# *Playful Coding: Game-Based Teaching for Programming Courses*

Ting An Lin, National Cheng Kung University, Taiwan

The Asian Conference on Technology in the Classroom 2016
Official Conference Proceedings

**Abstract**
In the Information Age, the Learn-to-Code movement in the Computer Science Education starts to go viral around the world in the. And educators start to encourage people learn programming in a early age. But learning programming is difficult for kids because of the abstract concept, which make kids hard to understand and be confused. However, the traditional curriculums of the computer program are adopted the teaching demonstration which ask students to learn by the demonstration teaching step. After the curriculum, Most of students are still not able to create their own programming project or fix the problems they encounter. These results usually make student lose their motivation to keep learning programming. Using Game-Based teaching in programming course can make kids understand the real problem through a situation simulation. And kids will also know the program development process by "playing games". Furthermore, it can inspire kids to learn to imitate and build up their logic thinking that can enhance their problem-solving ability.
The objective of this research is to study how to use games in the "Program the Word" which is a project to teach kids and teenagers programming in Chiayi Dongshi to guild kids to learn the introductory programming in the Game-Based environment. By observing the interactions of instructors and young learners to understand how to create a friendly teaching and learning environment for the programming courses.

Keywords: Game-Based Teaching, Programming Education, Interactive Learning Environment

## Introduction

In today's rapidly changing world, people must continually come up with creative solutions to unexpected problems. Collaboration, critical thinking, communication, problem-solving ability, and creative design are the important skills which are outlined by the Partnership for 21st Century Learning, are vital for today's students to survive in tomorrow's job market. According to that more and more countries focus on fostering young students with these skills.

Hartmann, Nievergelt, and Reichert (2001) considered that learning programming is the core competency in the Information Age. The other survey showed that kids can get recognition skills through the program curriculum such as inferential capability, logical thinking, planning, mathematical capability, and problem-solving skills(Maheshwari, 1997). In the United States, the government strongly promotes computer education in recent years. According to ACM Model Curriculum for K-12 Computer Science, the American government advocates kids to learn computer foundational concepts and simple algorithmic thinking. In September 2014, UK government started to introduce coding to the school timetable for every child aged 5-16 years old, they believe that we should start teaching these concepts at a young age to help produce future generations foster these competencies.

But it is difficult for kid to learn coding because of the abstract concept - knowing how to design a solution to a problem, subdivide it into simpler code subcomponents, and conceive hypothetical error situations for testing and finding out mistakes(Esteves, Fonseca, Morgado, & Martins, 2008). In order to let kids easily know the abstract concept and enhance their motivation, educators and researchers try to teach kid with game-based environment. Woodhead (2002) mentioned that games will let children be in a pleasant environment to enjoy the earning. Compared to traditional teaching environment, a game-based learning environment can encourage children to participate with initiative (DeVries, 2002). The game-based learning can let child get knowledge in a relaxed environment, and further strengthen their problem-solving ability and collaborative ability with games (Bruckman, 1998; Whitebread, 1997).In the previous researches had demonstrated that learning programming in context can increase learners' motivation(Savin-Baden, 2003; Savin-Baden & Major, 2004). For example, Board games provide a natural context for introductory data structures and algorithms, as both the layout of the game and the playing style are typically discrete(turn-taking)(Bezakova, Heliotis, & Strout, 2013). It has the potential to use tangible game as educational strategy for kid to learn abstract concept of programming.

We know the benefits of learning programming from a lot of educational researches, technological engineers and SC educators. But if we only teach kid texting syntax and practice the example from textbooks, it might be just a nightmare for young students who already have tremendous pressure academically in this generation. As Papert (1993) believed that the best learning experience for most people comes when they are actively engaged in things which are meaningful to them. According to several studies, they presented the successful result of using game-based strategies for kid to learn programming in the western countries. But are those strategies also suitable for Taiwanese teachers and young students to use or learn? It is a challenge for Taiwanese educators and teachers to build up a good environment and efficient instructional strategy to inspire our kid to learn. Just like Vicki Davis, a Computer Science Teacher and IT Director at Westwood Schools, said “Coding is something that every person and student can do, they just need the door opened”.

“Program the Word” is a project to bring learn-to-code movement to young learner in Taiwan. It is a nonprofit organization and has started to teach kids programming in Chiayi Dongshi since 2013.The project manager, Dr. When Yu, Su, mentioned that “Program the Word” organization is stepping in because our current education model is not making progress fast enough. He thinks that Taiwan must prepare today’s youth now in order for them to seize the opportunities of the future and this project will be one of significant starts for Taiwanese Computer Science Education. In order to keep make a massive impact, it is necessary to bring new methods into this environment to make the learning and teaching more and more well. The purpose of this research is to help “Program the Word” keep being a friendly learning and teaching environment by practicing the Game-Based strategy and also establish a set of effective teaching strategy as a reference for future study.

