International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 09 | Sept - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

HIGH_FLY GAME USING PYTHON AND PYGAME

Ms. Surabhi K S[1], Guru Santhosh P[2]

[1]Assistant Professor (SG), Department of Computer Application, Nehru College of Management, Coimbatore, Tamilnadu,India

[2]II MCA, Department of Computer Application, Nehru College of Management, Coimbatore, Tamilnadu, India

Abstract

This paper presents the development of a Flappy Bird clone using Python and Pygame, enhanced with multimedia features through MoviePy. The game offers user authentication, adjustable difficulty levels, and engaging multimedia effects, such as collision sound effects and video playback, to improve user experience. This implementation serves as an educational project for understanding game development concepts, user interface design, and multimedia integration. Because of the simplicity and engagement and more popular, many successful game titles have been imitated. This paper focuses on utilizing Python and the Pygame module.The Key elements of the original Flappy Bird game, such as obstacle avoidance, score monitoring, and gravity- based mobility, are all replicated in this game (High_fly). The design process for games, implementation techniques, performance evaluation, and user interaction are all covered in detail in this study

1. Introduction

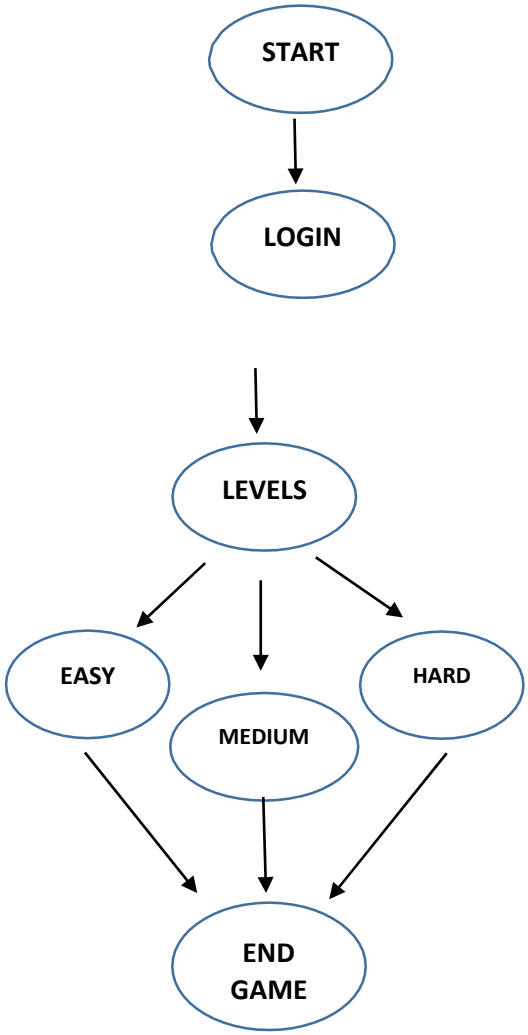
A growing number of people, both novices and experts, are interested in game development as a result of the appeal of straightforward yet addictive games. This game is inspired from Flappy Bird, which was initially developed by Dong Nguyen. In this work, a Python Flappy Bird clone 'HIGH_FLY' is designed and implemented, utilizing MoviePy for video effects and the Pygame library for visuals and music. By using multimedia to increase player engagement, the goal is to offer an approachable framework for learning game development ideas Casual gaming has become increasingly popular in recent years; Flappy Bird, which became a worldwide hit in 2014, is one example. The game's easy gameplay, challenging challenge, and straightforward mechanics are partly responsible for its success. Bird offers a difficult experience that calls both skill and exact timing.

The main objective of this project is to use Python and the Pygame library, a collection of modules made for game development, to create a Flappy Bird clone. This study aims to both recreate the fundamental mechanics of the original game and investigate potential ways to improve it by utilizing Python's versatility. Pygame is one of the most user-friendly and adaptable game creation frameworks available, which makes it a great option for beginners and enthusiasts alike. The goal of this project is to use Pygame to create a 2D bird- navigation game that incorporates key elements of gaming, like data storage, user interaction, collision detection, and scores.

