Literature Review of Game-Based Learning in Computer Science Education

Cain Rademan*
cain.rademan@gmail.com
University of Cape Town
Cape Town, Western Cape, South Africa

ABSTRACT

Programming is often difficult for first year Computer Science students. Students are required to develop problem-solving skills and grapple with new and abstract concepts. Teaching introductory programming well is thus critical in producing capable, motivated Computer Scientists suited to the demands of the modern world. Game-Based Learning is an area that has received significant attention in the past two decades, with the potential to motivate students and elevate their learning. This paper presents a literature review of the use of Game-Based learning in Computer Science education. Commercial or popular programming games are also considered. It was determined that there have been a significant number of studies investigating the design of educational programming games. The use of perspective and puzzle elements in these games was surveyed and analysed. It is suggested that the study of serious games in Computer Science could benefit from more comparative study, and theoretical and psychological underpinnings. It was also found that there are opportunities for research into the impact of first person perspective in programming games.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; *Computer games*; • **Social and professional topics** → **CS1**.

KEYWORDS

game-based learning, programming education, pov game, puzzle game

1 INTRODUCTION

Computer scientists are some of the most in-demand workers in the modern world. According to the US Bureau of Labour Statistics's employment projections, computing careers are expected to see significant growth in the coming decade [27]. Over the last twelve years, there has been a sustained growth in Computer Science undergraduate enrollment [5]. There is however, a skills gap between the requirements of modern jobs and the abilities of university graduates [35]. Additionally, high dropout and failure rates [6] within computing degrees mean that even if students enroll, many will not go on to graduate. Programming is often difficult for first year students, and frustration and disenchantment in this area is one of the key factors contributing to student dropout [22]. There are a variety of factors contributing to students' struggles, such as misconceptions about programming, unfamiliarity with syntax and weak mathematical abilities [30] . Providing students with engaging introductory programming courses is therefore critical to motivate them and ensure that they have the necessary skills to complete what is typically a very demanding degree [41]. Fostering problem solving skills in students is an essential part of Computer Science education. It has been shown that exposing students to problem solving before coding is an effective teaching method [19], and that high problem solving ability is correlated with higher programming ability [37],

1.1 Fundamental Programming Concepts

The 2013 ACM Software Development Fundamentals [33] identifies the essential competencies that an undergraduate computer scientist must develop: Algorithms and Design, Fundamental Programming Concepts, Fundamental Data Structures, and Development Methods. The competency "fundamental programming concepts" is related to basic concepts of programming languages that students must become proficient in. The fundamental programming concepts are listed as follows:

- Basic syntax and semantics of a higher-level language
- Variables and primitive data types (e.g., numbers, characters, Booleans)
- Expressions and assignments
- Simple I/O including file I/O
- Conditional and iterative control structures
- Functions and parameter passing
- The concept of recursion

2 GAME-BASED LEARNING (GBL)

Game-Based learning is the use of gameplay to produce defined learning outcomes. Plass et al note that it is usually assumed that the learning game is digital, but that this need not be the case [28]. Games utilising Game-Based learning are often referred to as Serious Games. Digital educational games are a method of implementing game-based learning, defined by Heintz and Law as games played on a computer serving educational purposes [14].

2.1 GBL in Computer Science Education

. The field of Game Based Learning often finds applications within the realm of Computer Science. It can be hypothesised that this is because the problem solving ability required to code can naturally be expressed through gaming. It is also possible that the flow state one achieves when coding is similar to the flow state achieved when playing an engaging game [8]. In their article "Teaching Introductory Programming from A to Z: Twenty-Six Tips from the Trenches" [41], Zhang et al emphasise the importance of exposing first time programmers to coding in an exciting, visual way, to motivate and engage students. Games are known to be engaging [38], so are an effective medium for accomplishing this task.

