Received: 17 February 2023; Revised:20 April 2023 Accepted: 26 May 2023; Published: 4 June 2023

Code Cody: A Game-Based Learning Platform for Programming Education

Mohd Suffian Sulaiman[1*], Muhammad Haziq Ikhwan Jamaludin[1], Zuraidah Derasit[2]

[1]*School of Computing Sciences, College of Computing, Informatics & Media, Universiti Teknologi MARA, Malaysia; suffian@tmsk.uitm.edu.my, haziqikhwan9@gmail.com*
[2]*School of Mathematical Sciences, College of Computing, Informatics & Media, Universiti Teknologi MARA, Malaysia; zuraidah.derasit@uitm.edu.my*

** corresponding author*

Abstract

Programming courses are an essential and challenging component of the education of future computer specialists. Many beginners struggle with the abstract nature of these courses, resulting in high failure rates. Mastering programming languages is incredibly challenging for non-computer science students, as the subject can be unfamiliar and complex. The abstract concepts and problem-solving skills required in programming pose significant hurdles for newcomers. This article proposes a development project for game-based programming education at the primary school level. Features of the application that enable students to learn programming by solving problems while playing games. This project follows the game development life cycle (GDLC) methodology, which encompasses different stages. It begins with initiation, where the initial concept and vision for the game are defined. Next is the pre-production phase, where the concept is refined, and a detailed plan is created. The production phase involves implementing programming code. Then comes the testing phase. Finally, the game is released to the end user. The result shows that all the app functionality testing works well, and it can be concluded that this app can be used as an alternative to learning programming compared to the traditional approach.

Keywords: game-based learning, unity, programming, programming education, game development life cycle

INTRODUCTION

Programming refers to the process of generating computer-executable software, apps, and scripts. It entails writing code in a computer-understandable programming language so that the machine can carry out certain activities or achieve a desired result (Guzdial & Guzdial, 2005). Programming education has become increasingly important in today's job market, as more and more industries require professionals who are proficient in coding and software development. However, traditional methods of teaching programming, such as lectures and textbooks, can often be dry and uninspiring. To engage students and make the learning process more enjoyable, many educators have turned to game-based learning to teach programming concepts (Bundhoo & Nagowah, 2022; Hong & Chu, 2017; Thiemann & Hamlin, 2022)

Game-based learning (GBL) is an approach to education that uses video games or game-like simulations to teach specific skills or concepts. In programming education, this can take the form of games that teach coding syntax and logic, or simulations that allow students to practise programming in a real-world environment. One of the main benefits of GBL in programming education is its ability to engage students and keep them motivated. Video games are inherently fun and challenging, and they can provide a sense of achievement and progress that is often lacking in traditional learning environments. By incorporating game elements into programming education, teachers can make the learning process more engaging and encourage students to stay motivated and focused (Chin, 2014; Su et al., 2019; Tacouri & Nagowah, 2021)

Despite the many benefits of GBL for programming education, it is important to note that it is not a silver bullet solution. GBL should be used with other teaching methods, like lectures and group projects, to make sure that students get a well-rounded education. Additionally, GBL tools should be carefully chosen to ensure that they align with the specific learning objectives and needs of each student (Mustafa et al., 2018; Noval et al., 2019; Rayner Tangkui & Tan Choon Keong, 2020)

RELATED WORKS

Chin, 2014 investigate the effectiveness of using digital games, specifically PowerPoint games, for the subject of Information and Communication Technology (ICT) on primary school students. The study found that PowerPoint games provide immediate feedback to students, which helps them learn concepts effectively and interestingly.

In 2021, Tacouri and Nagowah developed a mobile serious game called Code Saga to help undergraduate students learn programming concepts. The game uses mixed gaming approaches to make learning entertaining and educational. The study revealed that the majority of students preferred to learn programming using serious games compared to traditional learning methods.

Similarly in 2021, Theofilus and Widianto developed a traffic violations game-based learning to spread awareness of traffic law in Indonesia. According to the results of the questionnaires, the game can be a useful tool for traffic education.

In 2022, Bundhoo and Nagowah developed the mobile serious game called Gaming with OOP Learn to teach Object-Oriented Programming concepts to undergraduate students. The paper suggests that game-based learning has found great success in improving learners' ability to learn new skills and the capacity for retention as opposed to the traditional learning method.

Received: 17 February 2023; Revised:20 April 2023 Accepted: 26 May 2023; Published: 4 June 2023

In another study conducted in 2022, Thiemann and Hamlin proposed the use of game-based programming using MATLAB App Designer. The paper found that incorporating game design into first-year engineering programming courses can keep students excited about their coursework, engaged in challenging problems and proud of their accomplishments.

The previous studies on GBL are abundant but do not focus on the Standard Based Curriculum for Secondary Schools (KSSM) syllabus, especially for computer science foundation subjects. The proposed GBL, on the other hand, engaged the user to solve the problem and employed the KSSM syllabus so that it suited our secondary-level Malaysian education system.

METHODOLOGY

The game development life cycle (GDLC) methodology is employed as illustrated in Figure 1 to develop the game. Initiation, pre-production, production, testing and release are the five stages of this methodology (Theofilus & Widianto, 2021).

