**Programmer Group 1 10/10/2018**

[**https://github.com/paleohayduke/TrafficSimProject**](https://github.com/paleohayduke/TrafficSimProject)

**Report 2: 10/10/2018**

## Executive Summary

The architecture of our project is based on the idea of MVC, model-view-controller. We have a Model component that simulates the structure of roads and the automobiles that run on them. In this model, everything is measured in latitude and longitude and all data are stored in arrays of Roads, Nds and Auto objects. Our View consists of the Renderer and its subcomponents. The Renderer takes latitude and longitude coordinates of all entities in the physical model (roads and cars) and translates them into pixel coordinates on your screen. The renderer supports a 1:1 simTime:realTime mode as well as a fast-forward mode. The renderer allows you to scroll your view around the map by right clicking and zooming in and out with the mouse wheel. The renderer all supports capturing mouse coordinates for processing by the controller. The controller component of the system is the Simulation class. It controls the flow of time in the physical model by “stepping” through segments and telling the model to update its state and sending live model data to the Renderer.

Since the last report we have completed requirements, 1, 2, 3a,3b, 4,4a, and currently working on 5. The program was experiencing significant delays while calculating routes with A\* for the entire copperas cove/Killeen/ft.hood/nolanville area which motivated a reworking of the route generation and simulator classes to operate as separate threads sharing a buffer. The A\* algorithm was also improved by searching for minimum in the “queue” instead of keeping it continuously sorted. UML class diagrams have been completed for most components of the system. An effort must be started to create documentation for use of the system.

## Experiment

Experimentation during this phase has largely consisted of running the latest builds of the simulation and searching for bugs. By watching large quantities of traffic it became evident that certain node intersections were not linking with their neighbors correctly. This problem resulted in the development of a number of GUI tools to select nodes from the screen (to convert pixel cords to lat and longitude and select the correct node).

Introduction of stop signs, by themselves, went without a hitch but introducing collision detection (with calcDistance()) led to dead lock issues. The reason for the deadlock was found to be an incorrect ordering of a sequence of if else statements.

A SimClock class was designed to provide the Simulator with a way to keep track of simulated time. This clock tool allows us to do things like calculate the traffic flow rate as well as things like average length of trip. The ability to track time will aid in analysis of further experiments and helps fulfill project requirements.

Loading time was a common complaint and several methods were attempted to improve it. The improvement of A\* fixed a lot of the load time but map processing still took a considerable amount of time. The current method to improve load time involves processing the maps for use by the program and then saving them to file(2500 pre-generated route files were also created and saved) and then loading these on start up.

There has been one test on a method to display the traffic in separate lanes using offsets from the line. The process involves finding the reciprocal of the slope of the cars velocity vector.

Car accidents are also implemented and have been undergoing testing. So far they work as expected, the car stops and the traffic behind them is blocked. You can toggle accidents on and off.

## Results

The introduction of stop signs resulted in a second look at the tree structure that navigation is based on as well as the code controlling the Automobiles themselves. This work led to the successful implementation of stop signs as well as discovering other weaknesses and error in the code.

The program will rely heavily on the SimClock when performing traffic analytics (rates, etc…). Currently it can convert simulated time to the number of “ticks” that the program has run but needs further functionality(calculating hours between events, more sophisticated alarm system that Simulation can use to drive events). The clock also resulted in the introduction of a “set time” feature. The set time feature actually continues and speeds up the simulation for the amount of time to reach the new set time… this feature works well on small to medium size maps (1/4 size of Killeen) but is rather slow on the full size map of the entire region.

The change to A\*’s queue helped improve that specific component of the algorithm from nlogn to n. The performance boost from this is very noticeable. While it used to take about 15 minutes to load map and generate cars, it can now be performed in 36 seconds on our largest map.

Moving the route generation to a separate thread(s) greatly improved performance. We used the producer-consumer model to design the interaction of the threads. A number of producer RouteGenerator threads create routes based on current map and traffic levels (the A\* also takes traffic density into account) and store them to a buffer that Simulation is able to pull from to give an Auto a route.

The results of experiments on improving loading time reduced loading time from 15 minutes to 36 seconds.. This is so many orders of magnitude better than our first run of the program that took 45 minutes to initialize!

Currently the offset is processed by the Renderer which leads to skips in smooth Animation of the cars. A potential fix is to instead apply the offset to the actual waypoint that the car is following… when it reaches the adjustd waypoint position the waypoint will cycle and the car will travel towards the next way point in the appropriate lane. Once this work is done we can work on implementing multilane roads and a proper stop light.

