

## CS242 Final Project Proposal

### 1. Abstract

#### 1.1. Purpose

Simulate very simple traffic flows

#### 1.2. Motivation

I've always thought a traffic simulator would be interesting, and I want to fiddle around with Erlang and it's process model for concurrency. A small traffic simulator might be an interesting problem to attempt to solve with the language.

### 2. Technical Specifications

#### 2.1. Platform: Desktop

#### 2.2. Language: Erlang and Python (probably will use python for data visualization)

#### 2.3. Conventions: Will conform to Erlang conventions

#### 2.4. Tools: Vim, Erlang compilers/debuggers/runtime, EUnit for testing

#### 2.5. Audience: Geeks

### 3. Functional Specifications

#### 3.1. Features

- Stop Lights
- Custom maps with straight, two way roads and 4 way intersections
- Individual cars will have preset routes
- Microscopic traffic simulation
  - Each car makes it's own decisions. Each car it's own actor.
- Vehicles of different lengths

#### 3.2. Scope (Ideas not supported)

- Multiple lanes
- Merging
- Toll Booths or other interruptions of flow
- Stop Signs

## 4. Timeline

### 4.1. Week 1

- Get familiar with Erlang
- Create a car process
  - Awareness of speed and length, responds to messages
- Single car driving in a straight line
- No visualization expected except for debugging purposes

### 4.2. Week 2

- Add more cars and add some interaction between them
- Circular Traffic (straight line that loops around, if looping easy)
  - Eg. <http://www.mtreiber.de/MicroApplet.html5> with a single lane
- Begin implementing visualization
  - Post-mortem. Will visualize data dumped from simulation. Not real time

### 4.3. Week 3

- Add stop lights
- Cars only travel straight, never attempt to turn
- Heavy focus on visualization of traffic flow

### 4.4. Week 4

- Add car lengths
- Add custom car paths (cars can turn now)
  - Stop lights states:
    - Protected right turn
    - Protected left turn
    - Both lanes forward
- Finalize Visualization