2. Methodology

This project's methodology is divided into multiple crucial stages, all of which are intended to guarantee methodical development and quality control. These

phases are explained in detail in the sections that follow:



2.1. Logic Design & GameStructure

Game Loop: The main game loops through its operations, updating the screen and listening for events continuously. The two states of the game are ongoing gameplay and game over.

Event Handling: Pygame.event.get() is used to record important events, such as giving up, jumping (by pressing the space bar), and resuming the game. Based on these inputs, the game responds.

State of Game:

- **Active State:**The score increases an pipe moves across the screen when the bird passes a pipe.
- **Game Over State**: A collision video and a message informing the player that the game is over play when they crash into pipes or the screen's edges.

2.2. User Registration and Login

- **File-based User Management:**A simple credential management system is implemented using a text file to store usernames and passwords. Users can register, and their credentials are saved for future login attempts.The register() function writes the username and password to the file, while login() reads from the file to verify user credentials.

2.3. Database Management forHigh Scores

SQLite Database:

- A local SQLite database (highscores.db) is used to store the top scores for each difficulty level (easy, medium, hard).
- Three tables (easy_highscores, medium_highscores, hard_highscores) store usernamesand scores.
- The save_high_score() function inserts new scores into the respective table, while display_high_scores() retrieves anddisplays the top 5 scores.

2.4. Level Design & Difficulty

- **Level Selection**:Players can choose from three difficulty levels (easy, medium, hard), which affect the gameplay in terms of pipe speed and gap size

The level_selection() function displays a menu for level selection, modifying the gameplay parameters such as pipe_gap and pipe_speed based on the chosen difficulty.

2.5. Media and Assets

- **Sound Effects**: The game plays sound effects such as collision_sound on collisions using pygame.mixer.Sound().
- **Video Playback**: A collision video (collision_video.mp4) plays when the player loses, using the moviepy library's VideoFileClip to preview the video.
- **Image Assets**: Bird images,pipe visuals, and backgrounds are loaded by pygame.image.load(), and they are appropriately scaled and displayed during the game.

2.6. Collision Detection

- **Bird-Pipe Collision:** The game checks if the bird collides with pipes or the screen's edges using the check_collision() function. If a collision is detected, the game switches to the "game over" state.

  The bird's position relative to the pipe and its height is constantly checked to determine whether the bird has passed or hit the pipe4.

2.7. Game Flow

- **Login**: The game starts by prompting the player to log in or register. This is handled by the display_login() function. Level Selection: After login, the player chooses a difficulty level. Each level affects the game dynamics such as pipe speed and gap size.
- **Gameplay**: The game loop handles all aspects of gameplay: drawing the background, bird, pipes, detecting collisions, and updating the score.
- **Game Over**: When the game ends (collision or out-of-bounds), a game over message is displayed, the player's score is saved, a video is played, and high scores are shown.

3. Screen setup

The screen setup in the Flappy Bird Clone developed using Pygame is crucial for creating an engaging gameplay experience. The process begins with defining the screen dimensions, setting the game window to a width of 400 pixels and a height of 600 pixels. This configuration isimplemented using the pygame.display.set_mode() function, establishing the area where all game graphics will be rendered.

Color definitions play a significant role in the visual aspect of the game.White serves as the background color, black is used for text and outlines, and green represents the pipes, ensuring they stand out against the background.

To maintain a smooth gameplay experience, a game clock is initialized using pygame.time.Clock(), which regulates the frame rate. The setup also involves loading background images corresponding to different difficulty levels—easy, medium, and hard. These images are loaded into memory and resized to fit the game window dimensions using pygame.transform.scale(). This scaling process guarantees that the backgrounds fill the entire screen without distortion, enhancing the visual appeal of the game.During gameplay, the selected background is rendered at the top-left corner of the screen using the blitmethod,

allowing all other game elements, such as the bird and pipes, to be drawn on top. As the game progresses, these elements are updated and displayed, creating a dynamic and interactive environment for players.Finally, the game updates the screen in each iteration of the main loop.

