

Facilitating K-12 Teachers in Creating Apps by Visual Programming and Project-Based Learning

<https://doi.org/10.3991/ijet.v15i01.11013>

Chiung-Fang Chiu

National Chi Nan University, Nantou County, Taiwan
cfchiu@ncnu.edu.tw

Abstract—The use of apps in education has become popular in recent years with the development of technology. How to prepare K-12 teachers in creating apps for students learning is worth our attention. This paper presents how to apply visual programming and project-based learning to facilitate K-12 teachers in creating apps. Seven K-12 teachers with less computer background participated in this study. Data source including (1) K-12 teachers' project codes, (2) K-12 teachers' questionnaire, (3) K-12 teachers' interviews, and (4) questionnaire giving to K-12 students were collected and analyzed. The results reveal that K-12 teachers applied many programming concepts in the coding implementation in these projects. Most students gave positive feedback to these K-12 teachers' app projects. Furthermore, K-12 teachers were also very positive toward the creation of apps by visual programming and project implementation, which was reflected through K-12 teachers' questionnaire surveys and interview results. The above findings provide a practicable approach for future teaching to K-12 teachers or novice programmers on app programming.

Keywords—Programming, Application, Visual programming, Project-based learning

1 Introduction

Mobile technology has become significant part in our daily life. A wide variety of applications (apps) have been developed ranging from diverse purposes including entertainment, business, education, personal use and more. It has changed the way work, education, communication, and collaboration functions. Mobile learning using educational apps to solve temporal and spatial limitations and also support ubiquitous learning opportunities has become popular in recent years. Some reports show that many teachers and students have positive attitudes towards mobile learning or applying apps in education. For example, Domingo and Garganté surveyed teachers' perception about the impact of mobile technology in learning [1]. A total of 102 teachers of 12 different primary schools in Spain participated in this study. The result reveals that teachers thought facilitating access to information and increasing classroom engagement are the two main possible effects of mobile technology in the classroom. Jumaat, Tasir, Lah

and Ashari's conclusion from 11 articles indicates that students believed mobile technology will enhance their learning process and provide an effective way of learning [2]. Other research has yielded similar results: mobile learning facilitates access to information, increases classroom engagement, and provides new ways to learning [3].

The potential benefits of using apps in learning has been recognized in many disciplines. Aronin and Floyd suggested learning activities to introduce STEM concepts to preschool children through various play-based apps [4]. Beal and Rosenblum examined an iPad app for its effectiveness in helping visual impaired students studying algebra to solve mathematical word problems [5]. Visually impaired students and their teachers found the app and graphics to be easy to use and motivating. Students answered more mathematics problems correctly when using the iPad app. Soto-Martín and Lodeiro-Santiago integrated apps in the practice and teaching of conservation and restoration of cultural heritage [6]. Research of [7] utilized the mobile app to enhance the scientific inquiry experience of the students, especially in data gathering and complex scientific practices. Additionally, there are also reports about the benefits of using apps in early childhood education. For example, in the study by Kucirkova, Messer, Sheehy and Panadero, one app called “Our Story” was designed to support young children’s engagement in story-making activities [8]. The findings demonstrated that children’s individual engagement was more animated when using “Our Story” app than using other app software.

Applying commercial apps to education has great positive effects, however, some constraints may still apply. For example, the functions and contents of the apps might not match the purposes of the teacher. Additionally, schools usually have limited budget for buying apps for teaching. Consequently, if teachers can become more familiar with computer programming to develop mobile apps suitable for students’ learning condition and need, a better learning outcome may then be achieved. Considered that many teachers usually are non-majors and lack related computer programming background, they might face more challenges than youth when learning how to create apps. More research is needed in terms of selecting the most fitting programming creation environment and implementation approach.

Most novice programmers think programming is difficult to learn. Empirical studies show that programming syntax remains a major barrier to students [9-10]. Programming syntax is different from our everyday language syntax. As a result, most students find it non-intuitive and hard to understand. It is more common for students to spend more time dealing with issues of syntax rather than focusing on important algorithms and program structure designs. A visual programming environment provides easy-to-use interface that students can construct programs merely by manipulating graphical elements. It can reduce the cognitive load for novice programmers when programming concepts are first introduced to them [11]. Thus, one can expect the visual programming environment to be suitable for K-12 teachers when learning to create apps.

