Computer Graphics

Section: I Group: 7

Name	ID
1. Ahmed , Md. Rasel	17-33188-1
2.Dam Biprojit Kumar	17-33189-1
3.Hossain Nur	18-36279-1
4..Dola,Shionty Ghosh	18-38013-2

Table of Content:

Content	Page No.
1. Introduction	2
2. Project Proposal	2
3. Schematic Diagram	3-4
4.List of Object	**5**
5. Function to Represent the object	**5-8**
6.Interactive Function	**8**
7.Task assignment and code of function.	**8-63**
8.Conclution	**64**

Introduction

On the most fundamental level, OpenGL is a software interface that allows a programmer to communicate with graphics hardware. Of course, there is much more to it than that, and you will be glad to know that this book explains the finer details of OpenGL. But before we get our hands dirty and start coding, you'll need to know a little about the history of Computer Graphics and OpenGL.

The pushMatrix() function saves the current coordinate system to the stack and popMatrix() restores the prior coordinate system. pushMatrix() and popMatrix() are used in conjuction with the other transformation functions and may be embedded to control the scope of the transformations.

The math.h header defines various mathematical functions and one macro. All the functions available in this library take double as an argument and return double as the result. Let us discuss some important functions one by one.

The header file which is added to the program that one has written is basically what 'include stdio.h.' means. Stdio.h is known to contain the input and output operations like "printf" or "scanf" etc. "h" extension means the header file.

windows.h is a Windows-specific header file for the C and C++ programming languages which contains declarations for all of the functions in the Windows API, all the common macros used by Windows programmers, and all the data types used by the various functions and subsystems.

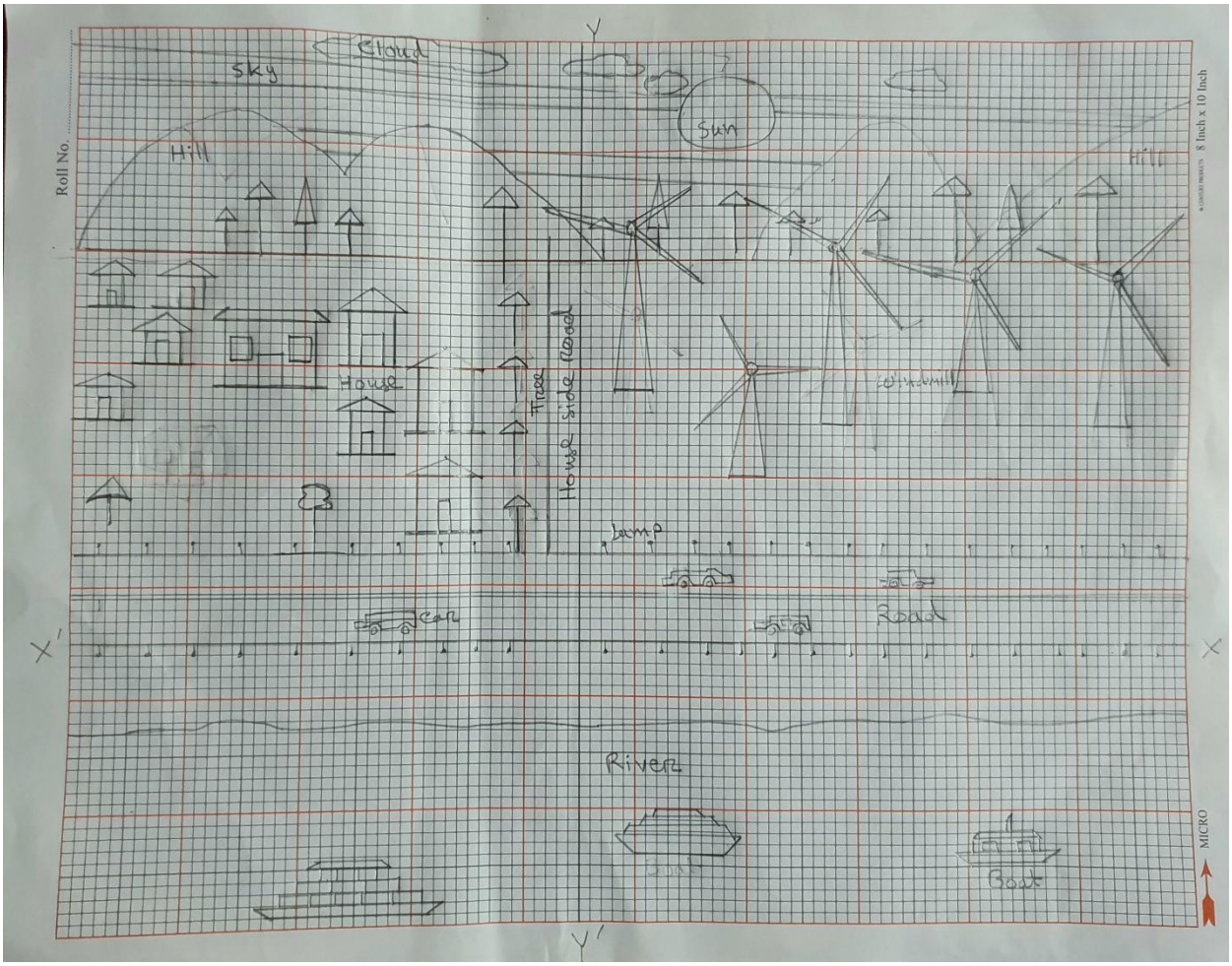
The **glTranslatef** function produces the translation specified by (x, y, z). The translation vector is used to compute a 4x4 translation matrix: The current matrix (see glMatrixMode) is multiplied by this translation matrix, with the product replacing the current matrix.

