Prototype for an Interactive Traffic Simulator

Jubin Philip Thomas¹, Harshal Sawant², Vinod Banthia³, Archana Patil⁴
1 - M. Sc. [Engg.] Student, Centre for Computer Graphics and Virtual Reality, 2 - Assistant Professor and Course Manager (Computer Animation and Games Development), 3 - Professor and Head, Department of Automotive Design and Engineering, 4 - Senior Lecturer, Computer Animation and Games Development Centre, M.S. Ramaiah School of Advanced Studies, Bangalore 560 054

Abstract

Traffic congestion is a crucial problem in large cities. It is normally caused by improper control of traffic signals, narrow roads, driving discipline etc. Many researchers have attempted to provide a solution for this problem by developing simulators. The literature review and study of these simulators reveal the lack of vehicle dynamics which is a major element to show the real-life situations. Thus there exists a need for an interactive traffic simulator that incorporates real life motion of 3D vehicle models. This paper presents a prototype for interactive traffic simulator, which incorporates real world vehicle dynamics using the Object Oriented Approach (OOD) of programming and Artificial Intelligence (AI).

The prototype aims to simulate a city environment along with vehicles having real world vehicle dynamics, AI for collision avoidance against obstacles and following traffic signal. Validation study on the implemented prototype is carried out with respect to real time traffic flow data.

The development results in an experimental setup for different end applications like town planning, traffic management etc. The users can test different scenarios of traffic flow pattern and derive the best solution for it. The prototype can be enhanced by considering environmental specifications and including fully fledged Graphical User Interface in order to provide user the freedom to create the environment of their wish.

Key Words: Artificial Intelligence, Object Oriented Approach, Traffic Congestion, Traffic Simulator and Vehicle Dynamics

1. INTRODUCTION

Over the last decade, world economy has constantly grown resulting in a continuously increasing demand for transportation. This demand for transportation is not just for passengers but also for the goods that help human being in their day to day activities. Transportation is considered as one of the most important factor, it provides a sense of freedom to move, which is one of the key elements in human life. It also helps in improving the quality of life. Due to this, the number of vehicles in the world is increasing with a high growth rate. Thus, it is not surprising that traffic has become an important factor in our daily lives. Lately, however, more and more downside of vehicular traffic is being experienced. As street networks have not been upgraded at such a pace as the demand for transportation, traffic jams have become a major problem in densely populated areas [1, 2 & 3]. The resulting delay has direct and indirect consequences.

Time spent on the road cannot be used for productive work. People, who have to travel a lot as part of their occupation, lose part of their income through traffic jams. This is because of being confined to a car in traffic jams, thus creating a stressful situation, which reduces the ability to work efficiently once the work place has been reached. Vehicles stalled in traffic jams use additional fuel. This is not only another cost factor to drivers, but also causes damage to the environment through pollution. Recurrent congestion largely restricts the independence of drivers. Instead of adjusting trips to activities, drivers have to plan their activities carefully in order to avoid congestion [4 & 5].

Traffic simulator can be used to help solve some of these traffic congestions and jams. Simulators like FreeSim [5], a platform independent, open source traffic simulator and CORSIM [6], a windows based traffic simulator are widely used. These simulators help in simulating the traffic and experimenting with different situations. The system analysis done through simulator thus at arriving the junction layout, town plan, signal management helps to avoid traffic congestions. In all terms, traffic simulation has proven to be an important tool to model and analyze the traffic state [7, 8 & 9]. But one of the most important aspects that has to be considered during the development of traffic simulators is to bring the real-world dynamics into it. Incorporating this dynamics is a great challenge as the vehicles in the simulation have to behave similar to the vehicles in the real-world. They have to behave in a similar way to human beings accelerating, decelerating and avoiding collision before it actually happens. For this purpose, the simulators must include Artificial Intelligence (AI) to show the similarity to the drivers’ behaviors during the real-world events. The simulator must also provide the facility for planning the infrastructure. The traffic management and town planning authority must be able to try different combination of road types and perceive how it affects the traffic conditions before deciding on the best one for a part of the city or town.