There have been several studies, with different nuances and perspectives, investigating game based learning in a Computer Science context [13, 24, 25, 39, 40, 42]. A common theme in the studies is that using games to teach Computer Science increases motivation, but the results on whether these games have a significant learning effect are mixed [24]. Learning games have been studied both in a digital and non-digital context. Hosseini et al present a study on non-digital game based learning in Computer Science, with positive results [16]. The level of integration of coding into the gameplay is varied. In some games, the player engages in what is basically a "regular" video game, but the player must answer coding questions to progress between levels. López-Fernández et al present a game of this kind in which teachers set the questions, with the results showing increased motivation among students but not increased knowledge acquisition [21]. In other games, coding and problem solving are an integral part of the game mechanics. An Example of this type of game in the academic world would be RoboBUG [23], in which the gameplay consists of the player identifying bugs in code. Commercial examples include popular games like Code Combat (playable in browser), 7 Billion humans and Exapunks (which can be be found on Steam [1]).

Bertram presents a paper discussing some of the psychological theories of learning and whether they support or oppose game-based learning, as well as a review of the methodologies used in research in this area [7]. The paper calls for standardisations in methodology for evaluating digital game based learning, and infers a need for randomized controlled trials (RCTs) in testing (RCTs are trials in which subjects are randomly assigned to one of two groups: an experimental group and a control group). This call for best practices within game-based learning is echoed in Miljanovic and Bradbury's literature review on the topic [24].

Dicheva and Hodge implement an educational game designed to teach students about the "stack" data structure [10]. Educational games with the subject matter of data structures is uncommon [24], situating this paper in a unique place in the literature. The study had positive results, showing statistically significant learning gains, as well as positive reactions from students.

2.1.1 Previous Literature Reviews. There exist some literature reviews precluding this paper on the topic of serious games for Computer Science education. These are analysed in this section.

In a 2018 paper "A Review of Serious Games for Programming" Miljanovic and Bradbury reviewed a number of programming games and assessed the educational content and methods by which the games were evaluated [24]. It was found that the games focused largely on problem solving and fundamental programming concepts, while there was a lack of representation of games focusing on data structures, development methods and software design. It was noted that since many of the games developed in studies are not released publicly, it can be difficult to independently review their content and gameplay. The study also found a tendency among researchers to assess whether players liked the game, and that the results on whether the games have a statistically significant learning effect were mixed. The paper mainly analysed games developed in a research context, omitting many publicly available games developed in a popular context.

Blanco and Engstrom [4] attempted to fill this gap by providing such a review of popular commercial programming games under the assumption that since these games are popular with a wide audience they have managed to provide an engaging player experience. As in Miljanovic and Bradbury's paper, they compared the games to the ACM/IEEE Computer Science curricula [34] to determine what aspects of programming they taught. Their results resembled those of Miljanovic and Bradbury, in that there was a strong representation of games teaching fundamental programming concepts, and an under representation of games teaching algorithms, design and data structures. It was also found that games of the "puzzle" genre dominated the results, with only two of the 20 studied games belonging to a different genre. They also identified 11 game design patterns that are used in commercial programming games. They hypothesised that their identified patterns could be used as helpful heuristics when developing programming games, since the games analysed had achieved large popularity and hence were engaging and appealing to players.

In another paper precluding Miljanovic and Bradbury, Valhdick et al provide a literature review of papers implementing educational programming games [36]. In the paper they attempt to classify 40 surveyed games into one of 3 categories: LOGO-like, Adventure games and General puzzles. LOGO-like games (so named after an educational programming language designed in 1967 by Seymour Papert) were classified as games where the player controls the movements of a character through simple commands that are dragged and dropped from a toolbar. This is a slightly confusing definition, since commands in Logo are not dragged and dropped, but entered as text-based programming commands [12]. Adventure games were defined as games in which the player commands a hero to explore the world, interact with other characters and collect objects. General puzzles were defined as a host of games falling under the category of simulations, real time strategies and maze games. Of the 40 games reviewed, it was found that LOGO-like games dominated, with almost 50 percent of games falling into this category. Unfortunately this paper is lacking in discussion of results, and there are mistakes in the writing, making it a less-than-ideal resource.