Project Proposal

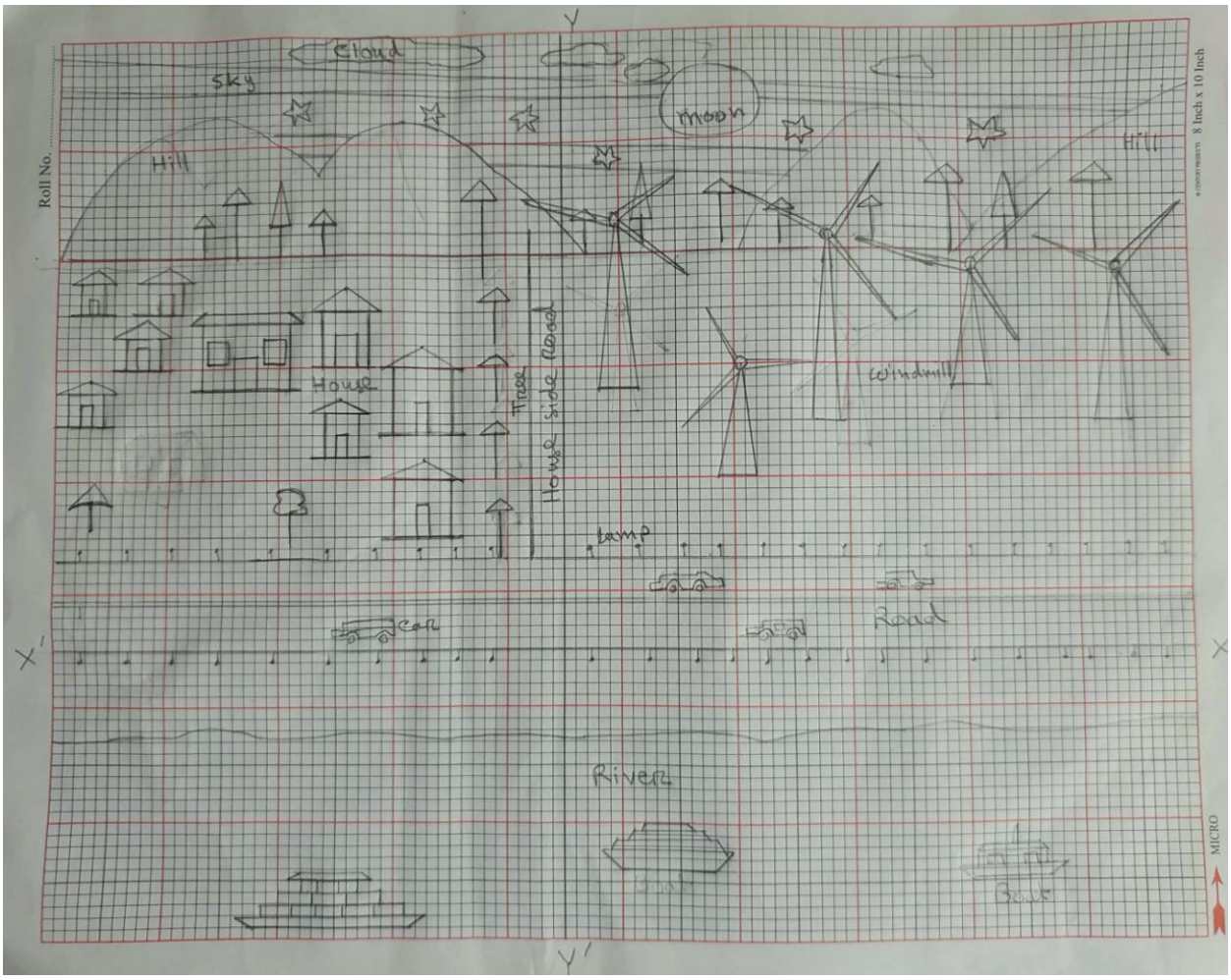
This is our project which actually describe a scenario that have been made with some content and interactive operation. In this picture there are some windmill. Windmills run all day and night, even in the rain. It is actually balancing the temperature of weather 24/7.There are two roads next to the windmill. There is some houses. The houses are surrounded by trees.

