

A Game-Based Approach to Teach Basic Python Programming

Graceline Jasmine S¹, Sujay Doshi², Alan K. Alex³, Yadukrishnan U⁴, Febin Daya J.L⁵

^{1,2,3,4} School of Computer Science and Engineering, Vellore Institute of Technology - Chennai Campus

⁵ Electric Vehicles, Incubation Testing and Research Center, Vellore Institute of Technology - Chennai Campus

¹graceline.jasmine@vit.ac.in

²sujay.doshi2020@vitstudent.ac.in

³alanalex.k2020@vitstudent.ac.in

⁴yadukrishnan.u2020@vitstudent.ac.in

⁵febindaya.jl@vit.ac.in

Abstract : This paper presents an innovative and interactive game-oriented educational platform employing the Unity game engine and C# scripting. The aim of this interactive 2D platform is to impart fundamental programming skills, specifically focusing on the Python language. This platform comprises of four challenging levels, each increasing in difficulty, providing a captivating and dynamic learning experience. The distinctive feature of the proposed platform is the usage of string matching with regular expressions to instantly evaluate the accuracy of the code entered by users, enabling them to develop their coding skills in a practical and interactive manner. The visually appealing graphics of this platform offer the user the ability to see the real-time effects of their code, rendering it more intuitive and user-friendly with real-time feedback. This platform is designed to be accessible to a diverse audience, including children and adults, making it an inclusive learning tool for people of all ages and backgrounds. The proposed platform is employed to instruct Python programming to a group of undergraduate students and the achievement of course outcomes (CO) is evaluated and contrasted with a traditional learning

approach. The comparative results are presented. The levels of attainment of COs experienced a notable improvement, demonstrating the effectiveness of the proposed learning platform, making it a valuable educational tool for coding novices.

Keywords: Game development; python programming; serious games

1. Introduction

The potential of games as educational tools is enormous, as they offer a captivating and interactive means for learners to actively participate in the learning process. Educational games are specifically developed to create an enjoyable and interactive learning experience, keeping students motivated and engaged while reinforcing significant concepts and improving retention. By providing instant feedback and a sense of achievement, games can also contribute to building learners' confidence and self-esteem. Additionally, games can be customized to cater to the diverse requirements of individual learners, providing a personalized and adaptable learning experience. These games can complement conventional teaching methods and provide a more efficient and engaging educational experience.

Minecraft Edu is an excellent example of gaming being used as an educational tool. It is a modified version of the popular game Minecraft, tailored for classroom use, and has been successfully employed to teach various skills such as architecture, geography,

Febin Daya J. L

Electric Vehicles, Incubation Testing and Research Center,
Vellore Institute of Technology - Chennai Campus
febindaya.jl@vit.ac.in

teamwork, and programming/coding. Within the game, students can build structures to learn about architecture and design, explore different biomes and landscapes for geography, and collaborate with their peers to enhance teamwork skills. In addition, MinecraftEdu enables students to create their mods and plugins, providing an interactive way to learn programming and coding. This approach allows users to develop essential skills while enjoying a fun and engaging experience and educators can offer a personalized and interactive learning experience.

Coding is an essential skill in today's digital era, as it forms the foundation of computer technology and has extensive applications across various industries. However, mastering programming languages and concepts can be a challenging task, requiring considerable time and effort. Besides, coding frequently involves trial and error, along with the need for debugging, which can be both frustrating and time-consuming. Furthermore, technology evolves rapidly, necessitating coders to continually update their skills to stay abreast of the latest programming languages and tools. Despite these hurdles, the rewards of learning to code are enormous, including expanded career opportunities and the ability to devise innovative solutions for complex problems. Game-based learning platforms offer a unique and effective approach to teach coding by providing an interactive and engaging learning experience. Incorporating coding into a game-based platform allows learners to practice coding while having fun, promoting active participation and enhancing problem-solving skills. Learners can receive real-time feedback, making the learning process more intuitive and accessible by seeing the impact of their code. Additionally, immediate feedback provided by game-based learning platforms can help learners identify areas for improvement and track their progress over time. Ultimately, game-based learning platforms present an innovative and efficient means to teach coding that has the potential to revolutionize the way this crucial skill is taught.