## Conclusion

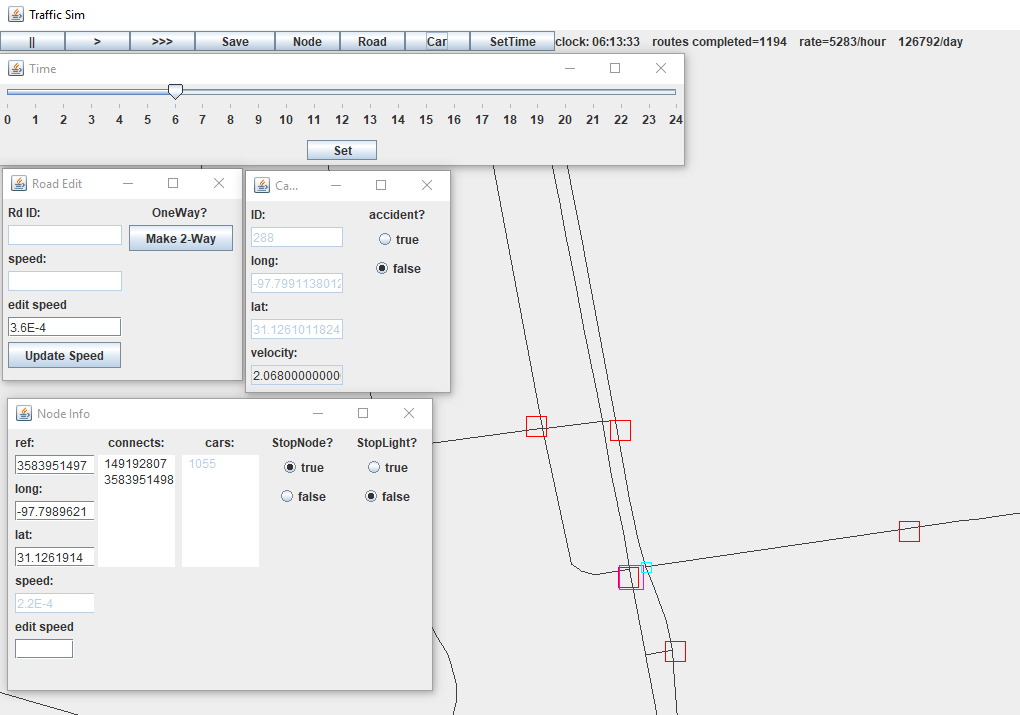
With most of the major project requirements met, we can begin to focus on designing tools to analyze traffic patterns and give recommendations to the user. The work on tools to debug the nodes can be built on to develop full fledged map editing tools (currently there are functions to convert a one way road to two way, change speed limits, enable/disable stopsigns and enable stoplights(not implemented yet)).

We also need to go through and eliminate all magic numbers; the variables they will be set to can be used in an options panel for live adjustment.

We also have a UML class diagram of the main components of our system included in the Appendix 1 of this document.

Appendix 2 includes the readme file that accompanies the project on our GitHub and includes all active Tasks along with all Completed tasks with dates.

