

Angry Birds Game in Python with PYGAME

Sagar V
CS-104 Project

Spring 2024–25

Abstract

This report is a summary of the development process of me making the Angry Birds Game in Python using PYGAME. It consists of the game's concepts, design, implementation, and the challenges encountered during development. The report seeks to offer a detailed overview that allows for understanding and playing the game without needing direct access to its source code.

Contents

1	Introduction	2
2	Modules	2
3	Directory Structure	2
4	Running the Game	3
4.1	Prerequisites	3
4.2	Rules	3
4.3	Gameplay	4
4.3.1	Loading Screen	4
4.3.2	Main Menu	5
4.3.3	Player Naming	5
4.3.4	Theme Selection	6
4.3.5	Bird Selection	6
4.3.6	PLAY	7
4.3.7	Winner	9
5	Customization	9

1 Introduction

This is a 2 player Angry Birds game where the goal is to destroy the opponent's fortress before your own is destroyed. You achieve this by using multiple birds of your choice and one special ability. It also consists of unique themes to add to the battle. Whoever destroys the other persons fortress *first, wins.*

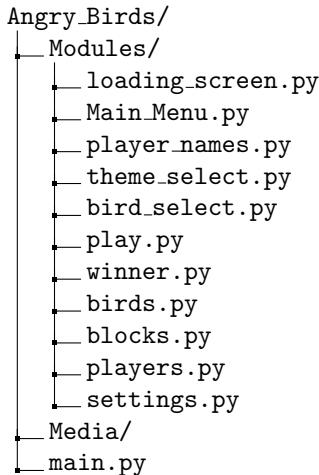
2 Modules

The external modules used are:

- **pygame-ce** - This is the basis of the entire game. It is the more frequently updated community version of the package pygame.
- **sys** - This is the package through which we control system actions, like closing the game window.
- **time** - This package provides various time related functions such as `time.sleep()` which pauses the execution for the given time
- **random** - This package is used to generate various sets of random numbers.
- **numpy** - This package is used to handle arrays. We use it particularly to store the block objects in an array
- **math** - This package provides various mathematical funtions like `math.sin()` or `math.cos()`

3 Directory Structure

The project directory is as follows:



- **main.py** – The main game loop

- **Media** – Contains the audio, image and font files
- **Modules** – Contains the core modules of the game
 - **loading_screen.py** – The module which contains and displays loading screen images and objects
 - **Main_Menu.py** – The module which contains and displays main menu images and objects
 - **player_names.py** – The module which displays the interface to enter the player names
 - **theme_select.py** – The module which displays the interface to select the theme of the game
 - **bird_select.py** – The module which displays and computes the interface to select birds of each player
 - **play.py** – The main game running module which displays the game. Contains the core mechanics of the game
 - **winner.py** – Displays the winner of the game
 - **birds.py** – Contains the **Bird** class which handles the bird objects in the game
 - **blocks.py** – contains the **Blocks** class which handles the block objects in the game
 - **players.py** – Contains the **Players** class which contains all the attributes of the player like name, score etc.
 - **settings.py** – Contains the constants and global variables used throughout the program

4 Running the Game

4.1 Prerequisites

Requires python already installed. If not installed, refer [Python download](#).

Also requires `pygame-ce` or `pygame` package. If not installed, refer [pygame docs](#) or [pygame-ce docs](#). After ensuring that both are installed, the game can be run by simply running any of the two:

```
python3 main.py
```

```
python main.py
```

4.2 Rules

- Press **ENTER** after entering each Player's name for it to be registered into the system.
- Each player can choose **upto 3 birds** of their choice. They receive any of those 3 birds by a **random algorithm** which cycles through them.
- Different birds do different damages to different blocks:
 - Red does **20 damage** to all Blocks

- Chuck does **30 damage** to **Wood** but **15 damage** to other blocks
 - Bomb does **30 damage** to **Stone** but **15 damage** to other blocks
 - Blue does **30 damage** to **Ice** but **15 damage** to other blocks
 - Stella does **40 damage** to **Block of your choice** but **10 damage** to other blocks
- If a bird goes out of bounds or slows down a lot, it does no damage.
 - Each block has 60 health. A block disappears when there is ≥ 60 **damage** done to it
 - The blocks are also randomly generated. They are the same for both players to keep it fair.
 - Each player gets a **single use** of special ability, **Mighty Eagle** in the game. Right click during bird flight before collision to activate the ability. The special ability instantly annihilates any block in its path.
 - The person who destroys the blocks first wins. If Player 1 destroys it first, however, Player 2 gets another chance to keep it fair.

4.3 Gameplay

Here we explore how it really is to play the game. The following order of subsections is the actual order of gameplay.