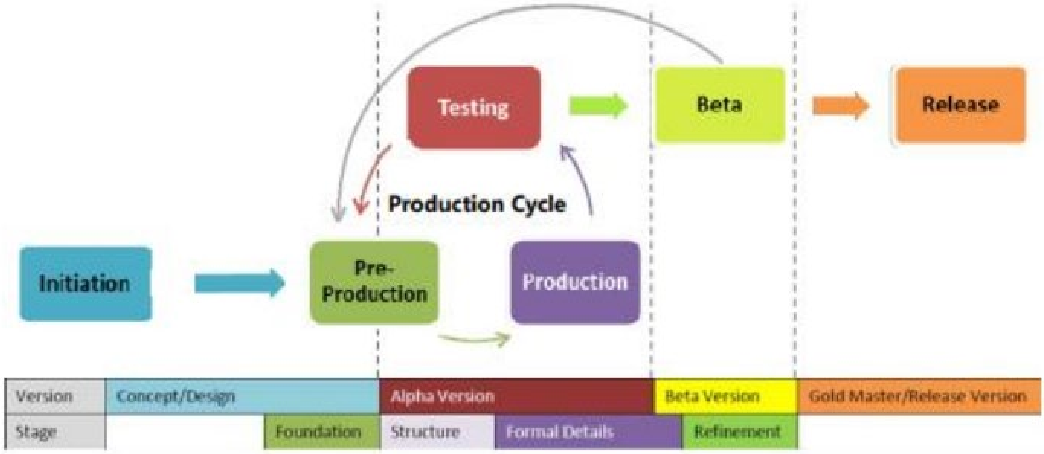


Figure 1: Game Development Life Cycle

Phase 1 - Initiation

The initial stage of the GDLC methodology is the beginning stage. Here, similar games are analysed, concepts are generated, and hardware and software requirements are determined. Figure 2 illustrates the Use Case Diagram for Code Cody (Azmy et al., 2021; Rasydan Ismail et al., 2019). The use case diagram describes the user interaction with the functional requirement of the application.

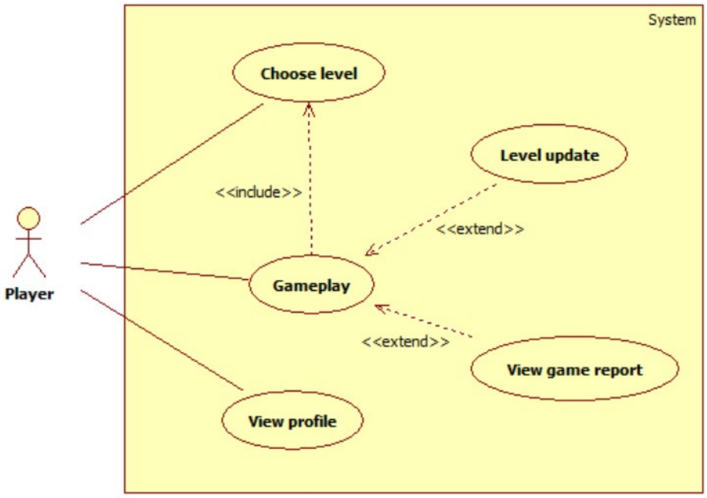


Figure 2: Use Case Diagram of Code Cody

Phase 2 – Pre-production

The pre-production phase is when game design is created, and it is the most important step in the game development process. Details of the game concept, game genres, game character, game mechanics, interactivity, game plot, enjoyment factor, and game features are all covered in game design. The game's graphics will lean towards a cute cartoon style, which will make the game's appearance more appealing and easier to capture the interest of young players. Furthermore, the navigation structure is an important aspect of the game interface because it takes the user around the game program.

Figure 3 illustrates the flow chart of Code Cody. Essentially, it outlined the actions that will be taken to construct a system that will be finalised during the creation of this game.

Received: 17 February 2023; Revised:20 April 2023 Accepted: 26 May 2023; Published: 4 June 2023

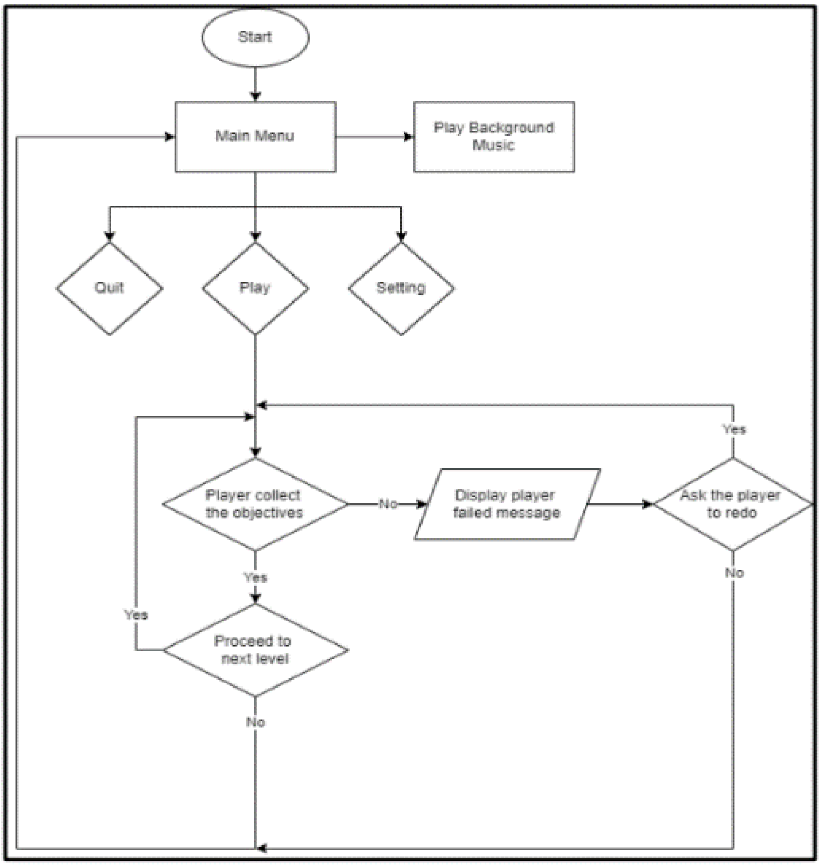


Figure 3: Flow chart of Code Cody

Phase 3 – Production

The production phase is the most important part of the game development process. Game assets and coding are two sub-phases that are considered to produce the game during this phase. Artwork, audio, video, maps, and other data are all examples of game assets. This game is developed using C# (C-Sharp) as the programming language through Unity 3D as shown in Figure 4.