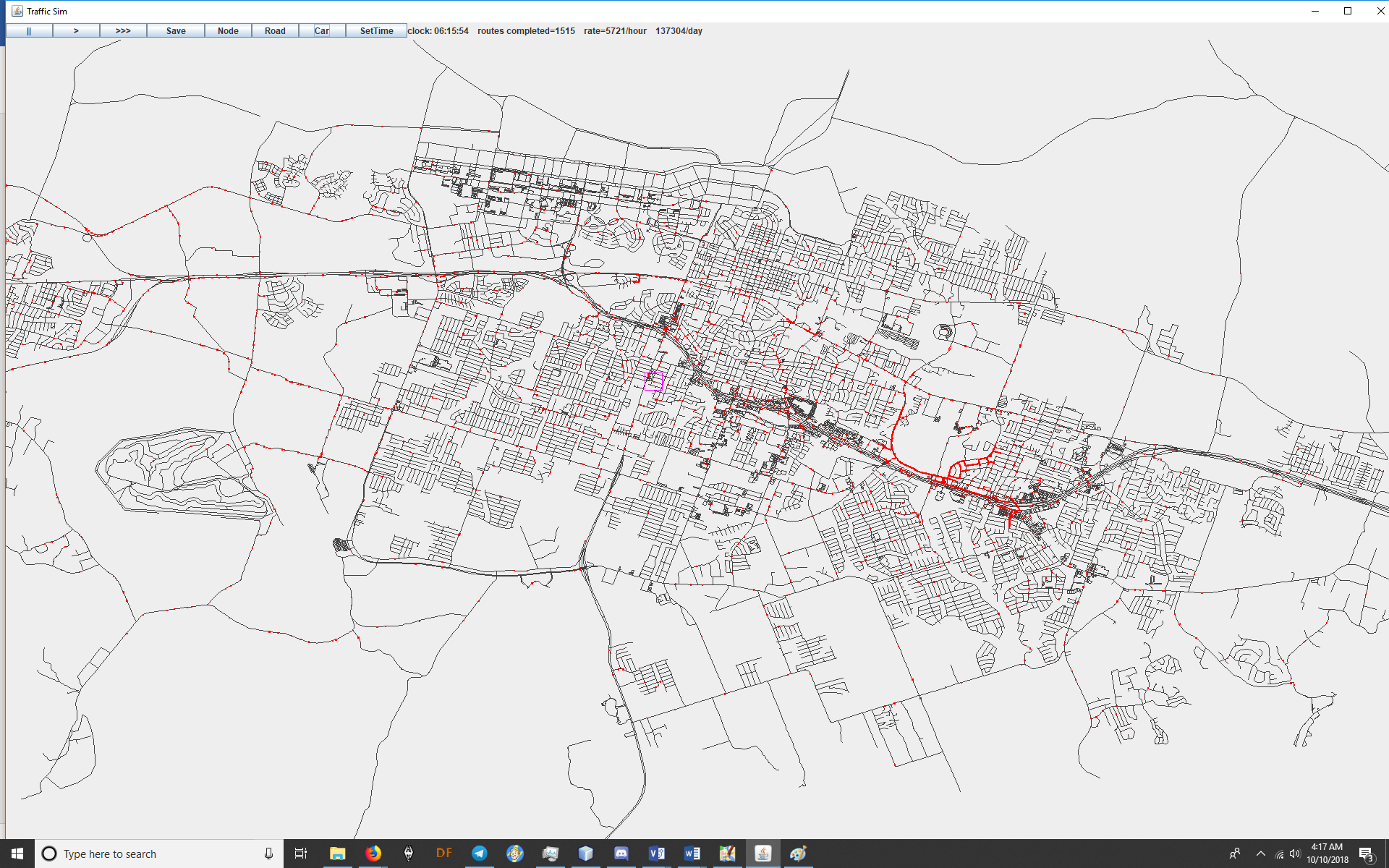
*Appendix A*

**

Tools so far.

https://cdn.discordapp.com/attachments/483778271104991256/499071472342663168/unknown.png

Example of total traffic rate.



Simulation on full size map. The red are cars.

*Appendix 1*



UML class diagram. Need sequence and state diagrams.

*Appendix 2 Project README*

*\*The follow document is a diary of the project. It includes both the planning and posting of tasks related to development of the project as well as marking the dates of their completion. Initial date that task was*

*Issued is in Announcements section… Need to turn this document into a Gantt chart.*

**# TrafficSimProject**

**TAMUCT COSC-3320-110 Programmers Group 1**

**TODO:**

-4 lane roads and visual road in correct lane offset. Lanes determined between

waypoint and lastwaypoint (between each two nodes). Calculate point a specific

distance perpendicular to a point on a line!

https://stackoverflow.com/questions/17195055/calculate-a-perpendicular-offset-from-a-diagonal-line

-Create simulation method to start the program paused without loading cars.

Give options to load map, pick number of cars and any other start options.

-Clock needs a "how many hours since (time)" function.

-Average trip length display

-Editing connections between nodes. Make connection display in NodeInfoFrame

a selectable list to jump between nodes.

-multi lane roads

-Display cars offset from center of road depending on their lane.

-Car passing

-Build Stop lights. manual placement with node info tool

-Ability to add new nodes

-Merge current tool windows into a menu that docks on the side of the window.

(like photoshop).

-Toggle hiding a node from pathfinding and eliciting reroute from cars

attempting to route through them.

-Implement "major disaster" scenario. Click a spot on the map and every car

will run away from it and attempt to path OFF of the map (evacuation simulation).

