
# TRIBHUVAN UNIVERSITY

# INSTITUTE OF ENGINEERING

## Pulchowk Campus

**A COMPUTER GRAPHICS PROJECT TITLED:**

# TRAFFIC SIMULATION

**Submitted by:**

ANUSHA ACHARYA (074bex406)

RAJIN PRADHAN (074bex434)

SAMAR SHRESTHA (074bex439)

SUDHAN SHRESTHA (074bex444)

**Submitted to:**

**Department of Electronics and Computer Engineering**

**March 5, 2020**

# Acknowledgement

We would like to express the deepest appreciation to Dr. Basanta Joshi for giving most valuable suggestion, helpful guidance and encouragement in completion of our project and providing us all possible assistance. He has been very helpful in completing this project work titled, “Traffic Simulation” as fulfillment for Computer Graphics Project. At last, we express sincere appreciation to other faculty members of Department of Electronics and Computer Engineering (DOECE), Pulchowk Campus, who helped and encouraged us in some way on completing our project.

Sincerely,
Anusha Acharya (074Bex406)
Rajin Pradhan (074Bex434)
Samar Shrestha (074Bex439)
Sudhan Shrestha (074Bex444)

# Abstract

The course of the subject ‘Computer Graphics’ has been designed to teach students the basic components required to generate images in computers’ display mechanism. By following the syllabus, one can learn the steps that are required to draw real world objects in a computer. The main objective of this project is to apply the syllabus in order to generate Computer based Animation. Our team has used Blender (an Open-source 3-D Computer Graphics Software) to create model of 3-D objects. We have used C++ Programming Language and OpenGL Library to produce computer based animation. We have applied various functions provided by OpenGL to make our project more capable of fulfilling the requirements stated.

# Table of Content

# 1. INTRODUCTION

## 1.1 INTRODUCTION

Computer Graphics is the complete process of generating and presenting graphical representations through use of computer. The graphical objects can be simple geometrical objects to real world objects. We use computer graphics as a means to deliver our ideas through interactive and interesting media rather than relying on text. And our project on Computer Graphics emphasizes on creating a small model of our world. This project portrays a traffic simulation. The simulation involves movement of computer generated objects in a computer generated environment.

## 1.2 BLENDER

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, and computer games. Some of the features provided by Blender include 3D modeling, UV unwrapping, texturing, raster graphics editing, fluid and smoke simulation, sculpting, animating and video editing. In our project, we have used Blender to develop models of object.

## 1.3 C++ Programming Language

C++ is a high-level, general-purpose programming language. It implements Object Oriented Programming through use of classes and objects. It also provides features like dynamic memory allocation, inheritance, polymorphism. C++ can also be viewed as an upgrade to C programming language. C++ is a very popular language and can be compiled through many compilers by use of many IDEs. The IDE that we have used for this project is Codeblocks.

## 1.4 OpenGL

Open Graphics Library or otherwise known as OpenGL is a cross-language, cross-platform Application Programming Interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a Graphics Processing Unit (GPU), to achieve hardware-accelerated rendering. OpenGL is available in many programming language.

## 1.5 GLUT

OpenGL Utility Toolkit, popularly known as GLUT is a free library of utilities for OpenGL programs, which primarily performs system-level I/O with operating system. The functions include window definition, window control, and monitoring of keyboard and mouse inputs. Routines for drawing a number of geometrical objects (including solid objects and objects with only outlines) are also available,

including cubes and spheres. GLUT can also be used to implement pop-up menus. The GLUT library has to be included in Integrated Development Environment (IDE). It provides various inbuilt functions to facilitate programming.

Some of the inbuilt functions that we have used in completing this project are:

**glutMainLoop ( ):** It is a function that loops within itself. It processes events and triggers callback when necessary. It can only be interrupted through the callback functions initialized before the call.

**glBegin (symbolic_constant) & glEnd(void):** The symbolic_constant refers to the type of object to be drawn from the vertices provided between glBegin and glEnd.