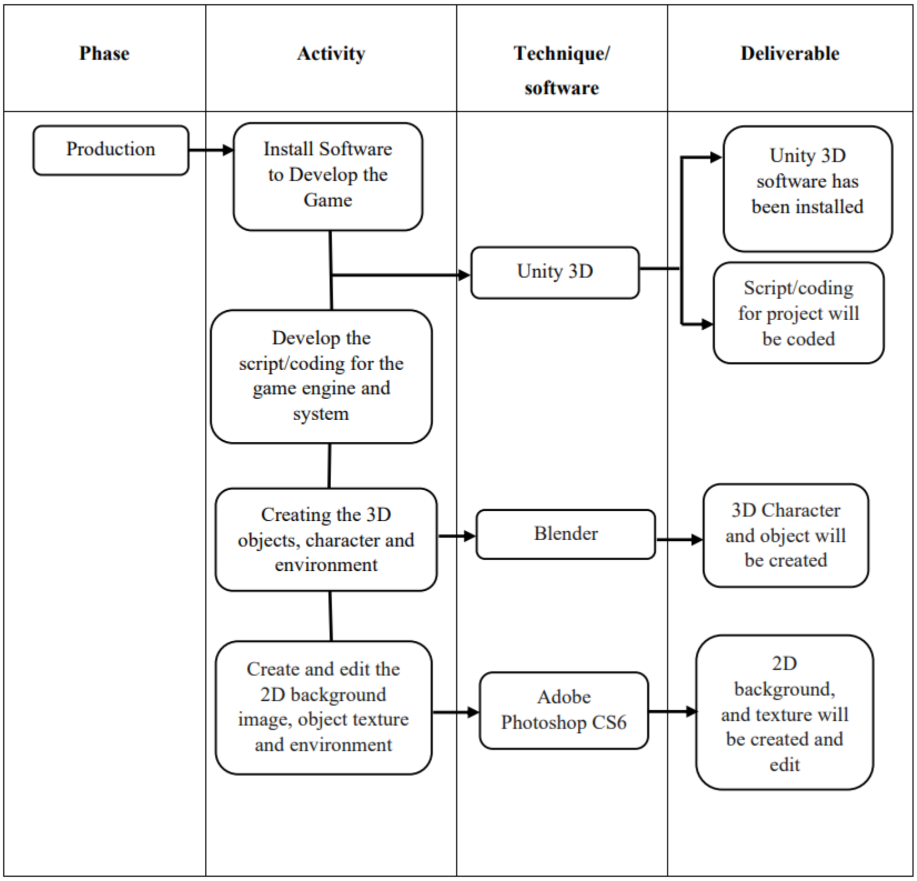

Phase
Activity
Technique/ software
Deliverable
Production
Install Software to Develop the Game
Unity 3D
Unity 3D software has been installed
Script/coding for project will be coded
Develop the script/coding for the game engine and system
Creating the 3D objects, character and environment
Blender
3D Character and object will be created
Create and edit the 2D background image, object texture and environment
Adobe Photoshop CS6
2D background, and texture will be created and edit

Figure 4: Production Phase Activity

Phase 4 – Testing

Unit tests, integration tests, system tests, and acceptance tests are the general stages of testing. However, this project will be focused on the system tests only. The acceptance test with the end user is not implemented because this study has not reached this phase yet. System testing is defined as examining how well the application satisfies the functional and non-functional requirements. Test cases are developed to execute the system tests. A test case is the specification of the inputs, conditions of the execution, testing procedure, and expected results. The test cases involved are for the main menu, options menu, level menu and pause menu as illustrated in Table 1, Table 2, Table 3 and Table 4 respectively. Other test cases are for questions 1, 2 and 3, tutorials 1, 2 and 3, game over the menu, wrong answer menu, score menu, level 1, level 2, level 3, credit, game manager, and score manager but not provided in this article.

Table 1: Main menu module test case

Objective	Expected Result	Actual Result	Status
Check for button availability	The main menu should have all quit, play and options buttons	As expected	Pass
Check for button functions	All the buttons should function well	As expected	Pass

Table 2: Options menu module test case

Objective	Expected Result	Actual Result	Status
Check for button availability	The main menu should have a volume toggle button	As expected	Pass
Check for button functions	The buttons should function well	As expected	Pass

Table 3: Level menu module test case

Objective	Expected Result	Actual Result	Status
Check for button availability	The level menu should have all level 1, level 2, level 3 and back buttons	As expected	Pass
Check for button functions	All the buttons should function well	As expected	Pass
Check for button animation	The buttons should have animations	As expected	Pass

Table 4: Pause menu module test case

Objective	Expected Result	Actual Result	Status
Check for button availability	The pause menu should have all resume, settings, menu and quit buttons	As expected	Pass
Check for button functions	All the buttons should function well	As expected	Pass

