CSE6730/CX4230
Spring 2022
Project Final Report

Team Number: Team 11
Team Members: Tusheet Goli, Matthew Dacey-Koo, Tushna Eduljee, Aarun Srinivas

Project Name: Georgia Tech Traffic Flow
Link to GitHub Repository: https://github.gatech.edu/tgoli3/gt-modsim-s22-team11

Link to Video Summary/Demo:
https://mediaspace.gatech.edu/media/%5BTeam+11%5D+Georgia+Tech+Traffic+Flow/1_1541a13m/254858153
YouTube Link: https://youtu.be/vrQESedT3Is

Abstract

In this project, we model traffic flow at Georgia Tech using campus road networks as they currently stand. This simulation will allow users to adjust various parameters such as traffic flow while putting in place terrain-changing obstacles such as road blockades and intersection shutdowns which alter traffic flow. Previous attempts at charting automobile routes around campus include PassioGO and GT Buses (which both map GT bus routes in real-time). While these help students navigate campus, a key challenge is smoothly navigating all road traffic around campus even under perturbations that change traffic patterns. As such, traffic patterns under perturbation can be observed and used to identify pain points in the campus road networks for city planning in the future. It can help provide statistical analysis to assist with the city road planning by giving insightful information about intersection congestions, placement of traffic lights, stop signs, etc. This insight can be particularly useful in developing re-routing strategies in situations of blockade or road closures for example, on game days, special events, etc. We aim to achieve this using a replica of the GT road map system and use the conceptual model of a microscopic traffic automaton model (IDM - Intelligent Driver Model) proposed by Treiber et. al. which is a multi-agent system of vehicles that act independently in response to a stimulus from the environment.

Project Description

With the refinement of our model over time, it may become helpful for future city planning that occurs in and around the Georgia Tech campus. It also could help assist in the rerouting of traffic for special events or during road accidents. We enabled this through our project as we ensured that the road systems in our model were as accurate as possible to those in real life. This includes road geometries, such as lane shape and layout, and key features such as one-way streets and traffic lights. These were some of the key real-world phenomena that we wanted to make sure we captured in our model, in order to make it as extensible as possible to real-life traffic dynamics at Georgia Tech. An ideal end goal for a project such as ours would be to assist in giving back to the campus and use the highlighting of the current pitfalls and areas for improving the current road systems around campus in order to benefit the faculty and students who live and work here.

System Description

The system that we modeled includes most of the Georgia Tech campus' main roads where automobiles such as cars and trucks can drive. This ranges from the intersection between North Avenue and Techwood Drive in the South East to the intersection of Curran Street and 14th Street in the North West. It is important to know that all Georgia Tech bus routes are also contained within this perimeter. These are the roads upon which we visualize all traffic through campus. As our simulation is meant for use in redirecting traffic given perturbations in regular flow (such as intersection blockages or road closures), the system features all of the same road rules as those present in the real world to keep it as realistic as possible. This includes all the same traffic patterns due to features such as one-way streets or limited turning lanes. Our system also includes accurate traffic lights, and important road signs (ie. stoplights, and stop signs) that we obtained through analysis from Google Maps and Waze APIs for all streets within the perimeter.

Literature Review

1. The paper titled "Trends in Real-time Traffic Simulation" discusses the various types of software tools that are being used to model traffic. Specifically, the paper analyzes 17 different traffic simulation software tools to determine the shortcomings and strengths of traffic simulation software as a whole. From its analysis, the paper comes to the conclusion that traffic simulation software still is unable to simulate the conditions present on complex heterogeneous road requirements due to the fact that it cannot identify patterns when presented with a small amount of real-time data. From this paper, we have identified several tools that could aid us in the development of our application. Specifically, the traffic simulation software tools Quadstone Paramics and SITRA-B+ might serve as good sources of inspiration for our project.

2. Another paper that offers some insights that could be utilized in our project is titled "Traffic Simulation Modeling of Rural Roads and Driver Assistance Systems." It discusses how microscopic traffic simulations can be used to analyze different road systems. Specifically, the paper attempts to utilize this simulation technique to analyze traffic present on rural roads and how driver assistance systems can affect traffic. The primary contribution this paper makes to the field of traffic simulation modeling is a simulation called RuTSim which stands for Rural Traffic Simulator. In particular, the paper utilizes RuTSim to simulate traffic present on a Swedish two-lane rural road and on a Dutch two-land rural road. In the chance that we would like to construct a model to describe a phenomenon that does not have much data, the ideas proposed in this paper could be very beneficial in that they can help serve as a starting point.

