EasyChair Preprint

№ 10906

Gizmos and Language: a Top-down Role-Playing Game for Basics of Programming Using Unity

Romeo John Perez and Denise Lou Punzalan

September 15, 2023

Gizmos and Language: A Top-Down Role-Playing Game for Basics of Programming Using Unity

ROMEO JOHN PEREZ, Developer/Author, College of Computer Studies - Gordon College
DENISE LOU PUNZALAN, Adviser/Co-Author, College of Computer Studies - Gordon College

This study presents the initial evaluation of Gizmos and Language, a 2D role-playing game, in teaching programming principles using block-based programming. The game employs a top-down perspective, a 2D graphics view that can be seen from either a side-scrolling or top-down angle. Block-based programming is a graphical language that uses graphical blocks to represent code, making it intuitive for learning programming concepts and facilitating the transition to text-based languages. The Agile methodology was utilized in the game's development, allowing for flexible and iterative development. The study targeted 112 first-year students enrolled in the College of Computer Studies department at Gordon College. The Game Experience Model (GEM) was adapted and employed to assess the game experience in terms of immersion, usability, visuals, performance, and novelty. Surveys and gameplay were used as research instruments, aligning with GEM. Descriptive statistics were employed as the research design to evaluate the data collection outcome. Based on the results of the GEM analysis from the 112 first-year students, the game provided a visually appealing environment that helped them become more familiar with programming concepts. Future testing with external participants may validate the game's effectiveness in various settings. However, the study recommends that games should not be the sole mode of instruction but can be valuable supplementary tools for teaching programming.

General Terms: Block Coding, Game Development, RPG, 2D Top-Down, Digital Games, Code Blocks

ACM Reference format:

Ben Trovato, G.K.M. Tobin, Lars Thørväld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Paper in word Format. *ACM Trans. Graph.,* 9, 4, Article 39 (March 2010), 4 pages.
DOI: 10.1145/1234

1 INTRODUCTION

Immersive learning through games has emerged as a promising approach to teaching programming and block-based coding. Block-based programming (programming by dragging and dropping blocks) has been widely used to ignite students' interest in coding and foster their computational thinking skills [1]. A 2D top-down RPG (role playing game) is one such game genre that has gained popularity for a lot of games developed in this category.[2] The topic of teaching computational thinking has gained significant attention which emphasizes the significance of individuals having a fundamental understanding of computer science concepts such as abstraction, automation, and solving complex problems, in order to be competitive in the job market [3]. Visual programming and game-based learning can enhance computational thinking and problem-solving skills in students and may be used to introduce them to programming [4]. However, most existing games have focused on the theoretical knowledge of computational thinking, ignoring the development of computational thinking skills [5]. Self-pacing, learner-controlled feedback, and real-time feedback are all principles of adult learning theory that can be incorporated into game-based learning to keep learners motivated and engaged.

Game developers and educators collaborate to create programming education games, but may face challenges due to developers' limited understanding of education and educators' misaligned expectations for game scope and usefulness. Using established software processes and educational best practices can enhance game development by improving teamwork, communication, quality, and reducing costs [3][4]. By integrating block coding into a 2D top-down RPG, first-year students of the College of Computer Studies at Gordon College have the unique opportunity to learn programming concepts in an interactive and immersive environment. This platform not only ignites their interest in coding but also cultivates their problem-solving and analytical skills, which are highly sought-after in the current job market. This study presents the initial evaluation of the game's effectiveness in teaching programming concepts and highlights the potential for further development and improvement. Additionally, there is a possibility for future testing that involves people outside the school to validate the game's effectiveness in different settings. The results of this study provide insights into the value of integrating interactive digital games into education and enhancing the learning experience for students.

