

Mini Project 4 – Write Up and Reflection

Everardo Gonzalez & Vienna Scheyer

Project Overview

For this project, we created a game in which the player has to reach the safe zone on the opposite side of the screen without getting caught by zombies. The zombies and players start at random locations on the screen, but the player always starts on the right side of the screen and the zombies always start on the left side of the screen (near the safe zone). The zombies always follow the player to make the game more difficult.

Results

The player is the red rectangle, the safe zone is the green rectangle, and the other four sprites are the zombies. Our game tries to induce anxiety in those who play it by limiting the control that the user has over the main player. The user has to press keys repeatedly to make the player move to outrun the zombies. The intense music adds to the stressfulness of the game.

This is a screenshot of the game in action:



Here is a screenshot of the win screen:



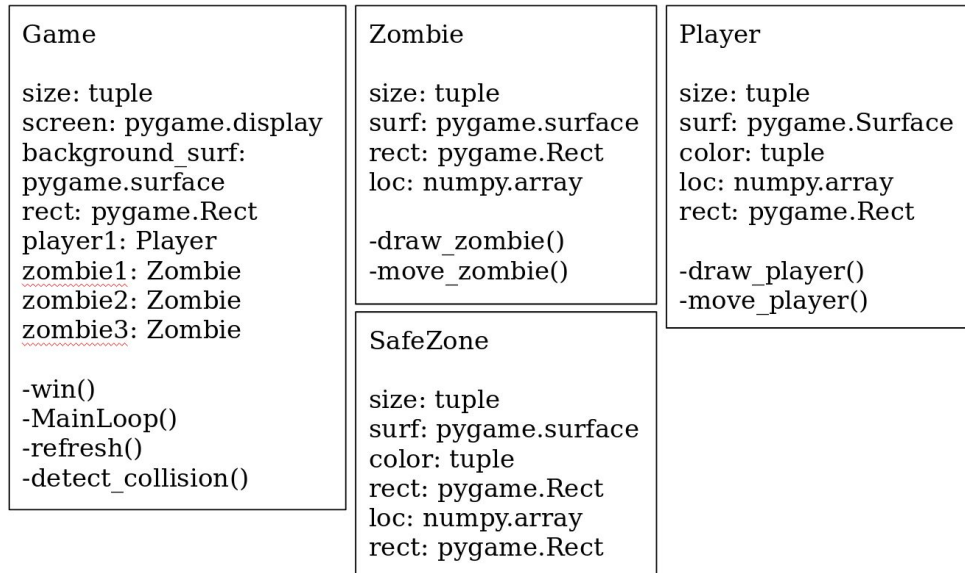
When the player reaches the green rectangle the game detects the collision, covers the sprites with the background, and displays the “You Win” text. If the player gets caught by a zombie, the game restarts. In other words, the game keeps running until the player wins.

Implementation

Our program consists of a Game class, a Player class, a Zombie class, and a Main Loop. In the Player and Zombie classes, we also have move methods that control the sprite movements. We initialize one player and 3 zombie objects. In the Game class, we create the background. We have a refresh method in the Game class as well, and we call this method from the Main Loop.

The player is controlled by user keyboard input, and the zombies are controlled by numpy vectors that constantly point towards the player. We create the vectors using the randomly generated zombie start positions relative to the player, and then convert these vectors to unit vectors. Each zombie is set at a speed when we initialize it, and then we use the unit vectors to determine the direction. We originally tried to make our own vectors, but this proved to be more complicated than we anticipated. We decided to use numpy vectors to make the zombie movement work better.

UML Class Diagram



Reflection

We met our goals of practicing pair programming and working on object oriented programming. We both have a better understanding of what object oriented programming is and we see how it can be helpful. We did nearly all of the programming together, which helped us agree on programming and design choices.

We realize that it would be beneficial to go back and improve the structure of our code. We focused on functionality and ran out of time to implement inheritance, but we can see how inheritance would be useful. In hindsight, it would have been a good idea to make a sprite group for the zombies to make the code more organized. We hope to modify this code to use inheritance after this project is over because we enjoy working on this game and we want to continue to improve this code.

User Feedback:

"This is so stressful!"
"Why did you do this to me?"
"This is so addicting!"