

# Developing Steps for Learning Programming through Gamification

Si Thu Tun

Department of Mechatronics Engineering  
Assumption University  
Samut Prakan, Thailand  
sithutuny8@gmail.com

Allapon Hutasingh

Department of Mechatronics Engineering  
Assumption University  
Samut Prakan, Thailand  
allaponhts@au.edu

Narong Aphiratsakun

Department of Aeronautic Engineering  
Assumption University  
Samut Prakan, Thailand  
Narongphr@au.edu

**Abstract**—Gamification has been widely used in recent years to enhance the learning experience of various subjects, including programming. However, there is a lack of clear and structured approach to implement gamification in programming education. This aim of this research is to develop a set of steps for efficiently learning programming through a game. The proposed steps cover the essential aspects of learning programming such as problem-solving, debugging, and understanding of programming concepts. The study also found that the proposed steps are flexible and also be applied to different programming languages and level of learners. The proposed steps also can be used as a guide for educators and trainers to implement the proposed step in programming education to make the learning process more engaging, interactive and fun. The study concludes the proposed step can be a valuable resources for educators and trainers to use gamification in education.

**Index Terms**—programming, gamification, education guide, interactive learning, learning steps, problem-solving

## I. INTRODUCTION

Gamification is the process of adding game-like elements to non-game activities. The idea of gamification has been present for an extended period [1]. The idea behind gamifying is to make the learning process more engaging and interactive, allowing learners to develop their skills in a fun and challenging way [2].

According to a study [3] shown that student who studied the programming through typical lecture method had trouble identifying key concepts, if not followed by an exercise immediately. Afterward, a new system is where instead of normal assignment, a game-like curriculum was created. Normal assignment were replaced by missions, where students gain experience to climb the leaderboard and earn the achievements. Majority of the participated student found that the system help them in learning process. The teaching staff also found improved interactions and motivation amongst the feedback. This supports the idea that the gamification can help learners to understand the programming concepts more meaningful. Another study also found that "Learning method becomes more fun but still do not forget the material given by the lecturer. So there is a balance in learning in the classroom, creating a comfortable learning environment and not boring." [4]. This highlights the potential of gamification that create the learning process more engaging and interactive.

Gamification is not limited to education, it could be applied in different areas such as: Marketing, Health and Fitness and Health, Game Industry, and Public services. A research stated that "The practice of medicine often involves tedious, repetitive, boring, and/or painful routines for both the practitioner and patient. Thus, with gamification, health workers can engage and collaborate more effectively." [5]. This show that gamification can be used improve the experience of both healthcare practitioners and patients by making medical routines more engaging and interactive. A research done in interactive marketing also found that "Gamification increases the psychological responses to advertising through user engagement with the game and transferring persuasion knowledge to the customer." [6].

However, despite the benefits of gamification, there is a lack of clear structured approach to implement in programming education. Thus, this study focus to develop a sets of steps for learning programming. The existing gamified programming education resources and methods [7] will be analyzed, the common elements and steps are extracted, and a guide can be proposed for educators, trainers and student to apply gamification in programming education.

## II. DESIGN STEP

Effective pre-planning is crucial for successful coding. It enables one to grasp the logic behind the program and identify potential challenges. Skipping this step can lead to oversights and errors in the final product. To ensure the game is developed efficiently, we will follow to the 5 main steps. Each step is designed to provide the learner with a comprehensive understanding and the necessary guidance to independently create their own game in the future. The following is the proposed steps to simplify the programming gamification. We will utilize a mindmap as a tool to organize and structure our plan [8].

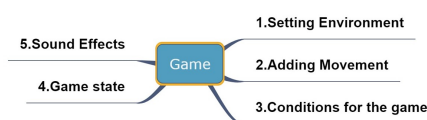


Fig. 2: Mindmap of the 5 Steps

### A. Step 2.1: Setting environment

The game environment plays a crucial role for determining the feel and functionality of the game. It serves as the foundation on which the game is built. In this step, a basic blueprint of the game should be established. This includes incorporating images of backgrounds, characters, and obstacles as a starting point, as well as considering important visual elements such as the pixel size and positioning of the images.