2 REVIEW OF RELATED LITERATURE

2.1 Evaluating the effectiveness of block-based and hybrid approaches for transitioning to text-based programming

A hybrid-based environment has improved the students learning curve when migrating to the text based environment. Students using the hybrid based environment are also able to effectively debug the code from seeded errors, outperforming students using the block-based environment by 28.6% on average.[6]

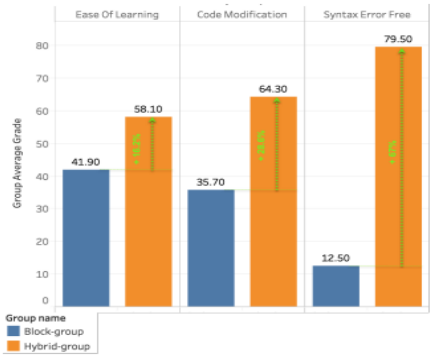

Fig. 1. Performance of Hybrid and Block Learning Improvement

2.2 Enhancing programming skills through problem-based game projects in engineering education using Scratch

The progress of the students who have taken the Introduction to Programming course in enriched or classical form in the fourth year Senior-project course are evaluated. The results show that by slightly improving the course curriculum through real-life game development projects in the Scratch environment, students' performance on the graduation projects improved significantly. An independent sample t-test was conducted to evaluate the hypothesis that students in the enriched Introduction to Programming course (experimental group) perform better than those in the classical Introduction to Programming course (control group). In this comparison, the dependent variable was the letter grade of the students determined after the completion of the course. The test was significant, $t(322) = -2.82$, $p = 0.005$, indicating that on average the students in the enriched Introduction [7]

2.3 Investigating the impact of gamification on language subject academic performance of junior high school students

Gamification has increased the academic performance of students enrolled in language subjects specifically in the Filipino class. It is proven that gamification can set a good mood and motivation for learners when used. The students in this study also exhibit favorable sentiments toward the usage of games in the classroom. It aided them in developing critical abilities, encouraging active engagement, and identifying their subject's strengths and limitations. It was also discovered that several issues need to be addressed, including a lack of good internet connections and the types of devices that students use. It is recommended by the researchers to study the effect of leader boards and badges on students' motivation and engagement.[8]

3 CONCEPTUAL AND THEORETICAL FRAMEWORK

This study develops a two-dimensional role-playing game that provides an immersive and educational experience for code-based programming. It aims to enhance students' learning by incorporating game design elements and a programming curriculum.

3.1 Game Development

Game development involves creating interactive entertainment software, and it can also be used for educational purposes by incorporating educational content. Educational games can make learning more engaging and interactive, providing an immersive experience for students. Games can also offer immediate feedback to students, allowing them to track their progress and identify areas where they need improvement. As a result, game development can be an effective tool for educators who want to enhance their students' learning experience.

Game development involves defining a vision and designing the game using flowcharts, wireframes, and prototypes, with feedback from stakeholders and players shaping the design. Experimentation with different technologies, styles, and assets helps achieve the desired design goals. Game development involves defining a vision and designing the game using flowcharts, wireframes, and prototypes, with feedback from stakeholders and players shaping the design. Experimentation with different technologies, styles, and assets helps achieve the desired design goals. The developed game, Gizmos and Language, is subjected to 3 different parts: Block Mode, for the traditional approach to block-based programming; RPG, wherein the player explores around the game to learn; and Sandbox Mode, wherein all of the blocks available to the game can be found.

3.1 Asset Making (Sounds, Graphics, Engine)

Asset making is a crucial part of game development, as it involves creating the visual and audio elements of the game. Art assets can be created using a variety of editing software, but for a pixel art world, Aseprite is a popular tool for creating and editing pixel art. Aseprite allows artists to create animations, characters, and backgrounds pixel by pixel, providing precise control over the final look of the game. On the other hand, music making involves creating and editing audio tracks and sound effects that will be used in the game. This process can be done using digital audio workstations (DAWs), such as FL Studio or Logic Pro, which allow composers to create and manipulate audio tracks, to create the desired soundscape for the game. The audio elements of the game are just as important as the visual elements in creating an immersive and engaging experience for the players. Development on dragging and dropping for the environment, the Blocks Engine 2 was utilized. It is a complete framework for creating games with a runtime visual block coding interface, enabling the players to code visually in the game using blocks. It is the primary component of the project wherein it is the base for establishing more blocks using its framework.

