14, and B15 in the yellow startup block program. All three are important building block programs, especially B14 and B15, not only important but probably the most difficult building block programs in ScratchJr as a whole. More time was spent teaching, and examples were used to aid the explanation. Then there's the green sound building block program, which teaches B41 and B42. The two building block programs are simple, but the environment can be a distraction, so not many students use the green sound building block program. Finally at the end of the red building program taught in B62, to ask students to pay attention to the difference of B62 and B54, but really want to use this building program in the story is not much, so rarely used by students. The analysis results are shown in Fig. 3. Because B14 and B15 this two blocks programs involve wide, the most available to 6 kinds of color change, if you use the two blocks, and use it correctly, that has learned of the parallel operation thinking concepts, events and conditions, namely C13, C14 and C15. There are also repetitive use and blending and abstraction and modularization in operational thinking practices, namely C23 and C24. Take the advanced work of student A25 for example, P3A25. He wrote a very long program number, but only used B11, B14, B15, B31, B61 and B63 building block programs. Blue and green didn't work, and purple did. But the whole process is correct, so that the role of the next action is continuous and smooth, indicating that his thinking logic is very clear.

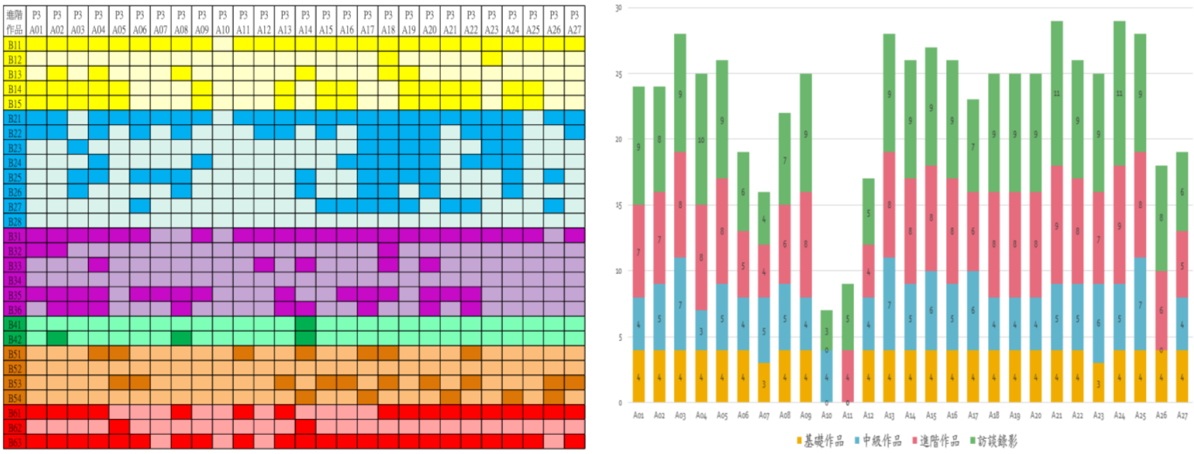


Figure 3 & 4. Block distribution in advanced mode (left) and students' CT ability in three stages.

After finishing the basic, intermediate and advanced works, the students were asked to use the iPad mini as the interview tool to interview each other's questions. Record the interview process using the iPad mini as analysis material. Take the interview process of student A16, P4A16 as an example:

[00:00] Interviewer: now, would you please tell me the outline of your story?

[00:02] Presenter: one day a rabbit and a turtle are racing to see who can run fast. The rabbit was far ahead. He thought the turtle was too slow, so he took a nap under the tree. Still undaunted, the tortoise ran on, and passed the hare. When the hare reached the end of the race, the turtle was startled to know that he was first.

[00:36] Interviewer: what are the building blocks for your character? Why?

[00:40] Presenter: I used orange blocks because turtles are slower and rabbits are faster. You can adjust the speed.

[00:51] Interviewer: which part do you think is the best part?

[00:54] Presenter: the action of a character.

[00:59] Interviewer: Do you think there is any other way to the character's building block arrangement?

[01:05] Presenter: You can also use the blue block, you can go to later let him move.

[01:15] Interviewer: You have ...
Did your story go according to your meaning?

[01:20] Presenter: Yes!

[01:22] Interviewer: Do you think it's difficult to use blocks to make character movements?

[01:27] Presenter: No.

[01:30] Interviewer: If you have more time, what would you like to add?

[01:35] Presenter: Make the story a little bit longer, with a little more content. [01:41] Interviewer: Do you think it's interesting to tell a story with SCRATCHJR? Where's the fun?