There is a river on the side of the road. Boats and ships are moving on the river. Cars are moving on both sides of the road. There are hills behind the house. Clouds are floating in the sky. The sun effects the windmills. We have made this project some functionally which is actually shown some animation in the project. In our project we have some object like hill,Sky,Hill,Sun,River,Car,House,Land, Cloud,Boat,Moon,Hill tree,Windmill ,Star, Lamp ,Tree Road ,House side Road.

Schematic Diagram



Day mood


Cloud
sky
moon
Hill
Hill
House
Tree
House side wood
windmill
lamp
car
Road
River
Boat
Roll No.
8 Inch x 10 Inch
MICRO
X
X'
Y
Y'

Night

List of Object:

i. Sky
ii. Hill
iii. Sun
iv. River
v. Car
vi. House
vii. Land
viii. Cloud
ix. Boat
x. Moon
xi. Hill tree
xii. Windmill
xiii. Star
xiv. Lamp
xv. Tree
xvi. Road
xvii. House side road

Function To Represent The Object:

Object	Function to represent object	Function Id	Group Member
Sky	Sky()	Sk-1	Dola
Hill	Hill()	H-1	Dola
Sun	Sun()	S-1	Dola

Land	Land()	L-1	Biprojit
River	River()	R-1	Dola
Road	Road()	Rd-1	Biprojit
Car	Car()	Cr-1	Rasel
car	Car1()	Cr-2	Rasel
House	house1()	Hs-1	Biprojit
House	house3()	Hs-3	Biprojit
House	house4()	Hs-4	Biprojit
House	house5()	Hs-5	Biprojit
House	house6()	Hs-6	Biprojit
House	house7()	Hs-7	Biprojit
House	house8()	Hs-8	Biprojit
House	house9()	Hs-9	Biprojit
House	house10()	Hs-10	Biprojit
Cloud	cloud()	Cd-1	Dola
Boat	boat()	B-1	Biprojit
Moon	Moon()	M-1	Nur
HillTree	hillTree();	HT-1	Dola
HillTree	hillTree2();	HT-2	Dola
HillTree	hillTree3();	HT-3	Dola
Tree	tree1();	T-1	Rasel
Tree	tree2();	T-2	Rasel
Lamp	RoadLeftLamp();	L-1	Rasel
Lamp	RoadRightLamp();	L-2	Rasel
Colud	cloud_one();	C-1	Dola
Colud	cloud_two();	C-2	Dola
Colud	cloud_three();	C-3	Dola
Colud	cloud_four();	C-4	Dola
Colud	cloud_five();	C-5	Dola