In this paper, the focus is on developing prototype for a traffic simulator which uses graphics along with real-world vehicle dynamics and artificial intelligence to help solve the traffic related problems. The prototype uses the object oriented programming approach because each vehicle has it’s individual dynamics i.e., the acceleration of a truck varies from that of a car. Using this approach, the vehicles can be modelled and can also provide dynamics to each vehicle depending on its type. The following sections discuss in detail how the requirement analysis was performed, the architecture of the prototype and various algorithms used for developing the prototype.
2. PROTOTYPE DEVELOPMENT

The aim of this work is to develop an interactive traffic simulator using object oriented approach of programming. The work attempts to incorporate the vehicle dynamics present in real world vehicles, AI for checking the signal and for collision detection with other vehicles, Response through performing appropriate task.

2.1 System Analysis

System analysis is the first step in development of an application, to understand the system behavior; to decide on the tools and approaches to be used. It helps in designing, developing and implementing the application in a faster and more reliable way [10]. As a part of system analysis, the traffic and signal properties at a junction were observed to understand vehicle movements, traffic management, signal properties, cycle time for signal lights and the traffic situation. The junction that was considered for this analysis is the signal at BEL circle, North Bangalore (12° 58' N Latitude and 77° 38' E Longitude) with cross type of intersection, i.e., two roads meeting at a point. One has three lane traffic flow (Outer Ring Road, East-West) while the other has two lane traffic flow (MSR Road, North-South). Fig. 1 shows images of the junction taken from Google Maps [11].

![Fig. 1 (a) Satellite image (b) Map image of BEL junction taken from Google maps [11]](image)

The traffic flow and signal timing were analyzed at three different times in a day. The analysis was done in the morning (9 am), afternoon (2 pm) and evening (6.30 pm), for 30 minutes. It was observed that the traffic flow was different for different time periods. Fig. 2 shows the BEL junction picture taken from the camera installed by the Bangalore Transportation Information System (BTIS) [12]. The results obtained by observing the traffic flow and signal properties are as follows.

![Fig. 2 Traffic at the BEL junction [12]](image)

<table>
<thead>
<tr>
<th>Heading</th>
<th>Light</th>
<th>Time</th>
<th>HMV</th>
<th>LMV</th>
<th>Rick</th>
<th>Bike</th>
</tr>
</thead>
<tbody>
<tr>
<td>South</td>
<td>Red</td>
<td>240</td>
<td>4</td>
<td>17</td>
<td>8</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>90</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>West</td>
<td>Red</td>
<td>240</td>
<td>9</td>
<td>20</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>90</td>
<td>5</td>
<td>11</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>North</td>
<td>Red</td>
<td>270</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>60</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>East</td>
<td>Red</td>
<td>240</td>
<td>14</td>
<td>15</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>90</td>
<td>7</td>
<td>9</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 1 Traffic flow and signal properties at 9.00 AM

<table>
<thead>
<tr>
<th>Heading</th>
<th>Light</th>
<th>Time</th>
<th>HMV</th>
<th>LMV</th>
<th>Rick</th>
<th>Bike</th>
</tr>
</thead>
<tbody>
<tr>
<td>South</td>
<td>Red</td>
<td>120</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>40</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>West</td>
<td>Red</td>
<td>120</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>40</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>North</td>
<td>Red</td>
<td>120</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>40</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>East</td>
<td>Red</td>
<td>120</td>
<td>9</td>
<td>7</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Green</td>
<td>40</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 2 Traffic flow and signal properties at 2.00 PM

At 6.30 pm: Traffic police was deployed in the evening (6.30 pm) to control the traffic signal. The police used to observe the traffic at each road and would change the signal based on the number of vehicles accumulated. More the vehicle accumulated at one particular road, more the duration of green signal for that road. This meant that the vehicles present in the other roads had to wait for a much longer duration and causing more traffic congestions.

From the data collected it could be concluded that even if the signal is manned or unmanned, traffic kept piling up for each cycle of signal. It was also observed that vehicles never followed lane discipline. The heavy vehicles usually were found on the right most lanes which caused the lighter vehicles to be held up behind
it. A traffic police alone is unable to manage the signal and the vehicle congestion. Thus there is a need for a traffic simulator that can simulate various scenarios and using the same, a suitable solution for the congestion must be derived.

2.2 Requirement Analysis

Requirements analysis is the first stage in any systems engineering process and software development process [9]. The traffic simulator must incorporate vehicle dynamics because that is the only way to show a simulation which resembles the real world physics. The input requirements for the prototype was identified as 3D models of car (LMV) and truck (HMV), as the focus was on developing the foundation for vehicle dynamics and this was done by taking these two types of vehicles and user input based on the data collected by observing the traffic flow and signal properties at BEL junction.

