Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Perceptions of Teacher and Students towards Integrating Computational Thinking into Language Education: A Pilot Study

Xiaojing WENG [a*], Ching-Sing CHAI[b], Morris S. Y. JONG[c], Gary K. W. Wong[d]
[abc]*Department of Curriculum and Instruction, The Chinese University of Hong Kong, Hong Kong*
[d]*Faculty of Education, The University of Hong Kong, Hong Kong*
*xweng@link.cuhk.edu.hk

Abstract: As a problem-solving skill, Computational Thinking (CT) is getting more and more popular beyond the spectrum of Computer Science. Scientists and researchers in the field of K-12 education particularly have a passion for it. Language skill, an essential ability for every individual, is one of the most significant capabilities which students need to cultivate. It is possible that students can improve their language skill to communicate when adopting or learning CT. Insufficient literature has explored the feasibility of connecting language education with CT activities. This study aims to investigate teacher and students' perceptions towards integrating CT into language education and explores how to improve the integration of CT into language education with the elements of programming. In order to achieve these objectives, the researcher conducted a pilot case study by interviewing one teacher and four students who have attended the maker space lesson in an elementary school in Hong Kong. The interview results show that the teacher and students hold positive perceptions towards integrating CT into language classes. In addition, their recommendations to better connect CT with language education through programming instruments were presented.

Keywords: Computational thinking, language education, constructionism, programming

1. Introduction

As a promising problem-solving ability rooted from Computer Science education, CT is adaptable to different aspects of people's life (Wing, 2006). Some researchers have been attempting to explore the power of CT. A typical approach for younger children to connect CT with programming practice is to use graphical platforms like Scratch, Alice, and Minecraft. Researchers have been examining how students' creativities and problem-solving abilities are cultivated through CT (Chang, 2014; Pellas & Peroutseas, 2017; Zhong, Wang, Chen, & Li, 2016). Given that computers have the potential to make learning more interesting (Jong, Shang, Lee, & Lee, 2008), studies have also been conducted to combine CT with writing, journalism, or poetry to probe some insights about the influence of CT (Zhong et al., 2016). However, inadequate investigations have been carried out on how CT influences students' language learning experience (Burke & Kafai, 2012), which deserves more attention both out of the consideration of students' language development and CT infusion.

Actions are required to fulfil the potential of CT in the area of language education. This research attempts to contribute to the literature by exploring the perceptions of students and teachers about the feasibilities of connecting CT with language education by adopting the CT instrument Scratch in English language learning class. Therefore, there are two targets for this study. One purpose is to understand students and teachers' perceptions on integrating English learning and graphical programming language learning after an intervention involving the infusion of CT into English learning. Then suggestions on integrating CT with language education will be put forward according to the research results.

2. Literature Review

Since Wing (2006) put forward the concept of Computational Thinking, research in this area has been thriving. Wing's work inevitably leads to discussions about the foundation issues of CT, which have not received a commonly recognized definition yet (Grover & Pea, 2013). Nonetheless, there is a deepening consensus of investigating the possible effects CT may have on the 21st century K-12 students (Yadav, Hong, & Stephenson, 2016). Another area of research revolves around designing valid instruments to assess CT (Lockwood & Mooney, 2017).

Second language education is one of the most challenging and essential parts of K-12 education. Acquiring fluency in the second language can greatly enhance one's market competitiveness. People who are capable of a second language have more job opportunities to work in the multinational corporations. Furthermore, the English language is likely the lingua franca for people to communicate all-through the world. Half of the world knowledge is stored or communicated through English language (Seidlhofer, 2009). We were unable to locate literature in our search through major research databases on the published study which employed CT as an independent variable to influence students' language learning. However, some insights about the cognitive aspects of students' language learning process in the context of CT could be found from the previous researches.

Burke and Kafai (2012) have investigated the possibility of introducing fundamental programming concepts to secondary school students in a writing workshop. In their study, students used programming tool Scratch to draft, revise, and publish digital stories made by themselves. While finishing their writing tasks, students studied the basic Computer Science concepts and recognized that programming and writing could operate in an interrelated approach while finishing the composition task (Burke & Kafai, 2012). Jenkins (2015) conducted a study to explore the possible relationship between language learning and CT by using computer-based microworlds to teach English poetry. The researcher employed an experimental design to compare students' performance in both of CT and poetic thinking before and after the intervention English lessons. Based on the research findings, students gained growth both in their CT and poetic thinking after the intervention. The researcher has presented some evidence that there is a positive correlation between the two thinking capabilities (Jenkins, 2015). If the integration of CT and students' language ability is feasible as the previous researchers have proposed, it is important to explore the possibilities of integrating CT into K-12 students' language education.