Unlike finishing a simple programming homework, creating educational mobile apps comprises several stages, such as students’ background analysis, learning content design, multimedia material collection, coding, debugging, and testing. Project-based learning emphasizes learning activities that are long-term, student-centered, and applicable to real-world issues and real-life practices. This learning process involves a series

of complex tasks which engages students' minds through working on open-ended projects, problem-solving, decision making, or investigative activities. These tasks are also necessary in mobile apps creation. Current literature demonstrates that project-based learning has been applied in various disciplines to improve learning outcomes or learning attitudes [12-14]. Therefore, in addition to visual programming environment, project-based learning was also selected in this study as a possible strategy for facilitating the learning process of K-12 teachers vis-a-vis creating apps.

2 Literature Review

2.1 Visual programming environment

Traditional textual programming languages have text-based interface that require precise and complex syntax which are unfamiliar and unnatural for beginners. The programming syntax is usually complicated and difficult to understand for novice programmers and syntax overload results in barriers to learning programming [9]-[10]. Easier to read and easier to write programs may relieve syntax complexity for learners. Instead of inputting text-based statements, visual programming environments often contain block-based programming elements which provide useful visual cues on how and where to use commands to develop programs. A program can be easily created just by dragging-and-dropping block-based programming elements so that the syntax problems are eliminated. Visualization helps learners understand the semantics of introduced constructs, elucidate principles of program structure and execution, and protects them from developing misconceptions. Study by [15] indicated that students who used the visual interface for programming spent less time off track and completed more of the activity's goals in less time when compared to those who used the textual interface for programming. Saito, Washizaki, and Fukazawa also compared the learning effects of text-based and visual-based programming and found that visual-based programming induces a larger change in attitude toward programming [16]. Therefore, research results suggest that visual-based programming is advantageous in a programming implementation environment for novice programmers.

Numerous studies have shown the benefits and feasibility of visual programming environments such as Scratch (scratch.mit.edu), Alice (www.alice.org), and Kodu (www.kodugamelab.com). Since they are easy to use and understand, visual programming environments have been used as lead-in courses for more advanced computer science courses. For example, research [17]-[18] taught Scratch in a CS0 course to improve the retention rate of at-risk majors and to stimulate interest and positive attitudes toward computer science. Scratch and Kodu were also used to teach important concepts of computer science for students [19]-[21]. Furthermore, research by Werner, Denner, Campe, and Kawamoto also pointed out that visual programming environment is helpful in the development of learners' computational thinking skills [22].

Previous research reported successful experiences of introducing visual programming for K-12 or pre-service teachers when creating interactive animated games or stories to facilitate teaching. For instance, K-12 teachers were taught Scratch, Alice, and

StartLogo in the workshop [23]-[25]. Considering that the participants of this present study are novice programmers, using visual programming environment to create apps may be better suited for them. App Inventor for Android (AIA) developed by Google, later transferred to MIT's Center for Mobile Learning, intends to enable everyone without programming background to develop Android mobile applications. AIA is an open-source web application and provides friendly visual interface which decreases the learning curve. Users, regardless of whether they major in computer science, can learn AIA successfully. Many study tried to introduce high school or university students programming concepts by teaching AIA. Teaching AIA would also be an interesting topic to attract students to choose computer science field in their future. Roy conducted a AIA summer camp for middle and high school students to introduce computing [26]. Students' disposition towards computing slight increased after the summer camp. Similarly, the research of [27] taught high school girls AIA and robotics in the summer camp to change their misconception that computer science is mainly about dealing with Office applications or 'fixing computers'. Furthermore, Perdikuri reported about senior high school teachers and students' experiences from using App Inventor in an introductory programming course [28]. Students participated in this study had been taught only Logo programming. The result shows that students and teachers had positive responses to the adoption of App Inventor in the classroom. App Inventor was also utilized to develop students' computational thinking skills through creating mobile apps [29]. In terms of teaching AIA to educators, Liu, Lin and Potter conducted a one-week AIA summer computing workshop for K-12 teachers in all subjects [30]. Teachers developed their apps for teaching with the help of workshop tutors. It reveals that K-12 teachers gained confidence in teaching computer science and in computing knowledge after the workshop. Hsu, Rice and Dawley also offered online mobile app design workshops for educators to assist their teaching, learning and work [31]. Educators commented that they enjoyed the workshops and had fun testing other people's apps. Those positive reports mentioned above suggest that App Inventor could be a programming language choice to teach K-12 teachers create their apps.

2.2 Project-based learning

Project-based learning is an instructional strategy that provides students the autonomy to learn, explore, and investigate throughout the learning process by encouraging them to do projects. As an instructional method, project-based learning is a student-centered learning pedagogy, instructors act like facilitators of knowledge. Students face authentic and motivating problems that require them to solve complex questions and to develop success skills [14]. Project-based learning enhances group work and benefits in social environment of education. It also motivates students for a greater design and application thinking [32].

