

MINI PROJECT ON MOVING CAR ANIMATION

NAME: Yogeshwar Patil

ROLL NO: SCOA13

INTRODUCTION

In computer graphics, use graphics.h which provide direct functions to draw different coordinate shapes (like circle, rectangle etc). By using these functions, we can draw different objects like car, hut, trees, etc. In this program, we will draw a moving car using line and circles.

Abstract

Main aim of this Mini Project is to illustrate 3D Car Animation using OpenGL Computer graphics. The concepts of OpenGL glut library and C++ has been used to create 3D Car Animation. We demonstrate all the

Abstract

Main aim of this Mini Project is to illustrate 3D Car Animation using OpenGL Computer graphics. The concepts of OpenGL glut library and C++ has been used to create 3D Car Animation. We demonstrate all the animation of 3D car including standalone car, driving in day/night, wheel effect, fog effect, animate in different directions, change of colours.

There is user interaction in this CG program. We have used input devices like mouse and key board to interact with program.

Implementation

- This program is implemented using various OpenGL functions and user defined functions which are given upcoming paragraphs.

To code this basic OpenGL Computer Graphics Mini Project. Projects we

defined functions which are given
upcoming paragraphs.

To code this basic OpenGL Computer Graphics Mini Project. Projects we need to understand the concept of car design and its movement. First, we need to code a car. There is different module for car coding - front module, body, window, wheel and silencer. We are not going to full code a glimpse of it below

Code for drawing front window -

```
glBegin (GL_TRIANGLES);  
    glColor3f (0.3,0.3,0.3);  
    glVertex3f (0.6, 0.5,0.6);  
    glVertex3f (0.7,0.65,0.6); //tri front  
window  
    glVertex3f (0.7,0.5,0.6);
```

Code for ignition system

```
glPushMatrix ();  
    glColor3f (0.7,0.7,0.7);
```

Code for ignition system

```
glPushMatrix ();  
    glColor3f (0.7,0.7,0.7);  
    glTranslatef (1.65,0.2,0.3);  
    glRotated (90.0,0,1,0);  
    gluCylinder(t,0.02,0.03,.5,10,10);  
glPopMatrix ();
```

Code for Wheel

```
glColor3f (0.7,0.7,0.7);  
glPushMatrix ();  
    glBegin (GL_LINE_STRIP);  
        for (theta=0; theta<360;  
theta=theta+20)  
        {  
            glVertex3f (0.6,0.2,0.62);  
  
            glVertex3f(0.6+(0.08*(cos(((theta+angle)  
*3.14)/180))),0.2+(0.08*(sin(((theta+angle)  
*3.14)/180))),0.62);  
        }  
    glEnd();  
glPopMatrix ();
```

```

glVertex3f(0.6+(0.08*(cos(((theta+angle)
*3.14)/180))),0.2+(0.08*(sin(((theta+angle)
*3.14)/180))),0.62);
    }
glEnd ();

```

This Computer Graphics Mini Project Projects not just show the animation of car the aim of project but also have title. In beginning to project there is title screen which **allow** to show case the name of projects, student name, guide name, college name and instructions for the program.

Various OpenGL functions used in this program.

glutInit (): interaction between the windowing system and OPENGL is initiated

glutInitDisplayMode (): used when double buffering is required and depth information is required

`glutInit ()`: interaction between the windowing system and OpenGL is initiated

`glutInitDisplayMode ()`: used when double buffering is required and depth information is required

`glutCreateWindow ()`: this opens the OpenGL window and displays the title at top of the window

`glutInitWindowSize ()` : specifies the size of the window

`glutInitWindowPosition ()`: specifies the position of the window in screen co-ordinates

`glutKeyboardFunc ()`: handles normal ascii symbols

`glutSpecialFunc ()`: handles special keyboard keys

`glutReshapeFunc ()`: sets up the callback function for reshaping the

`glutSpecialFunc ()`: handles special keyboard keys

`glutReshapeFunc ()`: sets up the callback function for reshaping the window

`glutIdleFunc ()`: this handles the processing of the background

`glutDisplayFunc ()`: this handle redrawing of the window

`glutMainLoop ()`: this starts the main [loop](#), it never returns

`glViewport ()`: used to set up the viewport

`glVertex3fv ()`: used to set up the points or vertices in three dimensions

`glColor3fv ()`: used to render colour to faces

`glFlush ()`: used to flush the pipeline

`glColor3fv ()`: used to render colour to faces

`glFlush ()`: used to flush the pipeline

`glutPostRedisplay ()`: used to trigger an automatic redrawer of the object

`glMatrixMode ()`: used to set up the required mode of the matrix

`glLoadIdentity ()`: used to load or initialize to the identity matrix

`glTranslatef ()`: used to translate or move the rotation centre from one point to another in three dimensions

`glRotatef ()`: used to rotate an object through a specified rotation angle

Interaction

In this graphics program we have user interaction with mouse and keyboard.

Mouse Interaction

Interaction

In this graphics program we have user interaction with mouse and keyboard.

Mouse Interaction

-> Press Right mouse button for menu and select submenu there.

- car model mode
- car driving mode
- fog effect
- wheel effect
- toggle light
- car colours
 - blue
 - red
 - green
 - black
 - yellow
 - grey
- Day mode
- Night mode

Keyboard Interaction

- Day mode
- Night mode

Keyboard Interaction

1. esc - Exit from the program
2. spacebar - Enter the main screen from start screen.
3. x- Rotate the car in 'x' direction
4. y- Rotate the car in 'y' direction
5. z- Rotate the car in 'z' direction
6. a- Increase the size of car in 'x' direction
7. s- Increase the size of car in 'y' direction
8. q- Increase the size of car in 'z' direction
9. u- Camera top view
10. f- Camera side view
11. left arrow key - Move car in forward direction
12. right arrow key- Move car in backward direction

Future Enhancement

8. q- Increase the size of car in 'z' direction

9. u- Camera top view

10. f- Camera side view

11. left arrow key - Move car in forward direction

12. right arrow key- Move car in backward direction

Future Enhancement

You can implement many enhancements in future like adding sound to it etc. Below are some future additions that can be added to projects for improvement.

