**@TrafficSim Data Collect**

**CSCI 6838 – Capstone**

**December 11, 2019**

Dilhar DeSilva

AtLink

Paul Bauer

AtLink

Dr. Gary Boetticher

University of Houston – Clear Lake

Mohammed Ausaaf Arshad

Arshad.ausaaf@gmail.com

Lydia Bierer

lbierer@gmail.com

Arif Lakhani

Ariflakhani7866@gmail.com

**Acknowledgements**

We would like to thank both of our mentors, Mr. Dilhar De Silva who is the Chief Technology Officer at AtLink and Paul Bauer who is a Quality Assurance Tester at the same organization. It is because of their continuous support, assistance and guidance that helped us complete the project. Paul is always very cheerful and helpful. He is very responsive to email and text communications. Our weekly meetings were quite informative due to the assistance of Dilhar and Paul.

A big thank you to Dr. Gary D. Boetticher, who is the guide for our project. He is an Associate Professor in Computer Science at the University of Houston – Clear Lake. Under his supervision, we learned a lot and his knowledge in the field of databases has helped us grow a mile.

**Table of Contents**

**Topic Page #**

**Introduction ……………………………………………………….……… 01**

**Market Analysis .…….…………………………………………………… 02**

**Requirements ……………………………………………………………… 03**

**System Architecture ……..………………………………………………… 04**

**System Design ……………………………………………………………… 05**

**Implementation ……………..……………………………………………… 06**

**Milestones …………………………………………………………………... 07**

**Conclusion …………………………………………………………………. 08**

**Future Work …………….………………………………………………… 09**

**References ……………….………………………………………………… 10**

**Glossary ……………………………………………………………….……. 11**

**Introduction**

As any motorist who has traveled on Houston’s expansive network of highways and roads can attest, traffic can be a huge source of frustration and problems. One example of an ongoing problem is that road construction is a constant in this city, the Unites States’ fourth largest. The project at large, of which our capstone contribution is a part, consists of taking state of Texas supplied traffic data and creating a simulation of said traffic using the UnReal game engine. This simulation produces a lot of data of which is lost once the simulation is stopped. Our mission was to capture the data produced in the simulation, populate several tables of a database and tentatively supply a graphical representation of the data of which later work can be built. Although outside the scope of our project, many analyses can be performed on this data once it has been collected and stored.

**Market Analysis**

There are several traffic simulation software packages available. They include: Matsim, SUMO, Aimsun, PTV Vissim, and FreeSim. They differ in their features and affordability. The following is a brief description of each.

MATSim, which stands for Multi-Agent Transport Simulation, “is an open-source framework to implement large-scale agent-based transport simulations and transport planning models” [1]. A developer can add to the repository of knowledge by writing Scripts in Java to create one’s own extensions. Here is where one can envision storing data created by the simulation into a database.

Simulation of Urban Mobility (SUMO) “is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks” [2]. The following data is produced by the simulation and sent to an output file: “edge lane traffic: edge or lane-based network performance measures, trip information: aggregated information about each vehicle's journey, vehicle routes information: information about each, vehicle's routes over simulation run, simulation state statistics: information about the current state of the simulation” [1].

The description from their web site stated: “Aimsun Next software allows you to model transportation networks small and large: from a single intersection to an entire region” [3]. These following data are collected and sent to an output file: “statistical simulation results…, output database definition, calculation of traffic statistics, experiment output, and map output” [1].

PTV Vissim “is a microscopic multi-modal traffic flow simulation package that allows you to simulate exact traffic patterns and displays all road users and their interactions in one model” [1]. The following are the data that the Vissim simulation includes in an output file: “scenario Management: Comparing scenario results conveniently, enhanced evaluation possibilities: Level of Service results for nodes, improved handling of the matrices through matrix editor, visualization highlights, vehicle record, vehicle travel times (raw data), pedestrian record, nodes (raw data), and signal changes” [1]. Again, the reviewers did not mention what type of file the output data file consisted of therefore, it would be difficult to ascertain as to whether output simulation data has been fed into a database.

“FreeSim is a freeway traffic simulator licensed under the GNU General Public License (GPL). FreeSim’s graphical user interface (GUI) runs within a web browser using the Adobe Flash plug-in and connects to a Java-based server application via a socket. The Java code uses a MySQL database for any persistent storage needed. FreeSim allows for different freeway systems” [4]. As mentioned in the previous sentence, FreeSim does have a mechanism whereby data produced by the simulation can be stored in a database.

**Requirements**

Here are the following requirements for the project:

1. The Vehicle Table must include a vehicle ID, location in X, Y, Z coordinates, current speed, and time from start of simulation.

2. The Collisions Table must include a collision ID, both vehicle ID’s (primary and secondary vehicle), location (X, Y, Z) where collision would occur, and time stopped for primary vehicle.

3. The IntersectionPoint (AtPoint) Table must include intersection ID, number of vehicles passed through intersection, and maximum speed of vehicle passing through the intersection.

4. The Intersection (AtPoint) Table must include currentIntersection ID, next Intersection ID, and time to get to the next intersection.

5. There must exist a mechanism that connects the data created in the UnReal simulation to the MySQL database.

6. Data should be graphically displayed on a webpage by using a package such as Google Charts.

**System Architecture**

Visual Studio with Game Engine codebase interfaces with UnReal 4.18 Editor. Code in Visual Studio was analyzed in order to find the variables that would populate the database when the simulation was run. Once the simulation was run, the next step was to collect the real time data about collisions, vehicles, and AtPoints (Intersections where the vehicles pass through periodically). This information is then fed into the database via the API using HTTP to send JSON objects. But before the values are stored in the MySQL database, the data needed to be parsed and converted into the appropriate data types for later analysis. Once the data was stored in the database, the next step would be to analyze the data to look for patterns in the traffic. One way to do this is to represent the data graphically through the use of a data dashboard. The flow of this process is graphically depicted below in Figure 1.0.