Unlike traditional classroom practices that are short and teacher-centered, project based learning emphasizes learning activities that are long term, student-centered, and integrated with real world issues and real life practices. Incorporation of real-world projects into the classroom provides a unique and priceless enhancement to the educational experience [33]. Project-based learning involves a series complex tasks which

engages students by working on open-ended projects, problem-solving, decision making, or investigative activities. It entails creating an environment in which individuals actively participate in the learning process and take responsibility for their own learning. It also fosters the ability to identify learning topics and the ability to find resources and evaluate the validity of these resources. Project-based learning can also improve learning effects, such as students' self-efficacy, cognitive strategy usage, self-organization, positive attitudes toward task value and group work [12, 13, 34]. Similar to constructivism learning theory, project development process enables students to construct their own knowledge through exploration and provides opportunities for meaningful learning about the subjects of their study [35]. Higher-order thinking skills instead of the mere memorization of facts can also be enhanced in project-based learning.

Project-based learning has been widely utilized in computer science courses. Pucher and Lehner pointed out that most teachers agree that teaching with project-based learning has numerous advantages, especially in the field of computer science [36]. In addition to applying technical knowledge and acquiring practical skills in programming, students also are able to get involved in the group project process and project management. The study by Qidwai demonstrated that project-based learning methodology in robotics class enhanced students' learning motivation and achievement [37]. Lara (2019) used project-based learning in a compilers course and required students to work in groups to build a complete compiler [14]. The results revealed that students were highly motivated and capable of identifying which important skills they needed improvement. In order to integrate artificial intelligence and machine learning concepts to a variety of undergraduate courses, Liao (2019) utilized various projects to be discussed in class [38]. Project-based learning is also well suitable for courses pertaining Big Data concepts [39]. Sillaots and Fiadotau introduced an application of project-based learning to game design and discovered that it was useful for technical skill acquisition and relationship-building [40]. In programming courses, project-based learning has been used to address the gap between abstract programming concepts and authentic programming tasks [41]. In order to address the wide range of students' abilities and interests, Jazayeri combined mastery learning with project-based learning to teach Python programming to novice programmers [42]. It revealed that students were motivated by doing projects. Project-based learning was also employed to facilitate interdisciplinary learning, such as in the fields of electrical engineering, computer science and applied physics. Most students felt that working on projects improved their ability to function on interdisciplinary teams and to communicate effectively [43]. In summary, project-based learning is an effective teaching strategy for students. For this present study, it is also believed that project-based learning would benefit K-12 teachers as they learn how to create apps.

3 Research Purpose and Questions

The purpose of this research is for K-12 teachers to learn how to create mobile apps by applying visual programming and project-based learning. K-12 teachers' attitudes toward this approach were investigated. The research questions are as follows:

- What are the components of the apps created by K-12 teachers?
- What are students' responses to the apps created by K-12 teachers?
- What are K-12 teachers' perceptions of using visual programming and project-based learning in mobile apps design?

4 Research Method

4.1 Participants

This study was conducted in one graduate course which was three hours per week. Students enrolled in the course were K-12 teachers. Seven students comprising 5 males and 2 females aged from 27 to 46 participated in this study. All of them had limited programming experiences before participating in this study. K-12 teachers were taught basic programming concepts using App Inventor. The version of AIA used in this study was 2.0.

4.2 Procedure

First, basic programming concepts, such as variables, conditional statements, loop controls as well as list structures, were presented during lectures and programming practices in the first half of the course. Then, other advanced program structures or concepts, such as procedure calls and functions, were taught to K-12 teachers in the second half of the course. Before the end of the course, K-12 teachers were required to design and implement app projects for the classroom subjects that they were teaching. They started to report and discuss their app project design every two weeks from the mid-term of the semester.

Each K-12 teacher invited 10 students to use and evaluate their apps after finishing the app projects. One questionnaire was designed to gauge students' opinions. Furthermore, another questionnaire was also developed aimed at examining K-12 teachers' perceptions of using visual programming and project-based learning in app creation. Two experts, including one elementary school teacher and one professor who taught App Inventor, were invited to assess the content validity and relevance of these two questionnaires. The Cronbach's Alpha internal consistency values for the questionnaires of "Students' responses to the apps created by K-12 teachers" and "K-12 teachers' perceptions of using visual programming and project-based learning" were 0.90 and 0.85 respectively.

4.3 Project codes analysis

The content of AIA project programs coded by the participants was further analyzed to examine whether programming concepts were applied in app project implementation. The program codes of each project in AIA were examined according to the following

Furthermore, after an analysis of the app projects, it was discovered that their app projects are less complicated so that advanced program structures are not necessary for these projects (i.e. figure 1). These results reveal that K-12 teachers, as programming novices, usually lack confidence in their programming skills. Providing more scaffolding, such as giving some programming examples or templates, may increase their confidence in coding advanced program structures.