**glVertex3f (x,y,z):** Specifies the coordinate of vertex for the object being drawn.

**glColor3f ( ):** This function is used to change the current value of color. The new color is the combination of the RBG (red blue green) values.

**glNormal3f ():** Specifies the x, y & z coordinate of new current normal.

**glLoadIdentity ( ):** It replaces the current matrix with Identity matrix.

**glPushMatrix ( ):** This pushes the matrix on top of stack, a step below. And duplicates that matrix to top of stack. i.e. two similar matrices are formed at top of stack.

**glPopMatrix ( ):** Replaces the matrix at top of current stack.

**glTranslated ( ):** Multiplies the current matrix by the translation matrix specified.

**glRotated ( ):** Multiplies the current matrix by the rotation matrix specified.

**glutInitWindowSize( ):** Is used to initialize the size of the window.

**glutInitWindowPosition( ):** Is used to specify the origin for window position.

**glutCreateWindow(string):** Is used to provide name to the window.

**glutDisplayFunc(display):** It is used to call the unique display function.

**glutKeyboardFunc (functionName**): It is a callback function. It sets the keyboard callback. Using this, the ASCII value of the key pressed or released can be known.

**glLightfv (lightNo., Attribute Type, Attribute matrix):** Sets the light source parameters.

# 2. OBJECTIVES

The purpose of carrying out this project were numerous. Some of the objectives of conducting this Project of Computer Graphics are:

1. To implement theoretical knowledge on Computer Graphics into practice.

2. To understand the algorithms mentioned in syllabus.

3. To correctly implement the algorithms through use of a programming language.

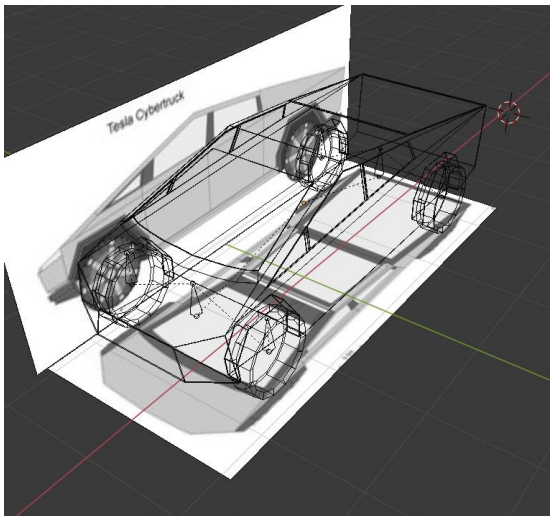
4. To generate objects (car, road, street lamp, etc), similar to that present in real world, in computer.

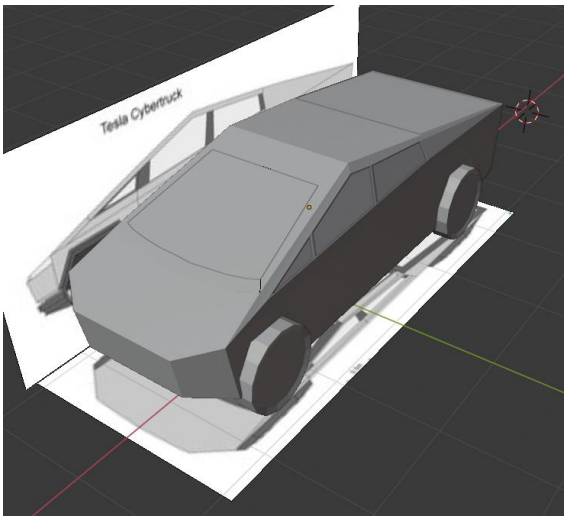
# 3. METHODOLOGY

Our project's development process can be divided into different phases. Each phase had distinct objective. Each phase consists of a number of steps.

