**CHAPTER 2**

**REVIEW OF RELATED LITERATURE**

Informatics (computer science or computing) as a science discipline, and in particular programming as an important part of that, is gaining at a lot of popularity these years in the field of education. As core subjects in a computer science (CS) major, programming subjects play an important role in a successful CS education.One of the greatest challenges faced by most students is the understanding of programming basics, especially for novices or for those who are in their first year of studies. Students develop algorithmic thinking or computational thinking at secondary and even at primary school. Using simple programming languages such as Scratch became accessible to young children (Maloney *et al.,* 2004).

Games have been used for educational purposes for decades. The popularity of games has led to the idea of using them in the learning of programming, taking advantage of the engaging features of games. In a very broad sense, two different approaches are used when using games or game-like elements in educational contexts: 1.) gamification and 2.) serious games. The term *gamification* is commonly defined as the use of game design elements in non-game contexts (Deterding*et al.,* 2011). Serious games are associated with a standalone game or platform, as well as indicated as a computer game (Djaout*et al.,* 2011). In this paper, we are going to use the general term to refer to game-based learning with main focus on teaching programming.

Game-based learning is concerned with using games not for entertainment, but for education purposes. Those who work within the field of gamification focus on identifying the context and conditions that support the integration of digital games within informal and formal learning environments. Educational scientists have pointed up several features of games that allow them to be used as learning tools. For example, games are engaging (Dickey, 2005) and motivating (Prensky, 2003). They also provide a lot of experiences (Arena and Schwartz, 2013) and an excellent feedback on performances (Shute, 2011). Finally, games support very well the learner centred education (Gee, 2005).

A contest can be seen as a part of game-based learning. Contests make the teaching of programming more attractive for students. Furthermore, programming with computer is one of the appropriate and effective ways to develop problem solving skills and computational thinking. During contests, students have the possibility to compare their abilities and learn from others. There have been many contests in programming throughout all over the world; most of them focus on algorithmic problem solving. Many teaching environments already contain game-like elements such as points, instant feedback and goals. However, there are engaging aspects in games that could be used in educational settings more widely. In well-designed games, even a failure can be a reward and triggers positive emotions (Ravaja*et al*., 2005). There is no off-the-shelf formula for designing successful games, nor is there one perfect learning environment. However, by deepening our understanding of the effects of games and game-like environments in educational settings, we can design and support more effective learning activities; and ultimately improve the learning of computer science.

The move from classical learning platforms to online contests and games can be explained as a way to ensure the best motivation as possible for their users. The online platforms can provide tools such as rankings, duels, discussion rooms, etc. to motivate their users to participate regularly. Finally, research on gamification in all its forms has proven that educational games improve the engagements of learners if the game is designed properly (Barata*et al.,* 2013; Nah *et al.,* 2014). Online platforms where students can learn programming are being developed all over the world, ranging from simple direct learning platforms to platforms of games that indirectly teach programming (Combéfis and Wautelet, 2014).

Lua is an extensible, lightweight programming language written in C. It started as an in-house project in 1993 by Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes.

It was designed from the beginning to be a software that can be integrated with the code written in C and other conventional languages. This integration brings many benefits. It does not try to do what C can already do but aims at offering what C is not good at: a good distance from the hardware, dynamic structures, no redundancies, ease of testing and debugging. For this, Lua has a safe environment, automatic memory management, and good facilities for handling strings and other kinds of data with dynamic size.

Some uses of Lua are for Game Programming, scripting in Standalone Applications, scripting in Web, extensions and add-ons for databases like MySQL Proxy and MySQL WorkBench and security systems like Intrusion Detection System.

In the last five years or so, it has been demonstrated that it is still possible to create fun and addictive and immersive game experiences in 2D. For example are hits such as Angry Birds(Rovio Entertainment 09), Peggle (PopCap Games 07), and Fruit Ninja (Halfbrick Studios 10) that are highly successful 2D games.

2D games tend to fall into a symbolic side on a scale of realistic to symbolic as such games speak to us in a move abstract level and players are actually comfortable with that, players tend to communicate in 2D in the form of letters, numbers, symbols and charts (Rasmussen 05).

