

ANGRY BIRDS 2_PLAYERGAME

Maradana Sai Siva Lochan

Spring 2024-2025

Abstract

This report outlines the development process of creating the game: Angry Birds: Two-Player Showdown, a 2D physics-based projectile game built using Python and Pygame. It covers the game's concept, mechanics, design, implementation structure, and the challenges encountered throughout development..

Contents

1	Introduction	2
2	Modules	2
3	Directory Structure	2
4	Running Instructions and Prerequisities	3
5	Game Play!	3
5.1	Game Navigation	3
5.1.1	Intro Screen	3
5.1.2	Main Menu	3
5.1.3	The Game!	4
5.1.4	Game Over	4
5.1.5	Quit	5
6	My Code	5
6.1	Customization in the Game	5

1 Introduction

The aim of this game is to destroy the opposing player's block structures using physics-based projectiles. Each player takes turns launching birds from their respective catapults, attempting to inflict maximum damage on the opponent's defense.

Victory is based on strategic aim, timing, and efficient destruction. While no traditional score counter exists, performance can be measured by factors such as number of successful hits, remaining blocks, or number of turns taken to dismantle the opponent's setup.

2 Modules

The external modules used are:

- **pygame** - pygame is a set of Python modules designed for writing video games.
- **Random** - A module that implements pseudo-random number generators for various distributions.
- **Sys** - A module that provides access to some variables used or maintained by the interpreter and to functions that interact with the interpreter.
- **Time** - A module that provides various time-related functions.
- **os** - A module that provides a portable way of using operating system-dependent functionality.
- **math** - A module that provides access to mathematical functions such as trigonometry, logarithms, exponentials, and constants like π . It was used to handle vector calculations and angle measurements in projectile motion and collision handling.

3 Directory Structure

The project directory has:

1. menu.py
2. main.py
3. blocks.py
4. angry_birds.py
5. projectiles.py
6. Images

- **main.py** - Manages the start menu interface, player name input, and transition into the game and it will start game by running
- **game.py** - Controls the core game loop, player switching, bird launching, collision handling, and game state transitions.
- **blocks.py** - Defines the Block class and handles the drawing, positioning, health bar, and falling animation of block structures.
- **angry_birds.py** - loads bird images and sets their initial positions and display on screen.
- **projectiles.py** - Manages projectile (bird) behavior including physics (velocity, gravity), movement updates, and collision side detection.
- **Images** - contains all images related to the game

4 Running Instructions and Prerequisites

install python first

Now install `pygame` by using command "`pip install pygame`"

The file can be run by:

```
python main.py
```

```
python3 main.py
```

5 Game Play!

5.1 Game Navigation

use play button to move from intro screen to main menu and also main menu to game play screen OR you may use enter keys

5.1.1 Intro Screen

The game starts with an Intro Screen^[1]:



Figure 1: Intro Screen

5.1.2 Main Menu

Here we need to fill the names of the players (2 players) before we choose to play

We can select any of these by pressing on these

We can also use key down or upper arrows or enter key to navigate^[2].



Figure 2: Main Menu

5.1.3 The Game!

- The game is designed for **two players**, each taking alternate turns.
- On a player's turn, they can **select one bird** from the available options and **drag it using the mouse** from their catapult to adjust angle and power.
- Upon releasing the mouse, the bird is launched and follows a realistic **projectile trajectory** with gravity and collisions.
- The bird can bounce off blocks depending on which side of the block it hits.
- The objective is to **hit and destroy the opponent's block structures**.

Some examples of the game screens[3][4].

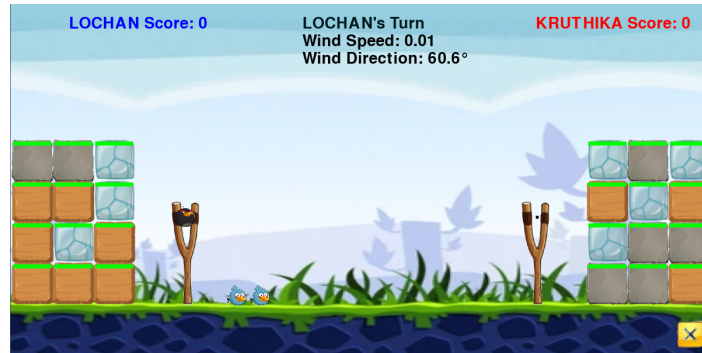


Figure 3: Game Starts!

