Angry Birds (Dual play) using PYGAME

Ahyan Hassan
CS104-Project

Spring 2024-25

Contents

1 Introduction to my Game:

In this game, two players are provided their own set of birds and fort of blocks. One who destroys other's fort will be winner of the game.

2 Modules

The external modules used are:

- `pygame-ce`
- `sys`
- `random`
- `math`

3 Directory Structure

The project directory is as follows:

```
.
├── resources
│   └── images
├── main.py
├── params.py
├── bird.py
└── block.py
```

- **main.py** - The main game loop
- **params.py** - Contains all the necessary parameters
- **bird.py** - Contains all details of birds in game
- **block.py** - Contains all details of blocks in game

4 Running Instructions

4.1 Prerequisites

Note: It is assumed that python is already installed.
The file can be run using the suitable command of two:
`python main.py`
`python3 main.py`
Ensure that `pygame-ce` is installed. If not then run
`pip install pygame-ce`

4.2 After starting

4.2.1 Intro screen

The game starts with the intro screen.

Figure 1: Intro Screen

4.2.2 Input screen

After 2 seconds, we will see the following image: Here the players need to add

Figure 2: User input Screen

their respective names and hit the play button to start the game !!

4.2.3 Main screen

After hitting play button you will see: Now you have to place the mouse pointer

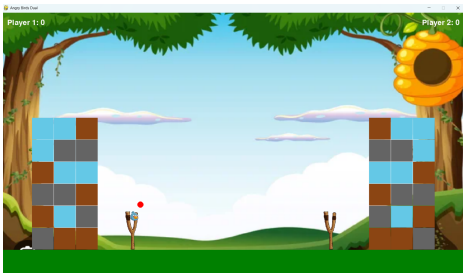

Figure 3: Game Screen

on the bird, drag and aim to enemy's blocks and release to shoot.
More you drag, more velocity you gain, upto some limit.
Players will get alternate chance.

4.2.4 Winner's screen

As soon as one player destroys complete fort of other, the game ends.
Winner will be displayed along with score as:

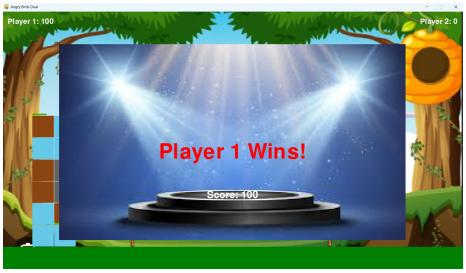

Figure 4: Winner !!

4.2.5 Play again

To play the game once more, you need to run the code again:
`python main.py` or `python3 main.py`

5 Various Implementations in the code

5.1 Blocks

I have generated 3 types of blocks using pygame.rect.
They have same hitpoints, but are damaged differently by different birds.
Three blocks are: ice, wood and stone. and all have hitpoints = 300

5.2 Birds

The game has 4 different birds that do different amount of damage depending upon block type.Birds have been generated randomly and distributed to both the player, each player gets turn simultaneously.
Details are given below:

5.2.1 Red

Does no discrimination :) gives equal damage of 100 to all the 3 types of blocks.
We can say it adjusts in every situations, while others are choosy.
That's why red is called the main hero of angry birds !!

5.2.2 Yellow (Chucks)

It loves wooden blocks the most.
Hence deals maximum damage to wooden blocks of 200 and to other blocks it does only damage of 50.

5.2.3 Blue

It loves cold icy blocks the most.
Hence deals maximum damage to ice blocks of 200 and to other blocks it does only damage of 50.

5.2.4 Black (Bomb)

It loves hard stone blocks the most.
Hence deals maximum damage to stone blocks of 200 and to other blocks it does only damage of 50.

5.3 Physical Mechanism

I have tried to make the game appear realistic.
For gravity, I have used iterative method with constant g.
For developing projectile simulated two components of bird position and bird velocity. For colliding effects, created a variable namely BOUNCE DAMPENING to reduce the velocity of bird and reverse it after colliding. Also bird vanishes after collision with block, so as to avoid crowding.

6 Project Journey

6.1 Learnings

Before this I had not much idea about pygame. But after making such game I have gained a lot of knowledge about handling various activities, bring realistic looks and also improved creativity level.

6.2 Challenges faced

Without facing challenges, no task can be completed, whether its easy or hard. I also faced various challenges for making this game. Some of them are :

- Choosing appropriate backgrounds for different screens.
- Editing images to convert them as desired (ie. remove background of elements, scaling of image etc.)
- Develop physical mechanics for bringing realistic looks.
- Generation and destruction of blocks.
- generation and distribution of birds to each player.

But I resolved these challenges upto my best.

6.2.1 Scope of improvement

As it is said "There is always scope of improvement".
I believe, if I start making game again, I'll definitely end up with certain improvements.
Some improvements I would try to make:

- Create better navigation options.
- Add more realistic appearances.

References

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

[2] Pygame CE Official Documentation
https://pyga.me/docs/.

[3] Sample Project
https://github.com/rennaMAhcuS/Maze/blob/main/report/report.tex

[4] Tutorial video
https://www.youtube.com/watch?v=AY9MnQ4x3zklist=LLindex=7