In addition, 2D is simply considered by some developers as a better platform in achieving certain artistic goals. Another important point is that there is a significantly less required in terms of assets in 2D than compared to 3D counterparts. For a small development team with limited resources, this can be a big deal for them. But sometimes work is done in 2D for some 3D games. User interface, heads-up display. and menuing system are most likely rendered in 2D . In fact, it is still considered to be a 2D media unless it is developed for a 3D television and that the final output is still a 2D screen for most games produced.

Additionally, looking through someone’s perspective who takes part in teaching 3D graphics programming, it is believed that focusing on 2D graphics is a good introduction to broader graphics topics. Beyond justification, 2D graphics provide gratification and allows developers to prototype quickly their ideas and mechanics.

Science advances by learning how things worked in the past and improves them. Video games also work in the same way in which they learn about the previous video games, improves them and creates better outcome.

Today, people live in a world with advance 3D graphics rendered by powerful computers. Some 3D games seem very real that they challenge the player’s ability to differentiate which is real and which is a game. However, there are still a lot of users that is looking for something that is a bit more classic, some want a simple game without the complexity of three-dimensional graphics. It is because of these players that 2D games are still alive but there is actually a lot of 2D games that are actually 3D game engine in disguise as everything on the screen is rendered as a two-poly quad or a square built from two triangles using OpenGL or DirectX.

One of today’s most powerful game engine is the Unity game engine which is available for download in the internet. Unity, is of course, a 3D game engine but the first thing that a user must know is how to treat an engine in order to understand how to build a 2D video game.

Nowadays, the traditional way of teaching programming is not attractive to students (Shri,R. , Wai,W. , Peter,C , 2006). For example, conventional lecture and tutorial classes are not adequate and provides only one-way learning environment. Most of the students nowadays prefer to have more freedom and self-learning approach. With the new teaching and learning pedagogy, most of the university are adopting student-centered learning approach. Computer games are one of the suitable tools to fit in the student-centered learning approach. This is because computer games are not only for entertainment, but it is capable to provide a substantial self-learning environment for the student. Object-oriented programming paradigm is one of the widely adopted programming paradigm in the IT industry, however it also one of the most challenging programming paradigms to the student to learn. Most of the computer science stream students are not able to understand and visualize the concept of the object-oriented programming paradigm. Thus, it is necessary to use computer game as the teaching and learning tool to reduce the complexity of learning object-oriented programming for the student as well as provides an efficient object-oriented design learning environment for them.

This study is providing an opportunity for students to participate in the learning process via computer games with the in-game function and interaction (Pieter, W., Herre, O (2013). This is because functionalities of a computer game are important elements when designing a computer game that could easily remember and accepted by players. Besides that, computer game is an effective tool to use in teaching and learning processes because computer game provides an opportunity for player to interact. Interaction is one of the essential elements in learning process. This is because interaction in a computer game will require the participants to provide their inputs, and construct their instruction. Participants also could get back the feedback or reaction from the game. This is vital in learning process. By participating in an interaction the learners could easily understand the concept or main point that carry out form the game.

Basically, the educator’s concern is how computer game relates to the teaching environment with minimum violation of the learning objective. This is crucial for the game designers to answer this question. Therefore, effective game design is a very important process for the game designer. Game industry is very wide and full of great opportunity businesses. In average, children could play up to 10,000 hours of game in a year nowadays. Because of this huge popularity, a lot of researchers are helping the game developers to explore the possibilities of using game as learning and teaching tools. Originally, the first education computer games have been designed as drill-and practice game for the learner. Normally the format of this type of games like problem or multiple choices question about so intended topic within the context of an unrelated story. However, these kinds of education games are different from the commercial game. This is because the drill and practice game only embedded with basic single goal compare to the commercial game with plenty of goals need to achieve. The most common game genre for the education games are simulations and role person games (RPG) (Luigi, C., Alfredo, S. 2008). The simulation games are suitable to apply to the education context because participant or learner will give full control of the gameplay and react as the game hero of the game. Base on this circumstance, student will have a simulation environment for them to practice on particular skill or technique. This is because RPG games normally required student act the character of the game and solve problem or puzzle within a circumstance where it needs learner to collect necessary informative information. Thus, this study proposes a simple role-playing game as teaching and learning tool to teach object-oriented programming in classroom teaching. The main objective is to demonstrate how efficient to teach and learn object-oriented programming via computer game.