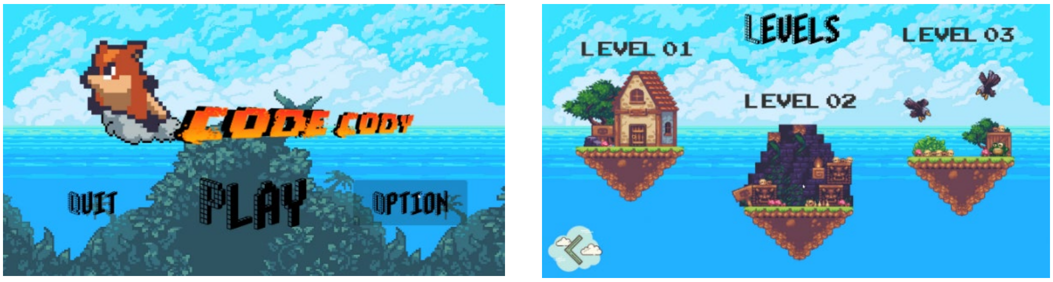
Phase 5 – Release

In the GDLC model, once all the previous phases are completed, the app can be released to the end user. However, this study has not reached this phase yet.

RESULTS AND DISCUSSION

This section presents the results of this project. The graphical user interface (GUI) of the game is as follows:

Figure 5 depicts the main GUI of Code Cody. In Figure 5 (a), the user has three options to choose from: quit, play, and options. In the option menu, the user can adjust the volume of the sound. Figure 5 (b) will appear when the user selects the play menu. The user can select from three levels to play the game, with each level increasing in difficulty. If the user hovers over the level's button, a floating animation will play. This animation is played using the math and IEnumerator functions, which enables the button's animation to play if the cursor is over the button.



(a) (b)

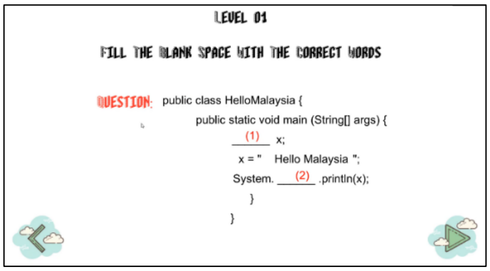
Figure 5: Code Cody's main GUI

Once the user has selected a level, the user will be asked the question depicted in Figure 6 (a). The user must understand the question and figure out how to solve the puzzle in the game. Figures 6 (b), (c), (d), (e), (f), and (g) show how the user will be led through the tutorial before the game begins.

The player is a red fox in the game. The user needs to move the player using the A, S, W, and D keys or the user can also use the arrow keys. The player needs to collect the objectives and bring them to the end point to finish the level.

If the user selects the incorrect answer while playing the game, as depicted in Figure 6 (h), the incorrect answer GUI will be displayed at the end of the game, as shown in Figure 6 (i). The user will then have the option to retry or quit the game. Each player will be given three lives. When the player collides with obstacles as depicted in Figure 6 (j), lives will be deducted. As shown in Figure 6 (k), once the player runs out of lives, the game will be over.

If the user selects the correct answer during gameplay as depicted in Figures 6 (l) and 6 (m), the completed level GUI will appear as depicted in Figure 6 (n). The number of coins collected will be displayed alongside the grade.



LEVEL 01

FILL THE BLANK SPACE WITH THE CORRECT WORDS

QUESTION:

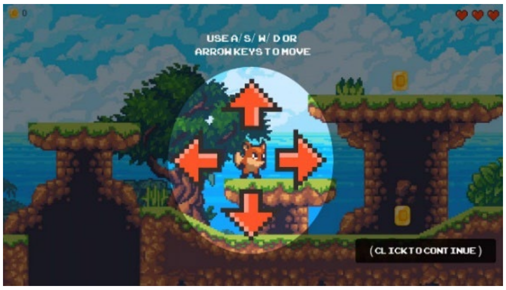
```
public class HelloMalaysia {
        public static void main (String[] args) {
    ___(1)___ x;
        x = "    Hello Malaysia ";
    System. ___(2)___ .println(x);
        }
    }
```

(a)

LEVEL 01
TUTORIAL

(CLICK TO CONTINUE)

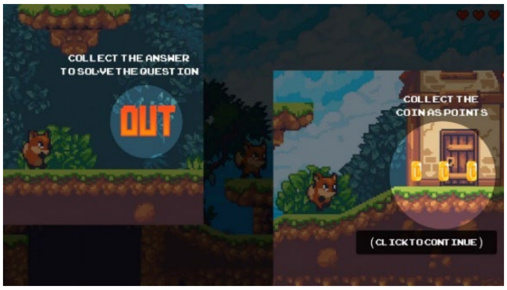
(b)



USE A/ S/ W/ D OR ARROW KEYS TO MOVE

(CLICK TO CONTINUE)

(c)

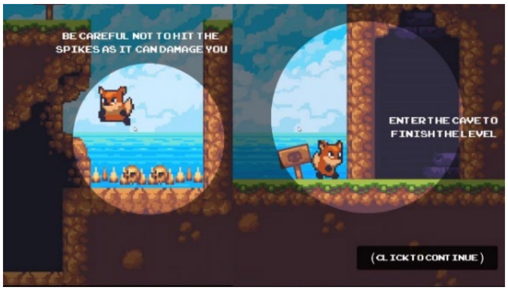


COLLECT THE ANSWER TO SOLVE THE QUESTION

COLLECT THE COIN AS POINTS

(CLICK TO CONTINUE)

(d)