The pygame.display.update() function refreshes the display to show the latest graphics, while the clock regulates the frame rate, limiting the game to a maximum of 120 frames per second. This structured and efficient screen setup is essential for delivering an enjoyable and responsive gameplay experience, laying the foundation for future enhancements and modifications.

4. Results

The "HIGH_FLY" game begins by initializing Pygame for graphics and sounds, and it connects to an SQLite database to store high scores. After loading assets, players are prompted to log in or register with a username and password. Once logged in, players select a difficulty level (Easy, Medium, or Hard), which adjusts the speed and gap size of the pipes in the game. The player controls a bird that falls due to gravity and must navigate through gaps in moving pipes by pressing the spacebar to make the bird jump. If the bird collides with a pipe or goes out of bounds, the game ends, showing a game- over screen and playing a collision video. The player's score is saved, and the top five high scores are displayed. Players can then choose to restart or quit the game.

5. Conclusion

The "HIGH_FLY" game exemplifies a well-structured approach to game development, integrating multiple programming concepts and tools to create an engaging user experience. It begins with the initialization of the Pygame library, which handles graphics, sound, and user input, along with loading essential assets like images and sound effects that contribute to the game's immersive atmosphere. The game establishes a connection to an SQLite database (highscores.db), enabling persistent storage of player high scores and enhancing user engagement through a competitive element.

A crucial feature of the game is its login and registration system, allowing players to create accounts and securely track their scores. Players can log in with their credentials, and new users can register, with their details saved in a separate file. This feature personalizes the gaming experience and ensures that multiple users can participate without data overlap.

Once logged in, players select from three difficulty levels: Easy, Medium, and Hard. This selection significantly impacts gameplay dynamics, as each level adjusts parameters like pipe speed and gap size, catering to various skill levels and enhancing replayability. The core gameplay involves controlling a bird navigating through moving pipes, with players using the spacebar to make the bird jump while gravity influences its descent. This mechanic creates an engaging challenge that tests players' reflexes and timing.

Collision detection is integral to the game, determining the outcomes when the bird collides with pipes or boundaries, leading to a "Game Over" state. A game-over screen is displayed, along with a collision video that adds visual interest and signals the end of gameplay. The game features a scoring system where players earn points for successfully navigating pipes, with scores saved to the database based on difficulty. High scores are retrieved and displayed, fostering competition and encouraging players to improve their performance.Overall, the "HIGH_FLY" game successfully combines various elements of game design, from user authentication and difficulty selection to gameplay mechanics and score

management, resulting in an engaging and replayable experience that showcases fundamental programming skills and the potential for future enhancements.

6. Acknowledgement

I would like to express our sincere gratitude to all those who have supported and contributed to the development of this research on high_fly which is a python game based project.

First and foremost, I extend our appreciation to my Advisor Ms. Surabhi KS for her invaluable guidance, support, and encouragement throughout the course of this project. Her insights and expertise were instrumental in shaping the direction and quality of my research.

7. References

[1] Pygame Community, "Pygame: A set of Python modules designed for writing video games," [Online]. Available: https://www.pygame.org/.

Nguyen, D. (2013). Flappy Bird. [Mobile Game] [2] D. Nguyen, "Flappy Bird," 2013. [Mobile game]. .GEARS Studio. [Online]. Available: https://play.google.com/store/apps/details?id=com.hamburgers.flappybird.

[3] Python Software Foundation, "Python programming language," [Online]. Available: https://www.python.org/.

[4]Mike Bertha. "Everything you need to know about your new favorite cell phone game, 'Flappy Bird' ". 2014.

[5] Patrick O'Rourke . "Flappy Bird is the ultimate mobile game ripoff". 2014

[6] Phillips, Tom . "Super Mario World player transforms game into Flappy Bird". 2022.