The objective of this proposed project is to create a platformer game that can teach coding to a broad range of individuals. By utilizing a platformer game's straightforward user interface, it is intended to impart the essential aspects of coding, such as conditional statements, basic input/output operations, looping, and iterative statements, among others, in an effective manner. The game is being designed with interactive features and a scoring system to maintain user

engagement and to deliver an immersive and interactive learning experience for beginners and learners from various backgrounds. The game's plot revolves around an astronaut who crash-lands on an alien planet and must learn the language and skills of the "Aliens" to survive. Players engage in quests and challenges, tackling puzzles and completing coding exercises that cover fundamental to advanced programming concepts. Unity's capabilities will be leveraged to create an engaging learning experience for the Python programming language.

The paper is structured as follows: In section 2, the background study is presented, encompassing related works and existing architectures. Section 3 outlines the project's implementation, including its research objectives, proposed work, and the approach methodology employed. Section 4 provides the results, and Section V concludes the paper.

2. Background Study

A. Literature Survey

In the contemporary job market, the demand for programming skills has become increasingly pivotal. Possessing proficiency in programming is no longer confined to the realm of technology-related professions; it has become a valuable asset across various industries. Various methodologies are being explored and reported in the recent literature to gain programming skills.

Aric et al discuss the importance of code review in software development, which involves manually examining source code by developers other than the author to improve code quality (Ardic et al., 2020). Despite its significance, code review practices are often not taught in Software Engineering education, leaving a knowledge gap. To address this, the authors propose a serious game approach to teach code review practices. They describe the learning objectives, design of the serious game, and companion quizzes. After conducting a preliminary experiment and gathering feedback, there is a significant improvement in the game prototype for integration into a software engineering course. Based on the results, the game is deemed ready for use in a software engineering course. There is an increase in self-learning approach among students during covid-19 pandemic due to online classes (Chang & Hwang, 2019). However, undergraduate students often struggle with introductory programming chapters, making it

difficult for them to understand complex concepts. To align teaching methodology with students' interests, a new method of learning for IT-related undergraduate courses is needed. An alternative approach called code saga is proposed with mobile serious game that covers key programming chapters based on the latest IEEE/ACM curriculum guidelines (Hamouda et al., 2019). Code Saga includes mixed gaming approaches to make learning educational and entertaining, along with fun assessment activities to assess students' understanding and a scoring mechanism to track their progress. Results from a test group at the University of Mauritius show that students prefer learning programming through serious games over traditional methods. Perutka & Lenka discusses the use of games to engage and educate participants in a MATLAB programming course (Perutka & Lenka, 2017). They introduced two games that were developed for this purpose and used with positive feedback from course participants. The games increased motivation and attentiveness among students, as well as improved memory retention of the source code, which can benefit future programming. The games cover various aspects of the basic programming course, including conditions, loops, and file manipulation. Overall, the use of games in programming courses can be an effective way to make learning more engaging and enjoyable for students.

The challenge of undergraduate students struggling to understand Object-Oriented Paradigms (OOP) using traditional learning approaches is analyzed and an alternative approach using mobile serious game is presented (Singh & Nagowah, 2021). To tackle this challenge, they develop a mobile serious game called "OOP Codes" that includes a story mode and mini-games to help students understand important object-oriented principles. A mobile application assesses and monitors students' performance on the principles learned from the game. After being tested by undergraduate students in Mauritius, the game was found to be an effective approach to learn OOP concepts by 46 out of 54 students. Therefore, OOP Codes is considered a useful asset for teaching and learning OOP concepts in IT-related courses for undergraduate students. Pombo & Lamas discuss the trend of using tablets or smartphones for children at home or in school, and the need for suitable teaching practices and content for such devices (Pombo & Lamas, 2022). They highlight the effectiveness of gamification and interactive activities in promoting engagement and learning, especially for younger students. The authors focus on the use of the Code