3. Unlike other papers which have worked to simulate traffic in the 2D space, the paper titled "Continuum Traffic Simulation" attempts to simulate traffic in the 3D space. Specifically, it attempts to provide a more accurate simulation of traffic in the real world in a fast and efficient way by describing the movement of several vehicles in a single computational cell. The paper further proved that its approach was much more efficient than state-of-the-art 3D simulation methods. This paper provides us with an idea of how to go about tackling 3D simulations in a fast and efficient way. Thus, the main idea presented in this algorithm which is approximating real traffic by describing the movement of several vehicles in a single computational cell could serve as a good starting point when constructing our own model of how traffic behaves.

4. The paper titled "Population-based simulation optimization for urban mass rapid transit networks" can help us in our simulation model for our project. Depending on whether we decide to look into metro rail-based transportation or road traffic, we can incorporate some hypothetical

future population-based work into our simulation. This could be helpful if we are motivated to focus on the city of Atlanta for our simulation. We could propose a way to create more efficient rail, bus, or roadways for our city based on some of the ideas for population-based simulation optimization discussed in this paper.

5. Wang et al. create multiple different models and compare them to data collected in China over a 10-month span. The researchers created the following models: Linear Regression, Spatial Lag, Spatial Error, and Time-Fixed Effects Error. To create these models, they consider both spatial and temporal features, creating equally-sized traffic analysis zones (rectangles for grouping data for analysis). The researchers found that the time-fixed effects error model, one that considers both spatial and temporal effects to be superior. From this, our group has identified a few models that we can pursue in our own research of traffic, along with possible inspiration for methodology.

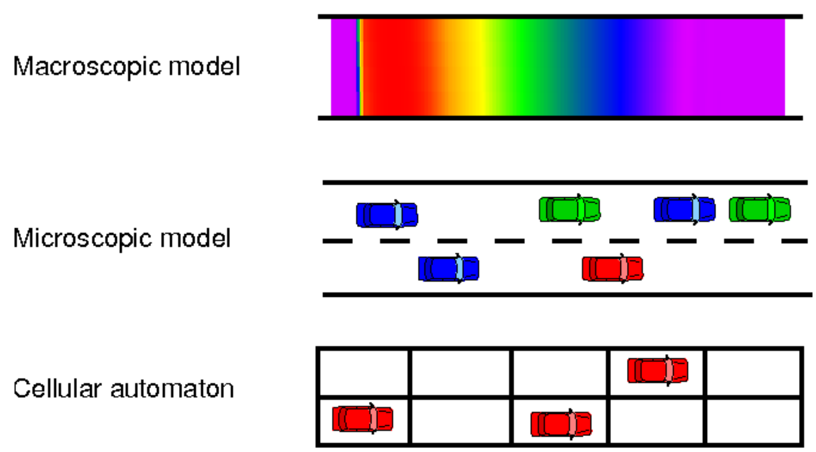
6. This link is a grouping of various resources used for a class on transportation at the University of Lisbon. Similarly, it provides many examples of models used to analyze various forms of transportation, for which our team can take inspiration to apply to our own individual topics. The methods and models listed are as follows: Multiple Linear Regression, Factor Analysis, Cluster Analysis, Generalized Linear Models, Panel and Spatial Regression Models, Discrete Choice Models, Ordered Models, Hazard-Based Duration Models.

7. Andrade-Michael et al. take a look at the bus vehicle and reliable driver scheduling problem. It attempts to take two NP-Hard problems - the Vehicle Scheduling Problem and Crew Scheduling Problem, and proposes an exact constraint programming model that is claimed to greatly improve efficiency when considering the reliability of drivers. This paper could be very useful if our team ends up pursuing research on local transportation systems like those of Marta or Georgia Tech. It'd be useful to utilize the proposed model and apply it to our local systems to determine where improvement could be made.

8. The paper titled "Modeling and simulation of highway traffic using a cellular automaton approach". If we do in fact aim to solve a traffic flow-related problem with our simulation, then we can utilize this paper to understand how cellular automata can be applied to simulate a traffic flow-related problem. Something like this can be helpful for us in describing the influence of a car accident in single-lane vs double-lane traffic flow models. Although this research project implemented the code in Matlab, we may use similar ideas to understand how to structure our inquiry in this space.

Conceptual Model

We firstly constructed a conceptual model of Georgia Tech's road system by utilizing a series of lane and traffic light objects. The lane object was implemented as a Python class and contains a queue that is used to keep track of all the cars currently driving on the lane. The traffic light object is also implemented as a Python class and is used to connect lanes coming from different directions together and direct traffic through their intersection. In addition to keeping track of the cars present in the lane, the lane object also contains a variable that can be set to close it off as well as a variable that can be set to determine the direction of traffic in the lane. The capacity of the lane is calculated using the lane's distance which will be determined upon instantiation and each car's length that is present on the lane. In a sense, this model of Georgia Tech's road system which uses lane and traffic light objects functions as a microscopic traffic model, i.e., a multi-agent system with independently acting vehicles where every i-th vehicle follows the (i-1)-th vehicle. This is because the movement of a car is directly dependent on its neighbors. In particular, if one car is located in front of another car, the back car's speed is dependent on how fast the front car is moving.



