Client Consultation:

- 1/31/22
    - I met with my client, Henry Westfall, to discuss a potential product he would want me to make for him. We both brainstormed what problems we had at school and how I could potentially code a solution to fix one or more of them. Finally, we stumbled upon a relatively straightforward one that would be fun to fix: he can't play the Chrome Dinosaur game on his computer because it's blocked. Thus, he proposed that I could create a game based off of the Chrome Dino game that he could play. The rules of the game would be similar to the original, but it would have different enemies and different art, at the very least.
- 2/1/22
    - I met with Henry to discuss the success criteria for this project. We decided that it would be best to have multiple ways of running the project on different platforms. Thus, we decided one of my success criteria would be having the project available on Windows 10/11, Linux, MacOS, and on browsers to make it accessible to the most amount of people possible. Next, we decided that to be faithful to the original Chrome Dino game and to enhance user experience, I could have audio play when important actions in the game occur (the game starting, when the player jumps, and when the player dies) because audio is played when those actions happen in the original Chrome Dino game. Third, we decided that there must be a way to assign a player a score based on how long they survived in-game. We decided the score would be in terms of seconds the player survived in-game (without the time the game spent paused) and that a high score would be stored within a file, which could be retrieved later. Fourth, we agreed that there must be a way for the player to lose the game. Henry decided to stay faithful to the original game and have the lose condition be if the player model hits an enemy cactus. Finally, we decided to have the game be an "endless runner" game, which means that it could potentially last forever if the player never hit a cactus and lost.
- 2/16/22
    - I met with Henry to discuss the difficulty of the final game that's produced. We decided to model it off of the original Chrome Dino game, where it starts out easy and gets harder and harder the longer the player stays alive. Initially, the speed of the cacti getting closer and closer to the player model would be slow, so it would be easy for the user to input a jump command using the keyboard to jump over the obstacle and continue. However, the longer the game progressed, the faster the cactuses would move towards the player. In order to avoid making the game impossible to play after a point, we are setting a max speed that the cacti cannot exceed. The speed is a vector with components that are floats, as opposed to integers, so they can be increased incrementally every time the score (which is the amount of seconds the game has elapsed) increases by 5 up until it hits its maximum at 75.

- 2/18/22
    - I met with Henry to discuss what exactly the enemy cacti would look like. In the original Chrome Dinosaur game, there were multiple types of cacti with different widths and different heights that presented a different challenge to the player trying to jump the dinosaur character above them. Thus, we decided to have a similar system in the final product: four different types of cacti with different lengths and different heights. Some would be easy to jump over (like the small single-spiked cactus or the large single-spiked cactus), while the others (the small triple-spiked cactus or the large double-spiked cactus) would require the player to be more precise when timing the jump to avoid hitting it. We decided that the first cactus that the player had to jump over would always be the small single-spiked cactus in order to give them a warm-up and let them get a feel of the gravity. The next cacti spawned would be randomly generated, however. I opted to get the art from the cacti from a website called flyclipart.com, which contained free artwork for non-commercial use. It fit the art style of the dinosaur model, so I opted to take the bundle of artwork from that website, which I then used to make my sprites.
- 3/4/22
    - I met with Henry to evaluate the final product and receive some feedback. He said that the project achieved all of the success criteria: there is a condition that a player can achieve to lose; scores are persistently saved; the game is available online, as well as in an executable form; the game has functioning audio; it stores high scores persistently; and the game can potentially last forever if the player simply jumps over the cacti. He was happy with the instructions that appeared on the screen and the ease of controls (ESC to pause/unpause, the up arrow to jump, and R to restart). However, he did have some criticism for me. I created the game with a base size of 1024 pixels by 600 pixels. While it worked well for my computer, the character models appear stretched on other computers with different aspect ratios. On his computer, the dinosaur model looked wider than it was on my computer. He also pointed to the performance on the HTML5 port of the game. The Godot Game Engine’s JavaScript export isn’t as optimized as its Linux or Windows executable exports because it’s still in beta. When playing at a resolution of 1080p, the game can lag and drop down to 30 frames per second from its target of 60. During the meeting, we also brainstormed possible ways to extend the game and add more features. He suggested adding in more enemy types, such as birds, that the player could duck below with the Down arrow key, because birds are an enemy type in the original Chrome Dino game.