Journal Report 10
11/25/19 - 12/8/19
Richard Zhan
Computer Systems Research Lab
Period 2, White

Daily Log

Tuesday November 26

I added a GitHub description of my project. I began working on implementing the ability to interact with $Events$ through my website viewer. $Events$ used to be stored by value in each of the $Cars$. I created a global variable $graphEvents$, which stores all of the existing Events. Now, each Car stores pointers to its $Events$.

Monday December 2

I altered the printing method that runs for each frame to include the global variable $graphEvents$. This has significantly slowed down the execution speed of my program due to high amounts of outputted data.

Tuesday December 3

I revised the printing method that I wrote to improve the runtime speed of the program. I now only print each $Event$ once, even if it exists in multiple frames. Although there are still many $Events$ that need to be outputted, this somewhat improved the runtime speed of the program.

Thursday December 5

I added the search bar in my web server that takes in an $Event$ id and then returns the state of the $Event$. This has significantly slowed down my web server's frame loading rate such that it is noticeable, especially in the later frames, when new frames are being loaded. The later frames tend to have many more events due to higher numbers of $Cars$ producing $Events$.

Timeline

Date	Goal	Met
11/11/19 - 11/17/19	Add the ability to see the history of $Events$ and $Cars$. Add real-time viewing of the simulation. Ask Mr. Kosek about JavaScript/Handlebars issue	No, I did not finish adding the ability to see the history of $Events$ and $Cars$. However, I did ask Mr. Kosek about the JavaScript/Handlebars issue. I found a better way to load the data from the output files.
11/18/19 - 11/24/19	Add the ability to see the history of $Cars$.	Yes, clicking on $Cars$ displays their current state.
11/25/19 - 12/8/19	Add the ability to see the history of $Events$.	Yes, I've added a search tool that allows users to input an $Event$ id to get the current state of the $Event$.
12/9/19 - 12/15/19	Create Manhattan-style, rural, and dense map input files and begin testing the effectiveness of DTD on them.	
12/16/19 - 1/5/20	Winter break	
Winter Break	I want to show a significant difference in time between my DTD/non-DTD cars for multiple types of maps (basic, Manhattan-style, rural, dense). These should be displayed on a JavaScript Web server, which can be interacted with by user (start/pause/click on objects to access current variables given a frame number).	

Reflection

This week, I added the ability to see an $Event's$ current state by searching for it on my web page displayer. Currently, it displays all of the $Event's$ data in a JSON format at the top of the web page. I originally outputted $Events$ for each frame, but realized that since $Events$ don't change (besides expiring) between frames, it would be faster to simply print all of the $Events$ a single time.

Next week I would like to test different types of maps with my DTD navigation system. In regards to my Winter Break goal, I still believe that I am on track to meet it.