Table 1. Analysis results of K-12 teachers’ app project codes.

Grade	Target Subject	Computing concepts used in project	Multimedia used in project	Quiz types used in project
1st	Chinese (reading)	Variables, Boolean statements, if statements, loop controls, lists	Pictures, audio	Multiple selection, matching
4th	Culture	Variables, Boolean statements, if statements, loop controls, procedures statements	Pictures, video, audio	Multiple selection, fill-in-the-blanks, matching
5th	History	Boolean statements, if statements	Pictures, video, audio	Multiple selection
6th	Chinese idioms	Variables, Boolean statements, if statements, procedures statements	Pictures, video, audio	Multiple selection, fill-in-the-blanks, matching
6th	Art	Variables, Boolean statements, if statements	Pictures, video, audio	Multiple selection
7th	English	Variables, Boolean statements, if statements	Pictures, video, audio	Multiple selection, matching
9th	Art	Boolean statements, if statements	Pictures, video, audio	Fill-in-the-blanks

5.2 Students’ feedback of the apps created by K-12 teachers

Table 2 shows students’ feedback of the apps created by K-12 teachers. K-12 teachers were required to reflect on students’ responses. A total of 70 students including 33 boys and 27 girls gave feedback about their experience using these apps. Their average age is 11 years old. The results is showed in Table 2. In terms of the learning content, most students felt that the learning content is readable (question 1). They thought the learning content is interesting (question 2). The learning topic suits them (question 3). As to the interface design of the app, most students indicated that it is easy to use (question 4) and beautiful (question 5). Furthermore, many students agreed that the app can boost their learning motivation (question 6). They thought the app is very practical and it increase their knowledge (question 7). Finally, students’ were asked about their intention to use these apps. Many students indicated they wanted to see the learning content in the app more times because it was quite interesting (question 8). With regard to the quiz design in the apps, the intention to play again was a little low (question 9). The reason might be that the mainly question type of the quiz are multiple choice questions. Furthermore, the mechanism of the quiz was not well designed. Because there were not many questions in the quiz pool so that the same questions were often selected again. Students’ feedback in the open-ended question indicated that these questions in the quiz

are not very challenging or interesting. Lack of challenge and interest might also explain low intention to try the quiz again. However, students were very willing to recommend the app to their peers (question 10).

Students' responses to the open-ended question are summarized below:

"I think this app is very interesting!"

"There are too few questions in the quiz; adding more questions will be better."

"The quizzes in the app are too easy to answer, and the number of the questions in the quizzes are too few."

"I am very surprised that my teacher can design and implement an app like this!"

"This app is great! My teacher is so brilliant."

K-12 teachers' reflection on students' feedback are summarized as follows:

"I am very glad that students like my app."

"Though this app may not be perfect, I applied what I learned this semester to my job. This app boosted my confidence in designing other digital courseware."

"Students' responses forced me to notice the importance of interesting content when using app to assist learning. It also helped me to learn that interesting content design should be taken into consideration so that apps can reach their optimal potential in assisting learning."

"Many of my students were very curious about mobile app programming. It encourages them very much to learn about the coding of mobile apps when they discovered that their teacher could create mobile app by himself. This effect was a little bit unexpected."

The above results indicate that most students gave positive feedback to the apps created by K-12 teachers. Most students felt satisfied with the learning content, learning topics as well as interface design of the apps. It was noted that students' intention to play the quizzes again was a little low (see Table 2, question 9) and we can thus assume that the app quizzes did not interest them. In addition, only multiple selection questions were used for the quizzes in several apps (see Table 1), which further accentuates the fact that the quizzes were not well designed. Because there were not many questions in the quiz pool, the same questions were often selected again. In summary, increasing the number of quiz types and questions in the app are recommended in future studies.

Overall, students' positive responses encourage K-12 teachers to continue to design courseware. It's important to note that students' feedback also prompted K-12 teachers to reflect further on the design of educational software. They realized the importance of attracting students' attention when designing educational software. This experience brought K-12 teacher's confidence in designing other educational software for teaching in the future.