## Literature Review

How to create a friendly environment for children to learn programming is an important issue in programming teaching education. There are four sections in this Section 2: programming language for kid, Game-Based learning in programming course and related strategy in teaching programming were discussed. And final section is a summary of this chapter.

### *Programming Teaching for Kid*

Learning to code doesn’t just mean kid can become a developer - it strengthens problem solving and logical thinking skills, and is useful for a range of other disciplines, careers and hobbies. The purposes of programming learning are to enhance kid’s ability of thinking logic and argument reasoning, using them to save problems around them. Papert (1980) considered that learning programming can let child enhance mind expanding and logic thinking. But most of computer learning courses are out-of-step with today’s need, they were not designed to help student develop as creative thinker (Resnick, 2007). The traditional course depends on explaining and demonstrating program examples by teacher, and students follow the steps to practice. The research pointed out that are two shortcomings in this pedagogy: First, many students can only finish the program exercises which they are learning, even students who perform well in the course often can’t face the challenge that they encounter in the future. Second, student easily loses their passion without interacting with teacher and peer, specifically the young aged student. According to the researches and surveys, there are summarized two main problems that cause student hard to learn programming:

1. Student’s lack of problem solving skill.
There are several reasons that are attributed to the students’ poor problem solving ability and an insufficient understanding of different programming control structure. The first one is that students are unable to break a problem down into sub-problems and identify the correct programming structures to these sub-problems (Lahtinen, Ala-Mutka, & Järvinen, 2005). That will make students don't really understand the real

problem statement. The other study revealed that novice spends less time to think about the real problem and busy to start doing programming. That make them easy fail and feel disappointed at final (Bishop-Clark, 1992). The second is that students don't know how to do the meaningful connection with previous experience and knowledge. Perkins and Martin (1986) consider that is attributed to students' "fragile" programming knowledge. The authors describe fragile knowledge in terms of missing knowledge, inert knowledge, misplaced knowledge and conglomerated knowledge. With "fragile" programming knowledge, novice can't appropriately use their knowledge to build up their programming structure. The final one is that novices usually don't know how to express their solution with their own logic structure. Bishop-Clark (1992) found that if novice can't organize knowledge to a structure, they usually can't answer the correct answer.

2. Student's lack of appropriate feedback from educator and peer.
Getting construction and corrective feedback are important in the programming learning. Gomes and Mendes (2007) considered that student get feedback in a instant interaction can reduce students to drop out the course. In the traditional education, students do not have less opportunity to interact with teacher and peer. Resnick (2007) proposed that teachers should support kid to interact and share idea with others when learning new knowledge.

## *Game-Based Teaching in Programming Course*

Many of us have grown up playing games, and in primary education games have a high presence in non-formal and informal segments of our learning. Unfortunately, in formal education, games are still often seen just as an unserious activity, and the potentials of game for learning often stay undiscovered. But in recent years, the researchers considered that there are specific educational domains where game-based learning concepts and approaches have a high learning value in CS education. These domains are interdisciplinary topics where skills such as critical thinking, group communication, debate and decision making are high importance. Garris et al pointed out that using game not only means having fun, it can be as a strategy to elicit the learner's motivation and skill with doing, reflecting, understanding and applying. The authors described how and when learning occurs when learners interact with a game (Figure 1) and believed that is a beneficial way for learners to learn through cycles of a game context (Garris, Ahlers, & Driskell, 2002). Programming is not a single skill but a complex cognitive activity, where a learner must simultaneously build and apply several higher order cognitive (such as abstraction), in order to solve a particular task. The Game-Based environment provides a construction tasks to make the programming more intuitive to learn(Jiau, Chen, & Ssu, 2009). There are also several success examples with using games in the programming course. In the RAPT (Reality and Programming Together) program has studied the use of game as an application to teach traditional computer science concepts in the CS course (Bayliss, 2007). In Lewis & Clark College, researchers used the board game to teach students the topic in traditional programming courses. These results indicated that there were positive and significant impacts of using games in introductory programming.

