

# Angry Birds Game Project Report

Sayan Mandal

CS-104

Spring 2024-2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Modules</b>	<b>2</b>
<b>3</b>	<b>Directory Structure</b>	<b>2</b>
<b>4</b>	<b>Running Instructions</b>	<b>3</b>
4.1	Installation . . . . .	3
4.2	Execution . . . . .	3
4.3	Game Navigation and Gameplay . . . . .	3
4.3.1	Main Menu Screen . . . . .	3
4.3.2	Game Settings . . . . .	3
4.3.3	Loading Screen . . . . .	3
4.3.4	Match . . . . .	4
4.3.5	Game History . . . . .	5
Controls4Blocks4Birds4Block Damage4Scoring system5Winning Condition5		
<b>5</b>	<b>Basic and Advanced Features</b>	<b>5</b>
5.1	Basic Features . . . . .	5
5.2	Advanced Features . . . . .	5
<b>6</b>	<b>Screenshots</b>	<b>6</b>
<b>7</b>	<b>Project Journey</b>	<b>8</b>
7.1	Key Learnings . . . . .	8
7.2	Challenges and Solutions . . . . .	8
<b>8</b>	<b>Bibliography</b>	<b>8</b>

# 1 Introduction

The main objective of making this game is to extend the single-player version of the original Angry Birds game to a turn-by-turn based two-player game where each player aims to destroy the opponents fortress by launching projectile from his slingshot.

## 2 Modules

The external modules used are:

- **pygame** – Used for game rendering, event handling, and graphics management.
- **math** – Provides mathematical functions such as trigonometry, used in projectile motion calculations.
- **random** – Used to generate random positions or parameters for objects.
- **numpy** - Used to create arrays of objects and work on them

## 3 Directory Structure

The structure of the project directory is as follows:

```
.
├── Scripts
│   ├── Game.py
│   ├── MainMenu.py
│   ├── birds.py
│   ├── blocks.py
│   └── utils.py
├── Media
│   ├── Fonts
│   └── Sprites
├── Data
│   └── GameHistory.csv
├── Report
│   ├── Screenshots
│   ├── Report.tex
│   └── Report.pdf
└── main.py
```

- **Scripts** – Various scripts that manages various parts of the game
- **Media** - Contains sprites and fonts used in the game
- **Data** – Contains player scores of the last 10 games played
- **main.py** – Main game loop
- **Report** - Contains report of the project

## 4 Running Instructions

### 4.1 Installation

Python must be installed on the device prior to this. Then the pygame module must be installed using the following command:

```
pip install pygame
```

### 4.2 Execution

To run the program, open the terminal in the directory containing "main.py" and then use one of the following commands whichever is suitable.

```
python main.py  
python3 main.py
```

### 4.3 Game Navigation and Gameplay

#### 4.3.1 Main Menu Screen

The Main Menu screen is first shown as the game begins. Here, the user can choose to:

- Play
- History(See the scores of past 10 matches)
- Quit

#### 4.3.2 Game Settings

After selecting to play the game, the Game Settings screen opens where the name of the players can be entered. The following game settings can also be changed:

- **Fortress dimensions** - Change the dimensions of the fortress (rows X columns).
- **Number of birds in queue** - To specify the maximum number of birds each player gets at a time. Note that once all the birds get used up new set of birds are randomly provided.
- **Include block gravity** – Option to include gravity for blocks which acts at the end of turn of each player
- **Include predicted projectile** – Option to display the predicted projectile upon dragging the bird on slingshot

#### 4.3.3 Loading Screen

After confirming the game settings, the loading screen appears. With a countdown of 3 seconds the match starts.

#### 4.3.4 Match

The match begins with player1(left) who has to aim his projectile at player2's(right) fortress. The turn ends after the projectile comes to a halt, or falls underneath the screen or spends a considerable amount of time outside the side boundaries of the screen. Then begins the turn of player2 who has to aim his projectile at player1's fortress and the game continues.

#### Controls

- Click and drag to stretch the slingshot.
- Release the mouse button to launch the projectile.

#### Blocks

The following table describes the type of blocks in the game:

Block Type	Weaker to	Number of Images
Wood	Red,Chuck	4
Stone	Red,Bomb	4
Ice	Red,Blue	4

Table 1: Types of Blocks

Blocks are randomly generated to create the fortress. Each player's fortress is the mirror image of the other to maintain fairness.

#### Birds

The following table describes the type of projectiles/birds in the game:

Bird Type	Strength
Red	Wood,Stone,Ice
Chuck	Wood
Bomb	Stone
Blue	Ice

Table 2: Types of Birds

Birds in the queue are randomly generated for each players separately. Birds in queue closer to the center of the screen are picked up first. Images of the Birds are taken from internet sources.

