



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING**

A Project Report

On

2D Flappy Bird Game Using Python and PyGame

Submitted By:

Arun Kumar Sauden (HCE081BEI009)

Submitted To:

Department of Electronics and Computer Engineering

Himalaya College of Engineering

Chyasal, Lalitpur

Feb-22, 2026

ACKNOWLEDGEMENT

I would like to sincerely thank everyone who has guided and supported us throughout the journey of developing this Flappy Bird game using Python and Pygame. This project would not have been possible without their encouragement, advice, and valuable suggestions.

I am especially grateful to my supervisor, Sushant Pandey, for their unwavering support and insightful guidance. Their patience, expertise, and feedback helped me tackle challenges, stay focused, and bring this project to completion.

I would also like to thank my friends and peers who shared their ideas, offered helpful discussions, and motivated me during the development process. Their input not only helped improve the game's functionality but also made the project a more enjoyable learning experience.

Any mistakes or limitations in this project are entirely my own, but the guidance and support of these individuals have played a significant role in shaping its success.

Thank you.

Arun Kumar Sauden (HCE081BEI009)

ABSTRACT

This project presents the development of a 2D Flappy Bird game using Python and Pygame. The game implements core mechanics such as bird movement with gravity, pipe obstacles, collision detection, scoring, and multiple difficulty levels.

Players control the bird's flapping to navigate through pipes, while difficulty settings adjust pipe speed and frequency for varied challenges. The game also includes a menu, restart option, and game-over detection to enhance the user experience.

This project demonstrates the use of Python programming concepts and Pygame for real-time animation, event handling, and interactive graphics, resulting in an engaging and fully functional game.

Table of Contents

ACKNOWLEDGMENT	i
ABSTRACT	ii
List of Figures	v
1. Introduction	1
1.1 Background Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Scope	2
2. Literature Review	3
2.1 Overview of 2D Game Development	3
2.2 Computer Graphics Concepts in Game Design	3
2.3 Python in Game Development	3
2.4 Pygame Library	3
2.5 Game Mechanics of Flappy Bird	3
3. Methodology	4
3.1 Requirement Analysis	4
3.2 Graphics and Animationn	4
3.3 Collision Detection and Game Logic	4
3.4 Transformation Composition	4
3.5 Testing and Debugging	4
3.6 Screenshots / Figures	4
4. Result and Analysis	8
4.1 Game Performance	8
4.2 Score System	8
4.2.1 Score Tracking	8
4.2.2 High Score	8
4.3 Collision Detection	8
4.3.1 Pipe Collision	8
4.3.2 Ground Collision	8
4.4 Menu and User Interaction	8
4.5 Performance and Discussion	9
4.6 Limitation	9
5. Conclusion and Future Enhancement	10

5.1	Conclusion	10
5.2	Future Enhancement	10
A.	Appendices	11
A.1	Key Controls (Example)	11
A.2	Additional Screenshot	11
	Bibliography	13

List of Figures

Figure 3.1	Main Menu of The Game	5
Figure 3.2	While Playing	6
Figure 3.3	When Flappy Bird Crash	7
Figure A.1	Playing in Medium Difficulty	11
Figure A.2	Playing in Hard diffuculty	12

1. INTRODUCTION

Flappy Bird is a popular 2D side-scrolling game in which the player controls a bird that must navigate through a series of pipes without colliding. The simplicity of its mechanics combined with its challenging gameplay makes it an ideal project for learning and demonstrating programming and game development skills. This project focuses on developing a Flappy Bird game using Python and the Pygame library. Python provides an accessible and flexible programming environment, while Pygame offers tools for handling graphics, user input, and real-time game interactions.

1.1 Background Introduction

Flappy Bird is a simple yet challenging 2D game where the player controls a bird navigating through pipes without collision. Its popularity and straightforward mechanics make it ideal for learning and demonstrating programming and game development concepts. This project uses Python and Pygame to recreate the game, implementing core features such as bird movement with gravity, pipe generation, collision detection, scoring, and difficulty levels. It provides hands-on experience with object-oriented programming, event handling, real-time updates, and game logic, combining technical skills with creative design.