The output requirements for the prototype were visualization of vehicles arranged in the order specified by the user in the beginning of simulation, the displacement of the vehicle based on the acceleration value, vehicle action based on traffic signal and preceding vehicle, collision avoidance with other vehicle and response through slowing down, stopping and starting.

The processes involved in creating the prototype was modeling of the vehicles, creating a virtual environment consisting of road, traffic signals etc., placing of vehicles according to user input, calculating velocity, displacement and cumulative displacement for each vehicle, real time collision detection with other vehicles and real time sensing of traffic signal.

2.3 Prototype Design

The system architecture for prototype of the traffic simulator is as shown in Fig. 3. The models like the city environment, car, truck etc required for traffic simulator were modeled using 3Ds Max. These models were then imported into Quest3D, an authoring tool. Here, various algorithms required to make the simulator function were written.

![Fig. 3 System architecture](image)

The different algorithms developed were algorithm for reading user’s input, algorithm for placing the vehicles and setting different dynamics related values, algorithm for calculating the velocity and displacement, and supplying the cumulative displacement to each vehicle and algorithm for checking collision with other vehicles and also to read the signal.

The algorithm for saving the user input to arrays was the first and foremost step as the user entered data was in text format. Once the data has been saved to the array, the system can read the data in an efficient and structured manner. The next algorithm placed the vehicles onto each lanes based on the input read from the array. This algorithm also set the acceleration value for each vehicle. The algorithm for calculating velocity and displacement was at the core, which calculated respective values for each vehicle and finally the cumulative displacement was given to each vehicle which helped it to move. The final algorithm was used to check for any collisions. Here, collision could be of two types, one is between vehicles and the other one, though not collision in real world, between vehicle and the signal. In this algorithm, the vehicles were made to follow particular action to avoid collision.

As the input to the simulation was varying in terms of number and type of vehicles and vehicle and signal characteristics, the best programming method to be followed was Object-Oriented Programming (OOP). It is a programming paradigm that uses objects and their interactions to design applications and computer programs [13].

Once the vehicles were placed at their rightful positions on the lanes using the vehicle object creation and placement algorithm, the next step was to calculate the velocity and displacement for each vehicle. Velocity is the rate of change of position. It is a vector physical quantity; both speed and direction are required to define it. The following equation was used to calculate the velocity:

\[ v = u + at \]

where, \( v \) is the final velocity, \( u \) is the initial velocity, \( a \) is the acceleration, \( t \) is the time difference.

Initially, \( u \) was taken as 0, acceleration value was specified by the user and \( t \) was the time difference between each calculations. Once the velocity was calculated, the next process was to calculate the displacement based on the calculated velocity. Displacement (vector), in Newtonian mechanics, specifies the change in position of a point in reference to a previous position. In simple terms, it’s the difference between the initial position and the final position of an object. The following equation was used to calculate displacement:

\[ s = ut + \frac{1}{2} at^2 \]

where, \( s \) is the displacement, \( u \) is the initial velocity, \( t \) is the time difference, \( a \) is the acceleration.

Initially, \( u \) was taken as 0, \( t \) the time difference between each calculations and acceleration was the value specified by the user. This displacement value was used to calculate the displacement at each time instance. But in order to move the vehicle cumulative displacement was given to the vehicle. Cumulative displacement was calculated by using the equation given below:

\[ s_{t+1} = s_t + \Delta s_{t+1} \]
where, $s_{t+1}$ is cumulative displacement, $s_t$ is the previous displacement value for the present time interval, $\Delta s_{t+1}$ is the accumulated displacement till present time interval.

Thus, the calculated cumulative displacement was given to each vehicle. This value made the vehicle to travel the required distance based on the user specified acceleration.

2.4 Programming of the Prototype

The implementation phase of any software development involves the creation of the entire system based on the conclusions of the system phase [9]. As concluded in the system design phase, the development phase for prototype of the traffic simulator involved the implementation of algorithms for reading user’s input, placing the vehicles and setting different dynamics related values, calculating the velocity and displacement, supplying the cumulative displacement to each vehicle, and checking collision with other vehicles and also to read the signal.

Algorithm for reading user input: The first algorithm that had to be implemented was to read the input given by the user. The input given by the user was in the form of text. The user was asked to enter a few details like number of cars and trucks on road, number of lanes required, position of the lanes, acceleration value for car and truck and positions of the vehicles on each lane in a text file.