Basically all computer games have to deliver certain of interaction and communication to the person who participates in the game play. In order to use computer game as learning and teaching tools certain question should be answer such as how the interfaces teach the learner, the story board of the game, the rule of the games.

The most important concern for computer game is how to retain the student focus on the gameplay. One of the significant aspects is the user interface. This is because without a user friendly and easily-understand interface, learners will get bored and frustrated easily, and thus end up leaving the game. Besides that, the user interface represent the first impression deliver to the learner it also the first thing to teach learner from the game. A clear and simple interface will make the learner having better and more comfortable feeling. Normally, most of the games are provide tutorial to the learner. Learner can explore themselves to the tutorial to get more detail of the gameplay. Therefore game designer should take consideration when create the tutorial for the learner. At this stage, learner will explore the entire game and gets amused as well. For example the game Banjo Kazooie, in the first level of this game, the player must demonstrate the ability to complete certain important tasks. Meanwhile, there is a tutor to help if the player gets stuck, but they don’t have to listen to the instruction (Pagulayan, Keeker, Wixon, Romero & Fuller, 2003). For this study, the researcher was applying 2D tile based design for the game interface. Another major concern for education game development is the overall game storyline and flow (Amy B.Adcock. , Ginger S, Watson. , GaryR. Morrison 2008). As in learner’s point of view, progressive learning is one of the effective learning methods especially for the topic that mostly interrelated. In order to include the progressive learning to the proposed game, narrative storytelling method had been adopted to deliver the game story and flow to lead the students to learn object-oriented programming. Students are able to learn step by step by completing each level in the game world. Nevertheless, game mechanics and rules are also another important elements need to be considered when developing an education game. Overall game mechanics for the proposed game are mini puzzle game. Each mini puzzle game is representing each covered topic with learning contents and the puzzle game as assessment to test the student understanding about the covered topic. Students need to complete the mini puzzle game as the pre-requisite to next level or next learning topic.

**Related Studies**

CodeCombat is a web turn-based RPG that teaches programming to learners as young as 6. Players will have to complete the typed code, in order to progress to the next level. CodeCombat uses typed code, not drag-and-drop blocks of code. Users needs to be registered first before they can play the game. Their courses are currently available in Python and JavaScript, with Java as their next goal for another programming language.

The game Ruby Warrior is a web-based RPG for learning Ruby and artificial intelligence. The player plays as a warrior climbing a tall tower to reach the precious Ruby at the top level. On each floor, the player must write a Ruby script to instruct the warrior to battle enemies, rescue captives and reach the stairs. Players must have their Facebook accounts for their progress to be saved.

CSS Diner is a free, simple, interactive web-based minigame for learning CSS. Players will have to enter the correct styling line of code based on the images displayed on the screen in the minigame’s CSS editor. There are 32 levels that will teach users the basics of how CSS selectors work. Each level gets progressively more complex, building on what the user have learned in previous lessons.

Untrusted is a lo-fi adventure game which challenges users to escape from a series of increasingly cryptic puzzle rooms by editing the very code that is running the game in the user’s web browser. The game uses JavaScript, making the game Untrusted stand out. It’s focus is on getting every players to really think like a programmer. Untrusted evokes the feel of classic cyberpunk literature and hacker culture, while also helping players improve their programming skills

Robocode is a programming game where the goal is to code a robot battle tank to compete against other robots in a battle arena in Java or .NET. The player is the programmer of the robot, who will have no direct influence on the game. Instead, the player must write the AI of the robot telling it how to behave and react on events occurring in the battle arena. Battles are running in real-time and on-screen. In order to run Robocode, Java 6 Standard Edition (SE) or a newer version of Java must be installed in the user’s system.

**List of References**

* Calabrese, Dave (2014). Unity 2D Game Development: Packt Publishing Ltd.
* Combelis, Berenesvicus and Dagiene (2016). Learning Programming through Games and Contests: Overview, Characterisation and Discussion : Vilnius University
* Lua Scripting Language. Retrieved from <https://www.tutorialspoint.com/lua/lua_tutorial.pdf>
* Pile, John (2016). 2D Graphics Programming for Gamers: CRC Press
* Seng and Yatim (2013). Computer Game as Learning and Teaching Tool for Object Oriented Programming in Higher Education Institution: Elsevier Ltd.