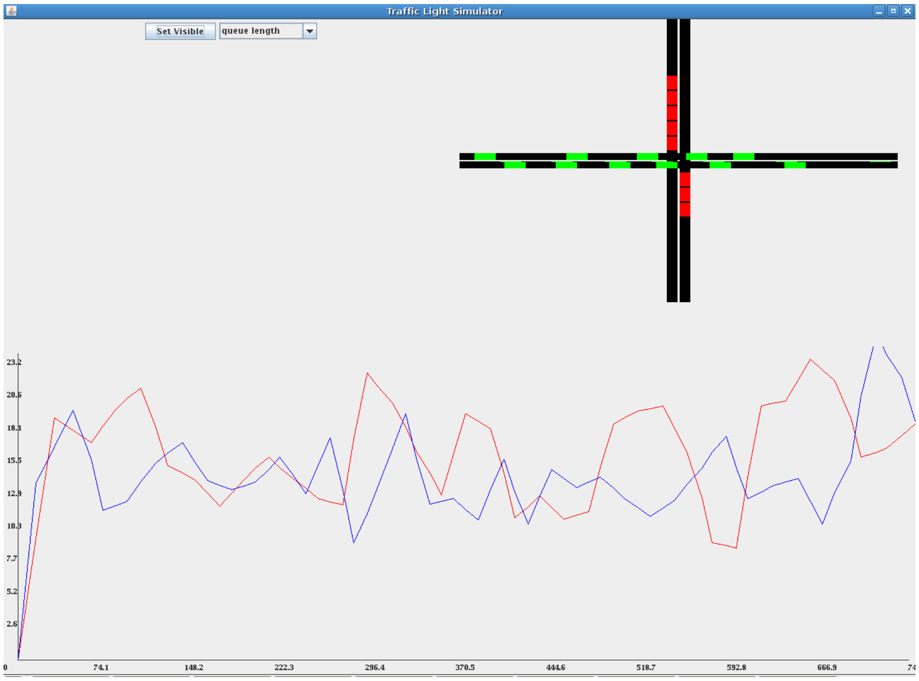
Traffic Light Simulation

Lynn Jepsen

TJHSST Computer Systems Lab 2005 - 2006

Abstract

This project is meant to simulate a busy traffic light. The cars follow all necessary rules of the road so that the simulation is realistic. The program will recognizes patterns in the intersection and makes the light as efficient as possible by minimizing waiting time and queue length plus maximizing green light usage for cars. The patterns in the intersection can change hourly to match the real world.



Results

This program looks like a realistic model of a traffic light. The cars speed up and slow down realistically, obey lights, and don't run into each other. The light algorithm decreases the number of back ups in the intersection, but it can still be improved. More variables should be added to the simulation so that it is as realistic as possible.

Background

Traffic intersections can be frustrating places. No one likes to sit behind a red light and many cars while we wait for one car to whiz by in the other direction. I have designed a simulation and a light algorithm that creates an efficient intersection. There are a many variables that effect a traffic light. They include number of lanes, speed of cars, number of turning cars, etc. I use mathematical models, that have already been developed by other people, to model how cars move on the road. My light algorithm is simply a matter at looking at previous information, and figuring out what worked best.

Procedures and Methods

I am using Java to program. I have finished a very simple simulation of a traffic light. It follows all basic rule of the road. It only has four lanes, but I plan to add in more. The light algorithm uses information from the intersection (like the number of cars stopped in each direction) to pick the best cycle length and ratio of green light time for each direction. This algorithm looks at a lot of previous information in order to make the best decision.