Mom, are we there yet?

Final Report

Team members:

Ignacio Rougier

Joshua Vigil

Kye Jones

Donald Poston

Team #: 18

Mentor:

Esteban Rougier

Introduction

The main motivations to work on this project were: a) to learn how to program on an agent-based software; b) to analyze a problem impacting the local community and c) to have fun. The problem that we chose for the project was to analyze the traffic flow at the intersection between Trinity and Diamond in Los Alamos, NM in the case that a roundabout was built at such intersection. We want to study this, because in general accidents are more probable to occur in traffic light crossings. Also, traffic light intersections tend to be more complex to maintain than roundabouts. Finally, roundabouts may present a viable alternative to traffic light crossings.

The modeling approach adopted in this project is based on an agent-based modeling software called Netlogo.

We had many resources about our project on roundabouts. We looked at libraries for books on it and we got a very big book on it. We also had a meeting with a county traffic engineer. He talked with us about all the stages of technology they used to count cars. Then he talked about the camera's they control to see the traffic. In the same room they are able to control the lights. He talked about the cars or machines they used to paint the lines. They have different tanks to put different paint in. Then they have this very small glass that is very slippery but reflect light so you can see the lines at night. That was our meeting with him. And of course we had the internet. To program we used net logo. We had trouble with all the complicated steps to be able to use it but with help from our mentor and our teammate Donald we were able to program net logo. We have learned a great deal from this project and thanks to our mentor and all the help he's given us we were able to complete our task.

Objectives

The main objective of the project is to determine if roundabouts are better or more efficient than traffic light crossings for this particular intersection.

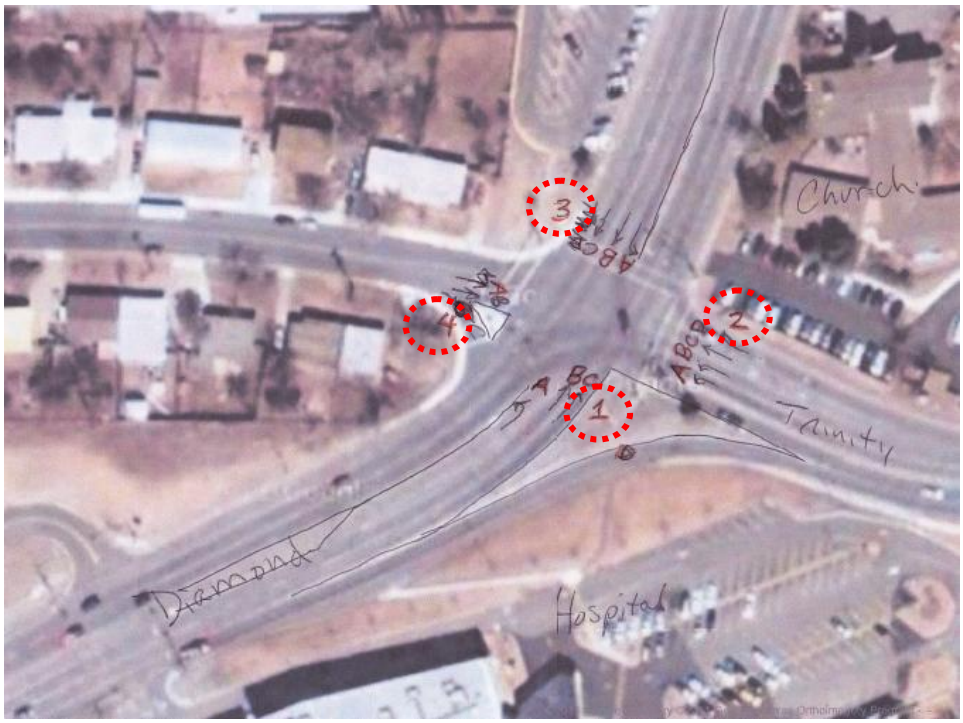


Figure 1. The current Trinity-Diamond intersection. The survey positions are also highlighted in the picture.

Survey

Before starting to design the simulation code, we took a survey of the total amount of cars per unit time going through the intersection at rush hour. There were four stations used during the survey, as shown in Figure 1. The number of cars going through lanes A, B, C and D were counted for a period of half an hour. The results obtained from the survey are shown in Table 1.

Table 1. Results of the survey. (Cars per hour)

Stations	Lane			
	A	B	C	D
Station 1	22	384	288	534
Station 2	124	68	80	304
Station 3	268	68	134	-
Station 4	12	38	10	-

This data will be used later as an input for the simulation code.

Program Procedure

The main parts of the program are described in this section.

Setup

The general layout of the program interface is shown in Figure 2. When the setup button is clicked, the program will create the roundabout geometry with all the decision points identified with a different color: turning decision points are shown as red squares and entering the roundabout decision points are shown as green stripes (Figure 2).

