Journal Report 10
11/18/19 - 11/24/19
Richard Zhan
Computer Systems Research Lab
Period 2, White

Daily Log

Monday November 18

I fixed a bug that froze the canvas after it reached the final frame. It wouldn't allow the user to select another frame to display. This was caused by the fact that I forgot to reset one of my variables after it reached the final frame and paused the screen.

Tuesday November 19

I researched how to interact with canvas elements by user-input clicking. I found that drawn objects did not have built-in clicking event listeners. There were two alternatives: iterating through each object to check if they had been clicked, or creating a second canvas that served as a hit map. I opted for the second choice since it would scale better as the number of objects increased.

Thursday November 21

I finished implementing the interaction with $Cars$ through clicking. When a Car (red circle) is clicked, it will display the current state (JSON format) of that Car at the bottom of the website. One of the biggest issues I ran into and fixed was the fact that scrolling through the screen would offset the clicking location. To resolve it, I searched up methods that provided the mouse's location relative to the canvas's top left corner.

Timeline

Date	Goal	Met
10/28/19 - 11/10/19	I would like to add functions to my GUI that would allow me to see the history of $Events$ and $Cars$. It would also allow me to see overall stats of the run as the program is executed in real-time.	I decided to spend time on reformatting how I printed the data in order to scale better as I progressed through the project. As a result, I did not pursue the original goal for these two weeks
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 $Events$.	
12/9/19 - 12/15/19	Create Manhattan-style, rural, and dense map input files and begin testing the effectiveness of DTD on them.	
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 a $Car's$ current state by clicking it on my web page displayer. Currently, it displays all of the data in a JSON format at the bottom of the web page. I can change the display into a more user-friendly format in the future, but for now, it is enough to suit its purpose.

I changed the time frame for the next journal to cover two weeks since we only have one Blue day next week due to Thanksgiving. For that work period, I would like to add the ability to see the information about *Events*. After that, I only need to test my DTD navigation systems on other types of maps.

In regards to my Winter Break goal, I believe that I am on track to meet it.