6.835 Project 4

Alyssa Waln

April 11, 2015

Troubles

One of the things that was odd to me (but I can't think of a better way to do it) is that the game is dependent on Leap input to keep going. As in, there isn't a main loop that can keep running without frames from the Leap. I thought of trying to mess with this, but I am a bit time-crunched and worried that such a large change would absolutely bork everything else.

To be honest, though, quite a bit of it was very straightforward (I've also done a *lot* of work with JavaScript before, so I'm used to the silly things it sometimes does). I didn't have much trouble implementing the basics.

Extension (and planned extensions)

The extension I implemented was to be able to use voice to put ships in (since I had a hard time pinching ships so far to the left, anyway). I had some trouble because I wasn't sure how to grab a particular ship (solved by more or less copying from $getIntersectingShipAndOffset$) and figuring out exactly where to put the ship was troublesome; just saying "here" in a corner doesn't specify how you wanted it oriented. I found that the most intuitive way (for me) was to have it just call into my hand, so I could rotate it freely from there.

I might play with this later, but one of the things I'd like to try is to make it play mind games with you, as in, if you choose a space, it might say something like "Are you sure you want to go there? What about here?" And with some probability, it will either be teasing or actually trying to get you to not hit something.

I also just thought of trying to make the computer impatient. If you take too long to fire, it could ask what the holdup is. Even more GLaDoS-y, it could take your turn away from you.

There was one other thing I tried doing for fun, which was to try and make the computer more "human" in a dumb way - literally (even writing it down makes it more obvious in retrospect that it was not a great idea). I wanted to make the computer have a "four-year-old" mode, where it could make mistakes, it could accidentally pick a square twice, and would sound unsure when it made moves. The idea was that in some ways, computers sound really "perfect," which might not be so natural (like always having the exact same speech cadence. Part of the reason GLaDoS feels a bit more human, I think, is because it is a human voice actor, just modulated). A computer that could make mistakes might be more relatable and enjoyable to talk to. I ended up scrapping this idea, though, because it occured to me that while being “perfect” was a less human feature of computers, it’s also the most *useful* feature of computers, so the trade-off didn’t really make sense (unless you very specifically wanted a computer talking companion rather than a computer that could actually get work done). I’m curious now about how speech systems add in variability. I think there is some, but computer voices still sound a bit flat, because tones are more about context than semantics (?).