2.1.2 Serious Python Games. It has been shown that using Python to teach introductory programming concepts (instead of a more complex language) is effective in facilitating learning [19]. We can hypothesise that because Python is simple and elegant, it is the ideal language to use in a serious game designed to teach introductory programming. There have been several studies providing implementations of games intended to do exactly this [17, 20, 32].

Escape from the Python's Den [17] is an educational game implemented within the popular game Minecraft. Unfortunately, the study from which it comes is extremely short and lacking in detail. The paper also crucially omits an evaluation of the success of the software.

Py-Rate Adventures is another such game [32]. It is a 2D platforming game which aims to teach basic programming concepts

using Python. It is playable by players who have no previous knowledge in Python. In this case, the use of coding in the game is not integrated into the gameplay, but is asked in the form of questions in between levels. The player must answer the questions to progress in the game. The evaluation in the study showed that players had a positive reaction to the game, and felt that it was beneficial in terms of learning. It is important to note that the study measured perceived learning of the players and so may not be reflective of the actual learning effect of the game.

2.2 Point of View (POV) in Programming Games

In video games, an important design choice is where the camera is placed. In its section on camera models, Ernest Adams' textbook Fundamentals of Game Design lists several types of perspectives used in games, such as First-Person, Third-Person, Top-Down, Side-Scroller and Isometric [3]. In this section, various examples of both commercial and academic programming games exhibiting each distinct perspective are listed. As noted by Miljanovic and Bradbury [24], many programming games developed in a research context are not made publicly available, which is problematic for researchers looking to test or independently evaluate them. This presents a problem when trying to ascertain the perspective used in different games, because it is not uncommon for the papers they are presented in do not explain the perspective, or to include confusing screenshots of the game (an example is [2]). It is much easier to document the perspective of commercially available games, because they are easily accessible and are well represented in terms of screenshots.

This is not a comprehensive listing, but it is helpful in ascertaining a general overview of the use of perspective in educational programming games. All of the commercial games are drawn from Blanco and Engstrom's literature review of mainstream programming games. Note that all of the commercial games listed can be found, either on Steam [1] or on Google Play [29] (in the case of mobile games).

Many programming games do not fall into one of Adams' perspective types [3], instead utilising either a text-based or abstract graphics approach. An additional category "Text Based/Abstract Graphics" has been included for these.

2.2.1 *Top Down.*

Commercial Games.

- 7 Billion humans
- Human resource machine

Research Games.

- Wu's Castle [11]
- BOTS [15]

2.2.2 *Text Based/Abstract Graphics.*

Commercial Games.

- Marvellous Inc.s
- Shenzhen I/O
- Robots: Create AI
- Spacechem
- TIS-100
- While True: learn()

Research Games.

- PyDiophantus [20]

2.2.3 *Side Scroller.*

Commercial Games.

- Spritebox: code Hour

2.2.4 *Isometric.*

Commercial Games.

- Else Heart.Break()
- Exapunks
- Algorithm City: Coding game for kids
- Coddy: World on algorithm Free
- Coding planets
- Gladiabots
- Lightbot: Code Hour

Research Games.

- Program Your Robot [18]

2.2.5 *Third Person.*

Research Games.

- Koelho et al's "Serious Game for Introductory Programming" [9]

2.2.6 First Person. There is a lack of implementations of first person programming games, both commercially, and in academia. Even outside the realm of game-based learning, there is a lack of study into first person perspective and its effects on player experience. Nacke et al attempted to measure physical responses to first person games using specialised equipment and correlated that with qualitative responses from participants [26]. However, the focus of the study was on investigating physiological responses as an indicator of psychological states of gameplay experience, and it did not compare the levels of flow produced in other perspectives. One might intuitively think that first-person games are more immersive, or yield a greater flow state then non-first-person games, but it seems that there has been little research in this area.