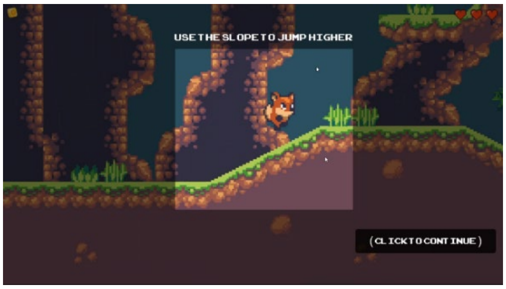


BE CAREFUL NOT TO HIT THE SPIKES AS IT CAN DAMAGE YOU

ENTER THE CAVE TO FINISH THE LEVEL

(CLICK TO CONTINUE)

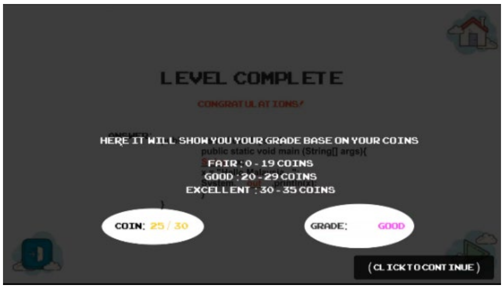
(e)



USE THE SLOPE TO JUMP HIGHER

(CLICK TO CONTINUE)

(f)



LEVEL COMPLETE

CONGRATULATIONS!

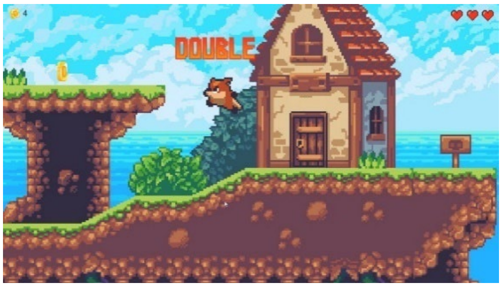
HERE IT WILL SHOW YOU YOUR GRADE BASE ON YOUR COINS
FAIR : 0 - 19 COINS
GOOD : 20 - 29 COINS
EXCELLENT : 30 - 35 COINS

COIN: 25 / 30

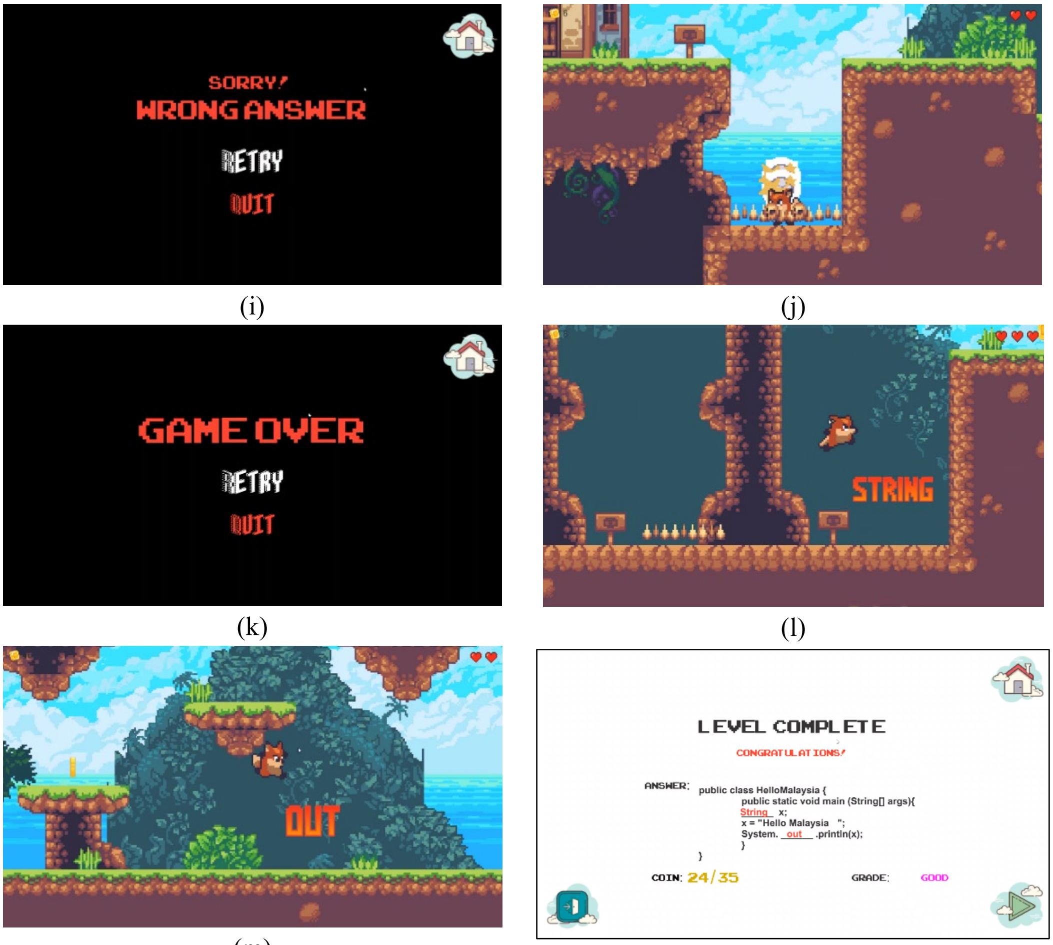
GRADE: GOOD

(CLICK TO CONTINUE)

(g)



(h)



```
public class HelloMalaysia {
          public static void main (String[] args){
          String   x;
          x = "Hello Malaysia   ";
          System. _out_ .println(x);
          }
}
```

(i) (j)