-Connect to GUI

-Set Git up to keep track of revisions.

**ANNOUNCEMENTS:**

10/9/2018 Cars are now offset to the right of the road to represent separate

lanes. Offset is in renderer... maybe do it in actual physical sim of world

to more easily handle turns (they "jump" at the turns now).

(consider this an EXPERIMENT)... I think instead could do this in the physical

model... insteadof calculating offset for the cars position every step... set

offset on the waypoint at begining of route leg and let car path to appropriate

location on its own.

10/9/2018 Can save processed map files as well as batches of direction files.

Loading time is a few orders of magnitude better.

Changed sort to find min in A\*. Fixed node intersections: they are using highways

now. Bunch of other little bug fixes.

10/8/2018 https://www.kcentv.com/article/news/local/killeen-neighborhood-it-takes-20-minutes-to-get-out-of-our-subdivision/600300276

The sim predicts this. The area in the link is consistently the most congested area

in MediumMap.osm.

10/7/2018 concurrency enabled: multiple route generators, running on seperate

threads, keep a buffer full of generated routes.

10/7/2018 A\* mode that takes speed limits into account... uses time instead of

distance as its metric.

\*Adjusted the traffic congestion heuristic on the A\*. make slider to adjust this

from a menu.

10/6/2018 Set time frame with a hour select slider introduced. Fast forwards to

whatever time you select. A car selector that draws a rectangle on a clicked car

if you have the car frame open. You can now set speed limits of individual nodes

from the node info screen...I'll make a tool to select groups of nodes by mouse

and by entire roads.

You can toggle accidents on a car to cause it to stop and block traffic.

Once we finish the current TODO list, we will have fulfilled all project

requirements.

Traffic Flow Rate is displayed in main window toolbar.

Roads now set speed from OSM file depending on whether they are a trunk,

primary, secondary or residential.

10/3/2018 Mouse wheel now zooms map while keeping focus of object under the

mouse cursor (just like Google maps). Pretty sweet.

10/2/2018 https://www.openstreetmap.org/edit?editor=id#map=19/31.08832/-97.71982

Register an account and start updating the map with traffic lights.

Mouse wheel zoom and map movement implemented. Zoom needs work.

10/1/2018 There are a few buttons, pause, play, fast forward.

Fast forward is button clicks^2.

Mouse position is now converted to correct long and lat. Red square dropped where

the engine thinks you are in lon&lat. Zoom has been fixed.

9/30/2018 Stop signs are working. Time for stop lights.

Cars now keep distance between each other.

A\* now takes into account traffic levels at time of route generation!

Pause and play functionality.

Placeholder Frames to hold data and select options-still empty.

9/29/2019 Consolidated intersections into one node. Next task is making stoplights

\*Nope... that broke a lot of intersections. set it back to old technique.

One way roads are now implemented. This causes an issue if u set the cars

to generate new routes from their destination waypoint if they just got to the

end of a oneway road that dead ends at the end of the map and a few other

circumstances. To deal with this i set up cars to spawn at a random new starting

location when they finish their route.

Pathfinding now takes distance traveled into account (A\*). Rebuilt GraphBuilder

to consolidate intersections into single nodes. Next step is stops.

9/28/2019 Figured out the methods of JComponent a little better and was able to

fix the frame rate issue in the Renderer. Works a lot better now!

List of nodes added to Nd that link to all cars who hold the node as a waypoint.

Might carry previous node to help figure orientation for handling stop signs.

Stop lights should be much simpler. Idea: build stop lights or stop signs

with mouse... map building tools.

New bug, sometimes the cars will cycle through more than one node when

updating waypoint.

-Check on the... d/D, im thinking its when D gets near 0 or something.

Pathfinding causes hiccups in frame rate on the full size Killeen map when

Random() picks two points that do not connect, leaving the algorithm to

(sometimes) search the \_entire\_ map.

I was thinking about signal time length for stop lights and started to

wonder, "should we add pedestrians?".

https://nacto.org/publication/urban-street-design-guide/intersection-design-elements/traffic-signals/signal-cycle-lengths/

A lot of the decisions on timing schemes depends on vehicle traffic

AND pedestrian traffic.

9/27/2018 The automobile can now update its position based on time and

velocity, this means it now moves smoothly between nodes.

You can now spawn variable amount of cars at once, setCars(int numOfCars).

Look at main() in TrafficSimProject for more details.