3.2 Game Design Document

The Game Design Document (GDD) is the blueprint from which a computer or video game is to be built. As such, every single detail necessary to build the game must be addressed in the document (or support documents). If it's not in the document, then it probably won't be in the game[9].

3.3 Game Experience Model (GEM)

The Game Experience Model or GEM is intended to be a tool for both analyzing and designing games. It should help one to feel confident that no part of the game experience is neglected in the design process. The GEM attempts to be the guide to the anatomy of the gameplay experience, helping the designers to pay attention to the equivalent ergonomics –making sure that the game is properly designed to suit the intended audience.[10] It is modified to align to the study and it consists of five primary components, namely Immersion, Usability, Visuals, Performance, and Novelty, which are essential in assessing the game's overall user experience.

Genre: 2D Top-Down RPG
Theme: Programming and code blocks
Platform: PC

Gameplay: A top down game with drag and drop block coding to solve quests and progress through levels. Players unlock new levels by mastering programming concepts.
Key Features: Explorational Interactive code blocks, sandbox mode to experiment with blocks, increasing difficulty and complexity of puzzles.

3.4 Vertical Slicing

Vertical slicing is the complete yet portioned implementation of a functional, valuable and demonstrable feature through all technology layers.[11] It showcases key gameplay mechanics, functionality, and visuals, allowing the developer to evaluate the direction and style of the project. The initial prototype is usually built using game development tools such as programming languages and game engines, like C# and Unity. It is an essential step in the game development process as it helps developers identify potential problems and make informed decisions about the project. A conceptual model of the research, shown in Figure 2, is based on relevant concepts, theories, findings, studies, and insights gained from them.

3.4 Conceptual Model

A conceptual model of the research, shown in Figure 2, is based on relevant concepts, theories, findings, studies, and insights gained from them.

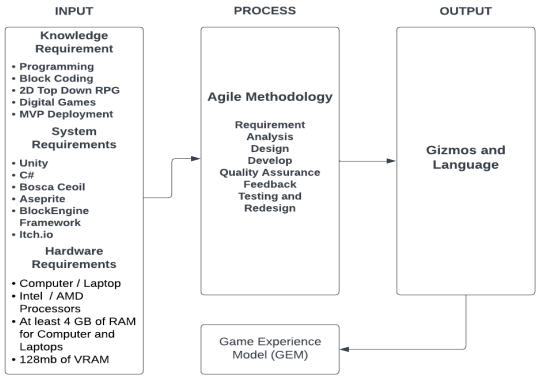

Fig. 2. Paradigm of the Study

4 METHODOLOGY

In this chapter, different steps needed to achieve the goal are outlined. These steps comprise of the research design, location of the research, tools used, confirmation of the tools' effectiveness, managing and gathering data, and the statistical analysis of the data.

4.1 Research Design

Descriptive statistics can be defined as an analytical approach to describing the data at a disposal. This approach aims to summarize and extract the data information and characteristics into certain numbers, graphs or tables.[12]

4.2 Respondents / Locale of the Study

This study targets the 112 first-year students enrolled in the College of Computer Studies department at Gordon College as the respondents. The data collection involved distributing surveys to each of the 112 respondents and was carried out at the school mentioned above.

4.3 Flowchart and Process Modeling

The focus of this section is to present the flow of the game and showcase the processes of each scene within the game

A flowchart is a diagram that shows every step, choice and outcome involved in a task. Each step is represented by a symbol, and connecting lines show the step-by-step progression through the task

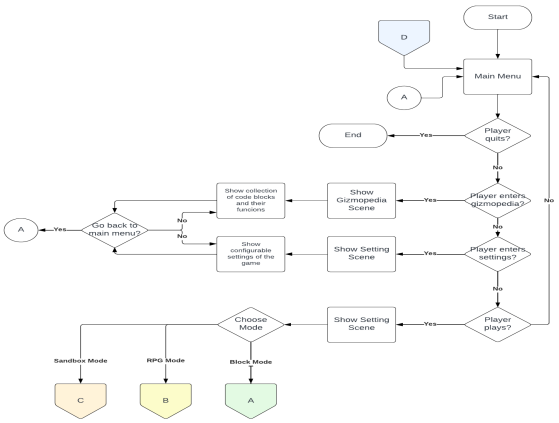

Fig. 3. System Flowchart\

Off page shapes navigation are as follows:

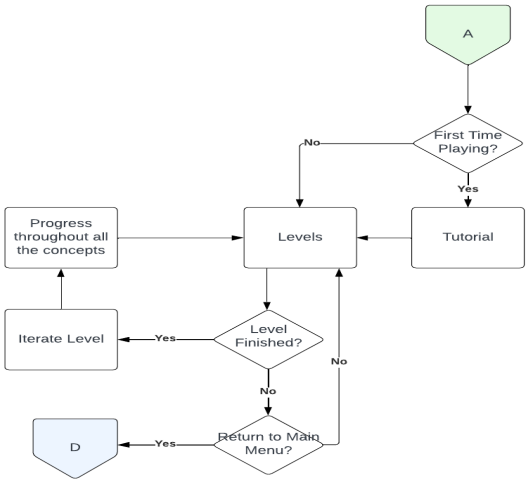

Fig. 4. Block Mode Flowchart

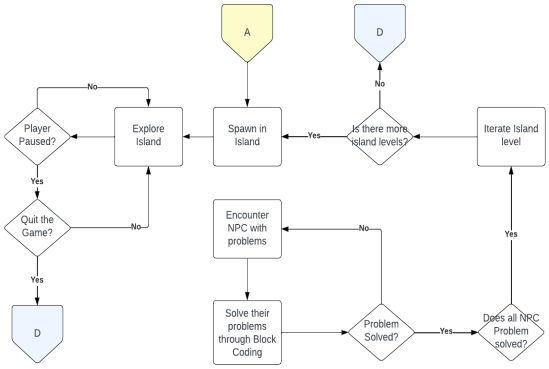

Fig. 5. RPG Mode Flowchart

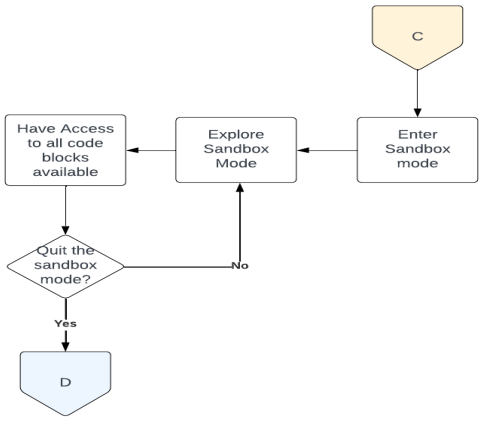

Fig. 6. Sandbox Mode Flowchart

4.4 Functional Decomposition Process

A functional decomposition diagram shows a top-down representation of a function or process in levels. Figure 4 illustrates functions of the game and then decomposes them into subfunctions.

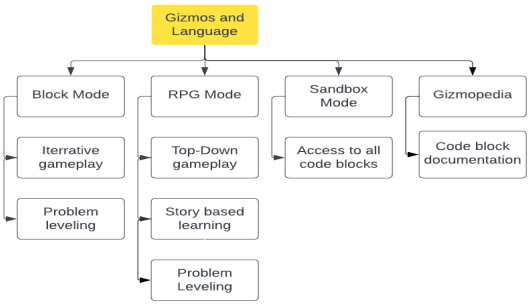

Fig. 7. Functional Decomposition Process

4.4 Data Flow Diagram

A data flow diagram is a visual representation of a system's boundaries, data inputs, and outputs that can be used to summarize and organize detailed details. The context diagram of the study is shown in figure 8.