For modeling the cars, we utilized the Intelligent Driver Model (IDM) which is a mathematical model for multi-agent vehicle movements developed by Treiber, Hennecke, and Helbing. Before exploring the mathematical models for the vehicle movement, we will define some variables.

x_i - position of the i-th vehicle
l_i - length of the i-th vehicle
d - distance/displacement
v_i - speed/velocity of the i-th vehicle
s_i - the distance between i-th and (i-1)-th vehicle
a_i - acceleration of the i-th vehicle
t - time

The speed/velocity equations are the same as the equations from mechanics physics, i.e., the equations of motion. They are as shown below.

$$\boldsymbol{v = \frac{d}{T} \implies d = vT}$$

Based on these equations of motion, we can identify the distance between two vehicles as well as the differential velocity between two back-to-back vehicles. Those equations can be given as follows.

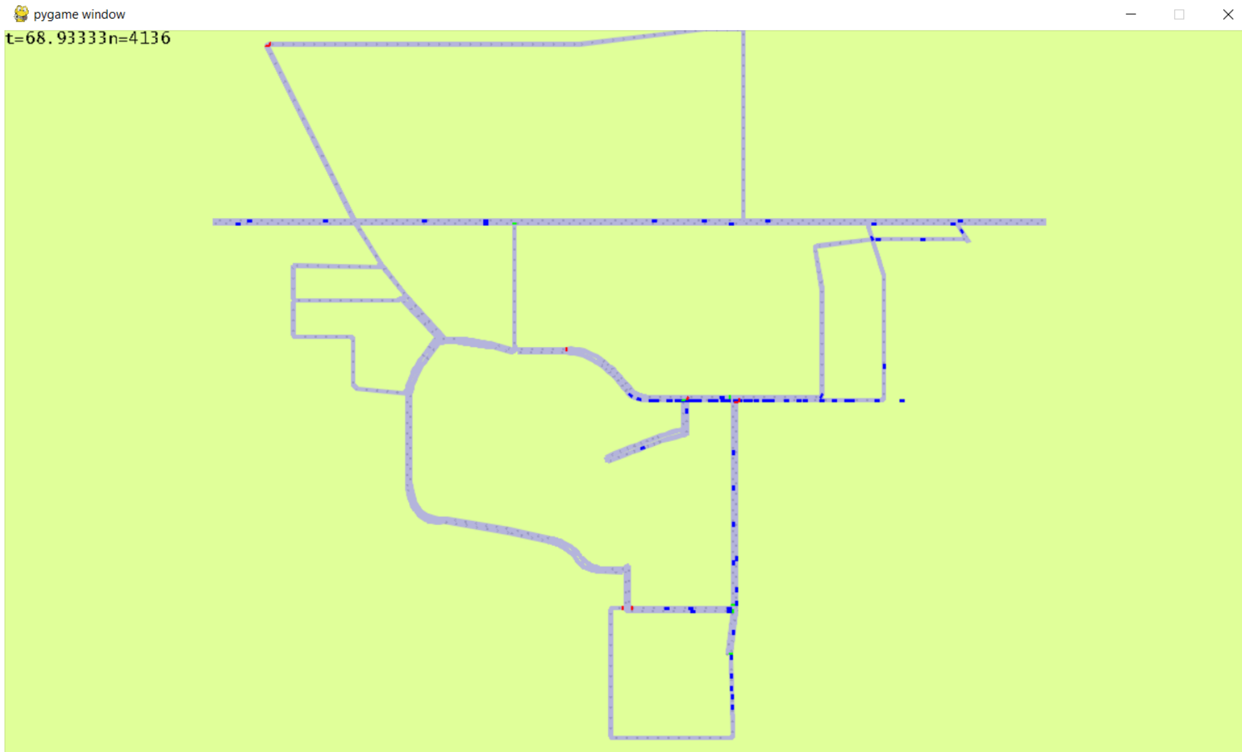
$$s_i = x_i - x_{i-1} - l_i$$
$$\Delta v_i = v_i - v_{i-1}$$

The acceleration of the vehicles is also governed by the IDM model proposed by Treiber et. al. This model assumes free roads and free interactions where the dynamics of the vehicle movements are given by the following equations. These equations govern the overall acceleration of the cars as a combined effect of the free movement on the roads as well as interactions between other cars on the road.