#### Block Damage

The damage that the projectile does to the blocks it hits is proportional to its normal velocity to the block. If the block is destroyed then the projectile losses some amount of its velocity, whereas it bounces with a velocity having a reduced normal component.

## Scoring system

Scores at each turn are assigned to the player based on the following equation:

$$score = \frac{damage \cdot health}{1000} + 3 \cdot B \quad (1)$$

where, damage = Health of the block reduced by the projectile,  
health = Health of the block before hitting it (maximum 100)

$$B = \begin{cases} 1 & \text{if projectile's first hit block is its strength and the projectile is not Red Bird} \\ 0 & \text{otherwise} \end{cases}$$

This is done to ensure reward for precise aim. Upon successfully destroying the opponent's fortress completely, the player is rewarded with an extra 20 points.

## Winning Condition

The game ends when one of the player's fortress is completely destroyed with an exception (as player1 destroys player2's fortress player2 is given a last chance before game ends as player1 starts the game first). This maintains fairness in the game.

When game ends the player with greater score wins. The scores of the players are then stored in history which can be viewed from the Main Menu. The players can then choose to either play again or return to the Main Menu.

### 4.3.5 Game History

On clicking "HISTORY" in Main Menu the names of players along with their scores are loaded for last 10 matches.

## 5 Basic and Advanced Features

### 5.1 Basic Features

- Slingshot mechanic using mouse drag.
- Basic projectile physics with parabolic motion.
- Collision detection with targets or environment.

### 5.2 Advanced Features

- Block Gravity that applies after each turn
- Predicted projectile on dragging the bird on slingshot
- Breaking the block at different extent based on its health
- Storing scores of players for last 10 matches

- Scoring system based on precision and accuracy of the destruction caused by the projectile
- Self-made background, slingshot, block and platform images

## 6 Screenshots

The following are some the screenshots from the game:



Figure 1: Title Screen

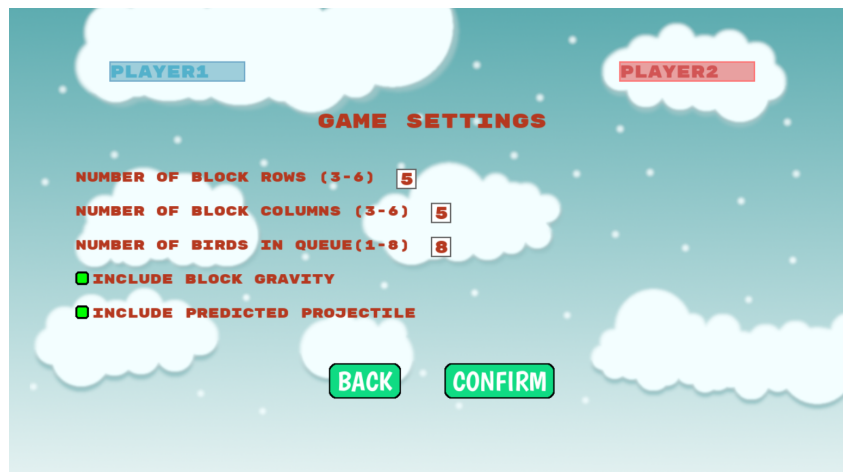


Figure 2: Game Settings Screen



Figure 3: Loading Screen



Figure 4: Game Starts



Figure 5: Game Ends



GAME HISTORY		
PLAYER1 71	v/s	114 PLAYER2
PLAYER1 308	v/s	274 PLAYER2
PLAYER1 111	v/s	117 PLAYER2
PLAYER1 91	v/s	103 PLAYER2
PLAYER1 100	v/s	115 PLAYER2
PLAYER1 263	v/s	255 PLAYER2

BACK

Figure 6: Game History

## 7 Project Journey

### 7.1 Key Learnings

- Understanding of basic physics and how to simulate motion.
- Hands-on experience with event-driven programming using Pygame.

### 7.2 Challenges and Solutions

- **Physics inaccuracies:** Fine-tuned projectile formulas and implemented frame-independent movement.
- **Collision glitches:** Adjusted hitbox size and used self-made collision detection utilities.

## 8 Bibliography

- Pygame Documentation: <https://www.pygame.org/docs/>
- Physics of Projectiles: [https://ocw.mit.edu/courses/8-01sc-classical-mechanics-fall-2016/796b8c575392e081439a7d4f10f820be/MIT8\\_01F16\\_chapter5.2.pdf](https://ocw.mit.edu/courses/8-01sc-classical-mechanics-fall-2016/796b8c575392e081439a7d4f10f820be/MIT8_01F16_chapter5.2.pdf)
- Pygame logo and Bird Images: Internet source
- Fonts :
  - <https://fonts.google.com/specimen/Boogaloo>
  - <https://fonts.google.com/specimen/Rubik+Mono+One>