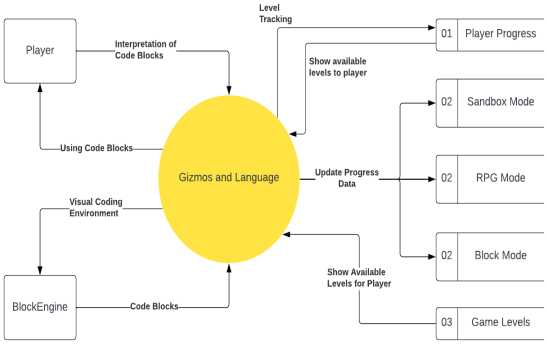

Fig. 8. Data Flow Diagram

4.5 Instrument

The study will use a survey tool that follows the Game Experience Model or GEM.. The survey will consist of a checklist and a Likert scale questionnaire to collect responses. The Game Experience Model is intended to be a tool for both analyzing and designing games. It should help one to feel confident that no part of the game experience is neglected in the design process. The GEM attempts to be the guide to the anatomy of the gameplay experience, helping the designers to pay attention to the equivalent ergonomics –making sure that the game is properly designed to suit the intended audience.[14] It is modified to align to the study and it consists of five primary components, namely Immersion, Usability, Visuals, Performance, and Novelty, which are essential in assessing the game's overall user experience. The Immersion component gauges game engagement, Usability evaluates user-friendliness, Visuals determines the appeal, Performance assesses technical quality, and Novelty gauges originality. These components guide the data gathering and validating stakeholders evaluating the user experience

4.6 Agile Methodology

The study used an agile approach, a modern software development methodology. Agile methods focus on customer satisfaction by delivering small, useful updates frequently and consistently. This makes improvements easy to implement. The agile software development lifecycle, shown in Figure 9, emphasizes customer satisfaction above all else. Agile development delivers rapid, iterative updates to produce a

high-quality product that meets customer needs.[13]

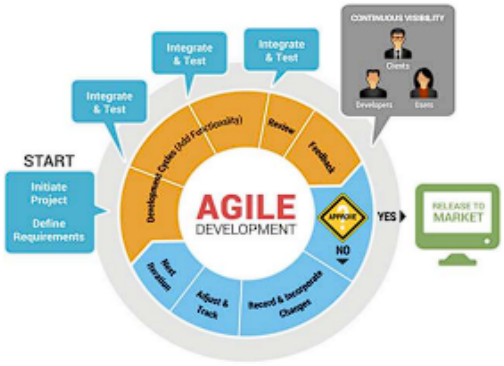

Fig. 9. Data Flow Diagram

4.5 Design / Gameplay

This section presents the game's architecture, in game functions, designs, and game flow screenshots.

Fig. 10. Main Menu of the Game

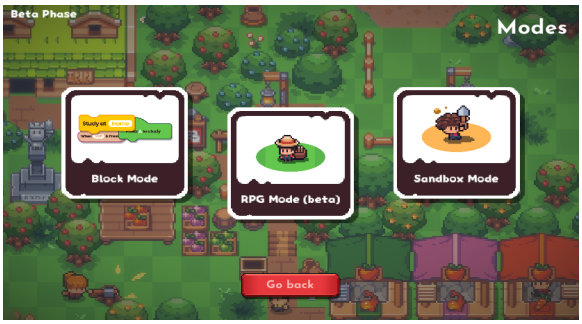

Fig. 11. Mode Selection

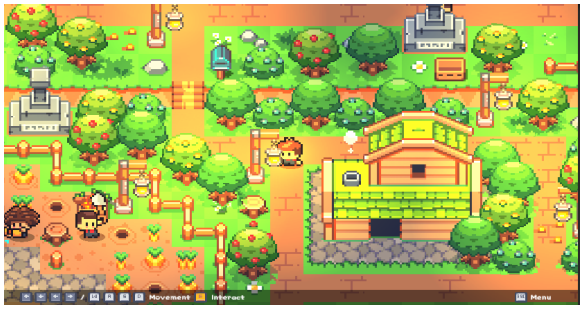

Fig. 12. RPG Mode (beta)

