Michael Scotto  
Michael Sun  
Igor Prokopenko  
Kevin Strileckis

Assignment 1 Agent/Wall Sensing

During the first project in our AI for game programming class, we learned a lot of things about trial and error as well as our own capabilities. We implemented a simple controller that allowed us to not only sense the heading and location of our subject, but also to affect the direction and movement of our subject. We also implemented wall sensors using raycast that made us able to track the location and distance of walls from our agent. We then created sensors that told us the location and distances of agents that we created in the environment, allowing us to track them with our movements. We finally created a system of pie slice sensors, which took in the angle and distances from the previous step and allowed us to trace the number of agents in each of four quadrants surrounding the subject. It was a taxing effort, but by the end of our project we all learned a lot about how to implement some basic game programming.

The controller was one of the easier parts, as we had implemented one in our tutorial project before. We simply assigned keystrokes to rotate in each direction, as well as to move backwards and forward based on a heading. We then had these two pieces of data output into our debug console. This was crucial to do first, as we needed some of this data to further work on the project.

The wall sensors operate using Raycasts, each of a length set in Unity. Three were created and reinitialized every call of function WallSense, and would check for any collisions with colliders that are tagged as "Wall." This function was called by the SubjectController's Update function, so that the wall sensors "sense" for walls every time the subject moved. When a wall is in range of the sensors, the function prints out to the log about how close it is to the sensor's origin and activates an animation where the subject blinks, letting the user know that a wall is coming up. A similar effect was applied to the agent detection, so that the agents would grow and shrink and change from blue to black and back to blue, so to catch the attention of the user.

Next, we added agents. These agents were stationary as it made it much easier to debug, and with these agents we used a circle collision to locate the agents in relation to our subject. We then updated these and read them into the console each time the subject moved, and effectively gave a new, constantly updating radar that gave us our agents, their location, and their distance from the subject. This knocked out the second part, and this was where we hit a road block.

Our problem was trying to implement the pie slice sensors in a separate object, when all along we should have been thinking about the steps we had already taken. After scratching our heads for a while talking about vectors and headings, we realized that if we could only make the angle values indicate the objects locations, we would be able to switch those locations into different quadrants. After some pouring on the internet, we found a way to take the cross products of the vectors and using that information, we changed our 0-180 degree repeating angles into a full -180 – 180 degree circle, which indicated, when split up into 90 degree chunks, which of the four quadrants (if any) an agent was located. After that it was a simple if loop that added up all of the agents in each quadrant, and returned them into the console.

By the end of this, at what is now 11:33 at night, we have all learned a lot of things. For one, procrastination hurts, and can make things seem a lot harder than they are. We also learned a lot about implementing ideas with not much but a few numbers to work with. Usually what you need to do can be found in what you can do, and as a group we found it very helpful to take our eyes out of the code and to think critically about what we had to do before we did it. We also learned that the internet is a great place, and if your questions are specific enough, there is a chance that someone else has already asked it.