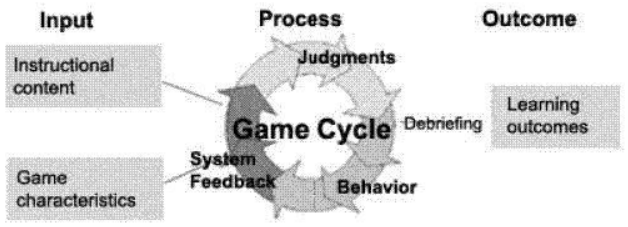

*Figure 1: Model of game-based learning by Garris et al (2002)*

***Potential Problem***

Using games in order to support the teaching of traditional outcomes in an introductory course is something that requires a good deal of planning and preparation. Bayliss (2007) mentioned that using games in an introductory course may not be very successful because here are some of the potential problems that may occur when trying to introduce games as an application area into computer science courses. The following things might be potential pitfalls that need to be avoided: “Game technology fails”- When using game with technology, we need test carefully that they can be demoed in the real situation. When these games worked, the students found them to be fun, but when they didn't work they were horrible and detracted from trying to learning how to solve problems through programming; “Not everybody likes games”- Games are easy to bring into assignments on graphical user interfaces due to their visual nature as well as in the study of algorithm through specific game algorithms such as character movement algorithms. But not every type of games would be suitable for students to learn. For example, there are a number of studies that show women are much less likely to enjoy violent (Wilson & Shrock, 2001). It is important for educator to think about the gender when choosing a game; “Games aren’t serious”- advocates that support the potential of Game-Based teaching and learning argue that games can positively impact students’ learning by providing an intrinsically motivating and engaging learning environment for students in ways that traditional school cannot (Meluso, Zheng, Spires, & Lester, 2012). But other opponents consider that game in the educational system suggests that games are for entertainment that they cannot be used to support student learning. It is important to think how to balance the “play” and ‘learning’ for assignment when using game to guild students to learn; “Not enough time”- The biggest hurdle to overcome in using games in introductory computer science courses is the amount of work that is involved for those who want to use games while meeting traditional course outcomes throughout the course. How to convey the programming concept to kid with games in a limited time is another point for educator to think about.

***Related Usage of Games in Teaching Programming***

There are many types of game that are suitable for teaching introductory programming, including computer game, board game(e.g. board, card, dice games), and games that are based on dedicated device(e.g. Lego robot, Cricket). According to Li and Watson (2011), the researchers divided the usage of games into three main categories: “Authoring-Based Approach”, “Play-Based Approach” and “Visualization-Based Approach”. “Authoring-Based Approach” uses game development as the student’s main learning activity. There are two features in this approach. First, it attempts to decrease the student external cognitive load by graphical and simple learning tool. The learner can gain an understanding of programming concept and create learner's own project further. Second, it focus upon problem solving aspects rather than syntax error. For example, Scratch uses blocked-based language for the young student to create their projects without taxing programming. Ozobot and Dash Robot are the toy robots with exciting coding adventure for kid to learn and they can also use blocked-based language to program their own project. "Play-Based Approach" is to convey knowledge by game playing. This approach used to consists of a series of tasks which are related to special concepts. By completing these missions and watching the strategy executing, learners can gain knowledge of concepts. For example, LightBot is a puzzle game where players must navigate a robot around a grid-based environment illuminating certain ground. Robot Turtle and Littlecodr are the board game to teach kid ages 4 and up about programming fundamentals (e.g if-else). Those games removed the need to type syntax and instead allowed learners to solve game tasks. They focus on underlying concepts and problem solving skills instead of syntax specifics. "Visualization-Based Approach" lies between previous approaches. The approach usually uses micro-worlds for the purpose of concept visualization and to demonstrate code execution in a visual context that there are no story, goal, or learning task to solve. For example, Turtle is the type of this approach. It not allow learners use it to create their own game. It is a tool for learner to see the effect of code execution.

Not all uses of game need to be complex, but it has to be practiced carefully in the classroom. Games are flashy and it is easy to become lost in sometimes. Student might immerse in playing and forgot to study the objectives of programming. Some lecturers think that the game are often perceived as unserious activity, and made the learning objectives would not be reached. All game assignment should directly relate to course outcome. It is important to use games in the correct context when using them as a educational tool and think the question at "what do we want that learners learn". Drake and Sung (2011) listed considerations that can guide the choice for appropriate games:

1. To design a game or to use exiting game.
2. To save class time to explain and get some emotional resonance.
3. The rule of game should be simple.
4. The game should be quick to play. e.g. 5-15minutes.
5. Too many players will be hard to test, two people are the best.
6. The hidden information, such as hands of cards not visible to the other players, should be avoided.