Karts app for game-based learning with 17 students aged 4 to 11 in Cape Verde. Pre- and post-assessments measure the familiarity with technology, app appeal, ability to play, conceptual understanding of coding, and in-class behaviors. The results show that children prefer haptic interfaces and find mobile phones and tablets easier to use than laptops. The gamification-based learning approach was effective in promoting interaction and learning outcomes, but further studies are needed to evaluate its long-term effects on children. Wong & Chou emphasize the importance of educational tools that combine entertainment and learning in their paper (Wong & Chou, 2007). They assert that the effectiveness of such tools hinges on their ability to be entertaining, as lacking this aspect may result in decreased appeal to students. Therefore, these tools must possess unique aspects that are tailored to the learning process. In this paper, the authors introduce their interactive Bomberman game-based teaching/learning tool, which merges modern education concepts with classical teaching and laboratory work. Students are required to read and write C codes in the game-based environment to control the movements of the Bomberman character and game music. This immersive learning environment is engaging and motivates students to practice extensively in order to win the game.

Chang & Hwang highlight the genre of educational games in their study, emphasizing that these games can be both entertaining and motivating for learners (Liu et al., 2020). They provide an interactive platform where learners can practice and challenge themselves while engaging in fun activities. To make educational games a formal tool for everyday teaching and learning, the authors have created Bomberman - a game that covers most concepts in the introductory C programming language. This game blends modern education concepts with classical teaching and laboratory work, allowing learners to read and write C codes and control the movement of the Bomberman character while enjoying game music. The game includes continuous challenges, an interesting storyline, and realism, creating an engaging learning environment for learners. Learners are motivated to extensively practice as they strive to win the game. The authors believe that Bomberman is an innovative educational tool that offers a new way of teaching C programming through a game-based digital learning approach. A game-based learning and open forms of teaching were examined for their impact on teamwork, social skills, and collaboration in diverse student groups (Hohl, 2019). Blended learning, open

space technology (OST), and gamification elements were utilized as open forms of teaching. A business game with play tokens was introduced, where interdisciplinary project teams managed a 3D-visualization task. Teams were continually evaluated and rewarded with play tokens. The business game was successful, as open forms of teaching promoted active learning and increased transfer competence. Transparent evaluation made learning progress tangible, leading to higher enjoyment and interest in learning. The game fostered personal responsibility, resulting in outstanding ideas and initiatives from students. On the other hand, CodeCraft, provides an intuitive graphical interface for writing programs, similar to other educational tools like Scratch and Alice, but stands out with its problem-based learning approach in an immersive 3D environment (Ventura et al., 2015). It is designed to be an autonomous learning game accessible to diverse student groups. The authors emphasize the use of an iterative development process and evidence-centered design to enhance and evaluate the effectiveness of CodeCraft in improving students' programming skills.

B. Existing System

The existing system of teaching coding via games includes the use of serious games as a method of learning for IT-related undergraduate courses. These serious games, such as Code Saga and CodeCraft, provide an immersive and interactive learning experience through intuitive graphical interfaces for writing programs (Rapeepisarn et al., 2008). They incorporate mixed gaming approaches, including problem-based learning in an immersive 3D environment, fun assessment activities, and scoring mechanisms to assess students' understanding of programming concepts and track their progress. These games are designed to cover key programming chapters based on the latest curriculum guidelines and aim to improve students' programming skills in an autonomous learning environment. The use of evidence-centered design and iterative development processes are emphasized to enhance the effectiveness of these serious games in teaching coding. These games offer an alternative and engaging approach to traditional methods of teaching coding, and their potential in reaching diverse student groups is being explored (Tacouri et al 2021). The game progression can be tracked using graph data structure. This will enable the game play being well played by the player thereby not interrupting their learning (Lawande et al., 2022). The proposed game platform

named as Code Quest follows in the footsteps of these discussed ideas and makes an attempt to take a learner-centric approach to teaching coding via serious games, addressing some of the limitations of the existing system and providing an engaging and effective learning experience for students.