From the dimension of English language education, this practice is designed with respect to task-based instruction (TBI) theory (Leaver & Willis, 2004). TBI recommends students to develop their language by conducting meaningful tasks. For example, making an appointment with a doctor, taking an interview, or completing a story. Ellis (2003) defined that a qualified task should have four features, including having real meaning, having 'gaps' to be fulfilled, needing leaners to use linguistic resources to finish the task, and having non-linguistic task outcome. Well-designed tasks can enhance English language learning in an effective way (Ellis, 2003).

One theoretical framework that supports this approach to develop CT is constructionism. Constructionism describes the process of gaining knowledge as "building knowledge structures" (Harel & Papert, 1991). Inspired by his teacher Piaget's constructivism which emphasizes learners' previous experience in constructing new knowledge, Papert's constructionism stresses that people gain new knowledge by engaging in doing and making artifacts, regardless of learning circumstances (Li, Cheng, & Liu, 2013). Constructionism provides some foundations for educational applications of programming in learning. With the spread of maker movements, constructionism emerges as a prominent theoretical foundation for learning by doing. In recent years, constructionism has extended the reach of its development to applied linguistics. For example, in 2004, the game SimCity was adopted as an instructional tool to teach Japanese high school students English with constructionist techniques (Pavlidis & Markantonatou, 2018). Nonetheless, applying constructionism in the computational context for language learning needs many further studies. Accordingly, this proposed research aims to pursue a fair understanding of such explorations. At this point, its feasibility is explored through qualitative interviews with students and the teacher.

3. Research design

3.1 Research Participants

This study was conducted in a primary school in Hong Kong. Students studied major subjects including Chinese Language, English Language, Mathematics, and General Studies. Besides, students also studied some technical subjects like Science, Technology, Society & Humanities, Art and Physical Education as the supplementary subjects. Even though because of the academic pressure and resource limit in elementary schools, major subjects occupied the largest portion of students' time, the development of STEM education encouraged school leaders to highlight the importance of Science and Technology. As a result, all elementary students in this school had a chance to take Computer Science as a selective interest-oriented subject once a week. There were 9 students study in this selected class when the research was conducted. The teacher called this class "Maker Space" and employed Scratch to teach students programming. Students came from different grades above grade 4, when was the starting grade for them to learn programming.

3.2 Learning Activity

Supported by TBI in language education and constructionism in CT development, the class consisted of the following phases:

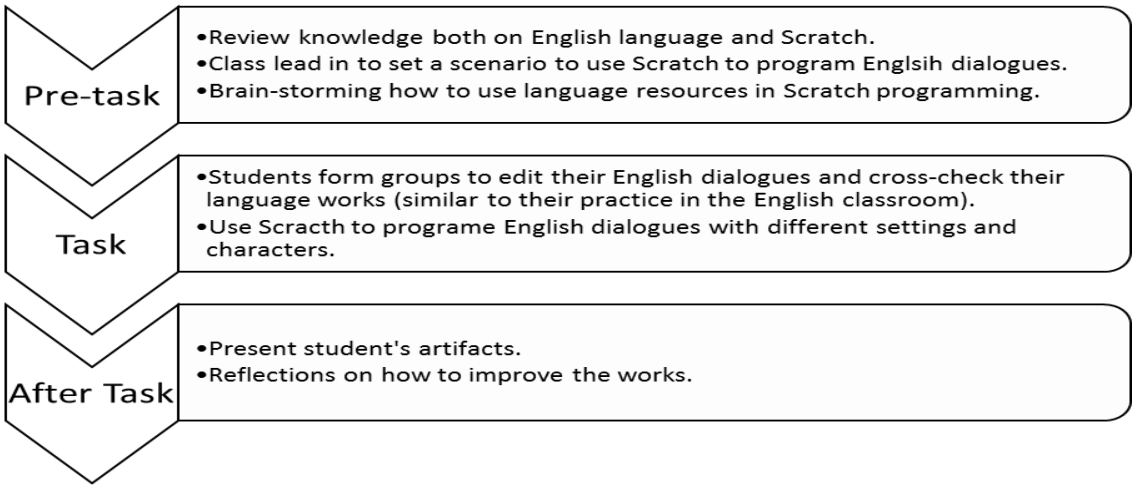

Figure 1. The One-Session Intervention Class.

3.3 Semi-Structured Interview with Students

Four students were selected to be interviewed (see table 1) after the intervention based on the criteria that interviewees should be different in age and gender, teacher's suggestion and researcher's classroom observation were also taken into consideration in this process. The interviews were conducted at the most convenient time for students after the lesson. Each interview lasted 10 to 15 minutes and all were audio-recorded with students' consent. The interview guide was adopted by the researcher to investigate student's opinions on (1) Scratch learning, (2) English language learning, (3) CT-infusing class and (4) their suggestions on how to better bridge CT and English education. Meanwhile, the researcher followed several rules to reduce the Pygmalion Effect. For instance, all the questions asked in the interviews were open-ended, no misleading questions were posed throughout the whole process, interviewees were encouraged to express their personal views freely, and questions were clarified more than once when inconsistencies arose.