Table 2. Students’ responses to the apps created by K-12 teachers (N=70)

Question	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	Mean
1. The learning content in this app is readable	81%	10%	7%	1%	0%	4.7
2. The learning content in this app is interesting	54%	26%	11%	4%	3%	4.3
3. The learning topic of this app suits to me	36%	43%	14%	4%	3%	4.0
4. This app is easy to use	63%	24%	11%	0%	1%	4.5
5. The interface design of the app is beautiful	41%	36%	11%	6%	6%	4.0
6. This app can boost my learning motivation	40%	31%	19%	6%	4%	4.0
7. I think the app is very practical and it increase my knowledge	59%	24%	13%	4%	0%	4.3
8. I am willing to see the learning content in this app several times	57%	19%	13%	7%	4%	4.2
9. I am willing to play the quizzes in this app several times	30%	37%	23%	6%	4%	3.8
10. I am willing to recommend my peers to use this app	59%	23%	13%	1%	4%	4.3

5.3 K-12 teachers’ perceptions about using visual programming and project based learning in app creation

Table 3 summarizes K-12 teachers’ responses in the questionnaire. First, the questionnaires were designed to gain an insight on K-12 teachers’ intention to use apps in education. They thought apps can boost students’ learning motivation, and can improve learning achievement (question 1). The use of apps in education is a future trend (question 2) and they are willing to use apps to assist teaching (question 3). In the interviews, K-12 teachers indicated that they would like to created their own apps and use them in their class. Overall, K-12 teachers have high intention to use apps in their teaching.

In terms of the learning of App Inventor, the visual and block-based programming of App Inventor can reduce K-12 teachers’ learning load vis-a-vis programming syntax (question 4). They felt it’s not difficult for them to learn App Inventor (question 5). Most K-12 teachers agreed that App Inventor is suitable for novice programmers to learn apps creation (question 6). The finding is similar to previous studies showing that learners with limited programming background were positive about the learning of App Inventor [28, 30, 31].

With respect to the experiences of app project implementation, K-12 teachers were satisfied with apps created themselves (question 7). The discussion and demonstration of their peers was useful to app design (question 8). They also thought that their app project was worth doing though it cost them much time and effort (question 9). It gives them much confidence when finishing their apps project (question 10). Overall, their apps creation abilities improved after the implementation of app project (question 11). It can be concluded that K-12 teachers responded positively vis-a-vis the learning of AIA programming and the app project implementation activities.

In an open-ended question of the questionnaire, some K-12 teachers indicated that this software development process and project design was a new and eye-opening experience for them. The complete project development process, which began with app

proposal design and app revision and ended with programming implementation, enhanced their understanding of project development and gave them food for thought. Some teachers wrote:

“The app project creation experience prompted me to evaluate whether the commercial educational software used in my class aligned with education theories about courseware development, or whether these softwares were designed just to let student have a good time.”

“Now, I realize the importance of paying attention to detailed elements when designing apps. For example, I will consider whether the interface design is convenient for learners and whether the play speed of voice file will influence the learning outcome. These factors were never in my mind before.”

The above comments also reinforced the argument that project-based learning could inspire deeper thinking and better decision-making. Instead of isolated knowledge, project based learning focuses on helping people apply what they have learned in different domains such as technical knowledge, practical skills in programming, interface design and multimedia courseware design [36, 44]. In others words, project implementation experiences enabled K-12 teachers to bridge the gap between theory and practice, helped them familiarize themselves with the process of creating apps, and boosted their confidence in creating their own apps for their teaching curriculum.

Table 3. K-12 Teachers’ responses of questionnaire and results (N=7)

Question	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	Mean
1. I agree that apps can foster students’ learning motivation or can improve learning outcome	43%	57%	0%	0%	0%	4.43
2. The use of apps in education is a future trend	43%	57%	0%	0%	0%	4.43
3. I am willing to use apps to assist teaching	57%	43%	0%	0%	0%	4.57
4. The visual and block-based programming of App Inventor can reduce K-12 teachers’ learning load vis-a-vis programming syntax	43%	43%	14%	0%	0%	4.29
5. It’s not difficult for me to learn App Inventor	43%	57%	0%	0%	0%	4.43
6. APP Inventor is suitable for novice programmers when learning app creation	29%	71%	0%	0%	0%	4.29
7. I am satisfied with my app project implementation	43%	43%	14%	0%	0%	4.29
8. The discussion and demonstration of my peers was useful to me when designing apps	71%	29%	0%	0%	0%	4.71
9. I think the app project was worth doing though it cost me much time and effort	57%	43%	0%	0%	0%	4.57
10. I gained much confidence when I finished the app project	57%	43%	0%	0%	0%	4.57
11. Overall, my app creation ability improved after the implementation of the app project	43%	57%	0%	0%	0%	4.43

6 Conclusion

With the advancement of technology in recent years, using mobile apps to facilitate teaching is popular. Many teachers in this study also agreed that apps can bring positive impact for students. Though K-12 teachers know the benefits of using apps to facilitate learning, how to develop apps by programming is difficult for them. Hence, considering that K-12 teachers usually have little programming background, levelling the learning curve of programming is important. In this study, visual programming language and project-based learning were utilized to help K-12 teachers create apps. App Inventor, which is one visual programming language, was selected to teach K-12 teachers how to develop apps. Basic programming concepts were taught first; project-based learning was then applied to facilitate apps creation.