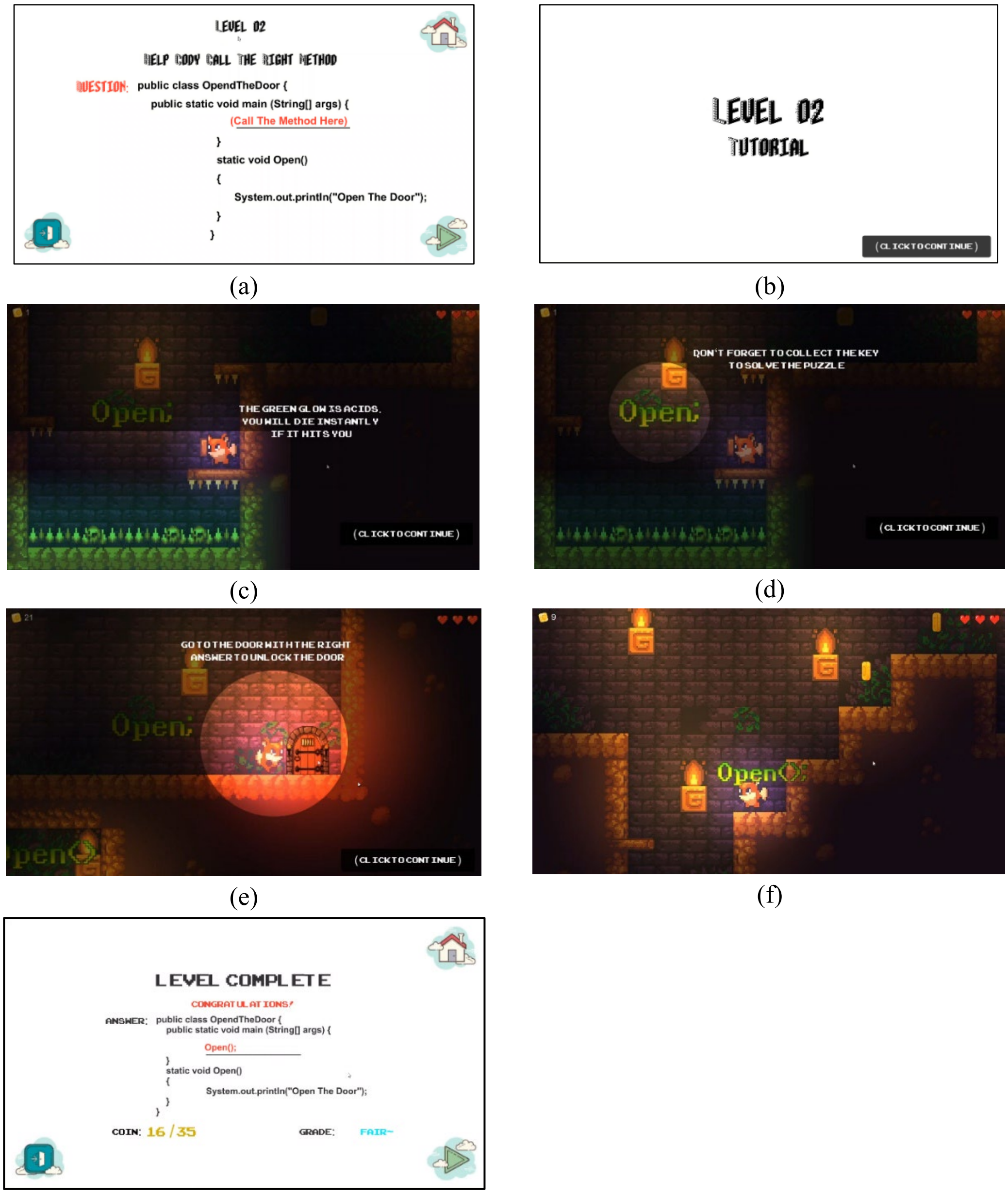
(k) (l)

(m) (n)

Figure 6: Level 1 of Code Cody

Figure 7 (a) illustrates the question for level 2. The user will then be given the tutorial before starting the game, as illustrated in Figures 7 (b), 6 (c), 6 (d) and 6 (e). The game environment for level 2 is based on a dungeon, which is usually a dark and eerie place. The user requires light to illuminate the player's view.

The game objective of the second level is to pick the correct answer as a key to open the door so that the player can exit the dungeon. The exit door will be opened if the player has the correct answer. The correct answer will follow along with the player once the player touches it. It will act as a key to open the exit door, as illustrated in Figure 7 (f).



LEVEL 02

HELP CODY CALL THE RIGHT METHOD

QUESTION:
```
public class OpendTheDoor {
    public static void main (String[] args) {
        (Call The Method Here)
    }
    static void Open()
    {
        System.out.println("Open The Door");
    }
}
```

(a)

LEVEL 02
TUTORIAL

(CLICK TO CONTINUE)

(b)

THE GREEN GLOW IS ACIDS.
YOU WILL DIE INSTANTLY
IF IT HITS YOU

(CLICK TO CONTINUE)

(c)

DON'T FORGET TO COLLECT THE KEY
TO SOLVE THE PUZZLE

(CLICK TO CONTINUE)

(d)

GO TO THE DOOR WITH THE RIGHT
ANSWER TO UNLOCK THE DOOR

(CLICK TO CONTINUE)

(e)

(f)

LEVEL COMPLETE

CONGRATULATIONS!