9/26/2018 It looks like routeFind() is fixed. You can enter a start node and

end node and it will automatically generate a path and animate the car taking

the nodes between both points. A new bug has come up, when you first run the

program, if you use the medium map or larger, the window will not update the

animation until after you MINIMIZE the window and then open it back up.

The renderer needs to be rebuilt and a controller/handler for setting up cars

needs to be written. The car controller needs to be able to generate a

defined amount of cars and give all of them random start and end nodes. The

controller needs to hold the cars in a list and update them all iteratively,

every "step". After they are all updated a list of all the cars positions needs

to be passed to the renderer to update the next frame.

9/26/2018 The demo now animates in a loop. I added a debug() method to the

Auto class. call the debug method after your car already has a waypoint assigned

to its waypointNode and you can manually traverse the nodes. waypointNodes

represent the next node the car is supposed to take on a route.

Debugging of the node system revealed that the map nodes are all correctly

linked. Since the node system is fine, the bug is most probably in

Direction routeFind(). \*see BUGS

9/24/2018 DEMO:

Once the demo loads, you can press a key into the console and hit enter to cause

a car to take a "step". Each step will traverse one node towards the car's

destination. The code is in the Simulation class under demo().

9/24/2018 There is a new Directions class. It has some stuff to build lists of

directions for the car to follow. Pathfinding algorithm needs to generate a

list of choices that the car will make at each node. The automobile can take the

direction list and it will travel from a starting node through the route with every

call of the .step() function. The renderer must be called separately to display

the car on the map. When we generate many cars we will update the cars position

in the geographic system and then form a list of coordinates (and possibly orientations)

along with type to send to the renderer to display.

There is a findRoute() in Directions but it is very buggy, see BUGS.

-ww

Cleaned up main and made a Simulation class to make things more manageable.

Need to fix protected vs private vs public on a lot of stuff.

9/23/2018 There individual nodes can be built into a tree structure that gives

every node a pointer to its connecting nodes along with edge weight data. I'm

working on an "Auto" class that can traverse the nodes while keeping track

of its current position. It takes a set of "directions" that are in the

form of a list of integers. each integer represents the index of the appropriate

connecting node.

-ww

9/22/2018 The map is displaying and is finding all intersections correctly. We've got

everything we need to set up a graph of the roads and then implement a

graph traversal algorithm.

-ww

**BUGS**

**COMPLETED TASKS**

COMPLETED 10/9/2018: -Create data structure to save processed and edited map data. Needs to be

enough to let GraphBuilder do its job.

COMPLETED 10/7/2018: -Create thread to handle path finding.

Create an array to hold 10 directions, as new cars are generated, they will pull

from the array to get their directions(disposing of source direction) and the

direction thread will then make a new direction, based on current traffic

conditions, to replace it... If there are no new directions, car can go inactive

until a new route is ready.

COMPLETED 10/7/2018: -makeDirection() in Simulation needs to be able to set a minimum distance

between start and end point.

COMPLETED 10/7/2018:-Adjust A\* to take speed limit into account

COMPLETED 10/7/2018: Set speed limit from OSM file.

COMPLETED 10/7/2018: change speed limits by Road

COMPLETED 10/6/2018: Traffic traffic load and flow rate. Increment counter as cars reach destination

and respawn.

COMPLETED 10/6/2018: Clock & Set Time

COMPLETED 10/6/2018: Speed limits \*by node

COMPLETED 10/3/2018: -Center camera to mouse on zoom

COMPLETED 10/3/2018: When hit detection is turned on, deadlock can occur at stops. Currently using

a bandaid to deal alleviate symptoms. Investigate and fix root cause!!!

COMPLETED 10/2/2018: -Zoom and map scroll by mouse button hold.

COMPLETED 10/2/2018 BUG: fix california stoppers

COMPLETED 10/1/2018: -Test play and pause button functionality.

COMPLETED 10/1/2018:-Test mouse interaction.

COMPLETED 10/1/2018:-Function to find a node based on its longitude and latitude.

COMPLETED 10/1/2018 BUG: -Fix the hit detection bug!!!!!!!!!!!!!!!

COMPLETED 10/1/2018 BUG: Map is displaying upside down because origin (0,0) is top left corner for the JFrame. Need to flip vertically.

COMPLETED 10/1/2018: zoom broken since redesigning renderer

COMPLETED 9/30/2018: pause/play functionality

COMPLETED 9/30/2018: Stop signs.