In summary, K-12 teachers were able to design usable apps through visual programming language and project-based learning. All K-12 teachers successfully created apps for the classroom subjects that they were teaching. K-12 teachers utilized many multimedia components in their apps, such as pictures, audios and videos. Many programming concepts, including variables, boolean statements, if statements, loop controls, were also used in the app projects even though more advanced program structures, such as list and procedures statements, were only used a few times. Additionally, many students responded positively to the apps developed by these K-12 teachers. Most students felt satisfied with the learning content, learning topics as well as interface design of the apps though many quizzes in the apps did not interest them. Consequently, it could be said that students thought these apps were of acceptable quality.

Furthermore, results reveal that K-12 teachers welcomed the creation of apps by visual programming and project implementation and they appreciated learning how to use App Inventor to create apps. They thought that the visual and block-based programming of App Inventor can level the learning curve via-a-vis programming syntax. These app project implementation experiences can enable K-12 teachers to bridge the gap between theory and practice. Project-based learning promotes deeper thinking and better decision-making and also provides opportunities for peer interaction and encouragement. The experiences in this study can be a reference for future teaching of App Inventor programming for K-12 teachers.

Though this research has demonstrated the aforementioned positive results, there are several limitations in the current study. Many K-12 teachers appreciated the opportunities to interact with peers and the course instructor during the project development process. K-12 teachers, however, were limited to discussing the project content design and sharing their experiences of project coding during class time due to geographical and time constraints. Furthermore, the number of K-12 teachers were few because the class was small. App project creation through teamwork may be helpful in increasing project quality and diversity because members can collaborate to create better apps and solve more complex problems. Due to the limited number of K-12 teachers in this sample study, independent work projects were adopted instead.

Based on the above considerations, future research can consider creating online learning communities and thus provide meaningful interaction without the limitations of time and space. Additionally, more K-12 teachers with different backgrounds may

be invited to participate in future studies to diversify research results. The development of relationship-building skills and communication skills during the app project creation process is also worth investigating. Finally, project creation through teamwork, as opposed to independent work project, could be considered as an alternative in future studies.

7 References

- [1] Domingo, M. G. & Garganté, A. B. (2016). Exploring the use of educational technology in primary education: Teachers' perception of mobile technology learning impacts and applications' use in the classroom. *Computers in Human Behavior*, 56, 21-28. <https://doi.org/10.1016/j.chb.2015.11.023>
- [2] Jumaat, N. F., Tasir, Z., Lah, N. H. C., & Ashari, Z. M. (2018). Students' preferences of M-learning applications in higher education: A review. *Advanced Science Letters*, 24(4), 2858-2861. <https://doi.org/10.1166/asl.2018.11078>
- [3] Churchill, D. & Wang, T. (2014). Teacher's use of iPads in higher education, *Educational Media International*, 51(3), 214-225. <https://doi.org/10.1080/09523987.2014.968444>
- [4] Aronin, S. & Floyd, K. K. (2013). Using an iPad in Inclusive Preschool Classrooms to Introduce STEM Concepts. *TEACHING Exceptional Children*, 45(4), 34-39. <https://doi.org/10.1177/004005991304500404>
- [5] Beal, C. R. & Rosenblum, L. P. (2018). Evaluation of the effectiveness of a tablet computer application (App) in helping students with visual impairments solve mathematics problems. *Journal of Visual Impairment & Blindness*, 112(1), 5-19. <https://doi.org/10.1177/0145482x1811200102>
- [6] Soto-Martín, O., & Lodeiro-Santiago, M. (2015). Apps in the practice and teaching of conservation and restoration of cultural heritage. *Procedia Computer Science*, 75, 381-389. <https://doi.org/10.1016/j.procs.2015.12.261>
- [7] Marty, P. F., Mendenhall, A., Douglas, I., Southerland, S. A., Sampson, V., Kazmer, M., Alemanne, N., Clark, A., & Schellinger, J. (2013). The iterative design of a mobile learning application to support scientific inquiry. *The Journal of Learning Design*, 6(2), 41-66. <https://doi.org/10.5204/jld.v6i2.124>
- [8] Kucirkova, N., Messer, D., Sheehy, K., & Panadero, C. (2014). Children's engagement with educational iPad apps: insights from a Spanish classroom. *Computers and Education*, 71, 175-184. <https://doi.org/10.1016/j.compedu.2013.10.003>
- [9] Denny, P., Luxton-Reilly, A., Tempero, E., & Hendrickx, J. (2011). Understanding the syntax barrier for novices. *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 208-212. <https://doi.org/10.1145/1999747.1999807>
- [10] Denny, P., Luxton-Reilly, A., & Tempero, E. (2012). All syntax errors are not equal. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, 75-80. <https://doi.org/10.1145/2325296.2325318>
- [11] Kaucic, B. & Asic, T. (2011). Improving introductory programming with Scratch? *Proceedings of the MIPRO International Conference*, 1095-1100.
- [12] Kaldi, S., Filippod, D., & Govaris, C. (2011). Project-based learning in primary schools: effects on pupils' learning and attitude. *International Journal of Primary, Elementary and Early Years Education*, 39(1), 35-47. <https://doi.org/10.1080/03004270903179538>