1.2 Motivation

The desire to gain more knowledge about computer graphics and game development is the driving force behind this project. A straightforward but engaging platform for learning concepts like animation, sprite management, collision detection, and real-time rendering is provided by the game Flappy Bird. This project offers an opportunity to learn more about smooth motion, graphical object design, and the importance of visual feedback in producing an engaging gaming experience through the recreation of this game using Python and the Pygame library.

1.3 Objectives

The main objectives of the project are listed below:

- To develop a 2D interactive Flappy Bird game using Python and Pygame.
- To implement core computer graphics concepts such as animation, sprite handling, and collision detection.
- To provide multiple difficulty levels and a responsive game interface with menu

and restart features.

- To enhance practical understanding of real-time rendering, event-driven programming, and graphical design principles in game development.

1.4 Scope

This project covers fundamental CG concepts used in 2D rendering:

- To showcase key computer graphics concepts like animation, sprites, and collision detection in a simple game.
- To build a fun and interactive 2D game using Python and Pygame.
- To gain hands-on experience in real-time rendering and interactive game design while making the project engaging.

2. LITERATURE REVIEW

2.1 Overview of 2D Game Development

2D games are widely used in learning environments due to their simplicity and effectiveness in demonstrating fundamental programming and graphics concepts. They focus on sprite-based rendering, basic physics, and event-driven interaction. Developing a 2D game like Flappy Bird helps in understanding the structure of game loops, object movement, and user interaction in real-time systems.

2.2 Computer Graphics Concepts in Game Design

Computer graphics plays a crucial role in game development. Concepts such as animation, sprite handling, collision detection, and rendering pipelines are essential in creating interactive environments. In 2D graphics, objects are drawn frame by frame to create smooth motion, which enhances visual experience and gameplay responsiveness.

2.3 Python in Game Development

Python is a high-level programming language known for its readability and simplicity. It is widely used in educational settings for teaching programming concepts. Python supports object-oriented programming and event handling, making it suitable for developing small to medium-scale games efficiently.

2.4 Pygame Library

Pygame is an open-source library built on the SDL (Simple DirectMedia Layer) framework. It provides modules for graphics rendering, sound handling, user input, and collision detection. Pygame simplifies the process of creating 2D games by offering ready-to-use tools for animation and real-time interaction.

2.5 Game Mechanics of Flappy Bird

Flappy Bird is based on simple mechanics involving gravity, upward thrust, obstacle generation, and collision detection. Despite its simplicity, it requires precise timing and coordination. Recreating this game helps in understanding how physics simulation and graphical feedback combine to create an engaging gaming experience.

3. METHODOLOGY

3.1 Requirement Analysis

The first step involved identifying the essential features of the Flappy Bird game, including bird movement with gravity, pipe obstacle generation, collision detection, scoring system, difficulty levels, and menu functionality. The overall objective was to create a smooth and interactive 2D game demonstrating computer graphics concepts.

3.2 Graphics and Animation

Sprite images were used for the bird, background, ground, and pipes. Frame-by-frame updates were applied to create smooth animation. Pygame's rendering functions were used to display graphical elements and manage screen updates.

3.3 Collision Detection and Game Logic

Collision detection was implemented using rectangular bounding boxes to detect contact between the bird and pipes or ground. Game-over conditions, restart functionality, and difficulty adjustments were incorporated into the logic.

3.4 Transformation Composition

Multiple transformations can be combined by multiplying matrices in order. This makes interactive controls simple: pressing keys updates a transformation matrix that is applied to all vertices.

3.5 Testing and Debugging

The game was tested multiple times to identify errors related to physics behavior, collision handling, and performance. Necessary corrections were made to ensure stable gameplay and proper functionality.

3.6 Screenshots / Figures

Add your actual screenshots to the `images/` folder and update filenames below.