3. Proposed Approach

Teaching Python using the proposed game based approach can be a fun and engaging way to introduce students to programming concepts. When teaching Python using games, it's important to provide users with clear instructions and guidance. It's also important to encourage experimentation and creativity, so students can explore and discover new programming concepts on their own. In order for the game to interact with the user, the following gameplay mechanisms were added.

A. Movement

Movement is the most basic mechanism which

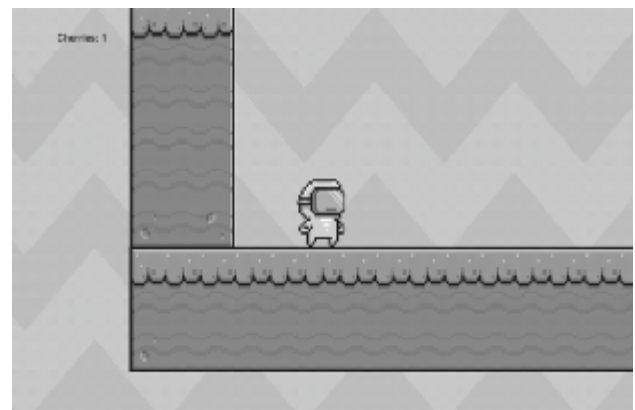


Fig. 1: Movement Controls in Gameplay

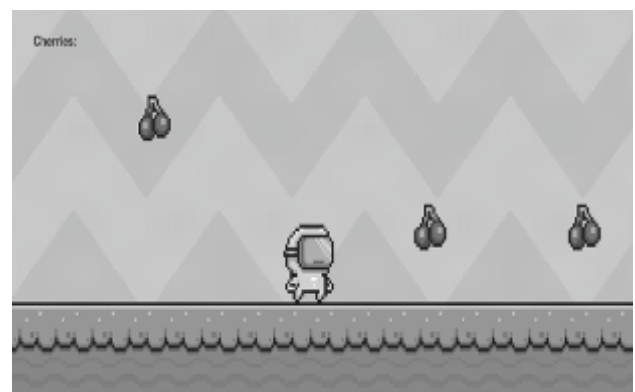


Fig. 2: Item Collection in Gameplay

allows the player to make the character move and jump around the platform as shown in Fig. 1.

B. Item Collection

Every game should have some kind of score mechanism that keeps the player playing the game and creates an intuition inside the player to keep on playing so that he can gain higher scores.

In the proposed game a food item is placed on the player's path as shown in Fig. 2, so that the player can collect and gain more points. It becomes more difficult to collect items when level proceeds, so that the player will not get bored by the game's simplicity.

C. Code Checkpoints



Fig. 3 : Code Checkpoint impact in Gameplay



Fig. 4 : Coding Arena – Test Case 1



Fig. 5 : Coding Arena – test Case 2

Algorithm 1: Code Checkpoints

```

Iterate through uncompleted levels
while uncompleted_levels_exist() do
    local current_level = get_next_level()
    Navigate through the level until the player reaches a checkpoint
    while NOT player_has_reached_checkpoint(current_level) do
        navigate_level(current_level)
    end
    Player reaches a checkpoint
    local exercise = get_programming_exercise(current_level)
    Solve the exercise
    while not exercise_solved(exercise) do
        if player_requests_hint() then
            display_hint(exercise)
            input=get_Player_Input()
            if regex(input,expectedCode)==true then
                return true
            else
                return false
            end
        end
    end
End the iteration

```



Fig. 6 : Level End Checkpoints of the Game



Fig. 7 : Traps in the Game – Type 1

This is the major hook of the game, where the player on reaching the code checkpoint as shown in Fig. 3, will be taken to a text editor and will be asked to write a basic python program. The player who gets stuck in solving the question can check for hints and after successful completion of the code, the player will be taken back to the checkpoint.

