Zombie Castle Defense Game

**Description**

In this game, the objective is to defend the castle by preventing zombies from breaking down the entrance and getting inside the kingdom. There is no winning, there is only seeing how high of a score the player can get by defeating as many zombies as possible as the game is endless. The player controls an archer and shoots arrows to defeat zombies. In the first minute, only regular zombies appear, in the second, fast zombies appear, and in the third, giant zombies appear. Fast zombies travel faster than normal zombies and while giant zombies travel slower, they have more health and do more damage to the entrance. As each minute passes, the health of newly spawned zombies increases, making the game more difficult. Once the player loses, their highest score is saved, and they have the option to play again.

**Animation/Interactivity**

The zombies have walking animations and travel from the right side of the screen to the left. For interactivity, the player is able to control the path of the projectile of the arrow by using their mouse. In addition, they can keep track of money earned from defeating zombies and use it to purchase upgrades through buttons. They can reduce the time between each shot, increase the velocity, inflict more damage, and repair the gate. When each upgrade is first bought, there is a visual update to the archer and arrows. For every purchase, the price of that upgrade increases. There are also royalty-free sounds I got from zapsplat.com and pixabay.com that I used for the shooting noise, arrow impact on zombie, zombie deaths, zombie damaging entrance, and a purchasing noise.

**Problems**

A problem I had was finding the launch angle of the projectile origin to the mouse. I shortly realized that a right triangle could be formed in between these two points and looked up a YouTube video on how to find an angle in a right angle given two distances—between the mouse’s Y and the projectile origin’s Y (opposite side), and between the mouse and origin (hypotenuse). I solved the equation to get the angle.

I took a lesson on Khan Academy on 2D projectile motions and understood the equations, yet I could not find a way to implement the rotation of the projectile as it goes through the path of the equation. I thought I could find a predicted rotation when the projectile reached the end of its path and increment it based on time, but this led to extreme and unnatural rotations. In the end, I settled for a linear rotation, which becomes more noticeable the higher the velocity.

A drawing of a person with a key

Description automatically generated with medium confidence**Drawing**