ELIE FRUNGORTS

**PRESS R TO PLAY THE NEXT ROUND**

The slightly useless story of my games creation (post creation)

**THE MENU**

The original plan for me was to create a neat looking menu screen with a reasonably functional game, throughout my time exploring all the possibilities of a menu, I quickly realized that creating each component would be far too much of a hassle and I quickly settled for a much easier JButton option, after many hours of working on how to get the menu to function and trying to minimize the booleans in the code, I quickly realized that a full page of booleans was too much. So I started exploring. What I quickly realized was that enumerated data types are in the most practical sense, a multi optional boolean. Put simply, the enum was a life saver. From having a 15-20 different booleans, I quickly simplified it to 3 and my menu was complete. Unfortunately, I was unable to implement high functioning menus as they were too much of a hassle and time was quickly running out.

**THE GAME**

Originally, my plan was to create a slow moving zombie game oriented around defending a building from incoming zombies. The main problem with implementing such a game was the issue of pathing. Each zombie would have to path from a random location anywhere outside the building to the nearest gate and then proceed into the building. Quickly realizing the problem, I decided to just try spawning zombies in the 4 corners of the screen. Upon realizing that direct pathing would likely be easier and much less of a hassle, I started my adventure. Originally, I tried to use trigonometric functions but realized that the calculations were very complex and slightly overwhelming. I quickly thought back to my grade 10 and 11 math classes and realized that similar triangles could easily do the same thing but without overwhelming the processor with unnecessary calculations. Now, I couldn’t get zombie spawning to work unless I had a way of modifying the game state. Luckily, enum saved the day. By creating an enumerated data type called game state I was able to seamlessly introduce pausing and game play without using an assortment of different Booleans. Now that I had the game switch from paused to unpaused, I was ready to spawn my first zombie. When I spawned the first zombie, two things were clear: 1. the calculations where faulty, and 2. the zombie would move very strangely when it overlapped with the main character. To fix the problem, I simply made impossible for the zombie to collide/overlap with the main character and thus, fixed the issues nearly instantly. Next was the issue of bullets. Thinking for a while, I thought of how I could add bullets into the game, especially when there was more than one type of weapon. I realized I could make the guns selectable using keys and thus, the magic of bullets was in the game. To introduce the ammo into the game, I made it such that, upon completing a wave, the game would randomly give the player an assortment of bullets. Now, to implement the waves, I created yet another enum that checked the in game state within the game.

using this I was able to add special waves that could be used to as rest and refresh round for those who might be running low on ammo or need a little rest period. I then realized I could have the player use a hotkey to start the next round, thus maximizing the rest periods. Unfortunately, I did not document that fact very clearly and when showing my friends my game over the internet, they all thought it was broken due to the lack of continuation to the next round. Luckily I told them to Press R to play the next round, and they all played it.

**THE END**

Long story short, enums save lives and documenting controls saves me a lot of trouble.