Enoch Huang

Experimental Game Design Final Game: Good Doggo

For our final project, my group decided to continue working on our Good Doggo game. In order to continue flushing out the game, we needed to give the player more things to do in the game and point them towards some goals. I, along with my group members, did a lot of research to figure out what kinds of quests could be viable and reasonable for a dog. This was how we ended up with the quests we have in the game today. I also spend a lot of time experimenting with the environment of the backyard scene. For example, I noticed that the grass seemed very blinding at certain parts of the day cycle, so I played around with the grass colors to find a color that was more pleasing to the eye. Furthermore, I implemented a day/night cycle into the game. I found a cartoonish skybox from the asset store (the skybox our artist made wasn’t working correctly for some reason, we couldn’t figure out why), and created a script that rotated the directional light with the skybox. This resulted in a day/night effect, which was going to be used for special night quests, which have not been implemented yet. Currently, there is a weird affect where objects still seemed to be reflecting a weird color at night so the objects don’t look completely dark. We are in the process of fixing this bug. Also, once Brittany finished the new house models, I implemented all of them into our actual scene and added mesh colliders to them. Karthik suggested that we have an automated camera system for our game in order to improve the immersion for the player. For a while, I played around with Unity’s CineMachine system, and for the most part I got everything working. I set up multiple smart cameras that could follow the dog, and when the dog ran into a certain location, the camera could change accordingly. However, I thought it was very weird that the player couldn’t control the camera. It felt very limiting as a player if I couldn’t pan around to see some things that were out of view. So, I decided to use a kind of hybrid camera system. For the most part, the player is able to control the camera however he wants by moving the mouse, but the camera will automatically move to avoid running into fences, the ground, and the houses. I was watching some gameplay videos of Mario Odyssey, and I noticed that they used a camera system similar to this, and I thought it would work well for out game. This new system resulted in a slightly automatic camera system while still giving the player the freedom to control the camera if he wants to. For Tiffany’s quest, I also programmed some simple scripts for the birds so they fly around Tiffany’s doghouse. Then, once Doggo scares away all the birds, Tiffany will walk out of the doghouse and begin the dialogue system that Camron created. Lastly, I worked closely with the other programmers in my group to debug and fix some small errors in the game. Some examples include fixing how the birds fly, making sure all the animations were working correctly, and making sure the fences weren’t overlapping. Looking back, I am very proud of what my group has accomplished, and I look forward to continue working on this game in the future.