Before running the simulation the user has to specify for each lane the number of cars that will go straight, that will turn right, and that will turn left on the roundabout. This numbers are entered in the boxes located at the right hand side of the screen (Figure 2). In this work these numbers were taken from the survey explained in the previous section.

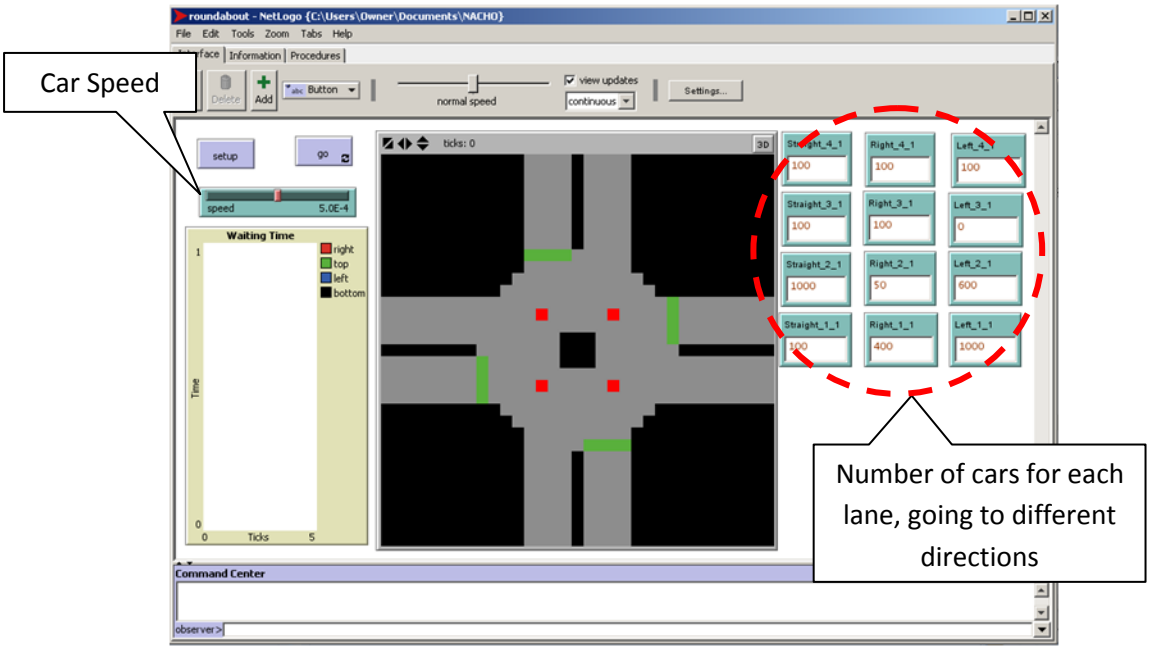


Figure 2. Locations where new cars are created.

Creating the Cars

Cars appear in four different locations, as shown in Figure 3. Before creating a new car on each location the program checks if there is a free space where to put it. If there is no free space, the program will wait until one is generated, which means that there will be no overlapping of cars upon creation.

As each car is created, it is given a predefined path to follow. In the current implementation of the program each car may have three different paths: left, right and straight as shown in Figure 4. For the sake of clarity, each car is given a different color according to its predefined route: straight=black; right=yellow; left=blue (Figure 4). In actual fact the colors of the cars depend on the location where each car is created. This can be seen in Table 2, where the RGB color code for each creation point is shown. This color code was implemented in order to help with the traffic logic inside the roundabout.

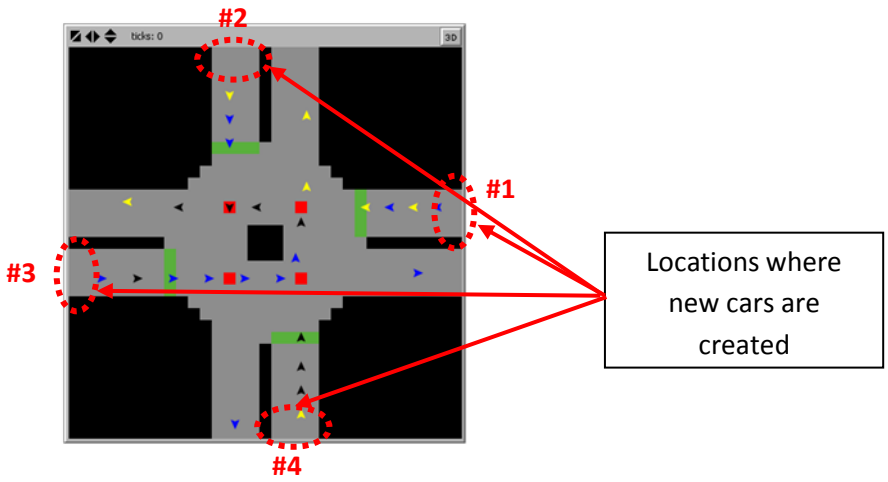


Figure 3. Locations where new cars are created.