[01:48] Presenter: Interesting, because you can create a variety of roles.

Student A16 used long sentences to make his story very clear. He uses the B35's invisibility to create an effect of distance, using two characters in the car representing turtles and rabbits. In his story, B53 is not only used to control how fast the characters move, but also B14 and B15 are used to control the order of dialogue. The best part is that he uses ScratchJr to express common stories. But it's important that the last sentence of the interview, ScratchJr, allows him to create all kinds of characters. That is, ScratchJr allows him to express his designs and ideas.

An ability map, illustrated as Fig. 4, was made of the number of abilities in the computational thinking framework reflected in the three-stage works and interview videos of 27 students. It can be seen from the graph that the students have more and more projects in the computational thinking framework after the basic, intermediate and advanced courses and interviews, and almost all the expression skills of the operational viewpoints in the framework of computational thinking is achieved. It can be found that only four students did not reach 50%, and two of them only completed one work. As for the other two students who did not reach 50 percent, one of them had a learning disability. Therefore, it is feasible to integrate computational thinking into the teaching of digital storytelling to improve the expression ability.

4. Conclusion And Future Work

Because this research is to use the National small information technology curriculum to carry out, but the computational thinking emphasizes that when the computational thinking is truly integrated into the human activities of the whole, computational thinking is as an effective tool for solving problems, everyone should master, everywhere will be used. Therefore, it is suggested that the future of the different disciplines can be integrated into the calculation of thinking, training and training. For example, the mathematics curriculum can take the four steps of computational thinking: Disassembly problem, pattern recognition, abstraction and algorithm, as the direction of the problem-solving process. It should not be confined to information technology courses. This study uses both the ipad Mini and the ScratchJr ipad app. But in this era of rapid information technology, there are a lot of things like the ipad Mini and ScratchJr ipad app, and probably a better learning platform than the ipad Mini, which is more convenient than the SCRATCHJR ipad app for visual programming. Statement Therefore, it is suggested that different tools can be used in the future to achieve better learning results. But computational thinking is a universal way of thinking and basic skills that should be actively studied and used by all. Therefore, it is suggested that teachers of different courses should be taught to design teaching courses, to carry out learning activities, or to be a team, with teachers from different disciplines, to design teaching courses together. This interdisciplinary subject integration teaching course is in fact the direction of future curriculum reform.

Acknowledgements

The research reported in this paper has been supported by the Ministry of Science and Technology in Taiwan under research project numbers MOST 104-2628-S-024 -001 -MY3 and 105-2511-S-024 -007 -MY3.

References

Chang, C. K., Tsai, Y. T., & Chin, Y. L. (2017, July). A visualization tool to support analyzing and evaluating Scratch projects. In *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)* (pp. 498-502). IEEE.

Clark, D. B., Tanner-Smith, E. E., & Killingsworth, S. S. (2016). Digital games, design, and learning: A systematic review and meta-analysis. *Review of educational research*, *86*(1), 79-122.

Faber, H. H., Wierdsma, M. D., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, *12*, 13-24.

Forte, A., & Guzdial, M. (2005). Motivation and nonmajors in computer science: identifying discrete audiences for introductory courses. *IEEE Transactions on Education*, *48*(2), 248-253.

Gorla, N., & Lam, Y. W. (2004). Who should work with whom?: building effective software project teams. *Communications of the ACM*, *47*(6), 79-82.

Han, K. W., Lee, E., & Lee, Y. (2009). The impact of a peer-learning agent based on pair programming in a programming course. *IEEE Transactions on Education*, *53*(2), 318-327.

Hayes, E. R., & Games, I. A. (2008). Making computer games and design thinking: A review of current software and strategies. *Games and Culture*, *3*(3-4), 309-332.

Kafai, Y. B. (2012). Learning design by making games: Children's development of design strategies in the creation of a complex computational artifact. In *Constructionism in practice*(pp. 87-112). Routledge.

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, *50*(4), 313-334.

Kalogiannakis, M., & Papadakis, S. (2017, August). Pre-service kindergarten teachers acceptance of “ScratchJr” as a tool for learning and teaching computational thinking and Science education. In *proceedings of the 12th Conference of the European Science Education Research Association (ESERA),«Research, practice and collaboration in science education». Dublin City University and the University of Limerick, Dublin, Ireland* (pp. 21-25).

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality.

Williams, L. A., & Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, *43*(5), 108-114.