

Software Lab Project Report

Swarupananda (22M2102) and Akshat (190110004)

November 22, 2022

1 Description of the game

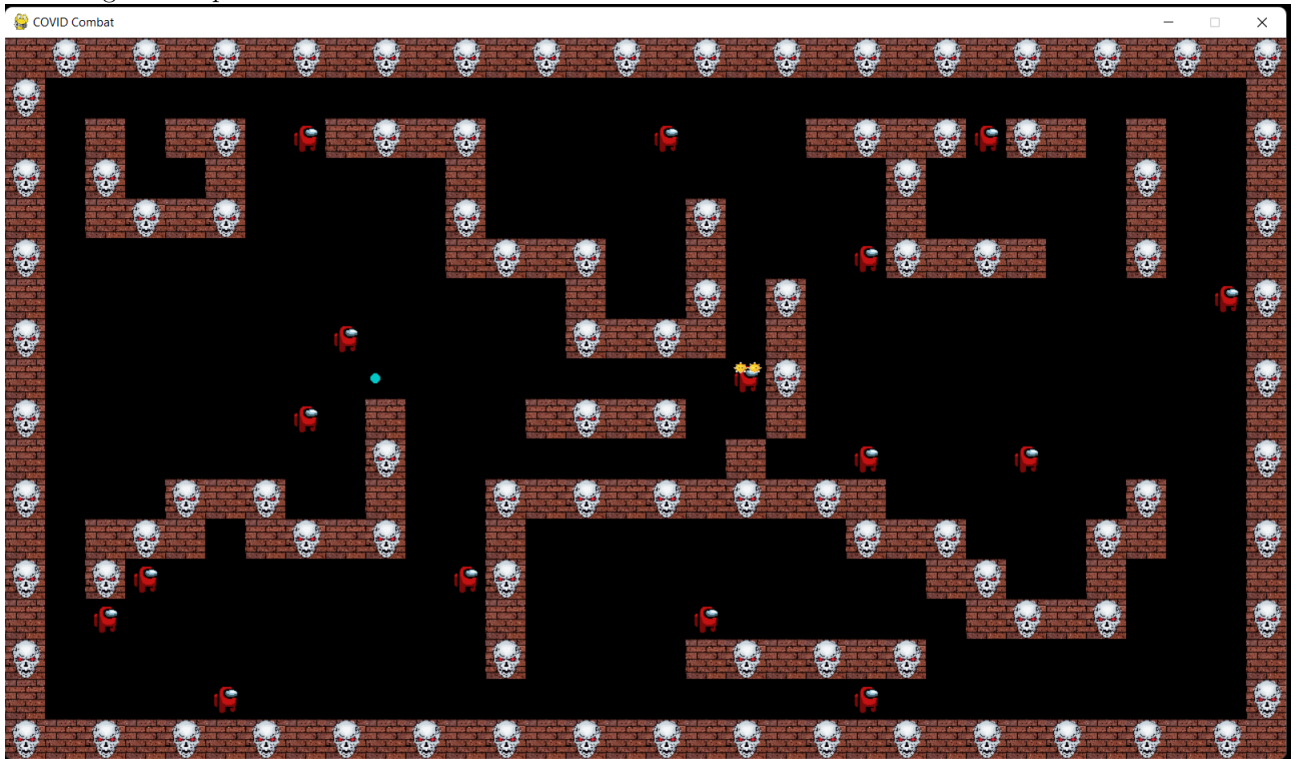
The name of the Game is “COVID Combat”. It is a 2D game. There is a battlefield consisting of empty places and obstacles through which the active objects (Player and Enemies) of the game can walk and interact. The Player of the Game has an unlimited supply of bullets using which he can kill the enemies. 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.

2 What have we achieved

We’ve made the 2D version of the game fully functional. This includes rendering the Battlefield i.e. the grid of obstacles onto the canvas. The Player and enemies have been added and their behaviors are all working as expected. Sound effects for background music and shooting bullets have been added. Collision detection between Player and enemies, bullets and enemies have been implemented and based on the collisions we also update the characters.

Raycasting for the 3D version of the game contains bug and not working properly. This needs some more work.

Following is a snapshot of the 2d version.



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.

4 High Level Algorithm

We first initialize the *enemies* array with certain number of active enemies. Then in an infinite *while* loop we do the following things:

1. Sleep for sometime
2. Check if the game is own or over or still running
3. Process the pygame event queue for user inputs
4. Update and draw the Player
5. Draw the grid
6. Update and draw the alive enemies
7. Flip the canvas

5 Documentation

Technologies Used

1. Python and Pygame: For coding the logic of the game
2. Latex: For creating user manual/documentation for the game
3. HTML-CSS: A relevant website to give details about the game.

Code structure

- main.py
 1. This imports the necessary python files and calls the `COVID_Combat.run()` method to start the game.
- COVID.combat.py
 1. Initializes the pygame module
 2. `run()` : Main game loop. It keeps on checking for user inputs (arrow keys for movements and space key for shooting bullets) and updates the canvas.
- Battlefield.py
 1. `render_2d_grid()` : Draws the grid on the canvas in 2D mode
- 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 the Player is currently 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 and projects them onto the screen in 3D mode.

6 Running Instructions

To run this game, you need *python3* and *pygame*. Installation instructions for *python3* 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. Use *up, down, left, right* keys for movements and *space* key for shooting a bullet.