2.3 Puzzle Elements in Programming Games

In commercial programming games, puzzle games are the overwhelming majority [4]. In academic papers, this is not as pronounced, although there is still a strong representation of traditional puzzle-type games. Even so, most of these at least contain puzzle elements. Adams' Fundamentals of Game Design lists puzzle games as those in which puzzle-solving is the primary activity [3]. It has been shown that programming improves cognitive reasoning skills [31]. It has also been shown that teaching problem solving before programming in an effective strategy in introductory programming courses to improve learning in students [19]. Clearly, problem solving and programming are two closely related and mutually beneficial areas. Since puzzle games offer players a chance to hone

their problem solving abilities, and offer logical and conceptual challenges to a player, it can be hypothesised that they are the optimal serious game genre for teaching programming.

3 DISCUSSION

Since there have been such a breadth and variety of Serious Games studied in research papers, researchers looking to develop their own educational game have an array of resources and references to guide them. The literature reviews on the topic are particularly helpful [4, 24, 36], providing useful insight into the methodological, implementational and pedagogical approaches of past research. Blanco and Engstrom's [4] identification of game design patterns used in popular programming games is of particular interest. They argue that there is not enough focus in academic research on creating educational games that are engaging and experiential for players, with more work being done into learning effect.

In terms of what camera perspective is most commonly found in serious programming games, it seems that the text based/abstract graphics, and isometric approaches are favored, both commercially and in research. There is a lack of study regarding the impact of first or third person games in programming games.

There is a lack of work in the literature on comparing the impact of the use of different game design principles on educational games. It is not known for example, which game genre and perspective is the most suited to teaching a particular domain. There has also not been enough study within the domain of programming into which level of integration between coding and gameplay is best. Some programming games have tight integration, with coding systems/puzzles being the core mechanics of the game. The idea of "Core Mechanics" is a concept discussed at length in Fundamentals of Game Design [3], and is described as "The data and the algorithms that precisely define the game's central rules and internal operations." It can be hypothesised that games in which programming is the core mechanic are better for learning since the player is focused on programming as the main activity. The opposite approach, in which the game mechanics and programming are segmented from each other, may not be as effective as the player may focus on the "fun", non-programming part of the game while only comprehending the coding enough to pass through the level. This is only speculation, and further work needs to be done in this area.

4 CONCLUSIONS

The problem of creating an educational programming game to teach introductory programming has been tackled before in numerous contexts. Studies have been undertaken teaching coding using a variety of different environments/languages, including Python, Java, and Visual Programming Languages. It can be seen that the majority of games seek to teach basic programming concepts, with comparatively few teaching data structures and algorithms. The implementation of an educational game about data structures and algorithms could be a fertile area for research. It can also be seen that throughout the literature, and commercially, puzzle games are the most popular genre of educational programming game.

Based on the survey of the literature, we can say that Game-Based learning is a broad field that could benefit from more solid theoretical and psychological underpinnings, and more comparative studies. What has been solidly established in the literature is that, in certain contexts, game-based learning can be an effective tool to increase student motivation and learning. It is still an open question exactly what type of game is best-suited for each situation. This author thinks that there is work to be done in determining the impact that different game genre and other game design principles have on flow, engagement, motivation and learning.

Specifically, within the realm of Computer Science GBL, the use of a first-person perspective has not been directly studied, and might yield interesting results surrounding the relationship between player immersion and learning.

REFERENCES

[1] [n.d.]. *Steam.* Retrieved June 3, 2021 from https://store.steampowered.com/

[2] Nicoletta Adamo-Villani, Marcus Oania, and Stephen Cooper. 2012. Using a Serious Game Approach to Teach Secure Coding in Introductory Programming: Development and Initial Findings. *Journal of Educational Technology Systems* 41 (12 2012), 107–131. https://doi.org/10.2190/ET.41.2.b

[3] Ernest Adams. 2014. *Fundamentals of Game Design, Third Edition.* New Riders.

