2/14 2 hours: review material, project setup, planning & implementation

2/16 2 hours: setup working test project. Wrote down thoughts and set goals for next two dev sessions before release 1.

2/18 5 hours:

I really dove into the project fully today. I was able to make some good progress and am trying hard to not focus too much on implementing a perfect design. I was able to implement a basic alternating pattern where cars are able to move successfully between road segments. I wanted to focus on the simulation aspect in the early stages of the product and did not want to mess with the ui portion. As a result, I found myself working around an increasingly-complex road creation script in the model class. This can obviously be improved with a builder and a few factories.

2/19 5 hours:

I was able to implement the basics of the intersections and lights. Lights transition between the states detailed in the notes. I used enums and was very happy with the extra functionality java provided me. In addition to holding the basic state representation, the enums currently hold a delay setting and the ability to return which direction is currently allowed to pass. The intersections currently are only implemented on East-West roads however. I hope to fix this issue next.

2/20 2 hours:

Worked on mostly the car behavior today. Cars will now no longer pass each other, though there are a few bugs related to passing cars between road segments that I’m still dealing with. The big problem I’m running into is the behavior at intersections.

2/21 3 hours:

Primarily focused on some intersection behavior which had been troubling me. Cars seemed to bounce and line up on top of each other due to an error in the function which calculated free space. One of the biggest issues I seem to be having is separating responsibilities between car and road. Is it the car’s responsibility to check the road for issues ahead? It probably is, maybe I should move that logic to the car and just have the road return a queue of cars.

I also changed the road interface to use a queue. Because cars cannot pass, the queue provides some functionality that seems useful.

2/22 3 hours:

Worked on a few road transition bugs where faster cars could skip a slower car at the point of exchange. I also began working on some of the customizable model parameters. It has definitely opened my eyes to some of the complexities which I will have to deal with. For example, the simulation time step will be pretty challenging to deal with. It may however explain some of the lurchy behavior I have now.

2/25: 5 hours:

After watching the lecture from 2/18 this weekend, I realize I made a significant mistake by assuming I could use the provided project code as a starting point. While I had made some significant progress, it had already become apparent that I was going to run into trouble with the very rudimentary timeserver provided and needed to stick to the implementations in the agent package.

2/27 4.5 hours:

Unfortunately, migrating my work from the previous release to use the timeserver proved more time consuming than I had hoped. I am now, from a functionality standpoint, about where I was last week. A fixed grid is supplied, cars move along it and do not pass each other, they wait for lights to allow them access to an intersection and the next road segment. I had to rework some of my logic, but it just seemed that fitting these pieces together took a substantial amount of time. Of course this is my own fault for incorrectly starting with the supplied project and trying to build it out.

With a free half hour, I quickly implemented the basics for normalizing the car position as well.

**WEEK 2 total: 14.5 hours**

2/28 3 Hours:

I spent most of my time today smoothing out some of the UI issues. As you mentioned in class, getting the UI working properly was very helpful from a debugging standpoint. I normalized the road and car lengths(though they appear quite small). I had a great deal of trouble with my simulation running VERY slowly as more cars were added. It seemed that the displaydelay, which was set to 50, caused too much delay between UI refreshes. Either the display delay was set too low or my modifications to the timeserver, which updated observers every time an agent was run.

2/29 4 hours:

I continued down my checklist of issues to address today. I set lights to start in a random state, previously they had all worked in unison. I also showed cars in intersections by modifying swing animator builder.

The biggest issue I ran into today was implementing brake distance. I believe I need to rethink how I am currently calculating distance and much of the car class/vehicle interface. Instead of working through the UI, I believe I need to beef up my unit tests to adequately address this issue. I am also beginning to realize I may have coupled the car class too closely with timeserver and road.

3/2 3 hours:

I worked on implementing the brake distance and generally focused on smoothing out the car movement. After a little investigation, I found the sortedset interface and decided it could help me greatly simplify some of the distance and free space calculation logic.

3/3 5 Hours:

I spent most of tonight working on ironing out some really tricky bugs related to car movement. It was pretty frustrating, but ultimately I’m glad that I can now begin the refactoring process in earnest.