The code compilation in the text editor as shown in Fig. 4 & Fig. 5, is done using regular expression (regex) which is a powerful tool for performing complex text processing tasks, such as searching for specific patterns, validating data inputs, and transforming text. The coding questions are created in such a manner that is related to the current situation of the game the player is in. The player goes hand in hand with the story line and increase in difficulty as the level increases. The overall content of the programming in the game is distributed in a way that teaches the player the basics of python programming. The content involves important topics like

- Printing
- Taking Input
- Manipulating input
- Performing Arithmetic operations
- Conditional statements

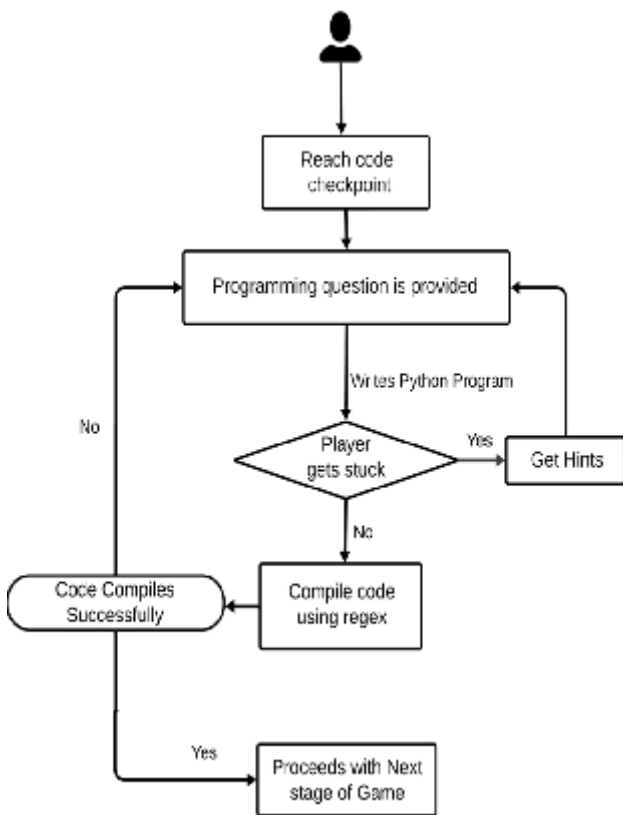


Fig. 8 : Flowchart of the complete gameplay



Fig. 9 : Traps in the Game – Type 2

- Loops

As the level proceeds the difficulty level of the questions will also get incremented. The final question will be like a final battle where the player has to build an entire application based on the topics that he has learnt in his previous questions along the game. The complete game play along with the code evaluation for leveling up is shown in Fig. 8.

D. Level Checkpoints

At the end of each level there are level checkpoints as shown in Fig. 6, where the player on reaching this level checkpoint will be taken to the next level and the player can proceed to the next level.

E. Traps

Traps are kept in order to make the game more interesting and challenging for the player. The player on hitting the traps as shown in Fig. 7 and Fig. 9, will eventually die and will be taken back to the previous checkpoint. Level 1 will not contain any traps since the player initially has to adjust with the game environments. Later on as the level proceeds certain traps are kept such that the player feels challenged in the game. This creates an intuition inside the player to continue playing and try to overcome these traps and challenges. (Jasmine et al., 2021).

F. Enemies

Enemies are created in order to make the game more interesting and challenging. It adds more fun elements to the game. These are hurdles that must be overcome in order to reach the exit to the level. They are simulated opponents that try to kill the player and stop his progress. So the player has to find some way to overcome these enemies on his path to the finish line. In our game, enemies as shown in Fig. 10 and Fig.

11, are represented by aliens of the planet in which the player crash landed on. These aliens try to stop and kill the player. There are different types of aliens. Some of them are slow and small, some of them are fast, and some of them are fast and big Graceline et al., 2024). As the level proceeds the difficulty level of the enemies will also get incremented which make the game more challenging and interesting than before.

G. Finish Lines

The major objective of the player is to escape from the alien planet. In order to do so, he has to reach to the



Fig. 10 : Miniature type enemies encountered in Gameplay – Type 1



Fig. 11 : Boss type enemies encountered in Gameplay – Type 2

place where his spaceship is and use it to escape to his home planet.

By that the game comes to an end, and will take the player to the end credits scene. In the end of level 4, after overcoming all the obstacles and hurdles, the player will reach the finish line as shown in Fig. 12, where his spaceship is located.