1. Addition of sound
2. More cars can be added into it.
3. Surroundings can be created in much better way by adding tree etc.
4. Multiple roads can be added with street light
5. Traffic signal can be added
6. Advance camera view
7. More colours can be added.



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC



File Edit Search Run Compile Debug Project Options Window Help

[] \TURBOC3\PROJECTS\MINI.CPP

1=[↑]

```
#include <stdio.h>
#include <graphics.h>
#include <dos.h>
#include <conio.h>
void main()
{
    clrscr();
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"C:\\TURBOC3\\BGI");

    for(int i=0;i<500;i++)
    {
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
for(int i=0;i<500;i++)
```

```
//body
```

```
//lower body finish
```

```
line(330+i,370,330+i,350);
line(50+i,370,50+i,350);
line(330+i,350,280+i,330);
```

1:1



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC



File Edit Search Run Compile Debug Project Options Window Help

[■] \TURBOC3\PROJECTS\MINI.CPP 1=1↑

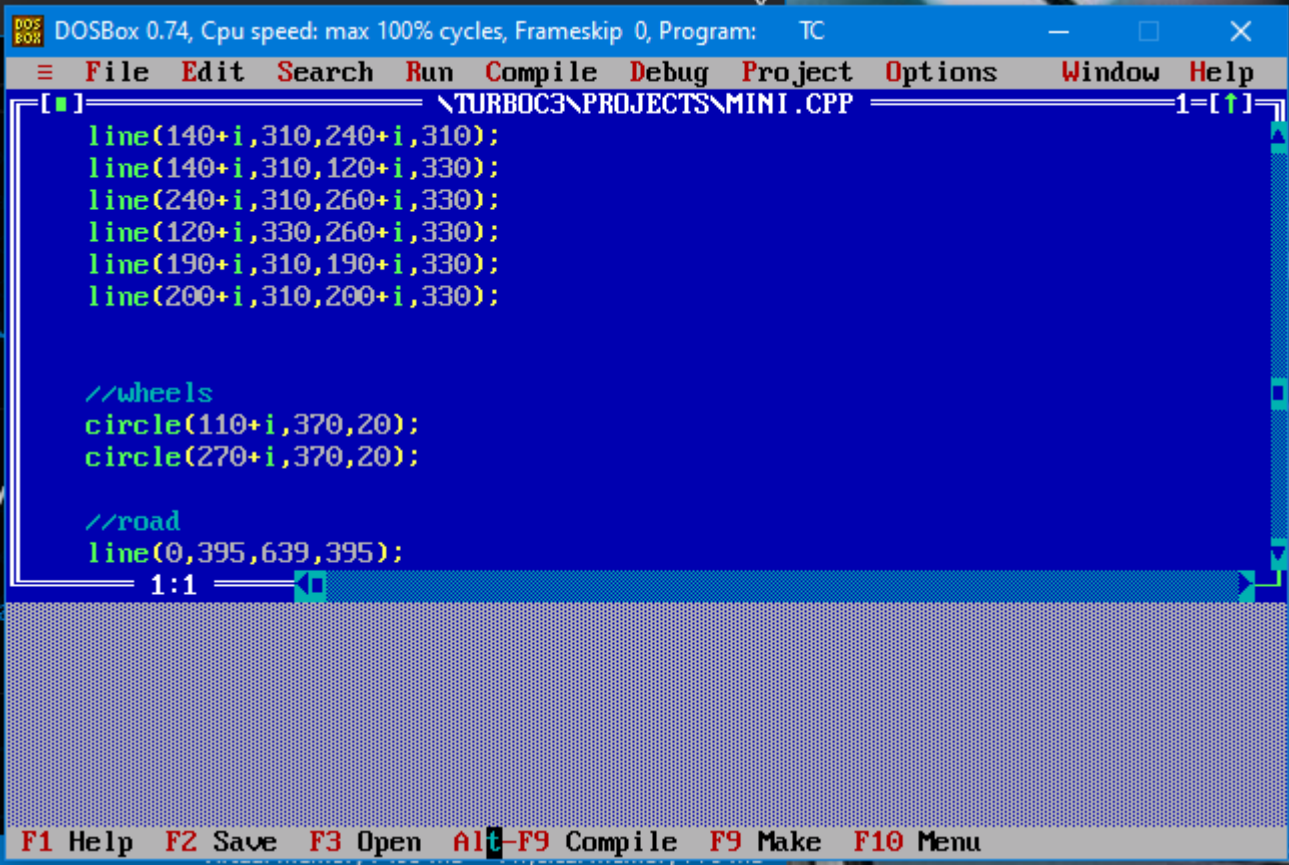
```
line(330+i,370,330+i,350);  
line(50+i,370,50+i,350);  
line(330+i,350,280+i,330);  
line(50+i,350,100+i,330);  
line(280+i,330,250+i,300);  
line(100+i,330,130+i,300);  
line(130+i,300,250+i,300);
```

//window

```
line(140+i,310,240+i,310);  
line(140+i,310,120+i,330);  
line(240+i,310,260+i,330);  
line(120+i,330,260+i,330);
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu





DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

File Edit Search Run Compile Debug Project Options Window Help

[■] \TURBOC3\PROJECTS\MINI.CPP 1=[↑]

//road

line(0,395,639,395);

delay(12);

cleardevice();

}

getch();

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