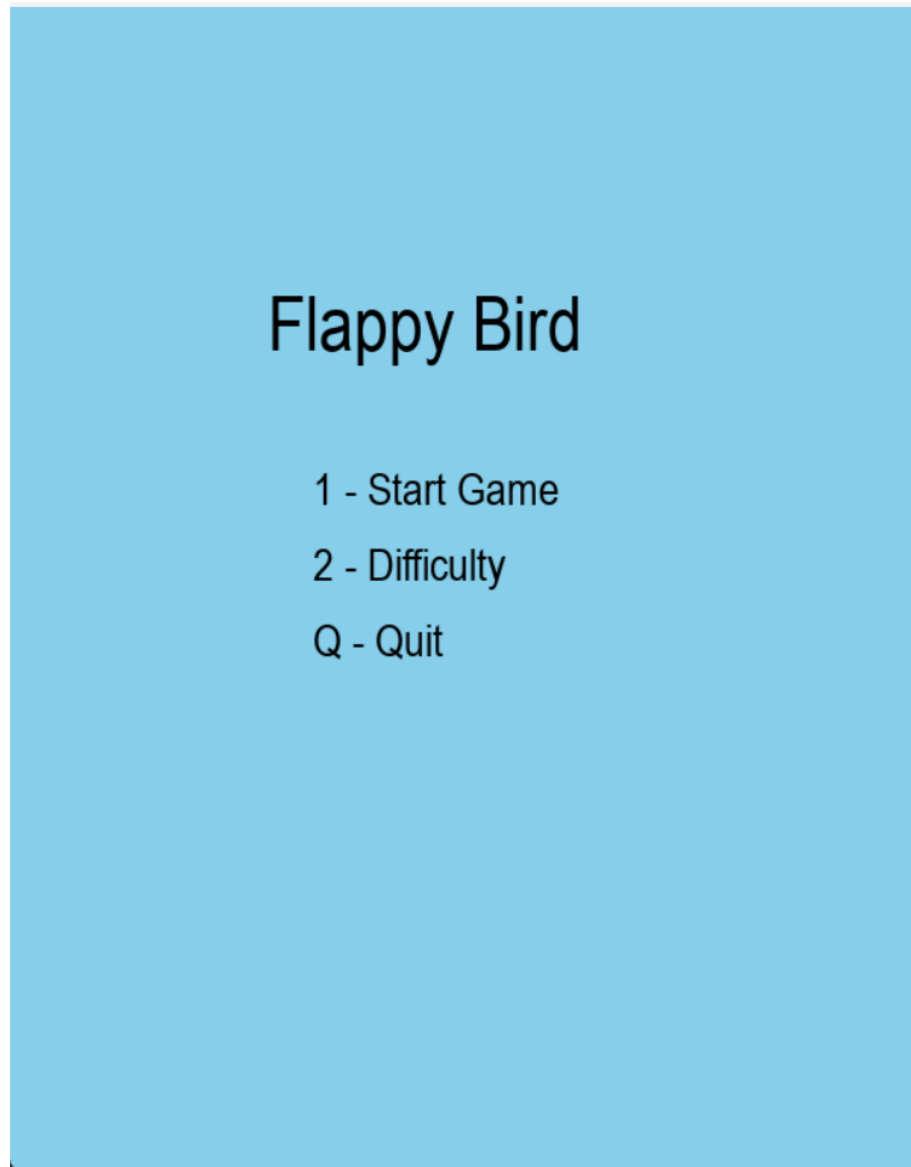


Figure 3.1: Main Menu of The Game

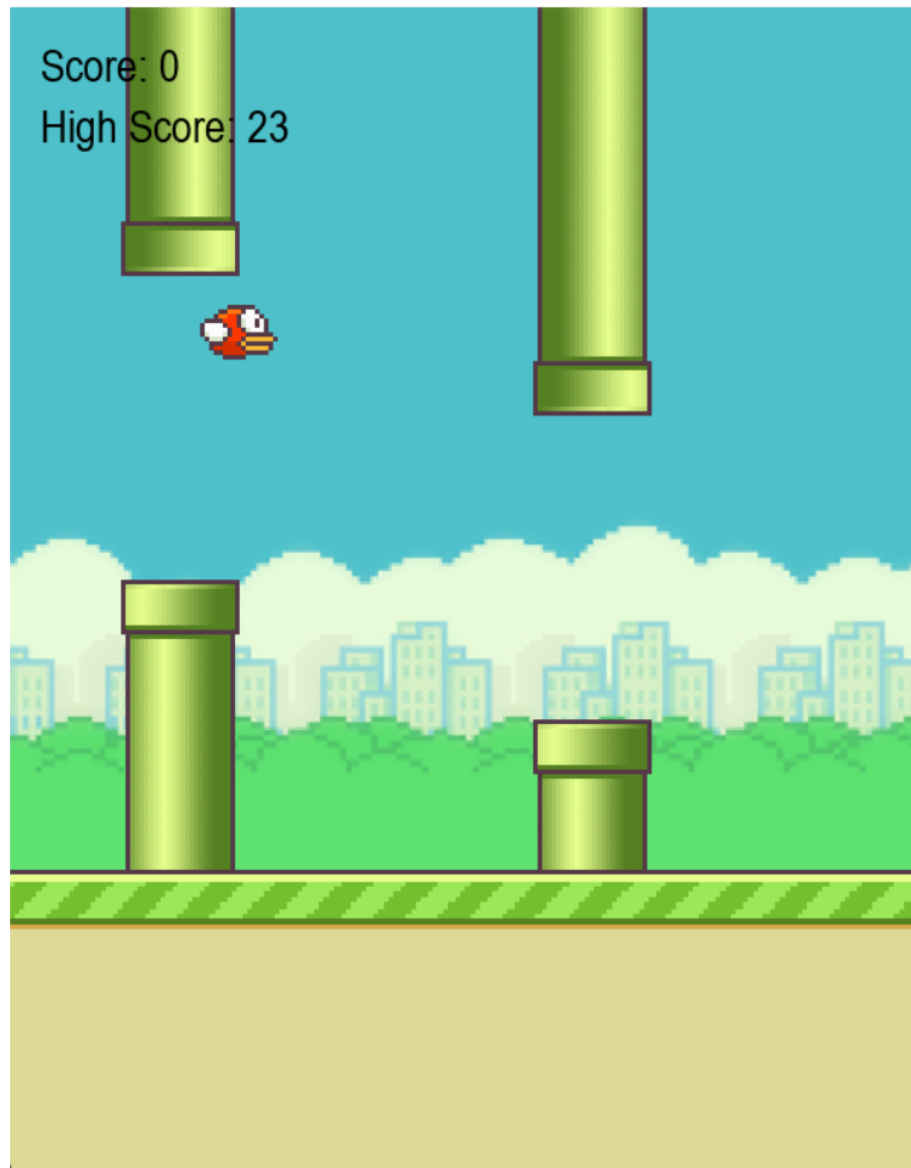


Figure 3.2: While Playing

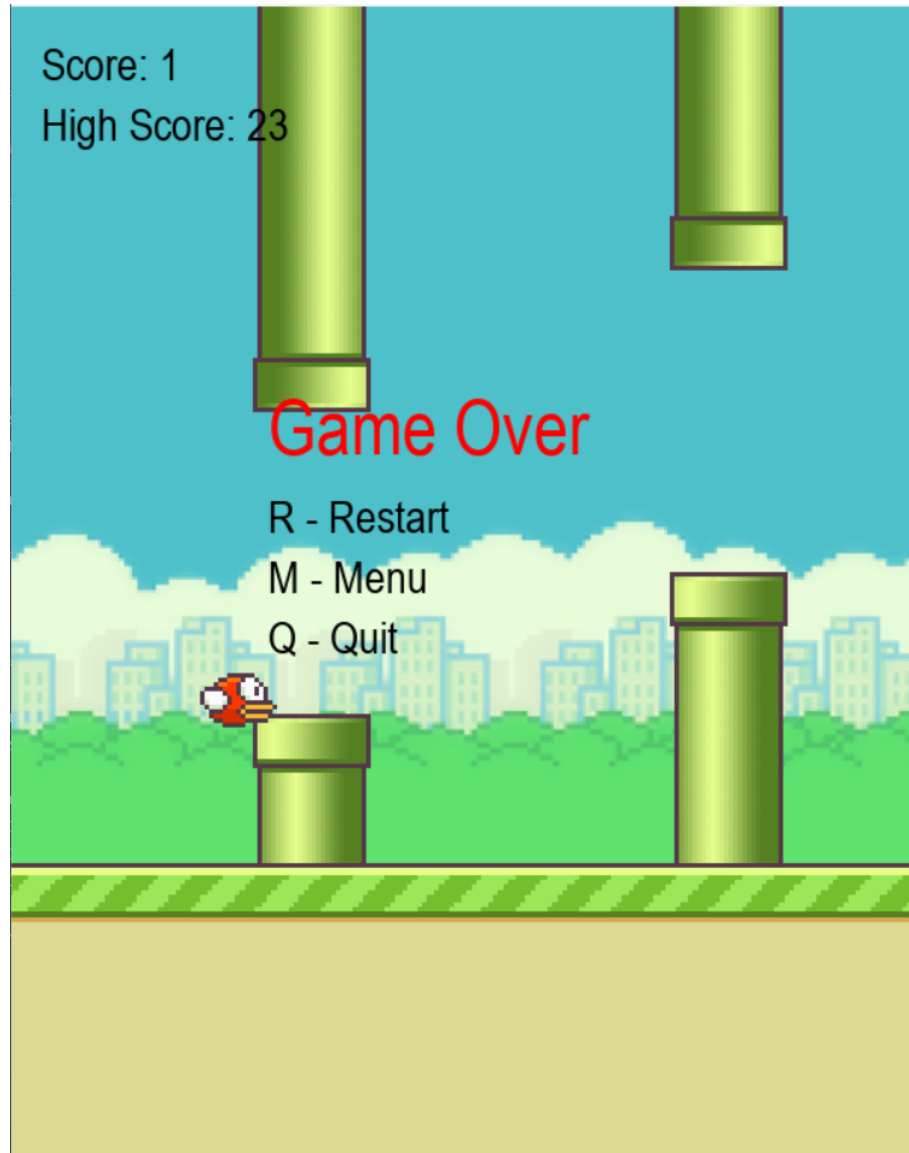


Figure 3.3: When Flappy Bird Crash

4. RESULT AND ANALYSIS

4.1 Game Performance

The Flappy Bird game developed using Python and Pygame runs smoothly at 60 FPS, providing responsive controls for the bird. The gravity and flap mechanics simulate realistic motion. Difficulty levels: Easy, Medium, and Hard affect pipe speed and spawn rate, giving players varied experiences suitable for both beginners and advanced users.

4.2 Score System

4.2.1 Score Tracking

The game accurately increments the player's score whenever the bird passes between pipes. This provides immediate feedback and allows players to track progress during gameplay.

4.2.2 High Score

The high score is stored persistently in a text file (`highscore.txt`) and is retrieved when the game restarts. This encourages replayability and motivates players to improve their performance.

