Aidan Lewis

Software Engineering

Homework 5

Reflection

1. I made some pretty significant design changes during the assignment. First, in order to add a second train, I added a train factory class. This way, if this were to be expanded to include even more trains, the train factory would provide a cohesive way of handling multiple trains. For the second part, I made changes in order to make my design aware of intersecting roads. To do this, I moved most of the functionality into the Road class. This class has a deque of cars. Once the top car on the deque reaches the end of the road, it’s popped from this deque, and added to the deque of an intersecting road. When it’s added to this new road, all of it’s previous observers are deleted, and this car is registered with the car just before it in the deque. The downside of this overhaul was that it required a pretty significant rework of the code, meaning that time was a definite tradeoff. However, once the infrastructure for this new approach was in place, it took significantly quicker than my previous attempts to actually code the functionality. With this approach, each road was essentially a self-contained linked-list of cars, and when moving from one road to another, the car simply had to be removed from one list and added to the back of the next. This made navigating the intersecting roads of part 2 much simpler.
2. I think in general my approach would scale up, since I implemented a general solution for connecting roads. With this general approach, any intersection could really be implemented (T-Junctions, four-way intersections, etc.), since one would simply have to note which roads intersected with which others, and then cars could transition smoothly between these different roads. A possible downside to my approach is that every intersection of two roads must be hard coded in. Ideally, I would have code that given a bunch of roads, could figure out where and what intersections existed.