

*Traffic Sim: Raynolds Addition / Barelas Stop Sign
Reconfiguration Proposal*

CS 351

Design of Large Programs

Instructor: **Joel Castellanos**

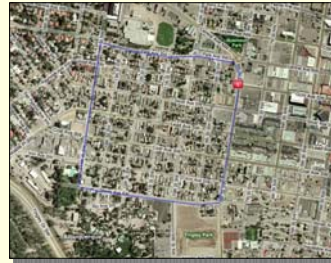
e-mail: joel@unm.edu

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering
Center (FEC) room 321

Lab Instructor: **David Godinez**

e-mail: dgodinez@cs.unm.edu



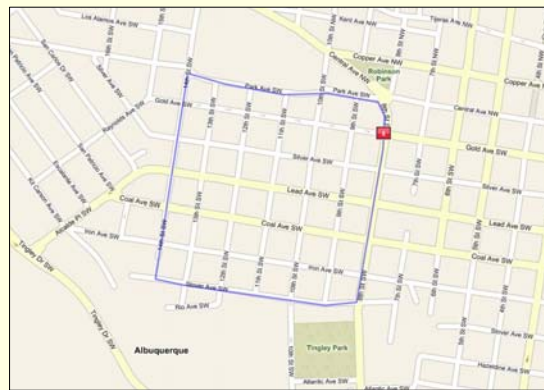
4/11/2009

Traffic Simulation: Topic and Scope

Raynolds Addition / Barelas Stop Sign Reconfiguration Proposal.

<http://www.cabq.gov/council/current-projects-studies/raynolds-addition-barelas-stop-sign-reconfiguration-proposal>

Release Plan 1:
Due Friday,
April 17th

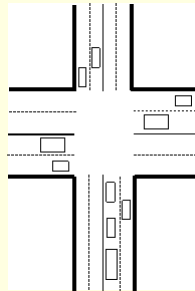


2

Simulation Type

Centrally controlled, object oriented, uniform time-step driven model with graphical user Interface.

We will NOT be creating an agent based model.



3

Agile Design



Planning/Feedback Loops



<http://www.extremeprogramming.org/>

4

Major Design Components

Person assignments are for Release Plan 1 only.

Main Control Class: Interface Control, Collision Detection, and Heartbeat.

- Jonathan M,
- TD,
- Santiago

GUI Class: Graphical User Interface.

- Annett,
- Patrick

Map Class: Roads, Intersections, Traffic Lights, Stop Signs, Speed Aprons, Stationary Obstacles (i.e. cars parked along the road), and Crash Sites.

- Basak,
- Trent

Traffic Objects: Cars, Trucks, City Busses, School Busses, Motorcycles, Bicycles, Pedestrians, and sheep.

- Omar,
- Jonathan B

5

Main Control Class: Ultimate Requirements

1. Entry point of Simulation.
2. Instantiate GUI is separate thread.
3. Instantiate, and provide getter/setter access to Map Class.
4. Provide getter/setter access to the host of Traffic Objects.
5. Provide heartbeat of Simulation. Each heartbeat:
 - a) Ask the Map for a list of newly created Traffic objects.
 - b) Ask each Traffic object where it requests to move.
 - c) If a Traffic object reaches its destination or exits the map area, then destroy that object.
 - d) Collision detection, & instantiate any resulting crash objects.
 - e) Update statistics: total number of crashes, average, and max time waiting at traffic lights and stop signs, ...

6

GUI Class: Ultimate Requirements

1. Be responsive to user controls.
2. Render all objects in the simulation with some simple icon that has a size proportional to its real size.
3. Provide an update method that updates object locations.
4. Provide a way for the user to start, stop, reset, and pause the simulation.
5. Provide an x -real-time setting for the simulation.
6. Provide a way for the user to select preset simulation configurations, and a way for the user to change these configurations.
7. Provide the user with a zoomable view of the simulation objects (perhaps at full-out view draw only roads).
8. Only the Main Control class will interface directly with the GUI.

7

Map Class: Ultimate Requirements

1. Provide getter/setter access to roads and many of their properties such as name, coordinates, speed limit at a given coordinate, and the ID of the next intersection in a given direction.
2. Provide getter/setter access to intersections and many of their properties such as whether it has a traffic light or a stop sign, the state of any traffic light that it has.
3. Provide getter/setter access and construct access to Crash objects and their properties.
4. Create Traffic Objects with an entry point (one of the incoming roads or a house within the map), using real probabilities at real times (i.e. rush hour, school start and let out).

8

Traffic Objects: Ultimate Requirements

1. Each object needs a full set of properties: entry point, destination (exit point), current location, max speed, size, turn radius, stopping distance, max acceleration, driver aggression, etc.
2. Each traffic object provides getter/setter methods for many of its properties.
3. The current location of an object can only be set externally by calling the constructor.
4. Provide an update method that is to be called each heartbeat. Within this method, the traffic object asks the Main Control Class for information about other traffic objects and asks the Map class about the location and state of map objects. Based on this information and its own properties, the *traffic object decides* the location into which it will move at the start of the next heartbeat.

9

Main Control Class: Release Plan 1

- Talk with the people working on the other segments and define a first pass workable interface.
- Instantiate the first pass of the Map Class.
- Instantiate a bunch of traffic objects.
- Implement a timer, and call each traffic object's update method each time the timer fires.
- Print the location (name of road and distance to the nearest intersection) of each traffic object at each heartbeat.
- Present results in PowerPoint form on Friday.

10

GUI: Release Plan 1

1. Release Plan 1 has no implementation of the GUI.
2. Release Plan 1 does not include a GUI layout design.
3. Review the GUI's Ultimate Requirements. Amend, modify, and clarify the requirements list.
4. Develop a first pass at the Interface Design. Including methods you will provide to the Main Control Class, and methods you want the Main Control Class to provide the GUI.
5. Present results in PowerPoint form on Friday.

11

Map Class: Release Plan 1

- Create data structures that represent all of the road names, and their intersections of the Raynolds Addition area.
- Represent all roads as straight lines with approximate distances between intersections that can be quickly estimated from the maps we already have.
- Each intersection that has a Traffic Light or Stop Sign must have one in the model. There is no need to make the red/yellow/green times accurate in release 1.
- Provide an update method to be called each heartbeat that creates traffic objects (cars only) with some probability at each entrance to the area. The probabilities do not need to be accurate.
- All roads have a 25 mile/hour speed limit everywhere.
- Present results in PowerPoint form on Friday.

12

Traffic Objects: Release Plan 1

- Provide a constructor for creating a car with a given *Start Location* and *Destination*.
- Provide an update method that to be called each heartbeat. In this method, check the location of every other traffic object.
 - If another car is in its way, it does not move.
 - If the object is not at an intersection, then it moves forward at 25 miles/hr.
 - If the object is at an intersection, then:
 - If there is a stop sign it stops for some number of heartbeats.
 - If there is a red light, it does not move until the light is green.
 - Otherwise, it chooses the legal turn that will take it nearest to its destination (regardless of traffic). The object then moves down the chosen road at 25 m/h if another car is not in its way.
- Present results in PowerPoint form on Friday.

13