4.3 Collision Detection

4.3.1 Pipe Collision

The game detects collisions with pipes accurately. Upon collision, the bird stops moving, the game ends, and the death sound (`dead.mp3`) is played.

4.3.2 Ground Collision

If the bird touches the ground, the game over state is triggered immediately. The bird's motion stops, preventing it from penetrating the ground, ensuring fair gameplay mechanics.

4.4 Menu and User Interaction

The main menu and difficulty menu provide a simple, intuitive interface. Players can start the game, select difficulty, or quit easily. After a game over, players can restart (R) or return to the main menu (M). Menus respond reliably to user inputs and maintain the selected game settings throughout the session.

4.5 Performance and Discussion

The Flappy Bird game runs smoothly at 60 FPS, providing responsive controls and realistic bird motion under gravity. Difficulty levels effectively adjust pipe speed and spawn frequency, while the scoring and high score tracking work reliably. The menu system and intuitive inputs (spacebar, R, M) enhance user experience, making the game engaging and enjoyable for players of varying skill levels.

4.6 Limitation

- Supports only single-player mode without progressive levels.
- Basic graphics and limited sound effects may reduce visual appeal.
- Collision detection uses simple rectangular bounds, which can feel slightly imprecise.
- Only few sound is implemented; additional audio could improve gameplay feedback.

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

The Flappy Bird game developed using Python and Pygame demonstrates how 2D game mechanics, physics, and user input can be effectively implemented. The game runs smoothly with responsive controls, engaging difficulty levels, and persistent high score tracking. It highlights the capabilities of Pygame for educational and recreational projects, providing an enjoyable experience while reinforcing programming and game development concepts.

