6.835: Mini Project 3

Nada Hussein

Spring 2020

1 Challenges

I had a lot of initial trouble with setup. It was hard to get web scripts working, as well as getting the LeapMotion and mic to process inputs the way I wanted them to.

I found it pretty difficult to smooth out the process of ship deployment. At first, figuring out the grab and pinch thresholds was hard, as sometimes the leap was not responsive at all, but sometimes it would continue holding onto a ship after I had tried to let it go. I was able to tune the parameters and arrive at a grab parameter of 0.85, or a pinch parameter of 0.85, which performed pretty well. I also smoothed out the cursor's movement by doing a simple average of the previous cursor and the current cursor position to get rid of any jumpy behavior.

I also encountered some challenges with both player's turn and CPU's turn. Initially, I struggled a lot with the speech recognition, as it took many tries for my program to hear what I was saying. I ended up solving this simply by using a separate headphone mic instead of my laptop's mic, and this made the process much smoother. Otherwise, I found this pset pretty straightforward and not very challenging.

Finally, it was a little difficult to debug this program since the Leap input always needed to be detected for the game to progress. Some features (like saying hit, miss, etc) don't require this, and it ended up being a little frustrating to need to hover my hand over the leap just to debug parts of my code that did not need gesture input.

2 Extensions

2.1 Detecting Lies

The first extension I implemented was a lie detection feature. When the CPU fires, the player responds with 'hit', 'miss', 'sunk', or 'game over', but the game responds based on the board itself. My implementation compared the user's verbal response with the actual status of the board, and if they did not match, the CPU calls the player a cheater or a liar, and then continues based on the board's result. This extension does not add any extra user interface, and will simply interject whenever the user lies about the CPU's firing location result. This

extension adds more interaction to the game, in that the user is directly responded to for lying, and this discourages them (hopefully) from attempting to cheat, or offers a funny feature that calls them out when they weren't expecting it.

2.2 Verbal Ship Placement

The section extension I added built on the user's options when deploying ships. Usually, the user has to grab and drag a ship to place it on the grid. I added a feature that allowed them to say 'put battleship here' or 'put patrol boat here' while pointing at a square on the grid in order to move the ship to that square. I did this in the processSpeech loop. If a user said the phrases I mentioned before, the code would enter a loop that finds the current location of the boat they want to move (battleship or patrol boat). From here, the program figures out the tile the user is at, and places the ship horizontally with its furthest left square being the one the user pointed at.

This extension was helpful for the user as I noticed that grabbing and dragging a ship could sometimes be finnicky. Having a user simply point to where they wanted to place a ship was a much more streamlined process, and allowed them to very quickly place their ships on the board and start the game with minimal frustration and troubleshooting.

This feature does not yet allow for rotation due to deadlines, but I would have added this to the extension given enough time.