A picture containing text, map

Description automatically generated

**Figure 1: Architecture**

**System Design**

As for the database design, the initial design the team came up with is depicted in Figure 2.0. As shown in this class diagram, there are four tables. The Vehicles table concerns itself with the individual vehicles, their location, speed, and time from simulation start. The Collisions table involves capturing the length of time stopped, the two vehicles involved in the collision as well as their location of the collision. The two intersections tables are related. The Intersections table is just a list of all the AtPoints and their potential subsequent AtPoints as well as the time to reach those points represented in seconds. The IntersectionPoint table captures information about the maximum speed traveled through a particular AtPoint as well as providing a count of the number vehicles to pass through the AtPoint.

A screenshot of a cell phone

Description automatically generated

**Figure 2: Design**

**Implementation**

The most interesting aspects of implementation is the visual representation of all the data generated by the simulation. Before beginning the project, once the simulation ended, there was no way to analyze what had happened in the simulation. By capturing and storing data created in the simulation, a pathway to analysis is created. Some examples of analyses performed might be answering questions such as, “at which region (represented by a range in coordinates) has the greatest number of collisions?” or “How many vehicles pass through a particular *AtPoint*?

The program was verified by viewing the output on the course website and by viewing the graphical representations of the data.

**Milestones**

This section describes the project milestones, their due dates, and their current status. The milestones in this report should match those on the team web site. If any milestone was late or not completed, you may provide an explanation.

**Key code:**

S1 = Mohammed Ausaaf Arshad

S2 = Lydia Bierer

S3 = Arif Lakhani

**Table 1: Project Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Due Date** | **Activity** | **Percent Contributions** | **Status** |
| Sept. 18 | Finish website | S2 (80), S3 (20) | **Done: Sept. 18** |
| Sept. 19 | Created class diagrams | S1 (40), S3 (60) | **Done: Sept. 19** |
| Oct. 3 | Created API | S1 (40), S2 (20), S3 (40) | **Done: Oct. 3** |
| Oct. 10 | Build the database | S1 (50), S3 (50) | **Done: Oct. 10** |
| Oct. 17 | Identified variables to insert into Db | S1(20), S2 (60), S3 (20) | **Done: Oct. 17** |
| Oct. 24 | Created and tested the tables | S1 (30), S2 (30), S3 (40) | **Done: Oct. 24** |
| Nov. 15 | Completed the front end | S1 (20), S3 (80) | **Done: Nov. 15** |
| Nov. 29 | Data Analysis | S1 (35), S2 (35), S3 (30) | **Done: Nov. 29** |

**Conclusion**

By capturing and storing the data created by the traffic simulation, one can keep tabs on the behavior of the simulation. From this data, predictions can be made. And later down the road, features of the simulation can be altered to simulate road closures. Then data can be collected and analyzed to see the effect of the road closures on the traffic that is simulated. The overarching goal might be to inform stakeholders such as the State of Texas the effects of making real life changes that affect traffic flow. For example, TXDOT might close off a particular road entirely to perform road construction. If the resulting detour created bigger problems for motorists perhaps in the form of delays and increased collisions, officials might decide to take a different tack of reducing that road’s lane count by one and working on the road in pieces rather than to shut it off entirely.

Weekly meetings with Paul kept our team on task. We utilized a myriad of communication methods to include Trello.com, email, text and Whatsapp. It seemed that a lot of time was taken during the planning stages to ensure that we understood the parameters of the project. This left less time for matters such as debugging code and perfecting the tables. Finding documentation for the UnReal engine proved challenging. This project required us to fit together several seeming disparate pieces of a complex puzzle. It is our opinion that the Capstone project which allowed us to work with industry mentors provided us with invaluable experience and insight into the software development process. We look forward to what the future holds as we all embark on our separate journeys post UHCL.

**Future Work**

Extending the graphical representation of the data is one area in which more work could be performed. Additionally, our mentor realized later that there were some variables from the simulation that he would have liked to have included in our tables. It was not until after were finished that this realization happened. Just when one thinks that he or she has accounted for most conditions of traffic, there seems to always be more to consider.

**References**

Please see the *ACM Citation Guidelines* document for details on how to set up references.

[1] Ejercito, Paolo M., et al. Traffic simulation software review. *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017.

[2] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. *Recent Development and Applications of SUMO - Simulation of Urban MObility*. International Journal On Advances in Systems and Measurements, 5 (3&4):128-138, December 2012.

[3] Siemens. Aimsun.next: Your Personal Mobility Modeling Lab. Retrieved December 8, 2019 from <https://www.aimsun.com/aimsun-next/>

[4] Miller, Jeffrey, and Ellis Horowitz. "FreeSim-a free real-time freeway traffic simulator." *2007 IEEE Intelligent Transportation Systems Conference*. IEEE, 2007.

**Glossary**

**AtPoint** – this is AtLink’s term for intersection point along a highway or road that provides a reference when a vehicle passes through it.

**Macroscopic Traffic Flow** – concerns itself with data such as density, flow, mean speed of a traffic stream.

**Microscopic Traffic Flow Model** – provides models of single-vehicle units