Table 1

Student Interviewees' Information

	Student 1	Student 2	Student 3	Student 4

Age	10	10	10	11
Grade	4	5	4	5
Gender	Male	Female	Male	Male

3.4 Semi-Structured Interview with the Teacher

The teacher took on three different roles in the school under study: he was primarily an English teacher for P5 students, but he was also the ICT coordinator given his interest in ICT. Besides, he taught Computer Science, in which he promoted Scratch programming. The interview with the teacher lasted approximately 40 minutes. Questions about his perceptions on (1) ICT, (2) STEM education, (3) Scratch programming teaching, and (4) the challenges/opportunities of using programming as a tool for English language teaching were discussed in detail. This interview was audio-recorded with the teacher's consent.

4. Results

4.1 Perceptions towards integrating English learning and graphical programming language

First, three out of four students thought the experience was interesting. For example, student 2 said "I think it was very interesting to study English dialogue with Scratch. I could conduct conversations with the computer. However, in our English class, our teacher tended to adopt the simplest approach. He often has us study English dialogue together with other students who sit in front or behind us nearby. I prefer using Scratch in English dialogue learning" and student 3 said "I think our class in which we studied English dialogue with Scratch was very impressive. In all my experience of learning Scratch, that infusing class was the most interesting one. I was able to make the characters by myself and control their movements. The best part was the way in which the computer reacted to my responses; when I replied to its question, it would continue to ask me questions or finish the dialogue. These different reactions were provided based on my answers". Both of them would like to engage in more such activity. However, Student 1 said that he thought using Scratch in English learning was excellent and he enjoyed the change from the usual English classes, but he did not want to have more of this kind of infusing class since he was confused about the learning focus. In the interview, student 4 said the class which adopted Scratch in English dialogue learning was neither very good nor very bad. He had learned English dialogue in that class through communicating with the computer. The class itself was amusing; he could solve all the programming problems he had come across easily through learning from the online community.

The teacher presented his positive perceptions towards this infusing class; he said: "I think the interventional class was good enough for me to achieve my teaching goals in helping students to finish their English dialogue Scratch projects while leaving them room for creativity."

Overall, it seems that integrating CT for English learning has the potential to improve interest of learning as an alternative in the traditional classroom teaching and learning. As students today are exposed to many different ways of learning, such an approach should not be discounted, especially for students with high interest in computing but low interest in traditional English lessons. Nonetheless, it was also clear that some students will focus more on the programming than English learning while some may feel no need for such intervention.

4.2 Suggestions for integrating CT into English education via a graphical programming language

Students suggested a range of further activities that were automated, more challenging and enriched through multiple contexts. Student 1 hoped in the future the computer could ask him English questions by itself rather than being limited to the questions and answers that he had programmed in Scratch before. He wanted to do something else next time. Student 2 regarded the English language requirements should be suitable. She hoped next time they could introduce more characters in

Scratch; just two sides of the conversation were not sufficient for her to think about more extended dialogues. Student 3 would like the computer to react automatically to his questions or answers without pre-designed programming. Student 4 hoped Scratch could make English learning activities into a game, for example, a Racing Game; while the racing car is running around the track, it could ask questions for its next operation and react accordingly after the answer was given. This would be more interesting than merely communicating with the computer.

In the interview, the teacher proposed that Scratch may help little in English learning compared with some other ICT scaffolders, like Powerpoint, Photostory or story maker, since it required the high-level thinking skill CT from students. But he suggested Scratch could be applied in the summary project or in a final assignment in English learning. For instance, after students gained some knowledge of vocabulary and grammar, they could use Scratch to make an animation, or they could use Scratch to introduce objects like tourist attractions in Hong Kong. He said: "In that infusing class, my goal was not to teach students how to learn English, but to teach them how to program with Scratch and develop their CT. However, I can see students' interest was piqued in the class. Just like how students draw their characters on the paper while they are writing, in this infusing class they can present more vivid characters in Scratch and conduct dialogues interactively. This can improve their confidence in this class and increase their interest in English."

5. Conclusion and Discussion

Generally speaking, students and the teacher welcomed the novel practice of utilizing Scratch in other courses. From the interviews, it was found that students expressed some interest in learning English with Scratch. During the infusing class, Scratch served as a tool for expressing English, which was consistent with the writing workshop designed by Burke and Kafai (2012). Hence, students not only cultivated CT by using this programming tool but also learned about the English dialogues involved.

