

## LAB SET 8

Write an animated program to construct a car like structure on a track and perform the following in a menu:

- a. Start the car.
- b. Change the direction of moving car from left to right and vice versa.
- c. Increase and decrease the speed.
- d. Stop the car.
- e. A sub menu to change color of the car body.

```
//Program
//Akarsh Singh
#include <stdio.h>
#include <stdlib.h>
#include <GL/glut.h>
#include <math.h>
//declare matrix of colors
float colors[4][3]={1.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0},{1.0,1.0,0.0}};

int body_color=0; //index to set body color
double s=0.0; // speed variable
double tr=0.0; // rotation with an arbitrary axis (tr,tr,tr);

void car()
{
    //car wheels
    glPushMatrix();
    glTranslatef(-0.9,-0.02,-0.2); // move the torus to left
    glutSolidTorus(0.01,0.03,25,25); // render a torus at the center
    glPopMatrix();
    glPushMatrix();
    glTranslatef(-0.6,-0.02,-0.2);
    glutSolidTorus(0.01,0.03,25,25);
    glPopMatrix();
    //car body
    glColor3fv(colors[body_color]);
    glPushMatrix();
    glTranslatef(-0.75,0.1,-0.3);
    glScalef(2.5,2.0,1.0);
    glutSolidCube(0.1);
    glPopMatrix();
    glColor3f(0.5,0.5,0.5);
    glPushMatrix();
    //front mirror
    glTranslatef(-0.6,0.1,-0.4);
    glScalef(0.5,2.0,1.0);
    glutSolidCube(0.1);
    glPopMatrix();
    //back mirror
```

```

    glPushMatrix();
    glTranslatef(-0.9,0.1,-0.4);
    glScalef(0.5,2.0,1.0);
    glutSolidCube(0.1);
    glPopMatrix();//dicky
    glColor3fv(colors[body_color]);
    glPushMatrix();
    glTranslatef(-0.95,0.1,-0.4);
    glScalef(1.2,2.0,1.0);
    glutSolidCube(0.1);
    glPopMatrix();
    //banet
    glPushMatrix();
    glTranslatef(-0.55,0.1,-0.4);
    glScalef(1.2,2.0,1.0);
    glutSolidCube(0.1);
    glPopMatrix();
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glColor3f(0.4,0.4,0.4);
    glPushMatrix();
    glRotated(20,tr,tr,tr);
    car();
    glPopMatrix();
    //Road
    glColor3f(0.0,0.0,0.0);
    glPushMatrix();
    glTranslated(-1.0,0.0,0.2);
    glScalef(4.5,1.2,0.0);
    glutSolidCube(1.0);
    glPopMatrix();
    glutSwapBuffers();
    glFlush();
}

void init()
{
    glClearColor(1.0,1.0,1.0,0.0);
}

void reshape(int w,int h)
{
    glViewport(0,0,w,h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    if(w<=h)
        glOrtho(-30.0,30.0,-30.0*(GLfloat)w/h,30.0*(GLfloat)w/h,-10.0,10.0);
    else

```

```

        glOrtho(-30.0*(GLfloat)h/w,30.0*(GLfloat)h/w,-30.0,30.0,-10.0,10.0);

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
    }

    void idle()
    {
        s=0.001;
        glTranslated(s,0.0,0.0);
        glutPostRedisplay();
    }

    void mainmenu(int id)
    {
        switch(id)
        {
            case 1:        //Start car
                           glutIdleFunc(idle);
                           break;

            case 2:        //Stop Car
                           glutIdleFunc(NULL);
                           break;

            case 3:        //Turn left
                           tr+=0.01;
                           break;

            case 4:        //Turn Right;
                           tr-=0.01;
                           break;

            case 5:        //Increase Speed
                           if(s==1.01)
                               break;
                           s+=0.001;
                           break;

            case 6:        //Decrease Speed
                           if(s==0.0)
                               break;
                           s-=0.0001;

        }
        glutPostRedisplay();
    }

    void colormenu(int id)
    {

