**Final Project README**

My final project replaces the first two questions of my board game from my midterm project with two fun minigames that the player must complete to move on to the next question/minigame of my board game. The overall purpose of my board game is to still reach the end position, but with the addition of minigames to give the player more hands-on fun with their keyboard and mouse.

The minigame that replaced the first question of my board game is the Computer Catching minigame, and all the programs I used for it are under the sprites, Minigame 2 and Minigame 2(2). In this minigame, a player tries their best to catch (click) on as many of the computers falling from the top of the screen as they can, and each computer they click on earns them one point. However, there is a burning computer that the player needs to avoid catching as it decreases their points by 1. In order to beat this minigame, the player must get over 10 points in 20 seconds. If the player can’t get to 10 points in 20 seconds, then they lose the minigame, and they also lose the whole game. Furthermore, the purpose of the Minigame 2 list was to add the items (normal computer or burning computer) that the player caught into this list. The purpose of the if conditions in my program is to sense if the player was able to complete the minigame successfully or not. The reason why I added another sprite called Minigame 2(2) was to add the programs that allowed the burning computer to fall at the same time the normal computers were falling from the top of the screen.

Moving on, the minigame that replaced the second question of my board game is the Safety Zone minigame. All the programs that I used for it are under the sprites: Minigame 1, Minigame 1 Villain, and Safe Zone. The purpose of this minigame was to let a player control the Minigame 1 sprite all the way to the Safe Zone sprite. However, there is a Minigame 1 Villain sprite that tries to catch the player as they move along the maze, and if this villain catches the player, the player ends up losing the game. If the player reaches the safe zone in time without being caught by the villain, the player can successfully finish the game and move on to the third question. The purpose of the right, left, up, and down blocks was to enable the player to move their sprite along the board. The purpose of using the forever loops in the Minigame 1 and Minigame 1 Villain sprites was to allow the player to keep playing the game until specific conditions were met.

In my board game, I also added some sounds to make the game feel more enjoyable. However, I noticed that it took up a lot of space, so I only added one song, which still makes the game more entertaining.

Finally, I want to talk about the complexity of my project. I was able to make my project complex by using a variety of different blocks and making them interact and work together using broadcast blocks to make the minigames flow with the rest of my board game. I used lists, global variables, sense blocks, if conditions, repeat loops, broadcast blocks, sound blocks, motion blocks, and many more. I wanted to demonstrate that even though I used a variation of blocks, I can make them work together to make fun minigames. I did this because I learned all the functions of these blocks from all the lab sessions we completed in this course and wanted to incorporate them into my project. I also wanted to state that my project is complex because it represents what I was able to achieve, understand, and learn from *Snap!*. I put a lot of effort and time into this final project, and I wanted to make the board game that I created in my midterm even better with more fun and difficult programs with the two minigames instead of the simple questions that I did previously. By changing two questions into two minigames, it took a lot of time and debugging to see if they worked, and overall, I feel proud of the work that I did for this project.