To provide more realism to the simulation the pre-defined paths for the cars are assigned randomly. This is done by using the input from the user shown in the right hand side of Figure 2. For example, if there were 100 cars going straight, 200 cars turning right and 50 cars turning left, the pre-defined path would be selected as follows: the Netlogo command:

set rnd_number random car_total

gives a random number (rnd_number) from 1 to car_total, where

car_total = 100 + 200 + 50 = 350

If the value rnd_number is anything between 1 and 100, then the car will go straight; if the value of rnd_number is between 101 and 300, then the car will go right; finally, if the value of rnd_number is between 301 and 350 then the car will turn left.

Table 2. Color code for each car depending on the creation location.

Predefined Path	Creation Location			
	#1	#2	#3	#4
Straight	0,0,0	1,0,0	2,0,0	3,0,0
Left	0,0,255	0,0,254	0,0,253	0,0,252
Right	255,255,0	254,254,0	253,253,0	252,252,0

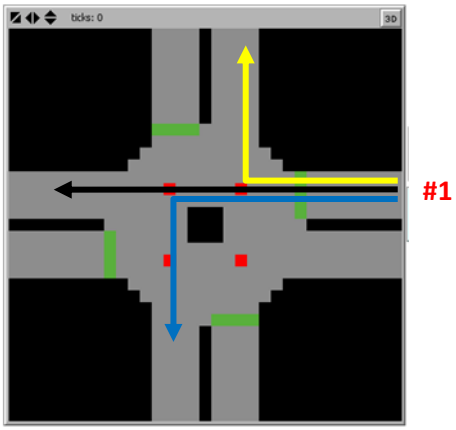
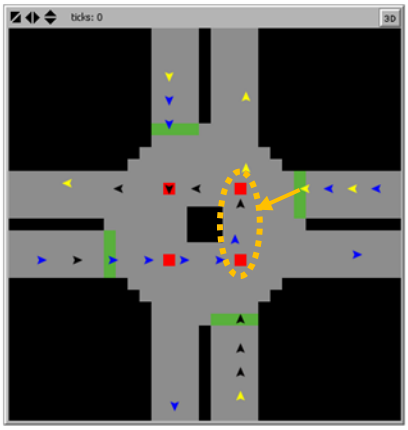


Figure 4. Example of the predefined paths for each car coming from location #1.

In order to avoid collisions between cars, before advancing each car the program checks if the patch right in front of the car is free. If the patch ahead is occupied then the speed of the car is set to zero; otherwise the speed of the car is set to the value specified in the slider control (Figure 2).

Cars entering roundabout

Before entering the roundabout it checks in an area for other cars on the patches ahead. If there are other cars it will stop and wait until the car ahead to move. At a certain point it either turns or goes forward according to its route. The car will continue until it exits the roundabout and dies.



The yellow car sitting on the green stripe waits for the area indicated by the dotted ellipse to be free of cars.

Figure 5. Example of car about to enter the roundabout.

Cars turning inside roundabout

The cars inside the roundabout are watching for other cars. If there are cars turning in there and they go in front of another car the first one will stop and wait until the second car moves. There are red points in the roundabout that mark where the cars will turn or they will stop in front of the red point. Some of the cars will go straight instead of turning.

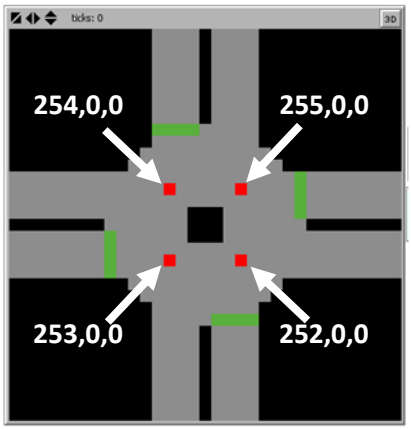


Figure 6. Color code for the turning decision points.