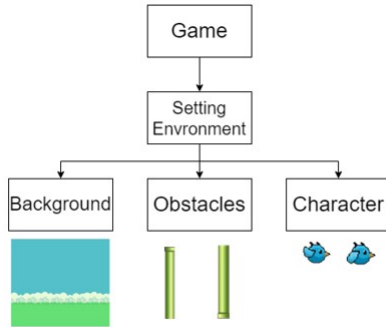


Fig. 2.1: Examples of Environment and some characters

### B. Step 2.2: Movement

The next step in the game development process is to implement the movement mechanics. This includes adding different types of movement systems to the game, such as walking, running, jumping, and climbing. In this step, inputs from the keyboard can be used to create the movement of our characters. This means that we will need to create a set of rules that dictates how the character moves, such as how fast they move, how high they jump, and how they respond to different inputs. Additionally, it is important to take into account how the movement of the characters and obstacles will be animated and displayed on the screen, considering the X and Y-coordinates of the images' pixels. This step is necessary as it adds an element of interactivity to the game and makes it more engaging for the player.

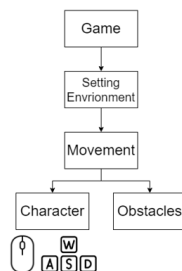


Fig. 2.2: Movements

Mathematical equations can be utilized to precisely calculate and control the movements of the characters and obstacles in the game.

$$Character.Y = Character.Y + Gravity \quad (1)$$

$$Obstacle.X = Obstacle.X + Speed \quad (2)$$

By applying gravity, the character will gradually fall down the screen by decreasing its Y-coordinate position as equation (1). The obstacles will move horizontally across the screen at a specific speed. This is controlled by the value of the speed that is added to the X-coordinate position of the obstacle as equation (2). After the movement mechanics have been implemented, different inputs can be used to control the character's movement.

### C. Step 2.3: Condition for the game

The next step is to add the game's conditions. The conditions in which the outcome of the game, such as determining if a player wins, loses, or gain points. This includes creating specific triggers that activate when certain actions occur, such as reaching a certain score or completing a level. Here it's important to consider how the conditions will be evaluated and what are the outcomes as a result. For example, if a player reaches a certain score, the game will count as win or player may advance to new level.

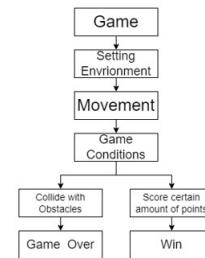


Fig. 2.3: Conditions for the game

### D. Step 2.4: Game States

After the game is functioning, different layers can be added to enhance overall experience. These layers can be main menu screens, game over screen, or different levels of the game. The menu screen will provide the player with options to start the game, view the instructions, or adjust the settings. The game over screen will be displayed when the player loses and allows to restart the game or view the earned score or a winning screen where the game states that the player has successfully cleared the stage or completed the goal. Another option can also be adding different stages of the game to provide more challenging and engaging experience for the player, as they progress through the game. This is the step where the game can be polished for a better dynamic experience.

### E. Step 2.5: Sound effects and Music

The final step in the game development process can be adding sound effects, background music, and songs. Sound effects can provide feedback to the player, such as the sound of jumping or hitting an obstacle. Background music and songs

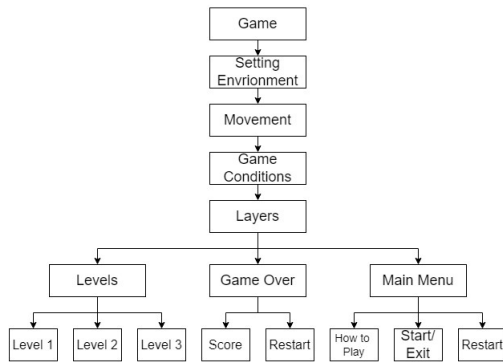


Fig. 2.4:Layers of The Game

are added to create more atmospheric and engaging experience for the player.