[4] Ander Areizaga Blanco and Henrik Engström. 2020. Patterns in Mainstream Programming Games. *International Journal of Serious Games* 7, 1 (Mar. 2020), 97 – 126. https://doi.org/10.17083/ijsg.v7i1.335

[5] Computing Research Association. 2019. 2019 Taulbee Survey.

[6] Theresa Beaubouef and John Mason. 2005. Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations. *SIGCSE Bull.* 37, 2 (June 2005), 103–106. https://doi.org/10.1145/1083431.1083474

[7] Lara Bertram. 2020. Digital Learning Games for Mathematics and Computer Science Education: The Need for Preregistered RCTs, Standardized Methodology, and Advanced Technology. *Frontiers in psychology* 11 (2020), 2127–2127.

[8] Chi-Cheng Chang, Chaoyun Liang, Pao-Nan Chou, and Guan-You Lin. 2017. Is game-based learning better in flow experience and various types of cognitive load than non-game-based learning? Perspective from multimedia and media richness. *Computers in human behavior* 71 (2017), 218–227.

[9] António Coelho, Enrique Kato, João Xavier, and Ricardo Gonçalves. 2011. Serious Game for Introductory Programming, Vol. 6944. 61–71. https://doi.org/10.1007/978-3-642-23834-5_6

[10] Darina Dicheva and Austin Hodge. 2018. Active Learning through Game Play in a Data Structures Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 834–839. https://doi.org/10.1145/3159450.3159605

[11] Michael Eagle and Tiffany Barnes. 2009. Experimental evaluation of an educational game for improved learning in introductory computing. In *Proceedings of the 40th ACM technical symposium on computer science education (SIGCSE '09)*. ACM, 321–325.

[12] Logo Foundation. [n.d.]. *Logo Programming.* Retrieved June 3, 2021 from https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html

[13] Stefanos Galgouranas and Stelios Xinogalos. 2018. jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming With Java. *Simulation gaming* 49, 6 (2018), 751–767.

[14] Stephanie Heintz and Effie Law. 2018. Digital Educational Games: Methodologies for Evaluating the Impact of Game Type. *ACM transactions on computer-human interaction* 25, 2 (2018), 1–47.

[15] Andrew Hicks. 2010. Towards social gaming methods for improving game-based computer science education. In *Proceedings of the Fifth International Conference on the foundations of digital games (FDG '10)*. ACM, 259–261.

[16] Hadi Hosseini, Maxwell Hartt, and Mehrnaz Mostafapour. 2019. Learning IS Child's Play: Game-Based Learning in Computer Science Education. *ACM transactions on computing education* 19, 3 (2019), 1–18.

[17] Ilish Kane, Chuy-Thuy Pham, Adam Lewis, and Vanessa Miller. 2019. Escape from the Python's Den: An Educational Game for Teaching Programming to Younger Students. In *Proceedings of the 2019 ACM Southeast Conference (ACM SE '19)*. ACM, 279–280.

[18] Cagin Kazimoglu. 2020. Enhancing Confidence in Using Computational Thinking Skills via Playing a Serious Game: A Case Study to Increase Motivation in Learning Computer Programming. *IEEE Access* 8 (2020), 221831–221851.

https://doi.org/10.1109/ACCESS.2020.3043278

[19] Theodora Koulouri, Stanislao Lauria, and Robert D Macredie. 2015. Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM transactions on computing education* 14, 4 (2015), 1–28.

[20] Dimitra Kouprítzioti and Stelios Xinogalos. 2020. PyDiophantus maze game: Play it to learn mathematics or implement it to learn game programming in Python. *Education and information technologies* 25, 4 (2020), 2747–2764.

[21] Daniel López-Fernández, Aldo Gordillo, Pedro P. Alarcón, and Edmundo Tovar. 2021. Comparing Traditional Teaching and Game-Based Learning Using Teacher-Authored Games on Computer Science Education. *IEEE Transactions on Education* (2021), 1–7. https://doi.org/10.1109/TE.2021.3057849

[22] A McGettrick. 2005. Grand Challenges in Computing: Education–A Summary. *Computer journal* 48, 1 (2005), 42–48.