COMPLETED 9/29/2018: Link to Road from nodes.

COMPLETED 9/29/2018: Change intersection node set up.. currently every intersection is made up of

two nodes... need to change to ONE node at each intersection... this will greatly

simplify stop signs and stop lights.

COMPLETED 9/29/2018: Alter localRoadBuilder to implement unidirectional node connections for

one-way roads.

COMPLETED 9/28/2018: Sometimes the cars will cycle through more than one node when

updating their waypoint. Might be a problem with node connections.

\*problem was with updating the waypoint and posNode of auto.

COMPLETED 9/28/2018: Fixed frame rate issue and the minimize-maximize bug

Renderer needs to rebuilt. Its doing a lot of unnecessary work.

COMPLETED 9/27/2019: Implement update of car position between nodes. Solve for x and y from

distance formula, Pythagorean theorem, vectors.

COMPLETED 9/26/2018: Implement a graph search.

COMPLETED 9/26/2018 BUG FIXED: findRoute() sort of works. Need to check if the problem is in the algorithm (likely)

or something with the linking of nodes crossing roads (in some test cases a path

will be generated that involves changing roads at intersections, but sometimes

the algorithm gets stuck in an infinite loop, need to review the code but im too

tired atm).

COMPLETED 9/23/2018: Building a tree out of the road and intersection data.

COMPLETED 9/22/2018: Renderer needs to be reworked. I just slapped it together to see if the program was getting

the coordinates of nodes correctly. This needs to be fixed so we can visually see what we are doing.

COMPLETED 9/21/2018: Determining order of nodes in a road. This is important for drawing the roads correctly on the map.

I am thinking going through list of nodes and then doing distance formula distance=sqrt((x1-x2)^2+(y1-y2)^2)

to find the shortest distances to two nodes from and one node in a road will work to figure their correct order.

There are lots of nodes on curves so that shouldnt cause a problem but some of the roads will be "ends" and only have one node connected to them.

COMPLETED 9/21/2018: Reimplement the method "private Nd makeND(long ref){ }" inside the "MapReader" as a binary search.

\*important to test the program with a map of the entire city of killeen to fullfil requirements.

O(n) vs O(log(n)) on test map at 28,000 nodes, that is 28,000/14 = 1000 times faster.

*Appendix 3 Meeting Minutes*

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Atendees** | **Discussion** | Meeting Times |

|  |  |  |  |
| --- | --- | --- | --- |
| 1-Sep | Ervin, Vincent, Michael, Juana, Walter. | This day we discussed potential project ideas that might make the job smooth and easy for the group. Also identified Unity as our user interface. | 1:30 hr |
| 10-Sep | Ervin, Vincent, Michael, Juana, Walter Va. | Added more to the discussion of potentially making this project with Unity. Ended up changing the User interface since manager did not allow us to  use such interface. Also identified jobs for all group members. | 1:00 hr |
| 17-Sep | Ervin, Vincent, Michael, Juana, Walter Va. | Requirements were given for project and group started to code the program. Currently in the design phase as group member are still hatching out  the details for program. Finished up disseminating jobs for group member and identifying strengths and weaknesses and use them towards our advantage. | 1:00 hr |
| 24-Sep | Ervin, Vincent, Michael, Juana, Walter Va. | Continued working on coding for the program utilizing netbeans. Will meet again on wednesday after class to touch up and identify potential upcoming assignments/task for the projects, group was also discussing time issues within the nodes and how slow the start up of the map was. For how slow it is we were thinking of saving the actual map into a database to speed up the process | 30 mins |
| 26-Sep | Ervin, Vincent, Michael, Juana, Walter, Va. | Discussing nodes and verifying how to make map deal with the distances being traveled along the roads. Making function statements will help with this issue. Identifying and see if we will use the start feature. Working on specific details to go to the next step which is making objects. Also discussed the use of peak times in the map to see traffic being tight around specific places. In addition figure out the GUI format. | 2 hours |
| 1-Oct | Ervin, Vincent, Michael, Juana, Walter, Va. | Discussed what else to do with the map, from adding traffic lights, stop signs and giving cars different colors to differentiate them from each other | 40 min |
| 3-Oct | Ervin, Vincent, Michael, Juana, Walter, Va. | Talked about setting up an UML, and discussed the report and what to put on the appendix. | 1:45:00 hr |

Appendix 3 Project README