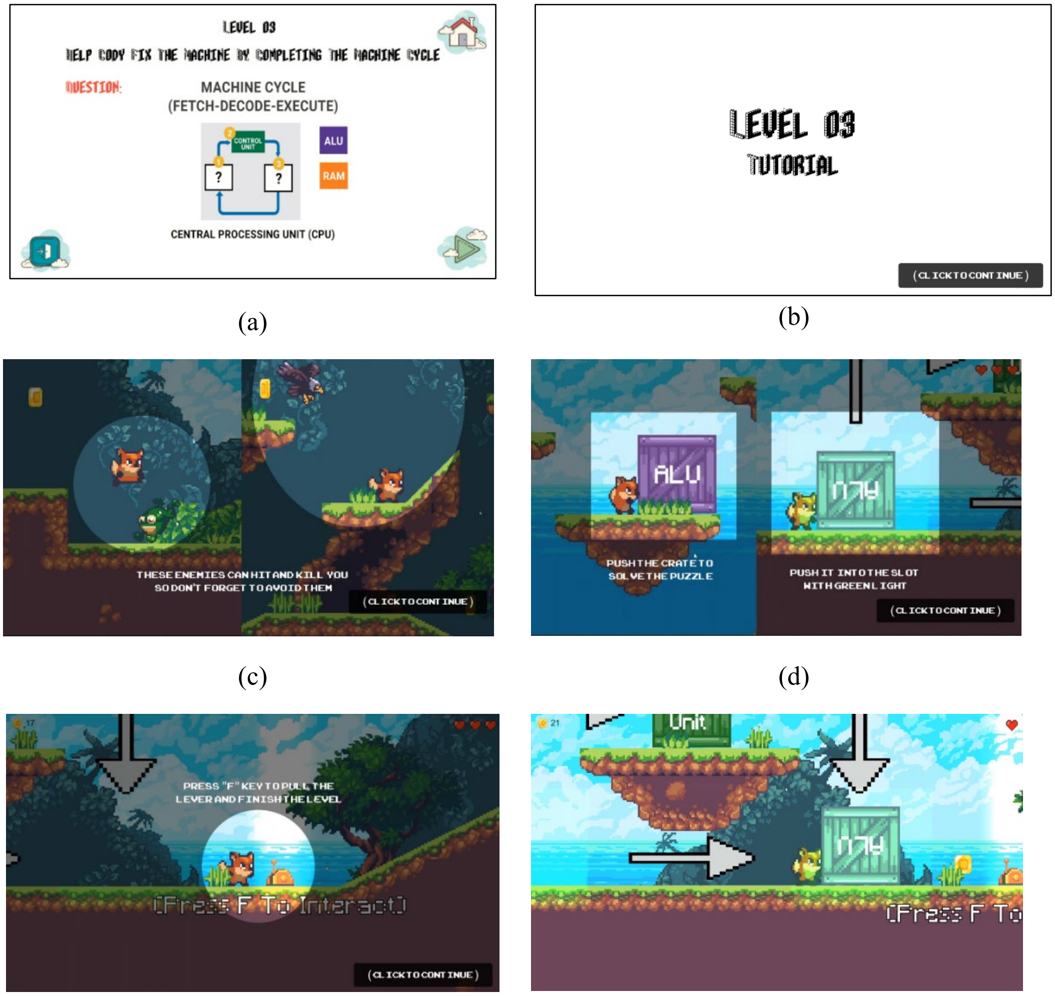
ANSWER:
```
public class OpendTheDoor {
    public static void main (String[] args) {
        Open();
    }
    static void Open()
    {
        System.out.println("Open The Door");
    }
}
```

COIN: 16 / 35   GRADE: FAIR~

(g)

Figure 7: Level 2 of Code Cody

Figure 8 (a) illustrates the question for level 3. Like the previous level, the user will be given the tutorial before starting the game, as illustrated in Figures 8 (b), 8 (c), 8 (d) and 8 (e). In this level, the player needs to push the crates to the correct location. After pushing the crate to the right places, the player will need to pull the lever to check the answer. If the answer is correct, then the score page will pop up as illustrated in Figure 8 (i) but if it is incorrect, the wrong answer will pop up.



LEVEL 03

HELP CODY FIX THE MACHINE BY COMPLETING THE MACHINE CYCLE

QUESTION: MACHINE CYCLE (FETCH-DECODE-EXECUTE)

(a)

LEVEL 03
TUTORIAL

(CLICK TO CONTINUE)

(b)

THESE ENEMIES CAN HIT AND KILL YOU SO DON'T FORGET TO AVOID THEM

(CLICK TO CONTINUE)

(c)

PUSH THE CRATE TO SOLVE THE PUZZLE

PUSH IT INTO THE SLOT WITH GREEN LIGHT

(CLICK TO CONTINUE)

(d)