4. Results And Discussion

The proposed game platform is an innovative approach to teaching basic Python programming



Fig. 12 : Finish Line implementation in the Gameplay

skills in a gamified environment. Through its engaging gameplay and interactive features, it successfully addresses the learning objectives of printing, taking input, manipulating input, performing arithmetic operations, conditional statements, and loops. The game provides an immersive learning experience that encourages active participation and fosters a deep understanding of fundamental programming concepts.

The results of the study demonstrate the potential of the proposed game approach as an effective tool for teaching basic Python programming skills. The

Table 1 : Course Outcome of the Python Programming Course

S#	Course Outcomes (CO)
CO1	Understand the working principle of a computer and identify the purpose of a computer programming language.
CO2	Learn various problem solving approaches and ability to identify an appropriate approach to solve the problem.
CO3	Differentiate the programming Language constructs appropriately to solve any problem.
CO4	Solve various engineering problems using different data structures.
CO5	Able to modulate the given problem using structural approach of programming
CO6	Efficiently handle data using flat files to process and store data for the given problem.

Table 2 : Assessment of Learning Performance

S #	Assessment Type	Weightage
1	Assessment Test 1 (AT1) (After 12 classes)	15
2	Assessment Test 1 (AT2) (After 24 classes)	15
3	Assignment Test 3 (AT3) (After 32 classes)	15
4	Project (P) (2 Weeks)	25
5	Final Assessments Test (FAT) (End of Course)	30
	Total	100

game's interactive nature allows learners to practice their skills in a safe and supportive environment, providing immediate feedback and opportunities for experimentation. The game's progressive levels and challenges are designed to scaffold learning, allowing players to gradually build their knowledge and skills as they progress through the game.

To assess the efficacy of the proposed game platform for instructing Python programming, a group of 60 undergraduate students were selected as test case. Table 1 outlines the course outcomes (CO) under consideration for this study. To ensure a fair evaluation between the proposed learning method and the traditional teaching approach, the same instructor conducted classes for two student groups, allowing for a comparison of their respective learning outcomes.

The assessment metrics users for both the classes are the same and are given in Table 2. The total contact hour for the course is 45 hrs. Table 3 presents the percentage of CO attainment for various assessment tests (ATs) employing the conventional teaching approach, while Table 4 displays the corresponding CO attainment percentages for the game-based approach.

The level of attainment of both the classes are analyzed and presented for comparison as shown in table 5. While both conventional teaching approaches and game-based methods have their merits, the comparative analysis suggests that game-based

Table 3 : CO Attainment - Conventional Teaching Approach

S#	Assessments	CO1	CO2	CO3	CO4	CO5	CO6
1	AT1	88.8	58.2	65.4	68.3	NA	NA
2	AT2	NA	63.3	68.2	64.7	NA	55.5
3	AT3	NA	72.4	71.6	65.4	NA	55
4	P	NA	73.1	70.9	72.3	71.2	80.1
5	FAT	89.3	72.2	69.3	68.5	NA	75.3
	Average	89.1	67.9	69.1	67.9	71.2	66.5

Table 4 : CO Attainment - Game based Approach

S#	Assessments	CO1	CO2	CO3	CO4	CO5	CO6
1	AT1	76.5	60.4	73.4	80.4	NA	NA
2	AT2	NA	65.5	76.2	76.8	NA	59.5
3	AT3	NA	74.6	79.6	77.4	NA	59
4	P	NA	75.3	78.9	84.3	76.4	84.1
5	FAT	79.3	74.4	77.3	80.5	NA	79.3
	Average	77.9	70.0	77.1	79.9	76.4	70.5

learning has the potential to offer a more engaging, adaptable, and effective educational experience. However, it is crucial to acknowledge that the effectiveness of either approach may vary depending on the specific subject matter, learner demographics, and the integration of technology in the educational context. A balanced approach that incorporates the strengths of both methods could pave the way for a more comprehensive and impactful learning environment. In this study, the first course outcome (CO1): Comprehend the operational principles of a computer and recognize the significance of a computer programming language, demonstrates higher achievement within the conventional approach. However, the remaining outcomes, emphasizing specific programming paradigms, exhibit a greater level of comprehension and engagement in the game-based approach. The comparative analysis of the attainment course outcomes for the conventional and the proposed game based approach is shown Fig. 13.