NOTE

- Each player may launch one bird per turn.
- Blocks have a defined **health level**, which decreases on each hit.
- When the bird hits block then it reflects and falls down and goes out of screen, Here ground collision is not considered
- The game continues with alternating turns until a victory condition is met (e.g., one player's blocks are all destroyed).
- There is no formal score system; performance can be judged based on **number of hits**, **remaining structures**, or **efficiency**.

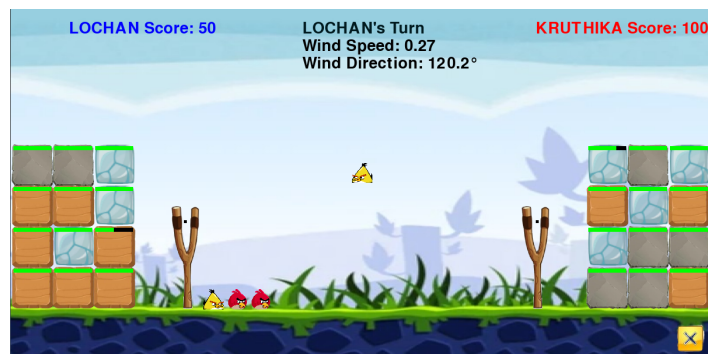


Figure 4: Game Play!

5.1.4 Game Over

The game ends when all blocks on one side are completely destroyed, declaring the opposing player as the winner. We can also end the game in between, Then the player with high score is considered as the winner The screen looks like this[5]:

Here a low varying wind will always blow and the angle of wind also varies over 360 degrees[6]:

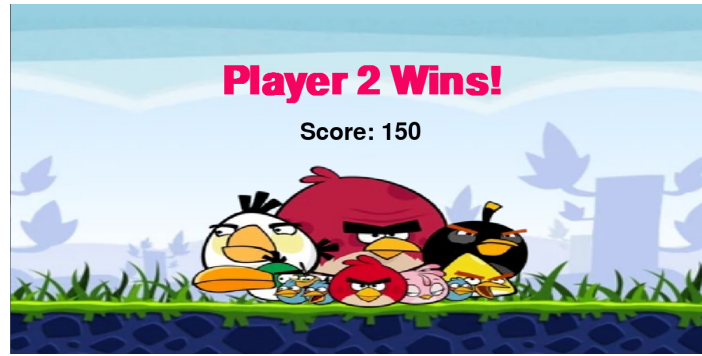


Figure 5: Game Over

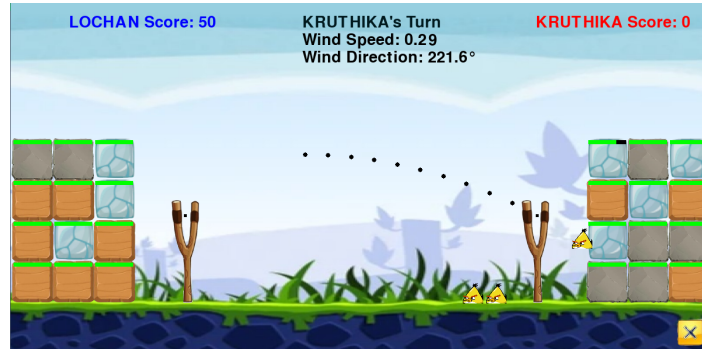


Figure 6: Fastest Solves

5.1.5 Quit

On clicking this X button, the Game ends and the program terminates.

6 My Code

The game was developed using Python and the `pygame` module, which enables real-time rendering, event handling, and sprite manipulation for interactive 2D games. The implementation centers around modular Python files, each handling a core part of the game logic.

To simulate realistic projectile motion, I used `pygame.math.Vector2` for velocity and direction vectors. Gravity is applied frame-by-frame, updating the vertical velocity of projectiles over time. When a projectile hits a block, the side of collision is detected using bounding box overlap, and its velocity vector is reflected accordingly to simulate bouncing.

Blocks have individual health bars, and when a block's health reaches zero, it is marked as destroyed.

The game uses mouse event tracking for drag-and-release aiming, switching turns between two players after each bird launch.

Throughout development, I frequently referred to the `pygame` documentation[1].

6.1 Customization in the Game

A list of all the special customization implemented in the game:

- Scores.
- Wind
- exit Button for easy navigation.
- enter keys usage

References

- [1] Pygame Official Documentation <https://www.pygame.org/docs/>.
- [2] Real angry birds game (I)(by professor) <https://github.com/estevaofon/angry-birds-python>
- [3] Real angry birds game (II)(by professor) <https://github.com/marblexu/PythonAngryBirds>