```

```

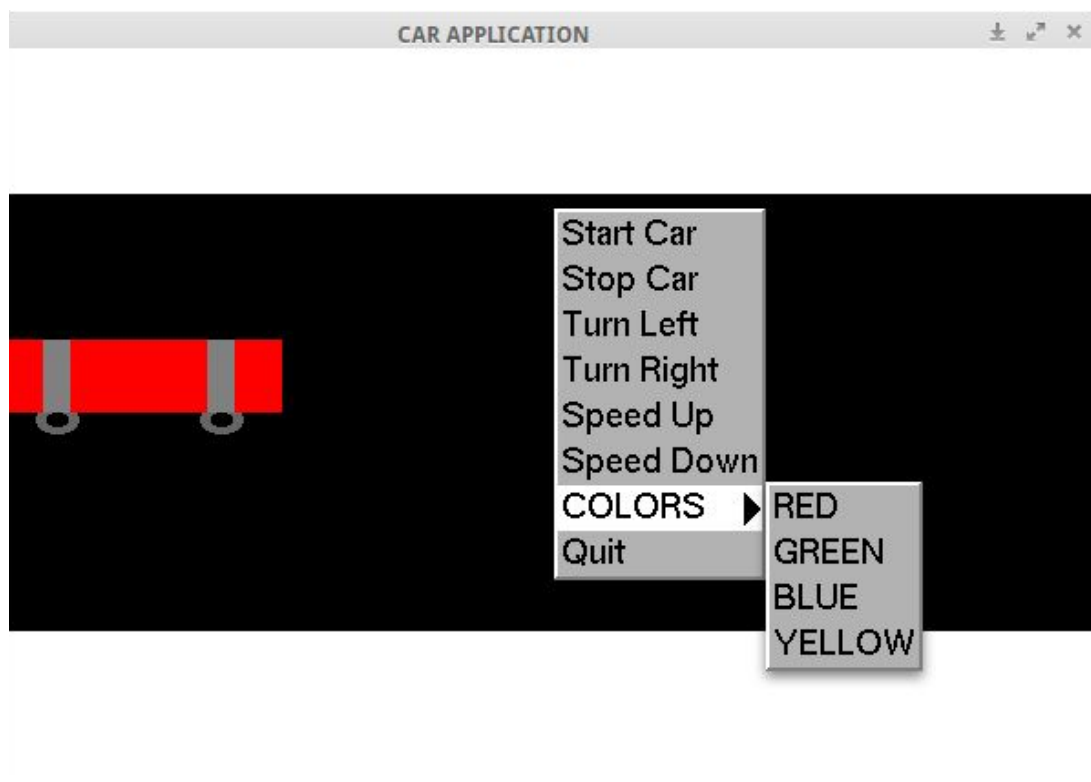
        body_color =id;
    }

int main(int argc, char **argv)
{
    int id;
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_RGB|GLUT_DOUBLE|GLUT_DEPTH);
    glutInitWindowSize(600,400);
    glutCreateWindow("CAR APPLICATION");

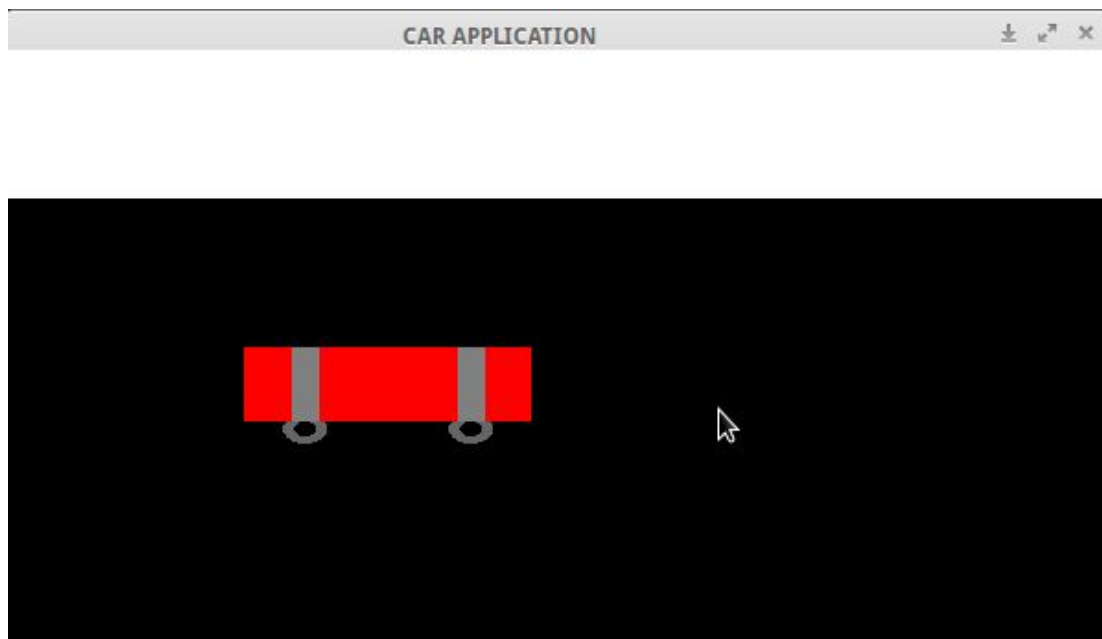
    glutDisplayFunc(display);
    init();
    glEnable(GL_DEPTH_TEST);
    id=glutCreateMenu(colormenu);
    glutAddMenuEntry("RED",0);
    glutAddMenuEntry("GREEN",1);
    glutAddMenuEntry("BLUE",2);
    glutAddMenuEntry("YELLOW",3);
    glutCreateMenu(mainmenu);
    glutAddMenuEntry("Start Car",1);
    glutAddMenuEntry("Stop Car",2);
    glutAddMenuEntry("Turn Left",3);
    glutAddMenuEntry("Turn Right",4);
    glutAddMenuEntry("Speed Up",5);
    glutAddMenuEntry("Speed Down",6);
    glutAddSubMenu("COLORS",id);
    glutAddMenuEntry("Quit",7);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    glutMainLoop();
    return 0;
}