$$\boldsymbol{\frac{dv_i}{dt} = a_{free\ road} + a_{interaction}}$$

$$\begin{cases} a_{free\,road} = a_i\left(1 - \left(\dfrac{v_i}{v_{0,i}}\right)^{\delta}\right) \\ a_{interaction} = -a_i\left(\dfrac{s^*(v_i, \Delta v_i)}{s_i}\right)^2 \end{cases}$$

That summarizes and dictates the fluent motion of the vehicles on the road and this is the mathematical model we utilized to simulate vehicle movements on the proposed Georgia Tech map system. This map system was made using python pygame where we replicated the Georgia Tech road system to simulate these cars on in. An example of our map with a simulation would be as follows:



Furthermore, as more cars join the lane, they will be affected by the cars in front of them. In addition to cars affecting the speed of surrounding cars, traffic lights can also affect surrounding cars. Thus, the combination of cars influencing other cars at each time step as well as traffic lights influencing cars makes this a multi-agent microscopic traffic model. It is however important to note that although this microscopic traffic model appears deterministic, most of the cars present in the road system do not behave deterministically as they sometimes will randomly speed up or slow down. Thus despite its appearance, there is some stochasticity present when talking about how cars affect other cars' speed.

Simulation Model

Our simulation is built upon the Python Pygame modules. This is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language. With our model including lots of traffic and individual actors, we wanted to make sure that the visualization would render seamlessly and without glitches due to the many independently moving parts of the simulation. As the crux of our conceptual model is mathematical equations, we found that the process of transferring it into our simulation and computer model was quite smooth. Unlike some different conceptual models, the key aspects of ours were tried and tested by those researchers who developed the equations, and this ensured that our simulation would work as intended from the start rather than requiring a lot of updates to the conceptual model itself. Additionally, most of the conceptual aspects that we included in our computer simulation (ie. the intelligent driver model for safe following distance) could be observed visually in order to ensure that they were working as intended and did not need to be updated or tuned for our use. The primary method that we did use for finetuning our simulation, however, was the visual observation of the vehicle and road dynamics.

Platforms of Development

As our project is specifically focused on Georgia Tech and its road planning system, we want to create a tool that both accurately represents the system and is easy to use for clients without background knowledge of the project. For this, we used python for front-end data visualization as well as for back-end data processing. We found python easy to code the backend, but also utilize the ease and power of pygame to be able to render an interactive and good-looking front end.

Front End

The front end is a pygame built in python. Pygame is a python module designed for writing video games and interactive UI screens. It utilizes the power of being a primary backend language that has the ability to render interactive visual displays and simulations. We created our simulation engine and GT road map system in the form of a pygame that can render cars moving on these roads.

Back End

The backend was also coded in python to work hand in hand with the pygame UI design. We have our mathematical models for our vehicle movements and roads encoded using python and the UI utilizes these models to render a clean simulation of our system.

These platform choices were made because we have experience working with python and pygame from previous classes and past projects. This also enabled a simple and clean coding experience with little to no effort for frontend and backend integrations since it was all in python. Using these platforms of development, we were successfully able to implement our simulation.