## Our Method

This research adopted AR (Action Research) methodology, which can be described as a family of research methodologies that involves an intervention or change on part of the researcher while research occurs. AR is a cyclical process that integrates the processes of planning, acting, observing and reflecting on the results generated from a particular project or body of work(Kemmis, McTaggart, & Retallick, 2004). This choice was made in view of the advantages of AR because it can provides the researcher with a large degree of flexibility to control changes, and help we find the critical problems and the of discovering how to improve the process

### *Data collection design*

There are going to carry out this research model (Figure 2) for a first step to understand a preliminary exploratory experience within a Game-Based learning and teaching context. In this Pre-exploratory process, the researcher was the main instrument of the observation and action. This participanting observation was an attempt to discover the meaning, dynamics and processes involved in the events. The observation will focus on the following these things:

1. How students and teacher interact with each other and with the environment:
2. the game-based classroom activities
3. the use of the types of games and
4. challenges, possibilities of teaching and learning within the game strategy.

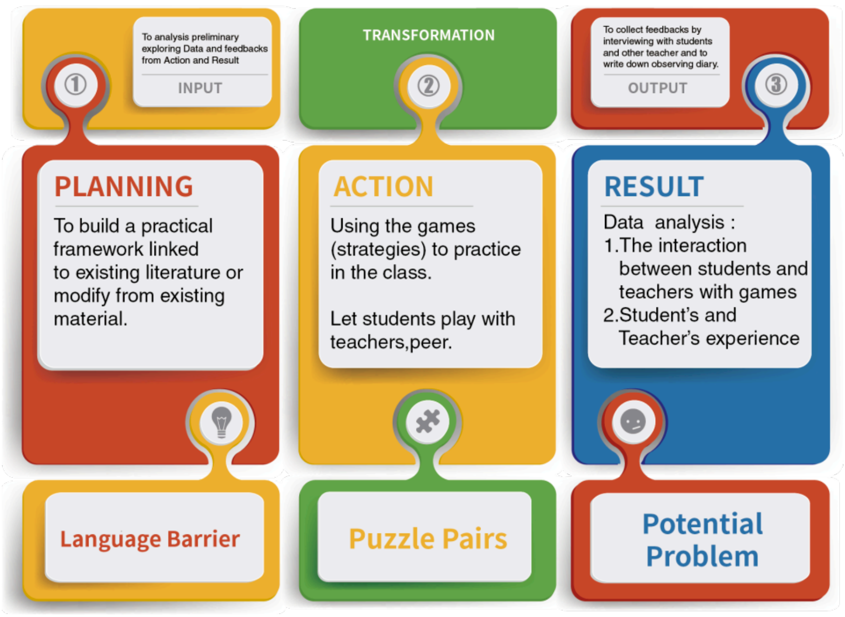
To analysis preliminary exploring Data and feedbacks from Action and Result
INPUT
①
TRANSFORMATION
②
To collect feedbacks by interviewing with students and other teacher and to write down observing diary.
OUTPUT
③
PLANNING
To build a practical framework linked to existing literature or modify from existing material.
ACTION
Using the games (strategies) to practice in the class.
Let students play with teachers,peer.
RESULT
Data analysis :
1.The interaction between students and teachers with games
2.Student's and Teacher's experience
Language Barrier
Puzzle Pairs
Potential Problem

*Figure 2 A model for collecting the preliminary exploratory data*

### *Participants*

Our participants are the teacher from NCKU and primary and junior high students in Chiayi Dongshi. “Program the World” organization cooperates with local Christian church and provide free courses for those kids and teenagers. Each a set of courses context usually be designed by 2-3 teachers and practiced in the classroom. We has different types of student in one classroom: beginners; students with some knowledge of programming (e.g.Scratch); students with relevant experience in programming projects(e.g. Arduino project). There are usually 15-20 students in a classroom. Most of them are male, the female are usually less than male.