PRESS "F" KEY TO PULL THE LEVER AND FINISH THE LEVEL

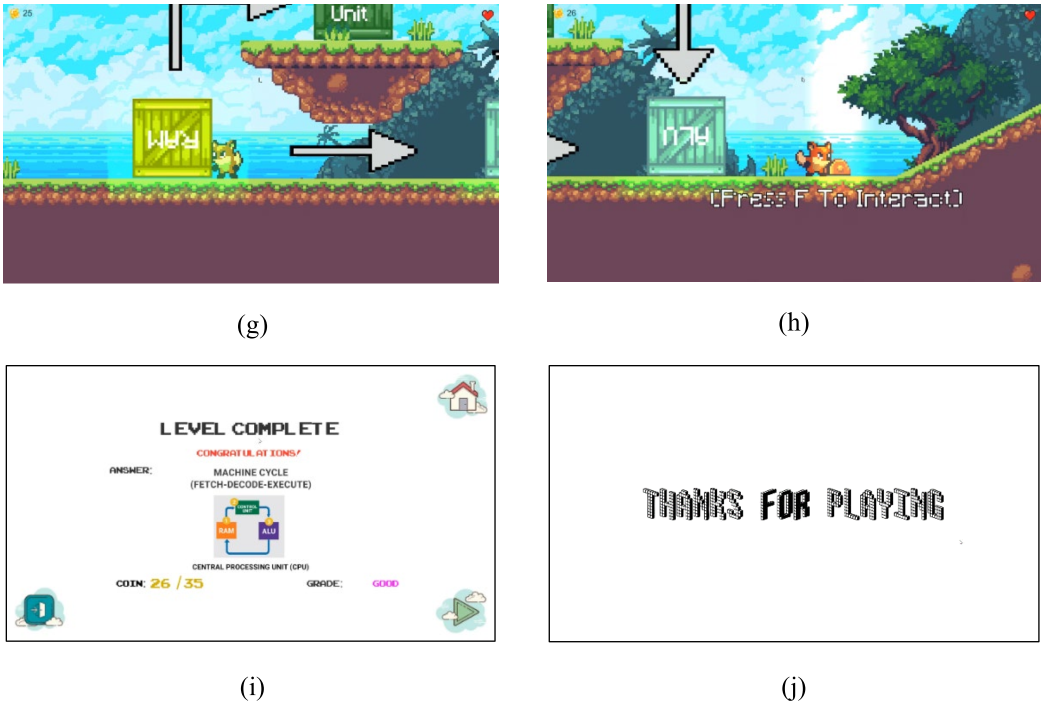
[Press F To Interact]

(CLICK TO CONTINUE)

(e)

[Press F To

(f)



(g) (h)

(i) (j)

Figure 8: Level 3 of Code Cody

CONCLUSION

This paper discussed the development of GBL for programming education namely Code Cody. GBL has the potential to revolutionise the way programming is taught in educational settings. By making the learning process more engaging, motivating, and interactive, GBL can help to ensure that students have a deeper understanding of programming concepts and are better equipped to apply them in real-world scenarios. GBL should not be used as the sole method of teaching programming, but it can be a valuable addition to traditional teaching methods. For further research, more questions and levels can be added so that the game can cover the whole syllabus. In the future, the game should also be played on mobile platforms.

ACKNOWLEDGEMENTS

The author would like to thank the College of Computing, Informatics & Media, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia for all the support.

REFERENCES

Azmy, I. H., Azmi, A., Sulaiman, M. S., & Mohd Yusop, O. (2021). Digital transformation in oil and gas industry: developing an OSDU third-party application. *7th International Conference on Engineering and Emerging Technologies*, *October*, 27–28. https://doi.org/10.1109/ICEET53442.2021.9659636

Bundhoo, D., & Nagowah, L. (2022). Gaming with OOP learn: A mobile serious game to learn object-oriented programming. *3rd International Conference on Next Generation Computing Applications*. https://doi.org/10.1109/NextComp55567.2022.9932243

Chin, S. M. (2014). Digital Game-based learning for teaching information and communication technology subject (The new subject in Standard Curriculum for Primary Schools (KSSR). *2nd IPGM International Conference*, *October*. https://www.researchgate.net/publication/282908010

Guzdial, M., & Guzdial, M. (2005). *Introduction to Computing and Programming in Python, A Multimedia Approach* (4th ed.). Pearson.

Hong, T. Y., & Chu, H. C. (2017). Effects of a situated 3D computational problem-solving and programming game-based learning model on students' learning perception and cognitive loads. *6th IIAI International Congress on Advanced Applied Informatics*, 596–600. https://doi.org/10.1109/IIAI-AAI.2017.96

Mustafa, M. Q., Hussein, S. K., Ali Raad, A., & Yussalita, Md. Y. (2018). Design and development of an interactive persuasive mathematics game for primary school children. *International Journal of Engineering & Technology*, *7*(4), 272–276. https://doi.org/10.14419/ijet.v7i4.19.22065

Noval, B. A., Safrodin, M., & Hakkun, R. Y. (2019). Battlebot: Logic learning based on visual programming implementation in multiplayer game online. *International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy*, 138–142. https://doi.org/10.1109/ELECSYM.2019.8901628

Rasydan Ismail, M. E., Ahmad Shukri, I. F., Azmi, A., Yahya, Y., Ismail, S. A., & Sulaiman, M. S. (2019). Development of Agronomist station system for water table management at Peatland. *International Conference on Research and Innovation in Information Systems, ICRIIS*.

Rayner Tangkui, & Tan Choon Keong. (2020). Enhancing pupils' higher order thinking skills through the lens of activity theory: Is digital game-based learning effective? *International Journal of Advanced Research in Education and Society*, *2*(4), 1–20.

Su, J. M., Li, M. J., Li, W. D., & Zhuang, X. Y. (2019). Building an authoring tool to create blockly-based programming learning games for elementary students. *8th International Congress on Advanced Applied Informatics*, *3*, 246–249. https://doi.org/10.1109/IIAI-AAI.2019.00056

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. https://doi.org/10.1109/IoTaIS53735.2021.9628484

Theofilus, G., & Widianto, M. H. (2021). Development of game-based learning traffic order for android with the game development live cycle method. *3rd International Conference on Cybernetics and Intelligent Systems, ICORIS 2021*, *January*. https://doi.org/10.1109/ICORIS52787.2021.9649553

Thiemann, K. A., & Hamlin, B. H. (2022). Implementation of game-based programming as a means to engage and excite students in first-year engineering courses. *Frontiers in Education Conference, FIE*, *2022-Octob*, 1–5. https://doi.org/10.1109/FIE56618.2022.9962382