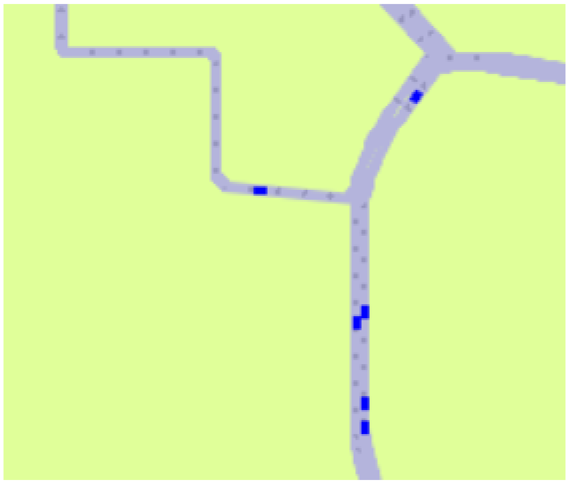
Experimental Results and Validation

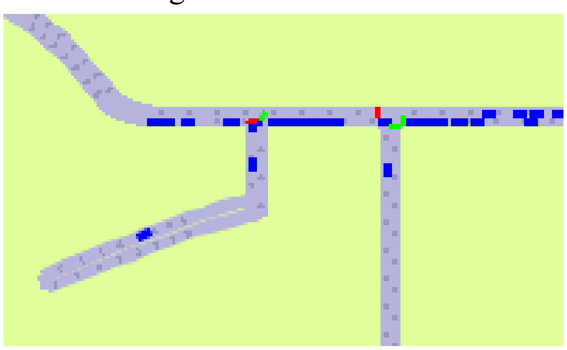
Experimental Goal

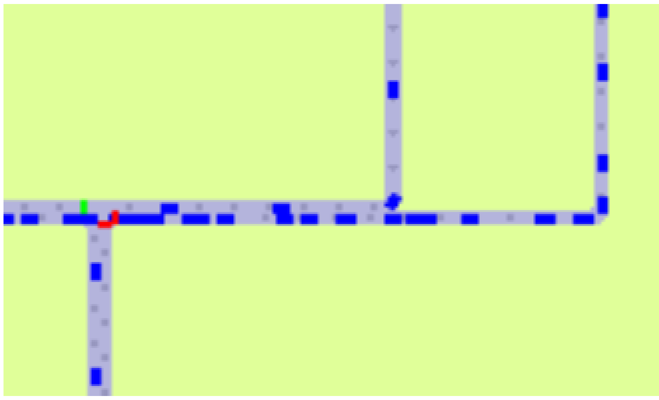
Using our knowledge of the campus as students, we located four points on campus that would be interesting to model. We wanted to look at the pedestrian impact as a function of the time of day. Each point also has a nearby parking deck for Georgia Tech workers and students, so each deck served as a nearby spawn point for cars that majorly affected the traffic.

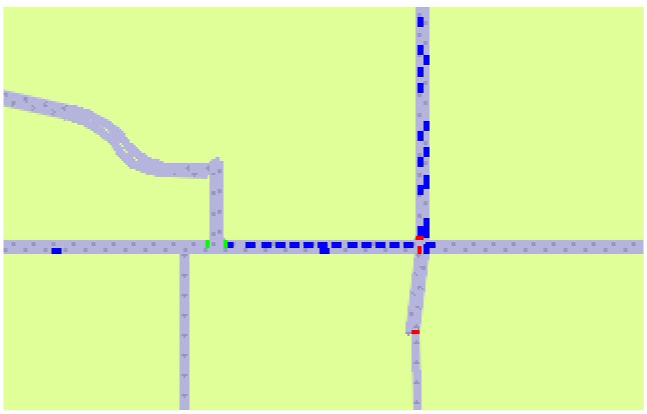
The four are as follows:

1. Right in front of the CRC

a. Busy crosswalk for students coming from the East to the CRC
b. Nearby Student Center Parking Deck
c. CRC Parking Deck
d. 

2. The Ferst and Fowler Drive intersection
   a. Crosswalk is one of the busiest on campus
   b. Klaus Parking Deck
   c. Peters Parking Deck
   d. 
3. The 5th and Williams intersection
   a. Poor road signage (one-way stop)
   b. Busy crosswalk for students entering/exiting campus to and from midtown apartments and Tech Square
   c. Tech Square Parking Deck

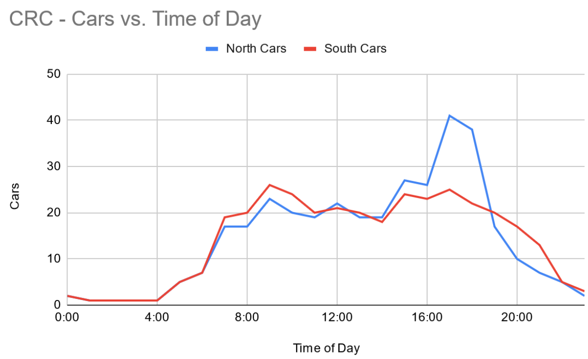
d. 

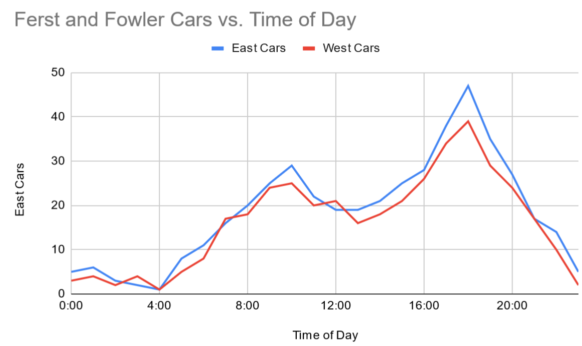
4. North Avenue along Tech Tower
    a. North Avenue Parking Deck
    b. Has a pedestrian bridge - to analyze this, we ignored the other crosswalk along North Avenue
    c. 