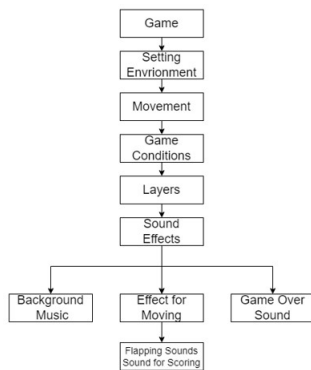


Fig. 2.5:Music and Sound Effects

### III. TESTING AND RESULTS

Our experiment will focus on creating a simple game, called Flappy Bird, using Python programming language.

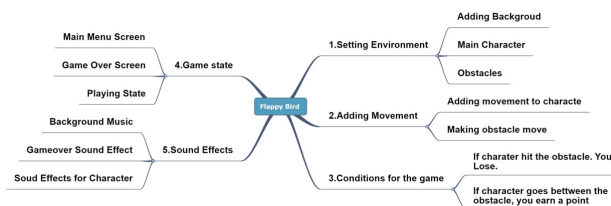


Fig. 3: Mind Map of The Game Flappy Bird

Through this process, we aim to gain insights into the process of creating basic games and understand how to apply 5-step approach outlined in previous section.

#### A. Step 3.1: Setting Environment

In this step, we added various images to the game, such as backgrounds, characters, and obstacles. Before adding the

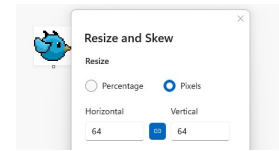


Fig. 3.1.1: Height and Width of the Bird

images, the size of each image is checked by loading into the paint software.

This is essential to ensure that the images are displayed correctly on the screen. After checking the size of the images, we set the screen size of the game and added the images. Understanding the width and height of the images is important for setting the images correctly in the game.

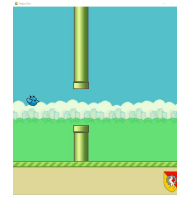
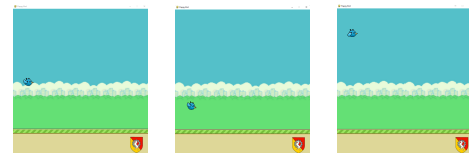


Fig. 3.1.2: Loaded Images

#### B. 3.2 Movement

In this step, we added the movement mechanics for the characters and obstacles. The initial position of the bird character is set to be in the middle of the background. We then applied gravity to the bird, causing it to gradually fall towards the ground from its vertical position as equation (1). In additional, a left mouse click is used as an input for the player. Clicking the button will add a positive value to the bird's vertical position, allowing it to fly upwards.



(a) Character in Starting Position (b) Bird falling due to Gravity (c) Bird going up due to input

Fig. 3.2.1: Movement of the bird

Similarly, the obstacles are set to move horizontally across the screen at a specified speed. The speed is set in a way that obstacles are challenging but not impossible to avoid. We also used the random library to create multiple different heights, the obstacles are generated randomly each time a new set of obstacles is created, adding an element of surprise to the game.

#### C. 3.3 Condition for the game

Here, we added the outcome of the game such as winning, losing, and scoring. Points are added when the bird successfully passes through the gaps. Game over condition is met

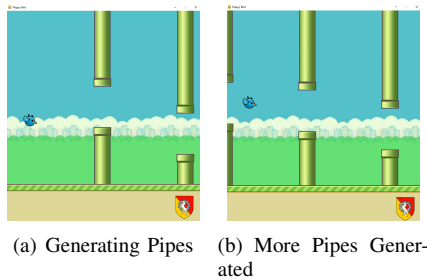


Fig. 3.2.2:Pipes moving through Right to Left

when the bird collide with the pipes. The player will win the game if he can goes through a certain amount of obstacles.