The algorithm had to read these data and store it into the arrays in order to make it accessible to other algorithm. Two arrays were used to save the data given by the user. The first array contained number of cars, trucks, lanes, and lane position and acceleration values of both car and truck while the second array contained the positions of each vehicle.

Algorithm for placing the vehicles and setting dynamics: To avoid collision between two vehicles some distance must be maintained between them. The minimum distance required between two vehicles was obtained by placing two cars at predefined position and setting dynamics on them. By using this algorithm, it was concluded that the optimum distance between each vehicle (front to back) should be one meter. Fig. 4 shows the snapshot of the two cars placed at a distance of one meter.

Algorithm for calculating velocity and displacement: After the placement of vehicles at their positions, the next step was to calculate the velocity and displacement for each vehicle. Velocity is nothing but the rate of change of position. By using the equations for velocity, displacement and cumulative displacement, each value was calculated. The cumulative displacement value made the vehicles move.

Fig. 4 Cars placed at a distance of one meter

The data entered by the user was saved into arrays by using the algorithm for reading user input. The next step was to place the vehicles in the order mentioned by the user. Initially, the number of cars, trucks and lanes were read from the first array to determine the number of instances that had to be created for each object (cars, trucks and lanes). Using the data from the first array, the instance of each vehicles and lanes were created. The algorithm then read the position of each lane from the first array and placed them at the appropriate positions. Then acceleration values specified by the user were assigned to the acceleration variable present in cars and trucks. Braking values for the vehicles were also calculated and used to decelerate the vehicles. For the traffic simulator, braking values were calculated by using the acceleration value.

The final step in this algorithm was to place the vehicles on the lanes. The user mentioned ‘C’ for cars and ‘T’ for trucks. The algorithm read the array, one column(lane) at a time. If the algorithm encounters ‘C’, it places an instance of car at that position or if it encounters ‘T’, it places an instance of truck at that position. If the user leaves a blank space or gives any other value instead of ‘C’ or ‘T’, the algorithm leaves that area blank and does not place any vehicle. Fig. 5 shows the snapshot of how the vehicles were arranged based on the user’s input.

Algorithm for calculating velocity and displacement: After the placement of vehicles at their positions, the next step was to calculate the velocity and displacement for each vehicle. Velocity is nothing but the rate of change of position. By using the equations for velocity, displacement and cumulative displacement, each value was calculated. The cumulative displacement value made the vehicles move.

Fig. 5 Snapshot of the vehicles placed according to user input

Fig. 6 shows the snapshot of the vehicles moving with their own velocity. It can be seen that the trucks move slower compared to cars as acceleration for trucks is smaller than the acceleration for cars.

Fig. 6 Snapshot of the vehicles in motion
To lay the ground works for the calculation of velocity and displacement, a single car was taken and the calculations were performed on it. A small algorithm was written to calculate the velocity, displacement and cumulative displacement. The car was made to travel for ten seconds with its velocity increasing, next ten seconds it was made to travel at a constant speed and finally, the last ten seconds decelerating. This was done to check if the dynamics was taking place properly based on the equations. For visualization, the car was made to change its color for every change in velocity. During the acceleration phase, the car was given green color, during the constant velocity phase, the car was given blue color and during the deceleration phase the car was given red color. Fig. 7 shows a series of snapshots taken to show the different phases of the car.

![Fig. 7 Snapshot showing different phases of velocity](image)

The velocity, displacement and cumulative displacement values at each second during the simulation were recorded into an array. The graph plotted based on these values shows that the rate of displacement is changing according to the change in velocity. The output graphs are shown in Fig. 8. Fig. 8 (a) shows velocity of the car over a period of 30 seconds. Fig. 8 (b) shows displacement of the car over a period of 30 seconds. Fig. 8 (c) shows acceleration of the car over a period of 30 seconds.

Algorithm for collision avoidance and reading signal:
The above explained algorithm had taken care of moving the vehicle based on the user specified acceleration. The next step in traffic simulator was to avoid collision between vehicles and also to read the signal and perform the necessary action. Before writing the main algorithm, a small algorithm was written using two cars. In this algorithm, the car that was in front was given lesser acceleration value when compared to the acceleration value for the car that was behind it. The collision avoidance algorithm was written for the behind car. The front car was made to travel at acceleration phase, constant velocity phase and deceleration phase for ten seconds each.

Depending on the phases of the front car, the behind car was made to accelerate or decelerate based on the distance between the cars. The behind car was made to decelerate if it comes within five meters from the front car. If the distance is greater than five meters, then the car continues to accelerate. Both cars changed their colors according to the phases. The snapshot of this is shown in Fig. 9.