### *Procedure*

According previous studies and data collections, we will separate the lesson into different levels with games and strategies that are suitable for teaching the major topics of programming. The students will be divided into different group by their previous experience of programming. Teachers will assist student play these games with others and let them do a simple project for testing. At the beginning, middle and end of the process, There will be some open-ended questions for students to understand the students’ views about the project they had developed and their learning in this environment. The teachers at the end of each session wrote a report with the key points of the each session. All interaction between teachers and students, which will be entirely recorded and saved at each session by video, as a tool to provide context at a later time for identifying the potential problems and next planning

## Conclusion

### *Findings and Future Work*

The following were several problem statements of teachers and student after preliminary exploratory. On the teachers’ part:

1. *To adjust to changing circumstance when in the class* –Sometimes, teacher 're going too fast will make students hard to catch up with teachers. Teachers have to adjust teaching schedule to fit students’ needs. But that will make teachers delay course progress.
2. *To keep teaching with enthusiasm* – Teachers have to pay more patients on each student because young students can’t perform well at first. How to make teachers keep teaching when they feel frustrated and exhausted is an important thing to think.
3. *Teachers’ experience transmission*– How to make the senior teacher’s teaching experience delivery to the new teacher is an important thing. That can make new teacher know the teaching environment and students quickly.

On the student’ part:

1. *The language barrier* - Most of students were afraid of the English interface. Some of them even lost the motivation when they can’t understand the English words.
2. *The fragile programming knowledge* – Most of students had already learned

the Scratch which is similar to App inventor, but when they use App inventor to create game. They couldn't bridge up previous experience to fix the problem.

3. *Having a strong dependency on teachers*- Some of students stopped thinking when they encountered the problems and just waited for teachers to come to help them.

In order to let teachers and students can overcome these problem, we will have to change some steps to improve our framework next time. The next work is to save those problems by AR research. We will keep come out different game-based teaching and to establish a system to help teacher and young learners in a friendly teaching and learning environment in programming course in a long-term.

## References

Bayliss, J. D. (2007). The effects of games in CS1-3. Paper presented at the Proceedings of the 2007 conference on game development in computer science education.

Bezakova, I., Heliotis, J. E., & Strout, S. P. (2013). Board game strategies in introductory computer science. Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education.

Bishop-Clark, C. (1992). Protocol analysis of a novice programmer. SIGCSE Bull., 24(3), 14-18. doi: 10.1145/142040.142052

Bruckman, A. (1998). Community support for constructionist learning. Computer Supported Cooperative Work (CSCW), 7(1-2), 47-86.

DeVries, R. (2002). Play in the early education curriculum: Four interpretations. Developing constructivist early childhood curriculum: Practical principles and activities, 13-34.

Drake, P., & Sung, K. (2011). Teaching introductory programming with popular board games. Paper presented at the Proceedings of the 42nd ACM technical symposium on Computer science education.

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2008). Contextualization of programming learning: A virtual environment study. Paper presented at the Frontiers in Education Conference, 2008. FIE 2008. 38th Annual.

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. Simulation & gaming, 33(4), 441-467.

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. Paper presented at the International Conference on Engineering Education–ICEE.

Hartmann, W., Nievergelt, J., & Reichert, R. (2001). Kara, finite state machines, and the case for programming as part of general education. Paper presented at the Human-Centric Computing Languages and Environments, 2001. Proceedings IEEE Symposia on.

Jiau, H. C., Chen, J. C., & Ssu, K.-F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. Education, IEEE Transactions on, 52(4), 555-562.

Kemmis, S., McTaggart, R., & Retallick, J. (2004). The action research planner.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. Paper presented at the ACM SIGCSE Bulletin.

Li, F. W., & Watson, C. (2011). Game-based concept visualization for learning programming. Paper presented at the Proceedings of the third international ACM workshop on Multimedia technologies for distance learning.

Maheshwari, P. (1997). Improving the learning environment in first-year programming: integrating lectures, tutorials, and laboratories. Journal of Computers in Mathematics and Science Teaching, 16(1), 111-131.

Meluso, A., Zheng, M., Spires, H. A., & Lester, J. (2012). Enhancing 5th graders' science content knowledge and self-efficacy through game-based learning. Computers & Education, 59(2), 497-504.

Papert, S. (1980). Teaching children thinking; teaching children to be mathematicians vs. teaching about mathematics. The computer in the school: Tutor, tool, tutee, 161-196.

Papert, S. (1993). The children's machine: Rethinking school in the age of the computer: Basic Books.

Perkins, D., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. Paper presented at the first workshop on empirical studies of programmers on Empirical studies of programmers.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. Paper presented at the Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition.

Savin-Baden, M. (2003). Facilitating problem-based learning: McGraw-Hill Education (UK).

Savin-Baden, M., & Major, C. H. (2004). Foundations of problem-based learning: McGraw-Hill Education (UK).

Whitebread, D. (1997). Developing children's problem-solving: the educational uses of adventure games. Information technology and authentic learning. London: Routledge, 13-37.

Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. Paper presented at the ACM SIGCSE Bulletin.

Woodhead, M. (2002). Work, play and learning in the lives of young children.