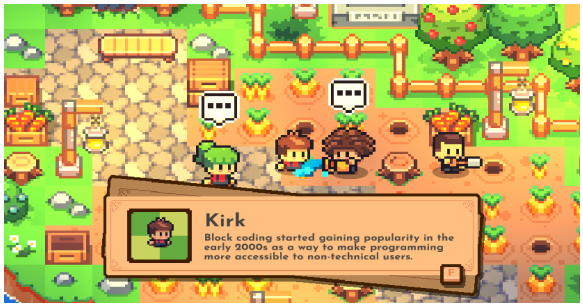

Fig. 13. RPG Mode Dialogue (beta)

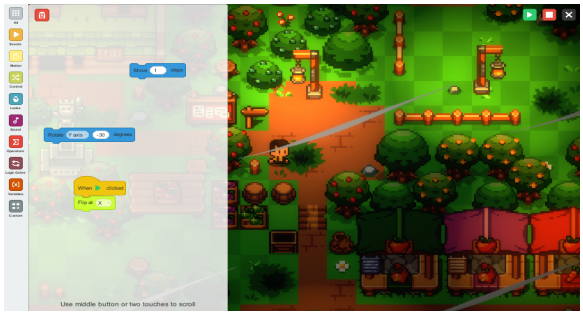

Fig. 14. Block Mode

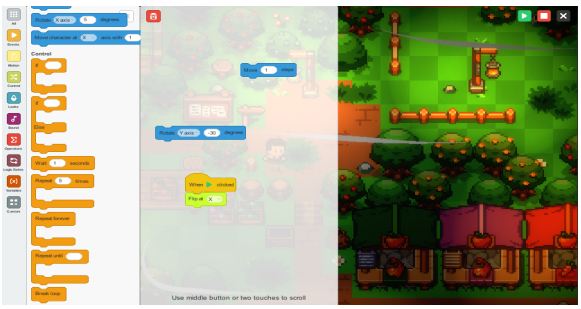

Fig. 15. Block Mode / Sandbox Mode. Blocks Panel

4.6 Data Collection

The initial evaluation of the study, which targeted primarily first-year students from the College of Computer Studies at Gordon College. A total of 112 student respondents participated in the study and rated (see Table 2) the respective criterion question based on their experience of the game's visuals, performance, effectiveness, and immersion using a Likert scale (see Table 1). Further testing could be conducted not only with students but also with other individuals outside the school to broaden the scope and generalizability of the study's findings..

Table. 1. Likert Scaling Method / Evaluation

Range	Descriptive Rating
4.50 - 5.00	Excellent
3.50 - 4.49	Very Good
2.50 - 3.49	Good
1.49 - 2.49	Fair
1.00 - 1.49	Poor

The table provided shows a scale method with different ranges of scores and their corresponding interpretations in line with the Game Experience Model. The scale method is a way of assessing performance or achievement based on a numerical score. The score ranges from 1.00 to 5.00, with 5.00 being the highest possible score..

Table. 2. College of Computer Studies Student GEM Evaluation Result

Modified Game Experience Model Evaluation		
Criterion	**Average Rating**	**Descriptive Rating**
Immersion	3.92	Very Good
Usability	3.91	Very Good
Visuals	4.24	Very Good
Performance	3.76	Very Good
Novelty	3.86	Very Good

As it was revealed from the student respondent's evaluation, the indicator "1. Immersion" was evaluated with a central tendency of 3.92, "2. Usability" rated 3.91, "3. Visual" rated 4.24, "4. Performance"rated 3.76, and "5. Novelty" rated 3.86, all of the criteria was evaluated "Very Good" as a descriptive rating using the aforementioned GEM.

5 CONCLUSION

In conclusion, the development of Gizmos and Language followed the Game Experience Model, which prioritized an engaging and educational experience. The game was designed to meet the needs of stakeholders and target audience, and the results of the study provide insights for further improvement.