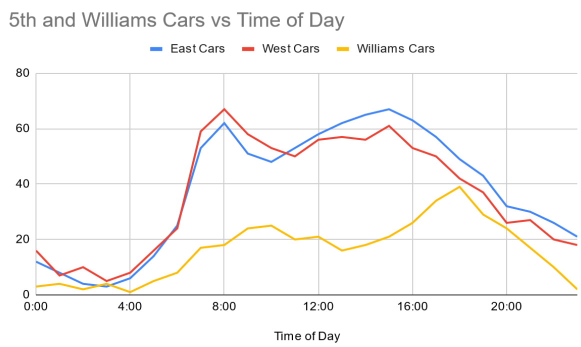
Procedure

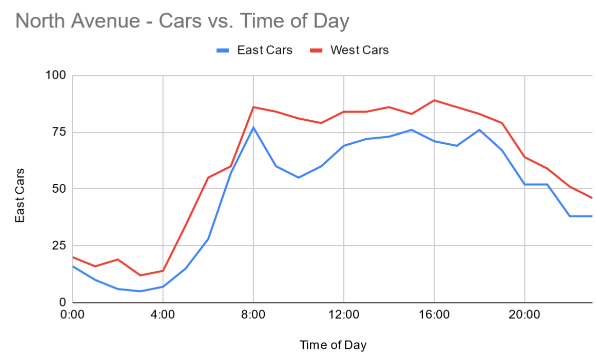
The simulator is set up to spawn cars at certain points. There are various entry points around campus along with parking decks. For the time interval, we had spawns follow a curve where they're at their lowest between 7 PM and 5 AM. Spawns then increase rapidly as people drive in for work or school, then decrease slightly until the end of the workday around 4 PM - 6 PM. These spawn rules are applied unilaterally to parking decks and street entrances, which may not be wholly realistic, but further implementation wasn't possible within the time constraints of the project.

Then, we set up an object at each point to track the number of cars passing by. An object at each location tracked cars for which one minute in the simulation represented one hour in real life. The collected data was then recorded, giving us the following graphs:









Validation & Analysis

The Georgia Department of Transportation provides datasets for us to compare car counts. For the point along the CRC, we can use a two-tailed t-test to compare the difference between the data. First, we took the logs of both datasets to mitigate the difference between the scales. With this, we calculated the 95% confidence interval of the difference between the data to be [-13.97, 10.88]. The test gave a two-tailed P value of 0.8039, which means the difference isn't statistically significant. Similarly, the other tests give us P-values all over 0.1, meaning we can proceed with analyzing our simulation on the basis that it mimics the real data to a strong enough extent. The data does represent what we were expecting it to, both on the basis of our real-world experience with the areas and as a result of us coding the simulator to attempt to mimic it as best we could. Even though the spawns were hard-coded, there were still enough variances in variables such as traffic light patterns and car paths to give us insight into Georgia Tech's traffic patterns.

Discussion and Conclusions

We learned from simulating Georgia Tech's road network, that the traffic flow on campus is a lot more complicated than we expected. Specifically, since the activity on Georgia Tech's campus is always changing due, we realized that to simulate Georgia Tech's network as well as possible we would have to ensure that vehicles get generated onto the map erratically. That way cars will enter and exit Georgia Tech's road network at random times, providing us with a simulation of how Georgia Tech's traffic flow behaves on average. In addition to learning about how complex Georgia Tech's road network traffic flow is, we were able to identify specific points that were susceptible to traffic jams. Thus, by outputting that knowledge, users of our simulation program could identify intersections that are susceptible to congestion, and then plan alternate paths with which they could reroute traffic to intersections with less reduce congestion. Users of our simulation program could use a similar approach to reroute traffic when road work needs to be performed at certain intersections and roads on Georgia Tech's road network.

Although our simulation was able to yield useful results, further work can be done to make our simulation much more similar to how Georgia Tech's road network functions in real life. Specifically, one change that can dramatically increase the accuracy of our simulation is to incorporate pedestrians and crosswalks into it. This is because, oftentimes, traffic jams occur at Georgia Tech not because of an overflow of vehicles, but rather because a large number of students have just exited their classes and are now walking through crosswalks to get back to their dormitory. Thus, since students are a main, if not primary, cause of traffic congestion at Georgia Tech, adding pedestrians and crosswalks could result in a significant jump in the accuracy of the results obtained from the simulation. In addition to incorporating pedestrians and crosswalks into the simulation, future work could consist of building a user interface that would allow users to control traffic flow, vehicle speeds, road blockages, etc. Since our simulation is an approximation of how Georgia Tech's road network behaves on average, it will not be the best tool for users who wish to understand traffic flow at Georgia Tech at a specific point in time. Therefore, by giving the user the ability to simulate the specific conditions with an added user interface, they can gain the exact understanding of Georgia Tech's road network that they need to accomplish their tasks. Last but not least, future work could also consider using different simulation models like a logic-based model, cellular automata, machine learning approach, etc.