5.2 Future Enhancement

- Add multiplayer mode to enable competitive gameplay.
- High Score different for different difficulty
- Introduce more advanced levels or progressive challenges.
- Improve graphics and animations for a more visually appealing experience.
- Track detailed statistics like average score, total games played, or achievements.

A. APPENDICES

A.1 Key Controls (Example)

- **Space:** Jump
- **R:** Restart
- **Q:** Quit
- **B:** Back
- **[1,2,3]:** 1.Easy 2.Medium 3.Hard

A.2 Additional Screenshot

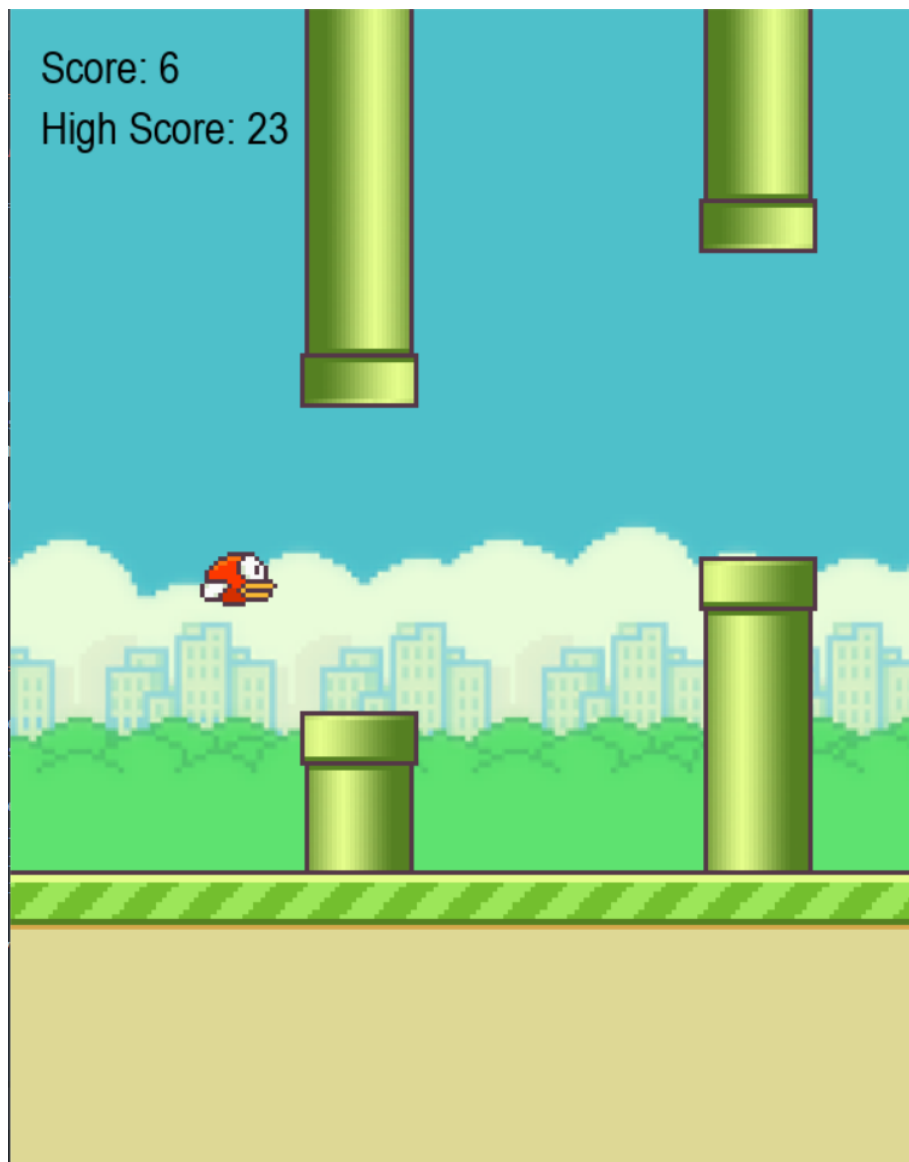


