Report and User Manual

Swarupananda Dhua(22M2102) and Akshat Gautam(190110004)

November 22, 2022

Contents

Chapter 1

Report

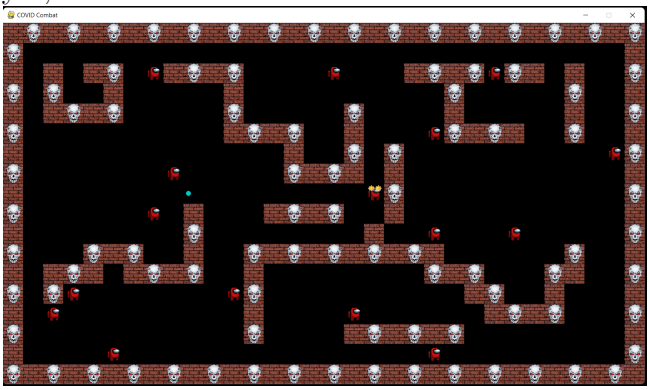
1.1 Description of the game

The name of the Game is "COVID Combat". It is a 2D game. There is a battlefield consisting of empty places and obstacles through which the active objects (Player and Enemies) of the game can walk and interact. The Player of the Game has an unlimited supply of bullets using which he can kill the enemies. On the other hand, enemies walk randomly and if the Player comes in contact with some enemy, the game is over. If all the enemies are killed, Player wins the game.

1.2 What have we achieved TODO

1. Grid for the playground has been made. 2. The Player and Coronaviruses have been added but their behaviours are not yet fully functional. 3. Sound effects (for background music and player move) have been added.

4. Collision detection between Player and Coronavirus, Shooting Bullets by the Player, Collision Detection between Bullet and Coronavirus.



1.3 Future Work

In *Camera.py* we've some experimental code using which we can cast the 2D grid as a 3D scene and convert this game to a pseudo-3D game. We can add other objects such as sky, floor, hills, trees etc. to make the game look better. Also, we can replace the Player and Enemy images by some realistic characters.

1.4 Documentation TODO

Technologies Used: Python and Pygame for coding the logic Latex for creating user manual/documentation for the game HTML/CSS for updating the status of the development.

Code structure is as follows:

- main.py -
    1. This imports the necessary python files, and also initiates the main function.
- COVID_combat.py
    1. Initializes the pygame module
    2. run() initializes the enemy position and direction
    3. run() also keeps on checking for user inputs (arrow keys)
- Battlefield.py
    1. draw the grid array by intializing a array
    2. render_2d_grid() draws the grids on the canvas
- Player.py
    1. update(enemies) - updates the bullet positions and checks collision between bullet and enemies. Based on collision check, enemies are set to dead if required
    2. change_position(delta_position) - updates player position based on the arrow key clicked.If new position contains an obstacle, position is not updated.
    3. shoot() - shoots a bullet in the direction in which Player is standing.
    4. rotate_viewpoint() – rotates the field of vision of the Player based on mouse events.This will be used in the 3D version of the game.
- Enemy.py
    1. update() - updates position based on random movement. Also checks wall collisions
- Camera.py
    1. take_snapshot() – The camera at the player's position casts rays to the objects in front of the player within field of vision.

1.5 Running Instructions

To run this game, you need *python*3 and *pygame*. Installation instructions for *python*3 can be found in *www.python.org* and for *pygame* in *www.pygame.org*. Once installed, the game can be run by invoking *pythonmain.py* from the root directory.