- [13] Ciftci, S. (2015). The effects of using project-based learning in social studies education to students' attitudes towards social studies courses. *Procedia - Social and Behavioral Sciences*, 186(13), 1019-1024. <https://doi.org/10.1016/j.sbspro.2015.04.205>
- [14] Lara, A., & Quesada, L. (2019). A project-based learning experience in a compilers course. *Proceedings of the ACM Conference on Global Computing Education*, 36-142. <https://doi.org/10.1145/3300115.3309502>
- [15] Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. *Proceedings of the ACM 11th Annual International Conference on International Computing Education Research*, 91-99. <https://doi.org/10.1145/2787622.2787712>
- [16] Saito, D., Washizaki, H. & Fukazawa, Y. (2017). Comparison of text-based and visual-based programming input methods for first-time learners. *Journal of Information Technology Education: Research*, 16, 209-226. <https://doi.org/10.28945/3775>
- [17] Rizvi, M., Humphries, T., Major, M. d Jones, & Lauzun, H. (2011). A CS0 course using Scratch. *Journal of Computing Sciences in Colleges*, 26(3), 19-27.
- [18] Rizvi M. & Humphries, T. (2012). A Scratch-based CS0 course for at-risk computer science majors. *Proceedings of the 42nd ASEE/IEEE Frontiers in Education Conference*, 1-5. <https://doi.org/10.1109/fie.2012.6462491>
- [19] Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010). Learning Computer Science Concepts with Scratch. *Proceedings 2010 International Workshop on Computing Education Research*, 69-76. <https://doi.org/10.1145/1839594.1839607>
- [20] Ward, B., Marghitu, D., Bell, T., & Lambert, L. (2010). Teaching computer science concepts in Scratch and Alice. *Journal of Computing Sciences in Colleges*, 26(2), 173-180.
- [21] Stolee, K. T., & Fristoe, T. (2011). Expressing computer science concepts through Kodu game lab. *Proceedings of the 42nd ACM technical symposium on Computer science education*, 99-104. <https://doi.org/10.1145/1953163.1953197>
- [22] Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: measuring computational thinking in middle school. *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, 215-220. <https://doi.org/10.1145/2157136.2157200>
- [23] Bell, S., Frey, T., & Vasserman, E. (2014). Spreading the word: Introducing pre-service teachers to programming in the K-12 classroom. *Proceedings of the 45th ACM technical symposium on Computer science education*, 187-192. <https://doi.org/10.1145/2538862.2538963>
- [24] Rodger, S., Dalis, M., Gadwal, C., Hayes, J., Li, P., & Liang, L. (2012). Integrating computing into middle school disciplines through projects. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 421-426. <https://doi.org/10.1145/2157136.2157262>
- [25] Ahern, T. C. (2009). Briding the Gap: cognitive scaffolding to improve computer programming for middle school teachers. *Proceedings of the 39th IEEE Frontiers in Education Conference*, M1H-1-M1H-5. <https://doi.org/10.1109/fie.2009.5350561>
- [26] Roy, K. (2012). App Inventor for Android: Report from a Summer Camp. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 283-288. <https://doi.org/10.1145/2157136.2157222>
- [27] AlHumoud, S., Al-Khalifa, H. S., Al-Razgan, M., Alfaries, A. (2014). Using App Inventor and LEGO mindstorm NXT in a summer camp to attract high school girls to computing fields. *2014 IEEE Global Engineering Education Conference*, 173-177. <https://doi.org/10.1109/educon.2014.6826086>