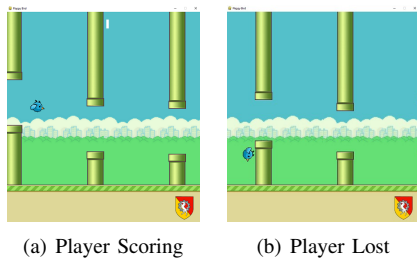


Fig. 3.3:Different Conditions of the Game

#### D. 3.4 Game States

Different kinds of menu screen are added here. The first is the main menu screen which has buttons for either starting the game or exiting the game alongside with some instruction for the game. When the start button is clicked, the game will move to the playing screen where the bird and obstacles will appear. If the win condition from previous step is met, the game will move into a winning screen or if the game over condition, is met the game will move to the game over screen instead where a player can restart the game and view the score.

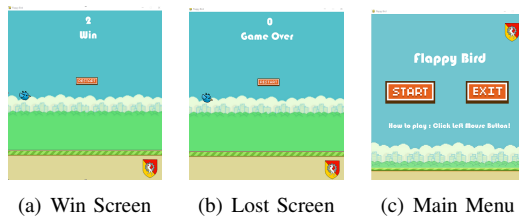


Fig. 3.4:Different Screens for the game

#### E. 3.5 Sound effects and Music

As a final step of the game development process, background music and sound effects are added to boost the overall experience.

#### F. result

During a programming course, the proposed steps were introduced to the students, and the response was mostly positive. Students expressed satisfaction with the five methods, and



Fig. 3.5:Video Link to Complete game

many were able to successfully create their own games using the step-by-step approach. The simplicity of the steps and the clear instructions made the learning process more manageable and easier to understand, especially for those new to programming. This positive feedback shows that these proposed steps can be an effective method to teach gamification programming. Here are some of their responses "shorturl.at/uzKS7".

#### IV. CONCLUSION

In conclusion, we were able to successfully develop the game, Flappy Bird, using the proposed steps outlined in this research. The proposed steps cover all the necessary aspects of game development, from pre-planning and design to adding movement, obstacles, and game conditions. We believe that by following these approaches the development process can be easier creating engaging and interactive learning process.

As future work, we can use these steps to create more complex games. We can incorporate different input methods such as controllers, joysticks, to increase the level of difficulty. Additionally, these steps can also serve as a foundation for other types of development such as creating animations and editing images using code.

#### REFERENCES

- [1] Malone, Thomas W. "What makes things fun to learn? Heuristics for designing instructional computer games." Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems. 1980.
- [2] Duggal, Kavisha, Anukool Srivastav, and Satvinder Kaur. "Gamified approach to database normalization." International Journal of Computer Applications 93.4 (2014).
- [3] Huang, Wendy Hsin-Yuan, and Dilip Soman. "Gamification of education." Report Series: Behavioural Economics in Action 29 (2013)
- [4] Rahardja, Untung, et al. "Gamification framework design of management education and development in industrial revolution 4.0." Journal of Physics: Conference Series. Vol. 1364, No. 1. IOP Publishing, 2019.
- [5] Sardi, Lamyae, Ali Idri, and José Luis Fernández-Alemán. "A systematic review of gamification in e-Health." Journal of biomedical informatics 71 (2017)
- [6] Noorbehbahani, Fakhroddin, Fereshteh Salehi, and Reza Jafar Zadeh. "A systematic mapping study on gamification applied to e-marketing." Journal of Research in Interactive Marketing (2019).
- [7] McGugan, Will. Beginning game development with Python and Pygame: from novice to professional. Apress, 2007.
- [8] Tanriseven, Isil. "A tool that can be effective in the self-regulated learning of pre-service teachers: The mind map." Australian Journal of Teacher Education (Online) 39.1 (2014): 83-98.
- [9] "How to Make Flappy Bird Game in Pygame?" GeeksforGeeks, GeeksforGeeks, 31 Aug. 2021,
- [10] Alp, Elit Cenik, and Mehmet Serdar Guzel. "Playing flappy bird via asynchronous advantage actor critic algorithm." arXiv preprint arXiv:1907.03098 (2019).