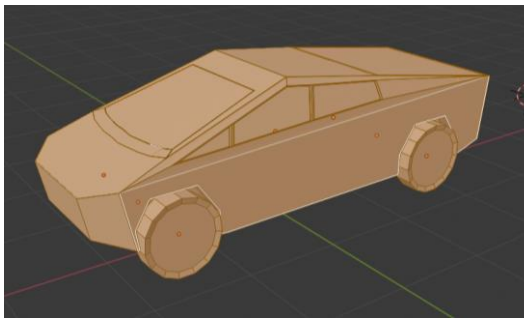
## 3.1 Phase 1

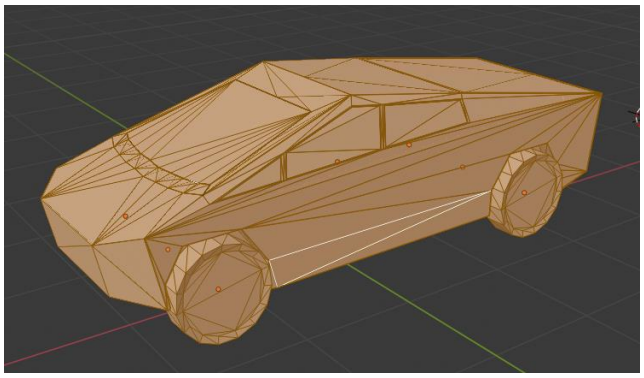
The first step was creating the objects. For simplicity, we modelled the 3D objects in Blender. At first a car was modelled using orthographic view as reference.




Then every polygon surface developed was converted into triangular surfaces for easy ray tracing and simplifying the normal calculation process.





Then the environment was modelled.

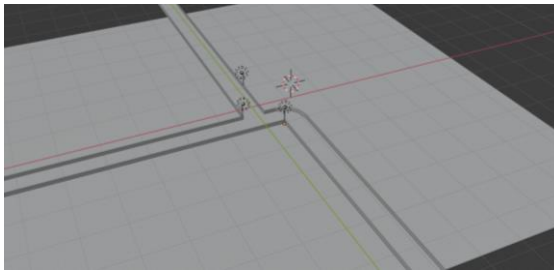


Fig. Non-rendered Environment

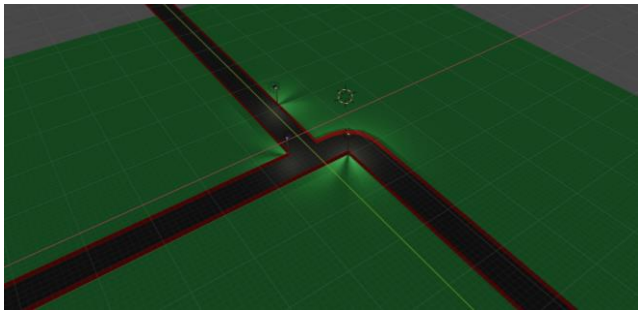


Fig. Rendered Environment

We also created a video through animation in blender. The purpose of creating this was to visualize what we were aiming our project towards.

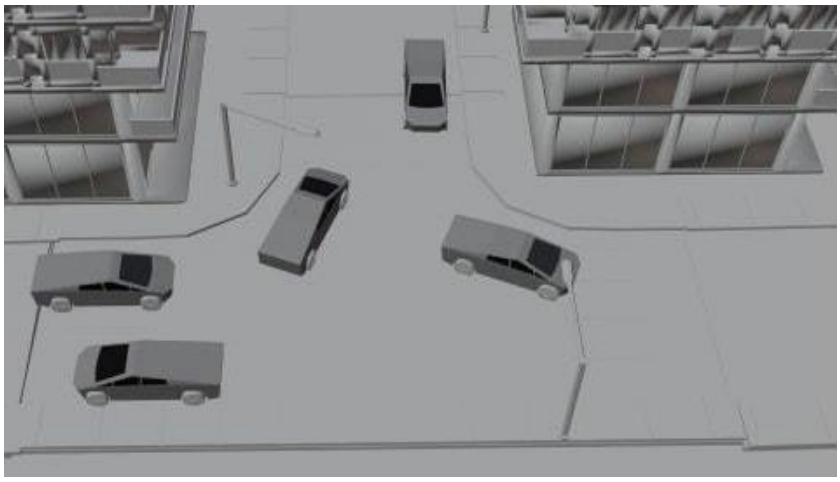


Fig. a Frame of the Animated Video

The created objects were exported to '.obj' file format. Use of Blender was limited to Phase 1 only.