[23] Michael Miljanovic and Jeremy Bradbury. 2017. RoboBUG: A Serious Game for Learning Debugging Techniques. 93–100. https://doi.org/10.1145/3105726.3106173

[24] Michael Miljanovic and Jeremy Bradbury. 2018. A Review of Serious Games for Programming.

[25] Mathieu Muratet, Patrice Torguet, Jean-Pierre Jessel, and Fabienne Viallet. 2009. Towards a serious game to help students learn computer programming. *International journal of computer games technology* 2009, 1 (2009), 1–12.

[26] Lennart E Nacke and Craig A Lindley. 2010. Affective Ludology, Flow and Immersion in a First- Person Shooter: Measurement of Player Experience. (2010).

[27] US Bureau of Labour Statistics. 2019. Employment Projections 2019-2020. https://www.bls.gov/emp/tables/emp-by-detailed-occupation.htm

[28] Jan L Plass, Bruce D Homer, and Charles K Kinzer. 2015. Foundations of Game-Based Learning. *Educational psychologist* 50, 4 (2015), 258–283.

[29] Google Play. [n.d.]. *Steam.* Retrieved June 3, 2021 from https://play.google.com/store

[30] Yizhou Qian and James Lehman. 2017. Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* 18, 1, Article 1 (Oct. 2017), 24 pages. https://doi.org/10.1145/3077618

[31] Ronny Scherer, Fazilat Siddiq, and Bárbara Sánchez Viveros. 2019. The Cognitive Benefits of Learning Computer Programming: A Meta-Analysis of Transfer Effects. *Journal of educational psychology* 111, 5 (2019), 764–792.

[32] Grigorios Sideris and Stelios Xinogalos. 2019. PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming With Python. *Simulation gaming* 50, 6 (2019), 754–770.

[33] The Association for Information Systems (AIS) The Joint Task Force: Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.

[34] The Association for Information Systems (AIS) The Joint Task Force: Association for Computing Machinery (ACM) and IEEE Computer Society. 2020. Computing Curricula 2005: The Overview Report.

[35] The Association for Information Systems (AIS) The Joint Task Force: Association for Computing Machinery (ACM) and IEEE Computer Society. 2020. Computing Curricula 2020.

[36] Adilson Vahldick, Antonio Jose Mendes, and Maria Jose Marcelino. 2014. A review of games designed to improve introductory computer programming competencies. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE, 1–7.

[37] Ashok Kumar Veerasamy, Daryl D'Souza, Rolf Lindén, and Mikko-Jussi Laakso. 2019. Relationship between perceived problem-solving skills and academic performance of novice learners in introductory programming courses. *Journal of computer assisted learning* 35, 2 (2019), 246–255.

[38] Eric N Wiebe, Allison Lamb, Megan Hardy, and David Sharek. 2014. Measuring engagement in video game-based environments: Investigation of the User Engagement Scale. *Computers in human behavior* 32 (2014), 123–132.

[39] Stelios Xinogalos. 2018. Programming Serious Games as a Master Course: Feasible or Not? *Simulation gaming* 49, 1 (2018), 8–26.

[40] Mirac Yallihep and Birgul Kutlu. 2020. Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and information technologies* 25, 2 (2020), 1237–1254.

[41] Xihui Zhang, John D Crabtree, Mark G Terwilliger, and Janet T Jenkins. 2020. Teaching Tip: Teaching Introductory Programming from A to Z: Twenty-Six Tips from the Trenches. *Journal of information systems education* 31, 2 (2020), 106–.

[42] Dan Zhao, Cristina Hava Muntean, Adriana E Chis, and Gabriel-Miro Muntean. 2021. Learner Attitude, Educational Background, and Gender Influence on Knowledge Gain in a Serious Games-Enhanced Programming Course. *IEEE transactions on education* (2021), 1–9.