![Fig. 8 Graphs showing the (a) velocity, (b) displacement and (c) acceleration](image)

![Fig. 9 Snapshots showing the collision avoidance](image)
Fig. 10 Graphs showing the (a) velocity and (b) displacement of behind car

Based on the above trial collision algorithm, the collision avoidance algorithm was built and incorporated in the traffic simulator. The prototype of traffic simulator was developed using object-oriented programming method. Thus, the algorithm was included in each vehicle and they perform their own collision avoidance. Fig. 11 shows the collision avoidance in the simulator.

Fig. 11 Collision avoidance between vehicles

A similar algorithm was written along with the collision avoidance algorithm for reading the signal. The vehicles distance from the signal was taken into consideration to implement this algorithm. If the vehicle reaches within the ten meters of the signal, the vehicle reads the signal. If the signal is red, then the vehicle starts to slow down, where as if the signal is green, then the vehicle continues to move at the normal speed. Fig. 12 shows the snapshot of the vehicles reacting to signal. Fig. 12 (a) shows the vehicles decelerating as the signal is red while Fig. 12 (b), shows the vehicles accelerating after the signal turns green.

3. VALIDATION OF PROTOTYPE

The verification process starts off as soon as the coding of the simulator starts. This process is known as unit testing. Unit testing is a software design and development method where the individual units of source code of the simulator was checked to confirm that they were working properly and they meet the output requirements mentioned for the simulator. Every time a new set of code was added, the simulator was tested to see if it causes any unwanted or undesired events.

Module testing was also done for the verification purpose. In module testing, analysis was done on every developed algorithm of the prototype. For each algorithm in the prototype, the system was checked to see if the appropriate actions that the algorithm must produce were happening.

Finally, integration testing was done to check if the simulator works properly on the whole. Integration testing is the phase of software testing in which individual modules are combined and tested as a group. During the integration testing, the simulator was given different inputs and the results were checked to confirm that the prototype was working as desired. As a test case for integration testing, the vehicles were given their acceleration and placed in an order. Fig. 13 shows the snapshot of the input for testing the system.

Fig. 13 Input for test case

Once the simulation was run using these inputs, the desired output was the cars and trucks arranged as mentioned in the text. The vehicles also have to move with the given acceleration. The cars should move at ten times the speed of trucks. Fig. 14 shows the output for the first test case.
The output values of the car and truck were retrieved and a graph was drawn. Fig. 15 shows the graph for velocity and displacement with respect to time. This graph is based on theoretically calculated output for velocity and displacement using the equations. Fig. 16 shows the velocity and displacement of the car with respect to time based on simulator values. Similar trend of results are observed for the truck simulator.

Thus from the graph of the test cases, it can be shown that the theoretical calculation based output for the vehicles were same as the output obtained from the simulator. It can also be concluded that the vehicle dynamics provided to the vehicles in this prototype is accurate.

4. CONCLUSION

There are many traffic simulators available commercially. But they do not have the dynamics of real world vehicles involved in them. Most of them show vehicles moving at constant speed. But in real-world each vehicle has its own acceleration limits. For a simulator that aims to help solve the various problems caused due to traffic congestion, vehicle dynamics is very much necessary as without these dynamics, it is impossible to simulate the real world traffic problems.

The prototype developed here allows the user to input the number of vehicles and in the order in which they are present on road. The user can change the order of the vehicles and check how each combination of vehicles contribute to or reduce the traffic. Vehicle dynamics is included in this prototype, with which the simulator can simulate various real world traffic problems in a realistic way. Each vehicle has its own acceleration property. The vehicles also avoid collision between vehicles by constantly checking the distance between each vehicle. Vehicles also have the artificial intelligence to read the signal and perform the appropriate actions based on it.

The level of dynamics in the present prototype can be further improved by taking into consideration the various geographical conditions, a GUI (Graphical User Interface) can be created to take the user inputs for different road types, setting the signal durations, pedestrians and weather can be included, other vehicles like buses, auto rickshaw, motor bikes etc can be added, over taking property can be given if the vehicle in front is moving slow, speed breakers can also be added to slow vehicles at intersections. The prototype intends to incorporate real world vehicle dynamics in traffic simulation. It can be converted into full-fledged traffic simulator to analyze traffic situation, traffic management, training traffic personnel, town/city planning etc.

REFERENCES