Table 5 :Comparative Analysis of Overall CO attainment

S#	Approach	CO1	CO2	CO3	CO4	CO5	CO6
1	Conventional Teaching Approach	89.1	67.9	69.1	67.9	71.2	66.5
2	Game based Approach	77.9	70.0	77.1	79.9	76.4	70.5

Furthermore, the proposed Code Quest's gamified approach to learning programming skills is particularly relevant in today's educational landscape, where digital technologies and gamification are gaining increasing attention as effective pedagogical tools. By leveraging the motivational aspects of gaming, Code Quest has the potential to enhance

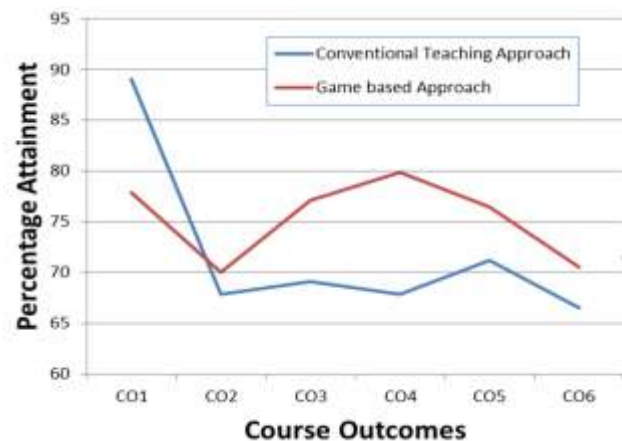


Fig. 13. Comparative Analysis of CO attainment

student engagement and motivation, thereby promoting a positive learning experience.

5. Conclusion

The paper presented, a unique game-based learning platform that utilizes the Unity game engine and C# scripting to teach basic programming skills using the Python language. The game's innovative feature of using string matching with regular expressions enables players to develop their coding skills in a practical and interactive manner, while the game's visual programming language makes it more intuitive and user-friendly. The proposed learning approach offers an immersive approach to learn Python programming concepts through engaging gameplay, real-time feedback, and visually appealing graphics, making it an inclusive and valuable educational tool for coding novices of all ages and backgrounds. With its innovative approach to teaching coding skills, this game based approach has the potential to make programming accessible and enjoyable for everyone. In future, the game could be transferred to a 3D environment and could be made more immersive with the integration of AR/VR technologies for a complete learner-centric and fun approach.