A big challenge in K-12 education is how to maintain students' sustained interest and attention. Constructionist learning theory ((Papert, 1980) seems to be implicitly addressing this issue by engaging students in building digital artifacts, of which is an engaging mediated process with emerging artifacts representing one's understanding. In this study, Scratch was adopted to meet students' needs and could keep children's interest. Learners can gain knowledge of other course being taught in the context of the Scratch environment in a subtle way, and they would not get bored when they were studying. In addition, since all the interviewees have learned Scratch and English both, they were willing to propose their opinions on how to take advantages of Scratch in English learning. This phenomenon could be regarded as supporting evidence for the positive influence of the class intervention.

Interviewees suggested some advice to connect CT with English education through graphical programming instruments in the semi-structured interviews as well. In general, these suggestions are pedagogical in nature. First, suggestions seem to be pointing towards automated pedagogical support to scaffold students in crafting or reshaping dialogues. Students thought it would be better if computers could ask people questions or give responses to people's requirements automatically rather than operating the pre-designed programming procedures. There was the proposal that researchers can gamification English learning activity while connecting CT with the English language, which would be more interesting compared with the current storytelling design. Second, when teachers are designing the learning environment they need to consider carefully about students' prior knowledge so as to prevent cognitive overload (Sweller, 1988) or students focusing on computation rather than language learning. From the point of English learning, one interviewee mentioned that both of learners' graphical programming skills and their English language capabilities should be taken into consideration. By doing so, students could reach their potentials in learning CT and English language. In the teacher's interview, he recommended that Scratch can be used as a summative tool to facilitate language tasks, this was mainly because he would like to reach a balance of the tool and content issues and keep class efficiency.

The feasibility of integrating language learning and computational thinking is a matter of pedagogical design in considerations of human cognition in general and students' prior knowledge in situ. Students and teacher in this pilot study hold a positive view or at least non-negative view

towards integrating language learning and computational thinking, this is premised on the fact that they have some computation skills and thus it did not result in cognitive overload. This is the limitation of this study.

Future studies around the topic of integrating CT into language education could be expanded to involve more elements. For instance, researchers can employ different scales with validity and reliability to evaluate students' learning gains both in the dimensions of CT and language education as learning outcomes of the infusing class. Furthermore, as research has proposed that teachers' pedagogical beliefs and their planning and conduct of computer-mediated classroom lessons are critical for students' learning process (Lim & Chai, 2008), professional development and communications are significant both for teachers who are pre-service and in-service. Assuming that teachers understand CT and know how to teach it effectively, students will have a chance to study CT in an efficient approach. As teachers are taking the responsibility to teach young generation new skills, we recommend that communities of practice should be built for teachers to support their constant learning.

References

Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (SIGCSE '12). ACM, New York, NY, USA, 433–438.

Chang, C. K. (2014). Effects of using Alice and Scratch in an introductory programming course for corrective instruction. *Journal of Educational Computing Research*, *51*(2), 185–204.

Ellis, R. (2003). *Task-based language learning and teaching*. Oxford: Oxford University Press.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Norwood, NJ: Ablex Publishing.

Jenkins, C. (2015). Poem Generator: A comparative quantitative evaluation of a microworlds-based learning approach for teaching English. *International Journal of Education and Development using ICT*, *11*(2), 153–167.

Jong, M. S., Shang, J., Lee, F. L., & Lee, J. H. (2008). Harnessing computer games in education. *International Journal of Distance Education Technologies*, *6*(1), 1–9.

Leaver, B. L., & Willis, J. (eds.) (2004). *Task-based instruction in foreign language education: Practices and programs*. Washington, DC: Georgetown University Press.

Li, Z. Z., Cheng, Y. B., & Liu, C. C. (2013). A constructionism framework for designing game-like learning systems: Its effect on different learners. *British Journal of Educational Technology*, *44*(2), 208–224.

Lim, C. P., & Chai, C. S. (2008). Teachers' pedagogical beliefs and their planning and conduct of computer-mediated classroom lessons. *British Journal of Educational Technology*, *39*(5), 807–828.

Lockwood, J., & Mooney, A. (2018). Computational Thinking in Secondary Education: Where Does It Fit? A Systematic Literary Review. *International Journal of Computer Science Education in Schools*, *2*(1), 41–60.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Pavlidis, G. P., & Markantonatou, S. (2018). Playful Education and Innovative Gamified Learning Approaches. In *Handbook of Research on Educational Design and Cloud Computing in Modern Classroom Settings* (pp. 321-341).

Pellas, N., & Peroutseas, E. (2017). Leveraging Scratch4SL and Second Life to motivate high school students' participation in introductory programming courses: findings from a case study. *New Review of Hypermedia and Multimedia*, *23*(1), 51–79.

Seidlhofer, B. (2009). Common ground and different realities: World Englishes and English as a lingua franca. *World Englishes*, *28*(2), 236–245.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, *12*(2), 257–285.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–36.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568.

Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, *53*(4), 562–590.