References

Andrade-Michel, A., Ríos-Solís, Y. A., & Boyer, V. (2021). Vehicle and reliable driver scheduling for public bus transportation systems. Transportation Research Part B: Methodological, 145, 290–301. https://doi.org/10.1016/j.trb.2021.01.011

Ding, D. (2011). Modeling and simulation of highway traffic using a cellular automaton approach.

Modelação da Procura de Transportes. Técnico Lisboa. https://fenix.tecnico.ulisboa.pt/disciplinas/MPTra/2020-2021/1-semestre/materiais-de-apoio

Pell, Andreas & Meingast, Andreas & Schauer, Oliver. (2017). Trends in Real-time Traffic Simulation. Transportation Research Procedia. 25. 1477-1484. 10.1016/j.trpro.2017.05.175.

Schmaranzer, David & Braune, Roland & Doerner, Karl. (2020). Population-based simulation optimization for urban mass rapid transit networks. Flexible Services and Manufacturing Journal. 32. 10.1007/s10696-019-09352-9.

Sewall, J., Wilkie, D., Merrell, P., & Lin, M. C. (2010, May). Continuum traffic simulation. In *Computer Graphics Forum* (Vol. 29, No. 2, pp. 439-448). Oxford, UK: Blackwell Publishing Ltd.

Tapani, A. (2008). Traffic Simulation Modeling of Rural Roads and Driver Assistance Systems (Ph.D. dissertation, Linköping University Electronic Press). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-12428

Treiber, M., & Kesting, A. (2017). *The intelligent driver model with stochasticity-new insights into traffic flow oscillations.* Transportation research procedia, 23, 174-187.

Wang, W., Yuan, Z., Yang, Y., Yang, X., & Liu, Y. (2019). Factors influencing traffic accident frequencies on urban roads: A spatial panel time-fixed effects error model. PLOS ONE, 14(4). https://doi.org/10.1371/journal.pone.0214539

Simulation Resources

Himite, B. (2021, September 7). *Simulating traffic flow in Python*. Medium. Retrieved April 28, 2022, from https://towardsdatascience.com/simulating-traffic-flow-in-python-ee1eab4dd20f

iMinichrispy. (n.d.). *IMinichrispy/GT-Buses: An IOS app for tracking the Georgia Tech Buses: Http://gtbuses.iminichrispy.com*. GitHub. Retrieved April 28, 2022, from https://github.com/iMinichrispy/GT-Buses

Microsimulation of traffic flow: Onramp. Microsimulation of Traffic Flow: Onramp. (n.d.). Retrieved April 28, 2022, from https://traffic-simulation.de/

Road traffic simulation software. – AnyLogic Simulation Software. (n.d.). Retrieved April 28, 2022, from https://www.anylogic.com/road-traffic/

Wikimedia Foundation. (2022, February 18). *Intelligent driver model.* Wikipedia. Retrieved April 29, 2022, from https://en.wikipedia.org/wiki/Intelligent_driver_model

