

Multiplayer Angry Birds-style Game in Pygame

April 29, 2025

Modules

I have imported every library given in the list which can be used by us. The following external libraries were used in the project:

- **pygame:** The core library used to handle graphics, events, audio, and game loop functionality.
 - I used pygame as a base to develop this project. I used it to setup of the game window, load and display images and handle mouse and keyboard events and implement some corresponding functions
- **numpy:** The fundamental library for numerical and scientific computing in Python.
 - used this for using some numpy functions.
- **math:** Provides mathematical functions used for calculating trajectories and angles.
 - used this for math functions in my program for calculating trajectories and angles etc in physics related stuff
- **random:** Used to introduce variability in obstacles and environment if needed.
 - used this to randomly generate three birds each side at bottom of slings and 6 blocks each side behind the slings
- **sys:** The **sys** module in Python provides access to system-specific parameters and functions.
 - used for sys.exit
- **time:** The **time** library in Python provides functions for working with time-related tasks.
 - used to calculate time of bird in projectile to make it disappear and other stuff
- **os:** Helps with loading assets like images and sounds.
 - used this for loading images (cuz images are everything in this game)

Directory Structure

- **codes:** Contains the codes(.py files) for game
 - **main.py** : Entry point of the game, initializes Pygame, manages main loop has the while loop where game takes place and deals with screen updates. Also calls functions from other .py files
 - **birds.py** : Contains the **AngryBird** class, physics logic, selection, launching, and trajectory prediction.
 - **blocks_info.py**: Has blocks class and deals with all blocks info and has function which assign blocks behind the slings randomly
 - **input_player1.py** : has logic(code) and function that deals with taking player1 name using user interface.
 - **input_player2.py** : has logic(code) and function that deals with taking player2 name using user interface.
 - **p1_p2.py** : has logic(code) and function that deals with giving a background image and showing player 1 vs player 2 interface (it shows their names).
 - **starting_game.py** : has logic(code) that displays game intro and navigate to other screen when we press enter
- **images:** Contains all images needed for the game
- **damage_multiplier.png** : see damage matrix which has a attribute named damage multiplier(to get idea which bird has advantage over which block)

Running Instructions

1. Ensure Python and Pygame are installed.
2. Run the game using the command:

```
python3 main.py
```

3. Controls:
 - Click and drag the bird backward to aim.
 - Release mouse button to launch the bird.
 - Players take turns launching birds.
 - Just wait for a second or two after opponents bird goes to rest then you can play when it disappears
 - For seeing which bird has advantage over which block(damage multiplier) check the damage_multiplier.png in the directory

- the damage you did to the block is added to your score
- game ends after one of player completely destroys other players fortress and he get a bonus of 1500 points in score
- destroying opponents fortress is important and decides the winner in this game but to have equality we make winner as player who has more score at end.

Basic and Advanced Features

Basic Features

- Bird selection and slingshot mechanism.
- Gravity-based physics for bird flight.
- Basic collision detection with obstacles.
- Turn-based multiplayer system.

Advanced Features

- Predicted trajectory line rendered while dragging the bird.
- Smooth switching between player turns.
- Modular structure for easy extensibility.



Figure 1: starting



Figure 2: take player 1 name



Figure 3: take player 2 name

Project Journey

- **Key Learnings:** Learned to handle real-time event-driven programming with Pygame, modular coding practices, and physics-based mechanics.
- **Challenges:** Predicting trajectory accurately while bird was being dragged; handling multiple players and turns without glitches.
- **Solutions:** Implemented trajectory calculation using physics formulas. Used flags and modular structure to isolate player states.



Figure 4: display player 1 vs player 2 (you will get your name)



Figure 5: inside the game

Bibliography

- Pygame Documentation: <https://www.pygame.org/docs/>
- Angry Birds Physics: <https://www.wired.com/2012/04/angry-birds-physics/>
- reffered some youtube vedios
- GeeksforGeeks : <https://www.geeksforgeeks.org/pygame-tutorial/>
- tutorialspoint : <https://www.tutorialspoint.com/pygame/index.htm>