Figure A.1: Playing in Medium Difficulty

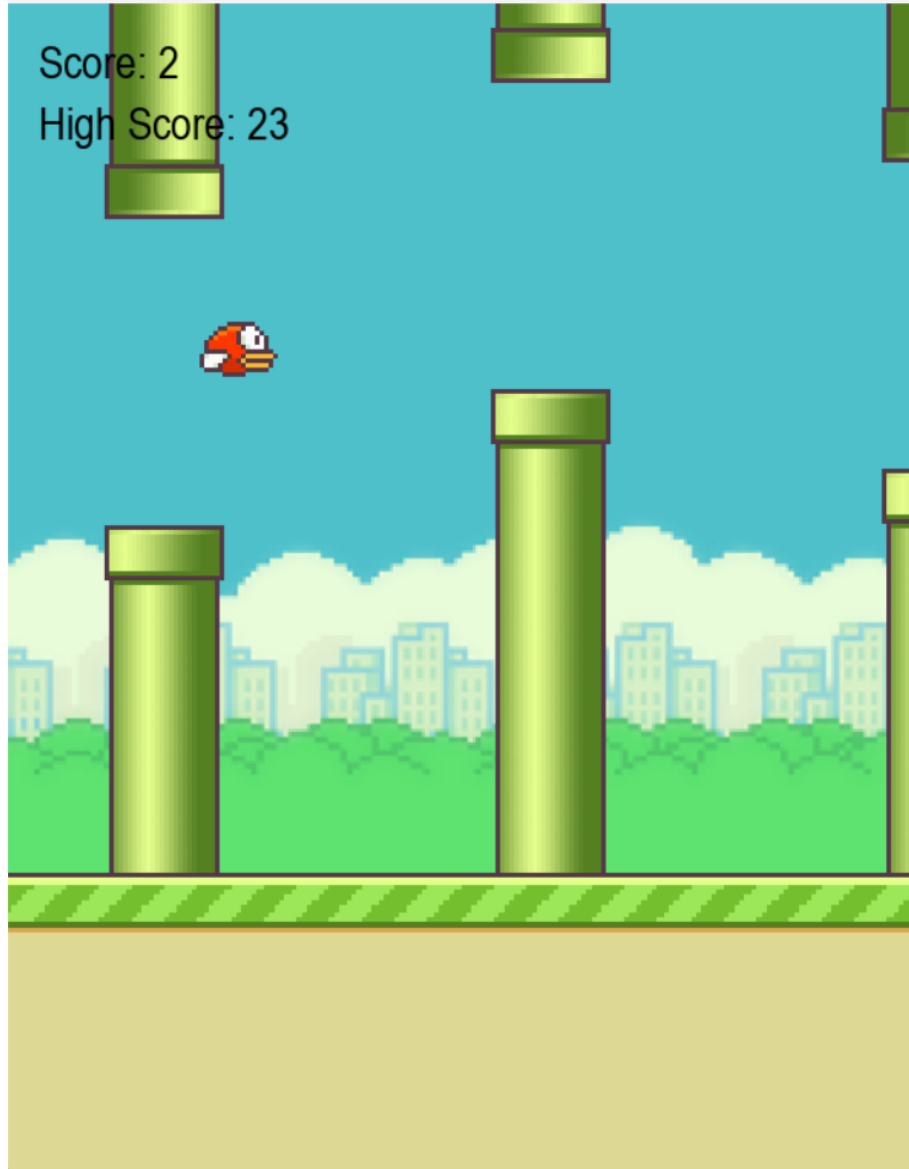


Figure A.2: Playing in Hard difficulty

Bibliography

Dong Nguyen, *Flappy Bird Game Concept*, https://en.wikipedia.org/wiki/Flappy_Bird, Accessed: 2026.

Real Python, *Creating Games with Pygame*, <https://realpython.com/pygame-a-primer/>, Accessed: 2026.

Gameplay Artist, *Flappy Bird in Python Pygame #1 – Game Setup and Basics*, YouTube video, available at <https://www.youtube.com/watch?v=XON1IA8MxQw&list=PLz7mmheDeooW795HsNzkLS6TFQSa4>, accessed February 2026.