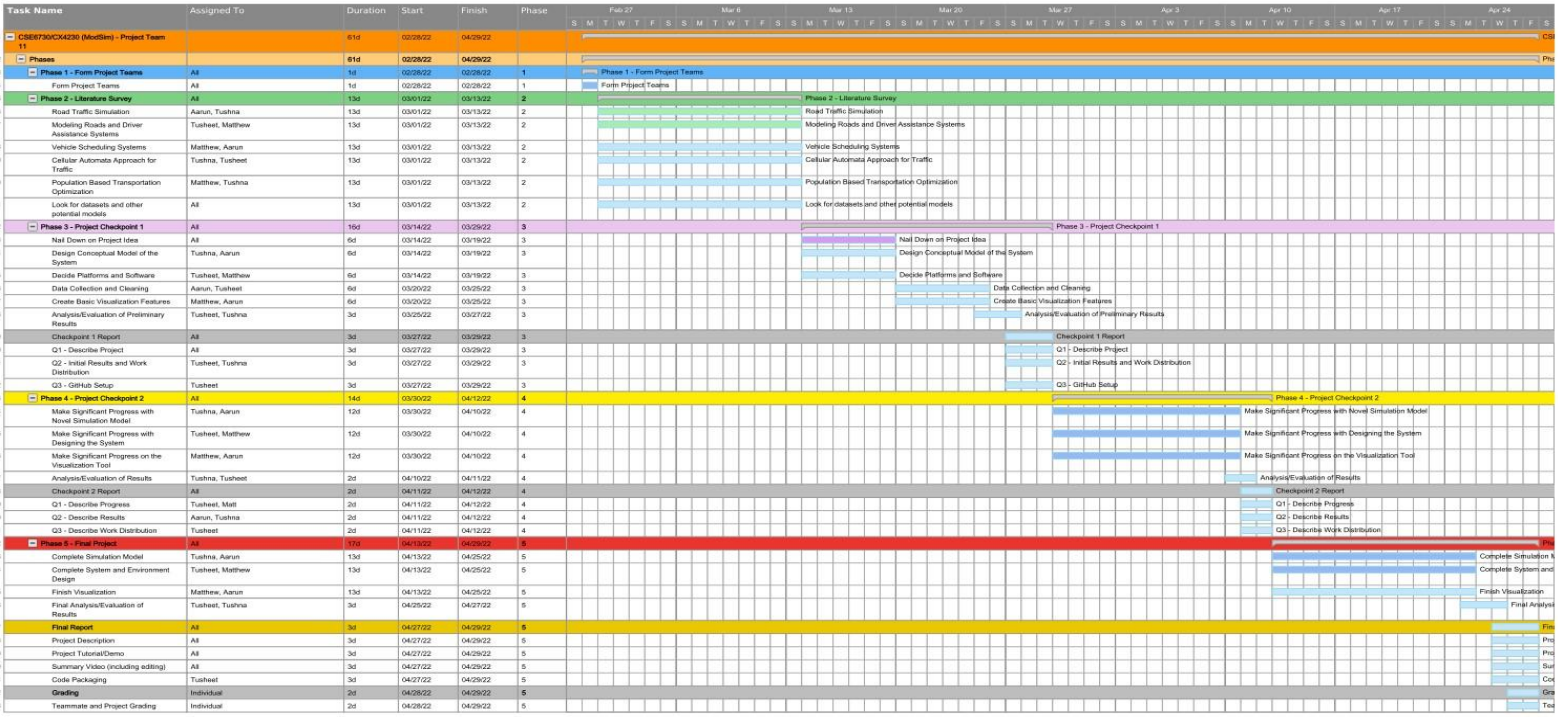
Datasets/APIs

Marta. (n.d.). Retrieved April 28, 2022, from https://www.itsmarta.com/app-developer-resources.aspx

Muonneutrino. (2017, August 2). *Wikipedia traffic data exploration*. Kaggle. Retrieved April 28, 2022, from https://www.kaggle.com/code/muonneutrino/wikipedia-traffic-data-exploration

Perez, A. (n.d.). *GT Buses API documentation*. GT Buses API Documentation. Retrieved April 28, 2022, from https://gtbuses.herokuapp.com/

Appendix: Division of Labor



The chart is also available on our team's GitHub repository - https://github.gatech.edu/tgoli3/gt-modsim-s22-team11/blob/main/charts/Gantt%20Chart.jpg

We have created a detailed Gantt chart to showcase all the progress we have made for this project and the task we plan on completing for the upcoming checkpoints with appropriate due dates and members assigned. For a brief summary, we have divided the entire project into five phases based on the five deliverable deadlines for the project. Please refer to the Gannt chart for more details.

Phase 1 - Form Project Teams (Done)
Phase 2 - Literature Survey (Done) - Gantt chart shows work distribution for the assignment
Phase 3 - Project Checkpoint 1 (Done) - Includes finalized project idea, conceptual model, platforms/software of development, and initial data cleaning and collection.
Phase 4 - Project Checkpoint 2 (Done) - Coded out the system being studied (GT road maps) within the bounds of the proposed conceptual model. We also got our basic simulation working (pygame) and are currently working on adding UI features that can change and disrupt the natural traffic flow to analyze the impacts this has. This is our main analysis and novelty aspect.
Phase 5 - Final Project (Done) - Complete implementing all the UI features, culminating progress and efforts from previous phases to deliver a complete project with a project report (with final results and analysis), tutorial/demo, and code packaging. All these tasks have been equally split up among the team members.

1. Tushna Eduljee - Worked on creating the simulation engine and pygame along with some road rules. Also, set up road intersections, traffic lights, stop signs, and traffic rules.
2. Matthew Dacey-Koo - Put a lot of effort into replicating the Georgia Tech road map. He also figured out the models for the road and road systems and implemented traffic flow features for the roads. He also analyzed the experiments and validated our results from available data.
3. Tusheet Goli - Helped with backend setup, set the codebase, and worked on implementing features like the simulation engine and road blockages. Implemented some of the simulation and system model and other UI features like blocking roads, modifying car speeds, creating diversions, etc.
4. Aarun Srinivas - Worked on creating the GT roadmap and getting the car flow working on this. Also helped with the simulation engine and model design.

All team members equally contributed to the final report document and the presentation/demo video. All work was evenly distributed and all members generally did an equal amount of work.