There are **back buttons** everywhere and **quit buttons** in some screens to enable smooth navigation:

4.3.1 Loading Screen

The game begins with a loading screen with a dynamically moving loading bar[1]



Figure 1: Loading Screen

4.3.2 Main Menu

This is the main menu from which we choose to either **PLAY** or **QUIT**[2]. There is also a mute/unmute button for easy volume control. There is also intro music which plays as soon as we enter this screen.



Figure 2: Main Menu

4.3.3 Player Naming

This is where we enter the names of the Players[3]. The selected box is highlighted by the corresponding player colours:
Yellow for Player 1
Red for Player 2

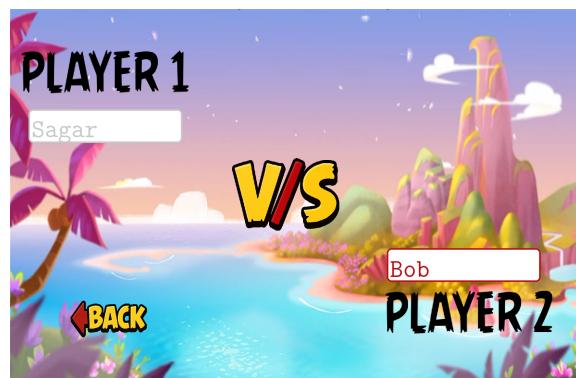


Figure 3: Naming Interface

4.3.4 Theme Selection

This is where we select the theme for the actual gameplay[4]. Hovering over a theme enlarges it with a blue border and the corresponding theme music plays.



Figure 4: Theme Select

4.3.5 Bird Selection

This is the unique bird selection menu. Each bird's information is displayed on mouse hover. Choosing it puts a tick over the bird.

The figure[5] shows when **Red** is selected and mouse is hovering over **Bomb**.



Figure 5: Bird Select

4.3.6 PLAY

This is the most comprehensive section of the game. This is where the actual gameplay happens.

This part starts off with a cool "FIGHT" animation[6]

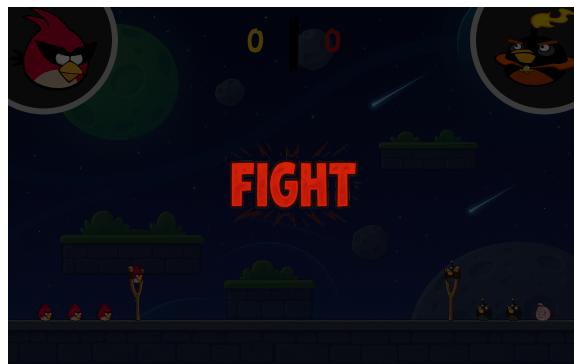
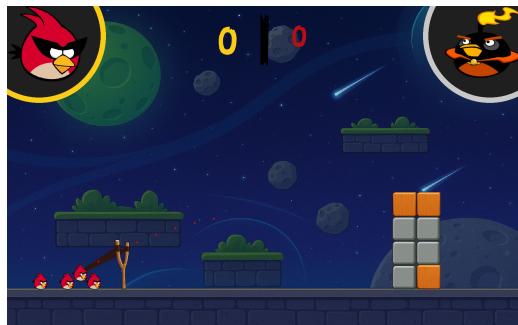


Figure 6: FIGHT Animation



(a) Player 1 Launching



(b) Player 2 Colliding

Figure 7: Gameplay Interface

A cool feature in the gameplay screen[7] is the **Player Avatar** in the top right and left of the screen.

The latest bird for launching shows up as the avatar in different styles for different themes. The Gameplay happens by a turn by turn basis, i.e., when Player 1 is launching, Player 2's slingshot and blocks are not visible, and vice versa.

In [7a], it is Player 1's turn. So, Player 1's avatar is highlighted and its score appears bigger than Player 2's. There are also tracking dots to help the players.

In [7b], the bird is colliding with block, the white highlight shows which block the bird collided with.



Figure 8: Special Ability Ghost Theme

The figure [8] shows us the special ability which can only be used once in a game by a player. It annihilates everything in its path.



Figure 9: Pause Menu

Figure [9] is the **Pause Menu** which pops up on clicking ESC key. It has volume controls and a **Quit** and **Back** button.

4.3.7 Winner

This is the Winner screen[10]. It displays the winner of the game with **confetti** and even a **crown**. It also has some great winner tunes to along with it.



Figure 10: Winner Screen

5 Customization

- Animated Buttons – size increases on hover
- Sound effects for button hover and clicks
- Menu options to change volume of background music and SFX independently
- Added a special ability, codename, **Mighty Eagle**
- Implementation of multiple themes – themed soundtracks, birds and special ability.
- A comprehensive bird select menu with further customization with respect to block damage in the form of **Stella**
- Added **Player Avatars** indicating which bird they have currently equipped and whose turn it is.
- Added slingshot, bird launch and bird collision sound effects
- Implemented a cool scorecard animation
- Custom font

References

- [1] Pygame Official Documentation <https://www.pygame.org/docs/>