Jonathan Clement

04/12/2022

Prof. Ryoya

Culmination

In the beginning of the development of Sirena, as a group we all did not want to oversee programming the project since it did not match any of our strong suits were. So, since I had the most experience, I decided to take up the role of lead programmer. At the beginning of the project my first difficulty was figuring out how I was going about designing the game around the concept. So, my solution was to research the previous projects that I have done to search for solutions. In the original project Idea, we had portals that transported the player from underwater to on. So, I had done research and had found a developer website by the name of Brackey's, where licensed developers help to solve coding conundrums. There I had found assistance for the topic of portals, but then an issue had occurred. This issue was a lack of focus on the concept of the game.

We had skipped out on working on the concept phase of the game and jumped to working on the design and development phase. When our advisor had brought this issue to our attention, as a team we had begun to poke holes in our own ideas. After a few weeks of refining, we had settled on the Idea of having a game focused on the pollution in the seas. We also changed the main character from an orc that walked on land, to a mermaid in the ocean. This change brought about a unique design philosophy that changed the way I thought about programming. I had to now begin to design with the intention of swimming being the primary focus. My first objective was changed from being a portal mechanic, to now getting swimming working. Getting swimming to work was long and tedious, it took me around 2 weeks to fully get it up and running smoothly and effectively. After I solidified the swimming and changed the code to work with a video game controller's input, I then had to begin designing the puzzles in the world.

Puzzles would be the primary focus after the movement, and the puzzles would have to make sense with the swimming code that I created. My advisor had brought up originally when we redesigned our concept that since swimming was our new focus, the puzzles had to make sense in an underwater environment and had to make sense to be completed with swimming in mind. Our reference used was a puzzle from the video game Skyrim where you must touch sensors that would open a door and you would have to swim through for a set time before the door closes again. There would be 3 doors and 3 sensors you had to swim through. So, I designed the puzzle to be able to be multiplied to fit however many sensors or doors we wanted to add. This was simple to code; if the player touched the sensor, raise the door to a specific position at a specific

speed. Then after a certain amount of time, do the previous door raise in reverse. This means that I set the door to go back to its original closed position at a specific speed.

The next design difficulty was dealing with the plethora of errors I was getting. I had made code that was accessing other coding scripts. The issue with this is that if an error happens it becomes difficult to troubleshoot due to not understanding where the error is exactly taking place. This was prevalent in the second puzzle. In this puzzle, the player must touch sensors which rotate their respective pillars. When the pillars were rotated to specific orientations, it would solve the puzzle and open the door. This code set me back by 2 weeks since it produced errors every couple of changes in the environment. And many times, I could not replicate what I had done in the test level which worked perfectly fine. In the end I ended up rewriting some of the code and made it simpler so that I was able to understand the errors if they popped up again. “Console.Log” is a line of code that prints out a message of whatever you want in your code, whether it be a number, value, true/false statement, or even custom text. This Helped me tremendously in troubleshooting my errors and problems.

The final thing that I had to do was comment on my code. Commenting code means to write descriptions in the code that make it easy for people to understand what is going on in specific parts of the code. This was important because my group members had to access the code at some point in time and they would not be able to understand the code since they did not sit down and watch me explain every part to them.

Overall, this was a great learning experience working in a team with set deadlines and problems that needed to be solved. It taught me how to problem solve when I cannot ask anyone for help. It has improved my skill of creating and/or repurposing code online to fit my wants and needs. It also taught me how to work collaboratively when errors or concerns happened, such as when a substantial portion of the work we had done was deleted. I believe that I can take these skills and apply them not only to the 2D concept art that I want to do, but also in a coding career. These skills not only apply to game design, but to design in general.