## 3.2 Phase 2

In second phase of our project, we dealt with the objects that had been obtained through first phase. Phase 2 can be divided into following steps:

a. We read the '.obj' file which was created from blender.

b. The file consisted of objects and their faces, vertices, texture coordinate & normal vector of faces. We read and stored all of them.

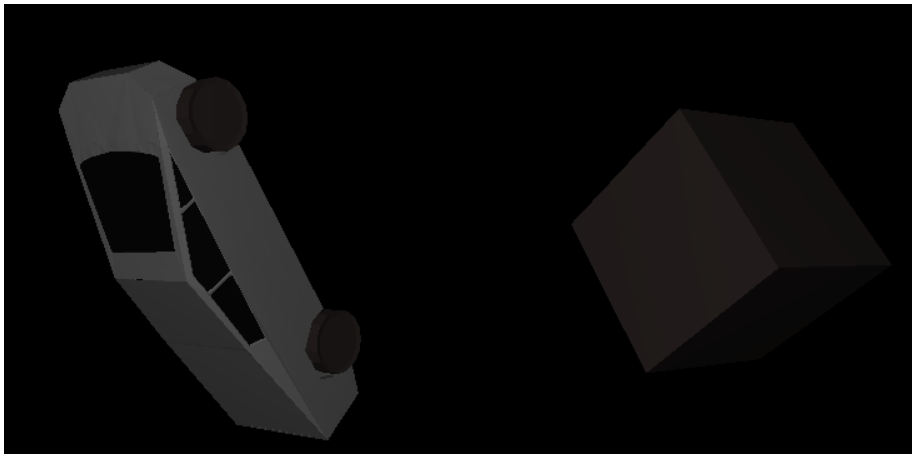
c. We plotted the vertices of each face, stored its respective normal vector and its texture coordinate.

d. For material, we only used ambiance, diffusion and specular attribute which was read from the '.mtl' file (we merged .mtl with .obj).

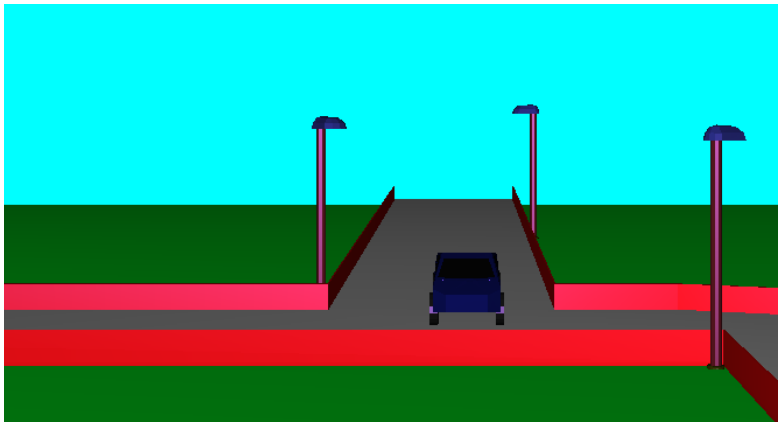
e. Similarly we applied those attributes.

At first, we loaded only a single object in OpenGL. We applied the above mentioned steps to load a cube modelled in Blender.

After observing appropriate result, we loaded multiple objects and provided a simple transformation for the objects to follow. The transformation included translation and rotation.



After successful loading of multiple object, we finally loaded our environment and car in OpenGL.



Loading of the Environment and the Car marked the completion of Phase 2.