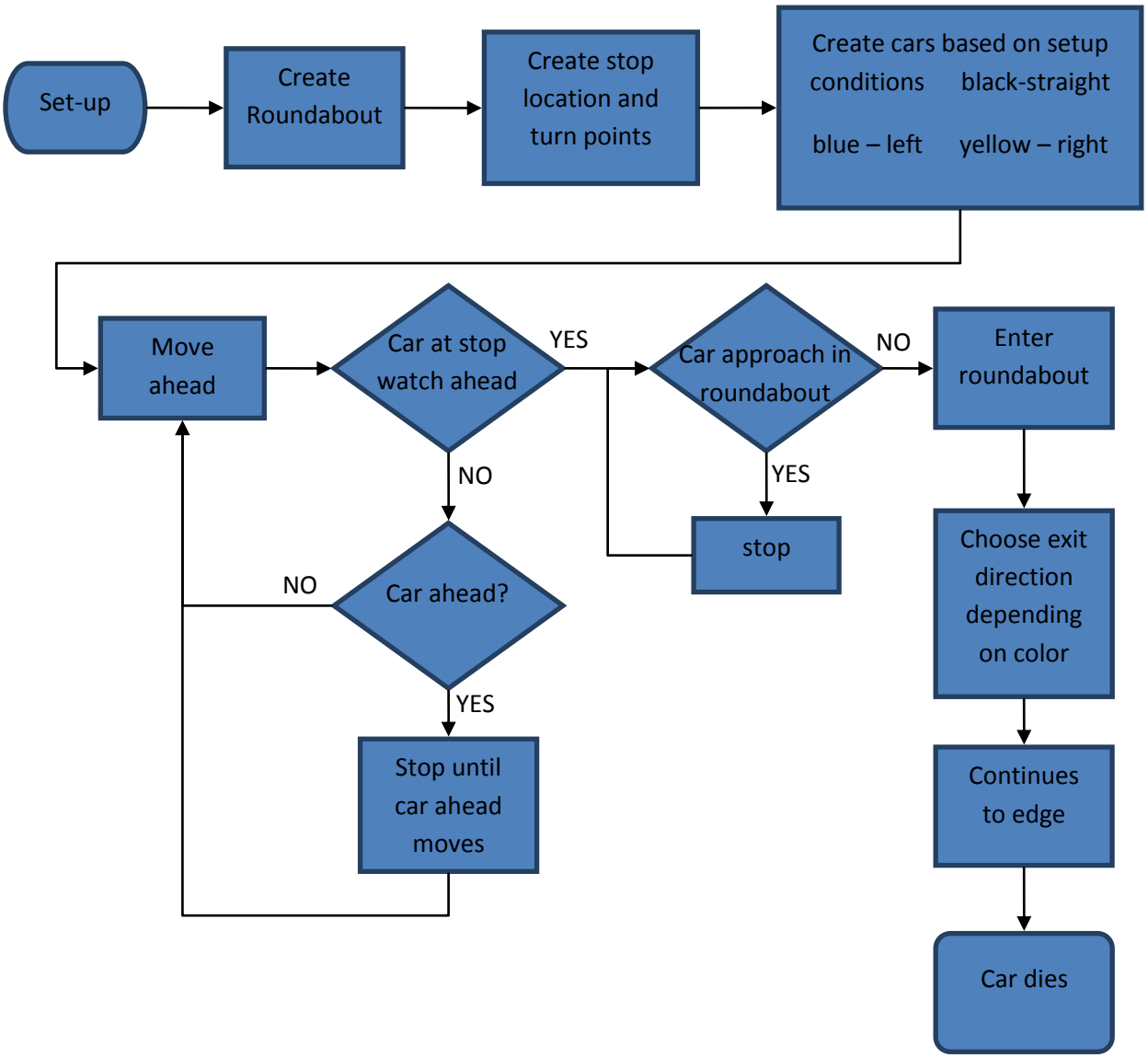
The four “red” points shown in Figure 6 will help to determine when the cars need to turn. For example: if a car created on creation point #1 (Figure 3) has to turn right, then it will do so only when it arrives to the patch with color code (255,0,0). If a car created on creation point #2 (Figure 3) has to turn left, then it will do so only when it arrives to the patch with color code (253,0,0), and so on.

Cars exiting simulation

When the cars make their turn they go straight. When they hit the simulation they will die.

Logic Flow

The logic diagram represents the logic embedded in the simulation.



Results

Our results were that the roundabout is faster than a regular cross section because there is constant movement. However, the cars in a roundabout may be more prone to get into an accident because to get into the roundabout you have to jump into a free space in the roundabout. This can cause sudden crashes but not as big of a crash due to the slower speed of the car. Because of this there is more interaction between the cars in the roundabout than cars in a cross intersection. The cars have to slow down before entering unlike the cross intersection where if you have a green light you can maintain your speed going through the cross

intersection. Cars entering the roundabout have to stop and wait until there is a gap in the roundabout. Cars pay attention to the cars in front and slow down to avoid an accident.

Conclusion

Our conclusion is that for a new undeveloped part of a city or town a roundabout would be a good choice. But for an already existing cross section not really because a cross section is smaller than a roundabout.

The roundabout shows that it reduces rush hour waiting due to the constant movement. The roundabout also reduces deadly crashes because of the slow speed and the angle of the crash which is a glancing blow. The roundabout needs a lot of money to build so you would want to keep the regular cross section until you had enough money. Our simulation was of a single lane roundabout and it was noted that traffic backed up because of turn taking involved. This back could be avoided by creating more flow with a two lane round about.