References

- Ardic, B., Yurdakul, I., & Tuzun, E. (2020). Creation of a Serious Game for Teaching Code Review: An Experience Report. *IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, 1-5.
- Chang, C.Y., & Hwang, G. J. (2019). Trends in digital game-based learning in the mobile era: A systematic review. *International Journal of Mobile Learning and Organisation*, 13(1): 68-90.
- Graceline, J. S., Jagadish, T., Shabrez, M., & Febin Daya, J.L. (2024). Classification of Organic and Recyclable Waste Using a Deep Learning Approach. *Lecture Notes in Networks and Systems*, 891.
- Hamouda, S., Edwards, S. H., Elmongui, H. G., Ernst, J. V., & Shaffer, C. A. (2019). RecurTutor: An Interactive Tutorial for Learning Recursion. *ACM Transactions on Computing Education*, 19(1), 1–25.
- Haritha, G., & Rao, R. (2024). A Holistic Approach to Professional Development: Integrating Kolb's Experiential Learning Theory for Soft Skills Mastery. *Journal of Engineering Education Transformations*, 37(2), 415–424.
- Hohl, W. (2019). Game-Based Learning-Developing a Business Game for Interactive Architectural Visualization. *11th International Conference on Virtual Worlds and Games for Serious Applications*, 1-4.
- Jasmine, S.G., Anbarasi, L.J., Narendra, M., & Raj, B.E. (2021). Augmented and virtual reality and its applications. In *Multimedia and Sensory Input for Augmented, Mixed, and Virtual Reality*. 68-85.
- Kaur, D. P., Kumar, A., Dutta, R., & Malhotra, S. (2022). The Role of Interactive and Immersive Technologies in Higher Education: A Survey. *Journal of Engineering Education Transformations*, 36(2), 79–86.
- Kulkarni, A., Deshpande, P., & Tokekar, M. (2024). Game-based Teaching Methodology for Active and Informal Learning. *Journal of Engineering Education Transformations*, 37(3), 129–142.
- Kunkolienker, K., & Kamat, M. V. (2024). Priority Based Brainvita: A Philosophical Perspective of Critical Thinking for Algorithm Generation using Game-based Approach. *Journal of Engineering Education Transformations*, 37(2), 225–233.
- Lawande, S.R., Jasmine, S. G., Anbarasi, J., & Izhar, L.I.. (2022). A Systematic Review and Analysis of Intelligence-Based Pathfinding Algorithms in the Field of Video Games. *Applied Sciences*, 12(11), 1-30.
- Liu, Z. Y., Zaffar, A.S., & Farida, G. (2020). Using the Concept of Game-Based Learning in Education. *International Journal of Emerging Technologies in Learning*, 15(14), 53-64.
- Martins, S. G. (2022). Car Race—An Engaging Interdisciplinary Python Based Project Activity. *Journal of Engineering Education Transformations*, 35(3), 45–51.

- Mondal, A. H., & Maity, R. (2023). Exploring the Effects of Game-based Learning. *Journal of Engineering Education Transformations*, 37(1), 98–105.
- Patil, Y. S., Suryawanshi, A. T., Kumbhar, S. G., & Mane, S. S. (2022). Implementation of a Team Game Tournament a Collaborative Learning Method and Study of its Impact on Learners' Development. *Journal of Engineering Education Transformations*, 36(Special Issue 2), 303–307.
- Perutka, K., & Lenka, S. (2017). New Computer Game in MATLAB for Educational Purposes. *Proceedings of the 28th DAAAM International Symposium*, 70-78.
- Pombo, N., & Lamas, D. (2022). Game-Based Learning for Young Children: A Case Study. *IEEE Global Engineering Education Conference (EDUCON)*, Tunisia, 133-138.
- Rapeepisarn, K., Wong, K.W., Fung, C.C., Khine, M.S. (2008). The Relationship between Game Genres, Learning Techniques and Learning Styles in Educational Computer Games. *Lecture Notes in Computer Science*, 5093, 497 -508.
- Rithvik, M., Narasimha Raju, K., & Ajay Ram, B. (2022). A Novel Game-Based Approach for Computer Networks. *Journal of Engineering Education Transformations*, 36(Special Issue 2), 313–318.
- Sarppa Raje, M., & Tamilselvi, A. (2024). Gamified Formative Assessments for Enhanced Engagement of Engineering English Learners. *Journal of Engineering Education Transformations*, 37(2), 500–507.
- Singh, N., & Nagowah, L. (2021). OOP Codes: Teaching Object-Oriented Programming concepts through a mobile serious game. *25th International Computer Science and Engineering Conference (ICSEC)*, 377 - 382.
- Tacouri, H., & Nagowah, L. (2021). Code Saga – A Mobile Serious Game For Learning Programming. *IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 190-195.
- Thombre, S., & Velankar, M. (2022). Gamification by Students: An effective approach to cyber security concept learning. *Journal of Engineering Education Transformations*, 36(Special Issue 1), 73–81.
- Ventura, V., Ventura, J., Baker, C., Viklund, G., Roth, R., & Broughman, J. (2015). Development of a video game that teaches the fundamentals of computer programming. *Southeast Conference 2015, Fort Lauderdale*, 1-5.
- Wong, W. T., & Chou, Y. M. (2007). An interactive Bomberman game-based teaching/learning tool for introductory C programming. *Lecture Notes in Computer Science*. 2968, 433 - 444.