1. The results showed that while the game was moderately immersive, there is room for improvement in terms of storyline and gameplay. The usability criterion scored well, indicating that navigation and accessibility were easy, but there is still room for improvement. The graphics were visually appealing and ranked first among the criteria, while technical performance was moderate but could be optimized. The game had some unique features, but more innovation could enhance the overall experience. These findings suggest that the results obtained from measurements, calculations, or specifications can be relied upon to a significant degree.
2. The success of the game depends not only on the experience but also on enjoyment and learning. Improving usability, adding more levels and mechanics, and expanding the development team could enhance engagement and retention. Given that game development is complex and time-consuming, allocating more resources could help realize the game's potential as an educational tool.
3. Gizmos and Language offer an innovative approach to familiarizing and teaching programming, fostering intrinsic learning and knowledge retention. While it may not be a one-size-fits-all solution, it provides an enjoyable and engaging way to learn. Further development and exploration of digital games for education could help struggling learners and provide more programming opportunities.

In summary, the study demonstrated that Gizmos and Language achieved its purpose of providing a fun and immersive platform for learning programming principles. Additionally, It is significant to emphasize that this study serves as an initial evaluation of the game's effectiveness as an educational tool. Further testing and evaluation not only among first-year students but also among other individuals outside the school could provide valuable insights into the game's potential and limitations. This could also help identify areas for improvement and expansion to cater to a broader audience and offer more diverse learning opportunities. However, it is important to note that while games can benefit some learners, they may not be effective for all, especially those who are struggling or less interested in programming. Games should be used as adjuncts and aids, not as stand-alone instruction. [15]

REFERENCES

[1] Tsung, S., Wei, H., Li, H., Wang, Y., Xia, M., & Qu, H. (2022, June). BlockLens: Visual Analytics of Student Coding Behaviors in Block-Based Programming Environments. In Proceedings of the Ninth ACM Conference on Learning@ Scale (pp. 299-303).

[2] Horsfall, M., & Oikonomou, A. (2011, July). A study of how different game play aspects can affect the popularity of role-playing video games. In 2011 16th International Conference on Computer Games (CGAMES) (pp. 63-69). IEEE.

[3] Forquesato, L. E., & Borin, J. F. (2018, July). Kids Block Coding Game: A game to introduce programming to kids. In Anais do XXVI Workshop sobre Educação em Computação. SBC.

[4] Kanika, Chakraverty, S., & Chakraborty, P. (2020). Tools and techniques for teaching computer programming: A review. Journal of Educational Technology Systems, 49(2), 170-198.

[5] Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G. J., Wang, M., ... & Delev, D. (2021). From gaming to computational thinking: An adaptive educational computer game-based learning approach. Journal of Educational Computing Research, 59(3), 383-409.

[6] Alrubaye, H., Ludi, S., & Mkaouer, M. W. (2019). Comparison of block-based and hybrid-based environments in transferring programming skills to text-based environments. arXiv preprint arXiv:1906.03060.

[7] Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. Computers & Education, 120, 64-74.
[8] Caballero, L. S., Ferrer, M. M., & Tiria, R. A. (2022). The Role of Gamification in the Academic Performance of Junior High School Students in Language Subject. Int. J. Sci. Res. in Multidisciplinary Studies Vol, 8(3)
[9] Baldwin, M. (2005). Game design document outline. Baldwin Consultant, 24.
[10] Suovuo, T. B., Skult, N., Joelsson, T. N., Skult, P., Ravyse, W., & Smed, J. (2020). The game experience model (GEM). Game User Experience and Player-Centered Design, 183-205.
[11] I. M. Ratner and J. Harvey, "Vertical Slicing: Smaller is Better," 2011 Agile Conference, Salt Lake City, UT, USA, 2011, pp. 240-245, doi: 10.1109/AGILE.2011.46.
[12] Chang, Y. T. (2022). Descriptive Statistics, Missing Data and Testing Assumptions. Introducing, Designing and Conducting Research for Paramedics, 119.
[13] Fitzgerald, K., Browne, L. M., & Butler, R. F. (2019). Using the Agile software development lifecycle to develop a standalone application for generating colour magnitude diagrams. Astronomy and Computing, 28, 100283.
[14] Russ, J. W., & Nunez, J. J. (2017). Unreal Engine 4 RPG.
[15] Hays, R. T. (2005). The effectiveness of instructional games: A literature review and discussion.