Colud	cloud_six();	C-6	Dola
Colud	cloud_seven();	C-7	Dola
Windmill	Windmill_One();	WM-1	Nur
Windmill	Windmill_Two();	WM-2	Nur
Windmill	Windmill_Three();	WM-3	Nur
Windmill	Windmill_four();	WM-4	Nur
Windmill	Windmill_five();	WM-5	Nur
Boat	boat3();	B-3	Biprojit
Boat	boat2();	B-2	Biprojit
Boat	boat();	B-3	Biprojit
Start	star1()	Str-1	Biprojit
Start	star2()	Str-2	Biprojit
Start	star3()	Str-3	Biprojit
Start	star4()	Str-4	Biprojit
Start	star5()	Str-5	Biprojit
Start	star6()	Str-6	Biprojit
	R_sky()	Rky-1	Nur
	DrawCircle() for hill	DC-1	Dola
	Circle()	C-1	Dola
	R_hill()	Rh-1	Nur
	Windmill_Blade()	WB-1	Nur
	R_land()	RLnd-1	Nur
	R_river()		Nur
	R_tree1()		Nur
	R_tree2()		Nur
	R_leftroadtree()		Nur
	RoadLeftLamp()		Rasel
	RoadRightLamp()		Rasel
	night_color()		Nur

	Rain(int value)		Nur

Interactive Function:

Function	Interactive Function	Function Id
Night_Display()	Night Mode	NM-1
glutDisplayFunc(display)	Day Mode	DM-1
Rainy_View()	Rainy mood	RM-1
Rain(_rain)	Start rain	Sr-1
	Stop rain	StpR-1
exit(0)	Exit window	Ex-1

Task Assignment and Code of Function:

Object of function	code

	void sky()
	{
	glBegin(GL_QUADS);
	glColor3ub(174,214,241);
	glVertex2i(0,550);
	glVertex2i(1000,550);
	glVertex2i(1000,1000);
	glVertex2i(0,1000);
	glEnd();

```
   glBegin(GL_QUADS);
   glColor3ub(214,234,248);
glVertex2i(0,650);
   glVertex2i(1000,640);
   glVertex2i(1000,690);
   glVertex2i(0,720);
   glEnd();

   glBegin(GL_QUADS);
   glColor3ub(214,234,248);
glVertex2i(0,850);
   glVertex2i(1000,800);
   glVertex2i(1000,820);
   glVertex2i(0,905);
   glEnd();


}
```

```
void sky2()
{
   glBegin(GL_QUADS);
     glColor3ub(0, 51, 204);
glVertex2f(-1.0f, 0.45f);
glVertex2f(1.0f, 0.45f);
   glVertex2f(1.0f, 1.0f);
glVertex2f(-1.0f, 1.0f);
   glEnd();
   glBegin(GL_QUADS);
   glColor3ub(214,234,248);
glVertex2i(0,650);
   glVertex2i(1000,640);
   glVertex2i(1000,690);
   glVertex2i(0,720);
   glEnd();

   glBegin(GL_QUADS);
   glColor3ub(214,234,248);
glVertex2i(0,850);
   glVertex2i(1000,800);
   glVertex2i(1000,820);
   glVertex2i(0,905);
   glEnd();


}
```

```
void Hill()
{
   glPushMatrix();

 glColor3ub(51, 153, 51);



  glBegin(GL_TRIANGLES);
 glVertex2i(48,850);
 glVertex2i(-135,500);
   glVertex2i(100,500);
 glEnd();
 DrawCircle(79,847,31,1000);//
 glBegin(GL_TRIANGLES);
 glVertex2i(47,848);
 glVertex2i(110,848);
   glVertex2i(90,500);
 glEnd();
  glBegin(GL_TRIANGLES);
 glVertex2i(110,848);
 glVertex2i(70,400);
   glVertex2i(280,400);
 glEnd();



  glBegin(GL_TRIANGLES);
 glVertex2i(230,758);
 glVertex2i(90,500);
   glVertex2i(250,500);
 glEnd();

glBegin(GL_TRIANGLES);
 glVertex2i(240,500);
 glVertex2i(390,500);
   glVertex2i(277,755);
 glEnd();

 glBegin(GL_TRIANGLES);
 glVertex2i(230,758);
 glVertex2i(277,758);
   glVertex2i(250,500);
 glEnd();
```

```
   DrawCircle(252.5,755,23,1000);//



glBegin(GL_TRIANGLES);
 glVertex2i(600,500);
  glVertex2i(750,500);
    glVertex2i(700,790);
  glEnd();
glBegin(GL_TRIANGLES);
 glVertex2i(690,500);
  glVertex2i(880,500);
    glVertex2i(772,790);
  glEnd();

glBegin(GL_TRIANGLES);
 glVertex2i(700,790);
  glVertex2i(773,790);
    glVertex2i(720,500);
  glEnd();
  DrawCircle(736,789,35,1000);//

    glBegin(GL_TRIANGLES);
 glVertex2i(800,500);
  glVertex2i(1000,500);
    glVertex2i(1000,890);
  glEnd();
  glPopMatrix();



}
```

```
void cloud_model_one(){

    glColor3f(1.25, 0.924, 0.930);

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Top
```

```
glPushMatrix();
glTranslatef(340, 225, 0);
circle(16);
glPopMatrix();

///Right

glPushMatrix();
glTranslatef(360,210,0);
circle(16);
glPopMatrix();


///middle_Fill
glPushMatrix();
glTranslatef(355,210,0);
circle(16);
glPopMatrix();

glPushMatrix();
glTranslatef(350,210,0);
circle(16);
glPopMatrix();

glPushMatrix();
glTranslatef(345,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
glTranslatef(340,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
glTranslatef(335,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
glTranslatef(330,204,0);
circle(10);
glPopMatrix();
```

```
    glPushMatrix();
    glTranslatef(325,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(320,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(315,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(310,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(305,204,0);
    circle(10);
    glPopMatrix();

    ///****Fill End****

}
```

```
void cloud_model_Two(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(305,205,0);
    circle(10);
    glPopMatrix();

    ///Top

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Right_Part
```

```
    glPushMatrix();
    glTranslatef(334,207,0);
    circle(10);
    glPopMatrix();

    ///Bottom_Part
    glPushMatrix();
    glTranslatef(320,207,0);
    circle(10);
    glPopMatrix();



}
```

```
void cloud_model_Three(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(300,200,0);
    circle(15);
    glPopMatrix();

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Top
    glPushMatrix();
    glTranslatef(340,220,0);
    circle(16);
    glPopMatrix();

    ///Top_Right
    glPushMatrix();
    glTranslatef(360,210,0);
    circle(15);
    glPopMatrix();

    ///Right_Part
    glPushMatrix();
    glTranslatef(380,200,0);
```

	circle(15);
	glPopMatrix();
	
	///Bottom_Right
	glPushMatrix();
	glTranslatef(360,190,0);
	circle(20);
	glPopMatrix();
	
	///Bottom_Left
	glPushMatrix();
	glTranslatef(320,190,0);
	circle(20);
	glPopMatrix();
	
	///Bottom
	glPushMatrix();
	glTranslatef(340,190,0);
	circle(20);
	glPopMatrix();
	