```

## OUTPUT



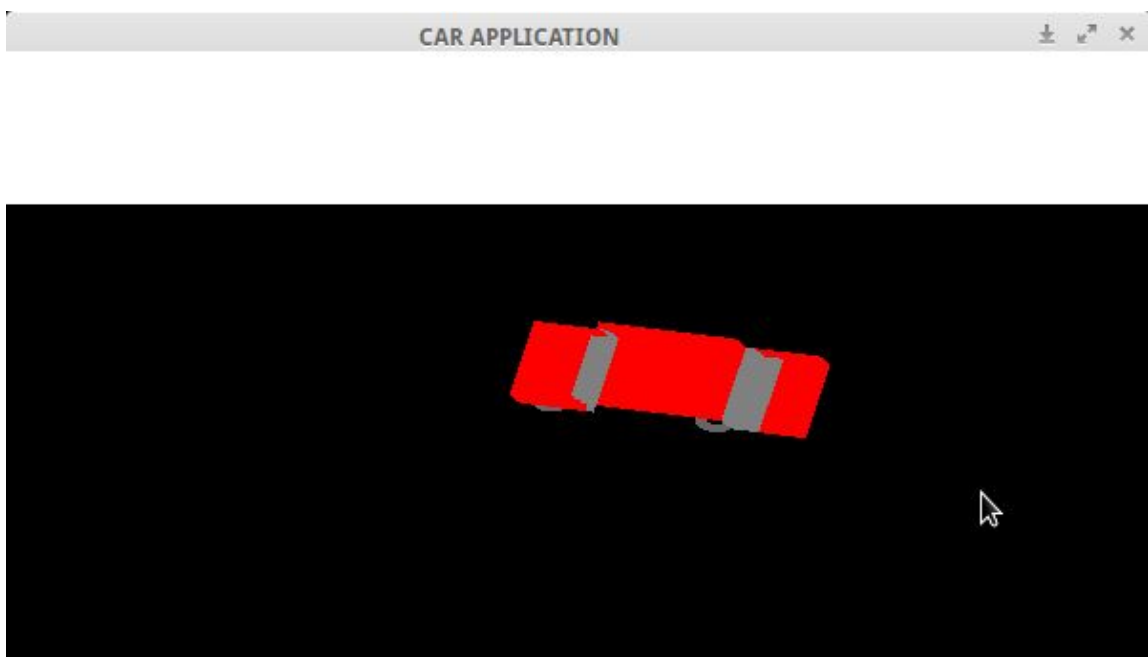
SCREENSHOT 1



SCREENSHOT 2: Start Car



**SCREENSHOT 3: Turn Left**



**SCREENSHOT 4: Turn Right**