## 3.3 Phase 3

Phase 3 is the final phase of our project. In Phase 3, we applied different algorithms to obtain required lighting conditions and object animation. Multiple objects were read on OpenGL and every objects were individually moved to obtain animation. Major steps involved in animation process were:

i. We created multiple objects to load Car and a single object to load Environment.

ii. After calibrating world coordinate, we fixed environment and calculated the paths for Cars to follow.

iii. Individual Cars were programmed to move along their respective path with respect to time.

We then adjusted the light parameters to obtain required lighting conditions.

i. Assign light attributes such as ambiance, diffusion, position and specular.

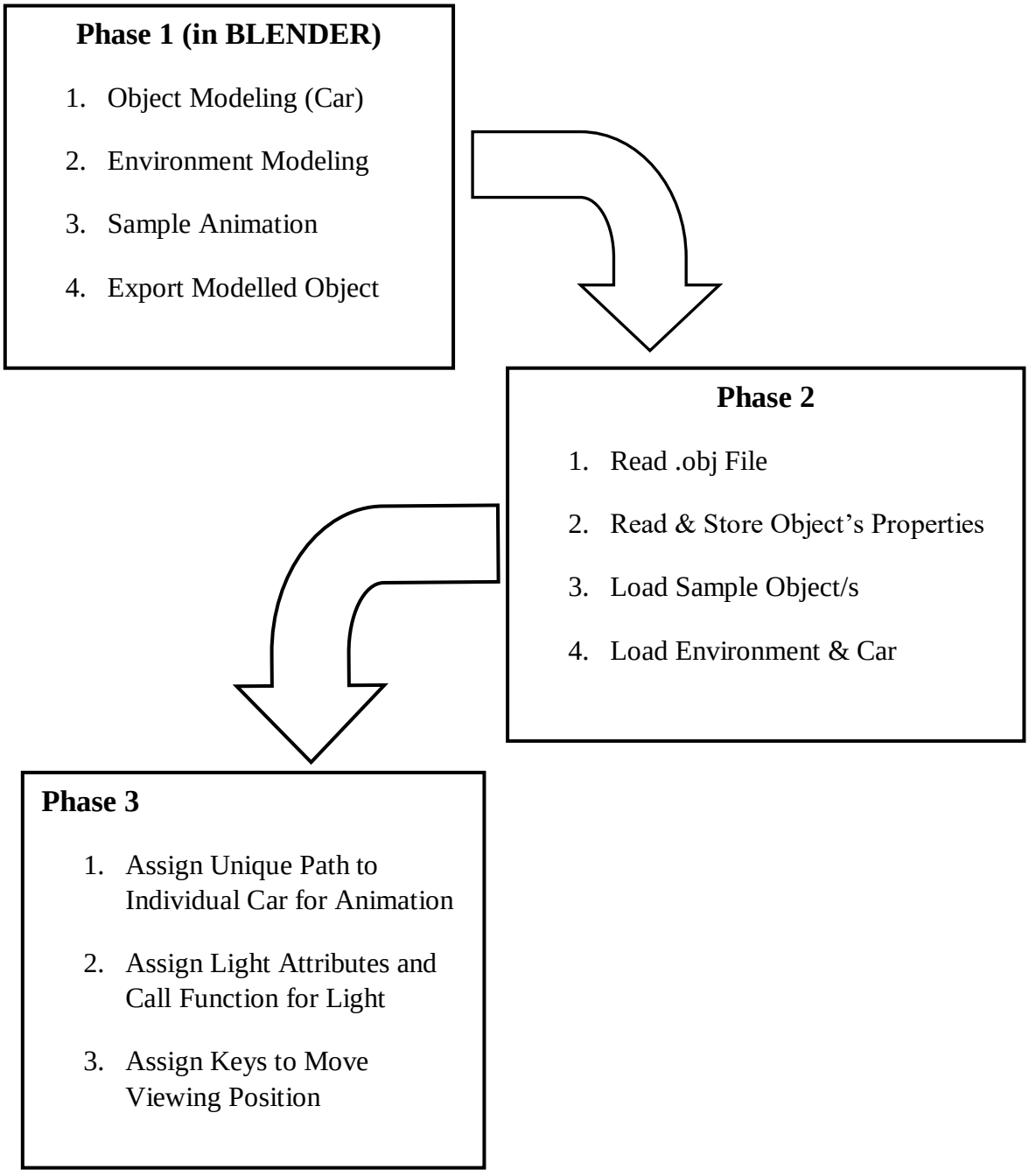
ii. Call function 'glLightfv(lightNo., attributeType, attributeMatrix )'.