- [28] Perdikuri, K. (2014). Students' experiences from the use of MIT App Inventor in classroom. Proceedings of the 18th ACM Panhellenic Conference on Informatics, 1-6. <https://doi.org/10.1145/2645791.2645835>
- [29] Turbak, F., Pokress, S. C., Sherman, M. (2014). Mobile computational thinking with App Inventor 2. *Journal of Computing Sciences in Colleges*, 29(6), 15-17. <https://doi.org/10.1145/2538862.2539023>
- [30] Liu, J., Lin, C. H., & Potter, P. (2013). Going Mobile with App Inventor for Android: A one-week computing workshop for K-12 Teachers, Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, 433-438. <https://doi.org/10.1145/2445196.2445324>
- [31] Hsu, Y. -C., Rice, K., & Dawley, L. (2012). Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology*, 43(1), E1-E5. <https://doi.org/10.1111/j.1467-8535.2011.01241.x>
- [32] Indiramma, M. (2014). Project based learning - Theoretical foundation of computation course. Proceedings of 2014 International Conference on Interactive Collaborative Learning, 841-844. <https://doi.org/10.1109/icl.2014.7017882>
- [33] Sanger, P. A. & Ziyatdinova, J. (2014). Project based learning: Real world experiential projects creating the 21st century engineer. Proceedings of 2014 International Conference on Interactive Collaborative Learning, 541-544. <https://doi.org/10.1109/icl.2014.7017830>
- [34] IŞIK, Ö. & GÜCÜM, B. (2013). The effect of project-based learning approach on elementary school students' motivation toward science and technology course. *Hacettepe University Journal of Education*, 28(3), 206-213.
- [35] Tseng, K.-H., Chang, C.-C., Lou, S.-J. & Chen, W.-P. (2013). Attitudes towards science, technology, engineering and mathematics (STEM) in a project-based learning (PjBL) environment. *International Journal of Technology and Design Education*, 23(1), 87-102. <https://doi.org/10.1007/s10798-011-9160-x>
- [36] Pucher, R. & Lehner, M. (2011). Project Based Learning in Computer Science – A Review of More than 500 Projects. *Procedia - Social and Behavioral Sciences*, 29, 1561-1566. <https://doi.org/10.1016/j.sbspro.2011.11.398>
- [37] Qidwai, U. (2011). Fun to learn: project-based learning in robotics for computer engineers. *ACM Inroads*, 2(1), 42-45. <https://doi.org/10.1145/1929887.1929904>
- [38] Liao, W. & Guzide, O. (2019). Teaching artificial intelligence for undergraduate students: A project-based approach. *Journal of Computing Sciences in Colleges*, 34(3), 132-132.
- [39] Gruss, R., Farag, M., Kanan, T., English, M. C., Zhang, X., & Fox, E. A. (2015). Teaching big data through project-based learning in computational linguistics and information retrieval. *Journal of Computing Sciences in Colleges*, 31(2), 260-270.
- [40] Sillaots, M. & Fiadotau, M. (2018). Using project-based learning to teach learning game design: The example of LIFE project. 12th European Conference on Game-Based Learning, Oct. 4 - 5, 2018. Sophia Antipolis, France.
- [41] Peng, J., Sergis, S., Wang, M., & Sampson, D. (2017). Combining Smart Web-based Learning Environments with Teaching and Learning Analytics to Support Reflection on Project-based Programming Education. Proceedings of the 26th International Conference on World Wide Web Companion, 379-385. <https://doi.org/10.1145/3041021.3054158>
- [42] Jazayeri, M. (2015). Combining mastery learning with project-based learning in a first programming course: an experience report. Proceedings of the 37th International Conference on Software Engineering, 351-358. <https://doi.org/10.1109/icse.2015.163>

- [43] Gorlatova, M., Sarik, J., Kinget, P., Kymissis, I., Zussman, G. (2013). Project-based learning within a large-scale interdisciplinary research effort. Proceedings of the 18th ACM conference on Innovation and technology in computer science education, 207-212. <https://doi.org/10.1145/2462476.2465588>
- [44] McManus, J. W. & Costello, P. J. (2019). Project based learning in computer science: a student and research advisor's perspective. *Journal of Computing Sciences in Colleges*, 34(3), 38-46.

8 Author

Chiung-fang Chiu received the B.S., M.S., and Ph.D. degrees in information and computer education from National Taiwan Normal University, Taiwan. She is currently an assistant Professor in the Graduate Institute of Curriculum Instruction and Technology, National Chi Nan University, Nantou, Taiwan. Her research interests include computer science education, educational technologies, and teacher education.

Article submitted 2019-06-10. Resubmitted 2019-09-26. Final acceptance 2019-09-28. Final version published as submitted by the authors.