With aim of making our project more interactive, we have assigned keys to move and rotate the viewing position (rather the entire sets of objects).

- Keys W & S move the x-axis.
- Keys A & D move the y-axis.
- Keys Q & E move the z-axis.
- Keys U & J rotate about x-axis.
- Keys I & K rotate about y-axis.
- Keys O & L rotate about z-axis.

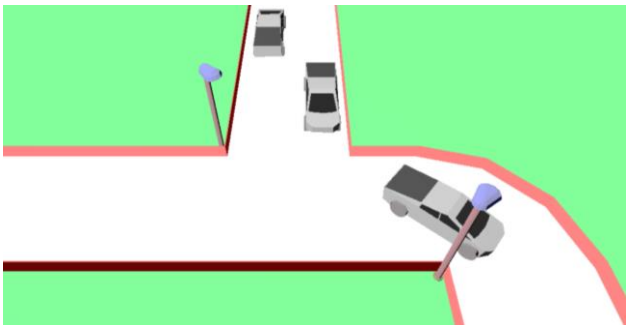
Here, x-axis is perpendicular to screen towards viewer.

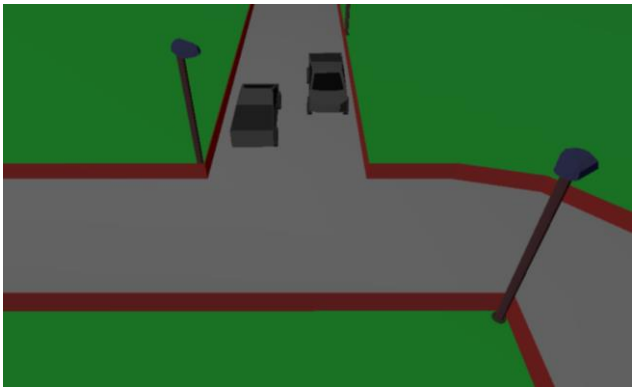
## Project's Work Flow Diagram



# 4. RESULT

The final product of our project includes an interactive traffic simulation model. Many modelled cars run on the road in the modelled environment. Each car has its own set of instructions for animation. The simulation is available in a day and a night mode. These modes can be toggled through M & N Keys. The viewing position can also be changed through specified keys.





# 5. DISCUSSION & CONCLUSION

In the process of completing this project, we followed many steps. We made use of various platforms. We used Blender to model our objects and export them to object files. We used Codeblocks to perform programming in C++ Programming Language. We used GLUT to effectively load the objects, animate them, and produce a lighting (illumination) condition.

We made use of various inbuilt functions provided by GLUT. And also used user defined functions to create required output. We tried using Illumination techniques to create a day and a night effect, and this was a success. But using street lamps as source of light turned out to be a failure. Although, we were able to use keyboard keys to navigate the simulation.

After successfully completing this project we can conclude that the algorithms included in syllabus can be applied to create a functional graphical output. We can conclude that we can , now, model objects in Blender, import them in openGL, and apply motion, transformation and illumination.

# 6. REFERENCE


i. http://freeglut.sourceforge.net/

February 2020

ii. Computer Graphics with OpenGL, Fourth Edition

by D. Hearn, M. P. Baker & W. Carithers

iii. Computer Graphics, C Version, Second Edition

by D. Hearn & M. P. Baker