	///****Fill End****
	
	}

	void cloud_one(){
	glPushMatrix();
	glTranslatef(cx,700,0);
	cloud_model_one();
	glPopMatrix();
	
	}

	void cloud_two(){
	glPushMatrix();
	glTranslatef(bx+-200,750,0);
	cloud_model_one();
	glPopMatrix();
	
	}

	void cloud_three(){
	glPushMatrix();
	glTranslatef(ax-80,670,0);
	cloud_model_Two();

	glPopMatrix();
	
	}

	void cloud_four(){
	glPushMatrix();
	glTranslatef(dx+300,675,0);
	cloud_model_Two();
	glPopMatrix();
	
	}

	void cloud_five(){
	
	glPushMatrix();
	glTranslatef(ax+-300,770,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void cloud_six(){
	
	glPushMatrix();
	glTranslatef(cx+-500,710,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void cloud_seven(){
	
	glPushMatrix();
	glTranslatef(cx+-200,780,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void sun()
	{
	glPushMatrix();
	glTranslatef(480,800, 0);
	
	glBegin(GL_POLYGON);
	glColor3ub(249,215,28);
	for(int i=0;i<200;i++)
	{
	float pi=3.1416;
	float A=(i*2*pi)/200;

	float r=45;
	float x = r * cos(A);
	float y = r * sin(A);
	glVertex2f(x,y);
	}
	glEnd();
	glPopMatrix();
	
	}

	void Windmill(){
	
	///Windmill_Stand
	glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	Windmill_Stand_Model();
	glPopMatrix();
	
	///Windmill_Motor
	glColor3f(0.11, 0.23, 0.36);
	glPushMatrix();
	glTranslatef(380,650,0);
	circle(10);
	glPopMatrix();
	
	///Windmill_Rotary_Blades
	glColor3f(0.11, 0.23, 0.36);
	glPushMatrix();
	glTranslatef(380,650,0);
	Windmill_Blade();
	glPopMatrix();
	
	}

	void land()
	{
	glColor3ub(224, 181, 79);
	glBegin(GL_QUADS);
	glVertex2i(0, 120);
	glVertex2i(1000, 120);
	glVertex2i(1000, 550);
	glVertex2i(0, 550);

	glEnd(); }

	void River()
	{
	glColor3ub(0, 255, 255);
	glBegin(GL_QUADS);
	glVertex2i(0, 120);
	glVertex2i(1000, 120);
	glVertex2i(1000, 0);
	glVertex2i(0, 0);
	glEnd();
	
	glBegin(GL_POLYGON);
	glVertex2i(200, 120);
	glVertex2i(400, 120);
	glVertex2i(300, 140);
	glEnd();
	
	glBegin(GL_POLYGON);
	glVertex2i(600-150, 120);
	glVertex2i(800-150, 120);
	glVertex2i(700-150, 140);
	glEnd();
	
	glBegin(GL_POLYGON);
	glVertex2i(750, 120);
	glVertex2i(800, 120);
	glVertex2i(850, 140);
	glEnd();
	
	glBegin(GL_POLYGON);
	glVertex2i(850, 120);
	glVertex2i(900, 120);
	glVertex2i(950, 140);
	glEnd();
	
	glBegin(GL_POLYGON);
	glVertex2i(900, 120);
	glVertex2i(970, 120);
	glVertex2i(1000, 140);
	glEnd();
	}

	void road()
	{
	glColor3ub(66, 66, 49);

```
glBegin(GL_QUADS);
glVertex2i(0,330);
glVertex2i(1000,330);
glVertex2i(1000,280);
glVertex2i(0,280);
glEnd();


//House side road

glColor3ub(66, 66, 49);
glBegin(GL_QUADS);
glVertex2i(300,330);
glVertex2i(330-10,330);
glVertex2i(300-10,550);
glVertex2i(270,550);
glEnd();
    //White line
glColor3ub(255, 255, 255);
glBegin(GL_LINES);
glVertex2i(0,304);
glVertex2i(1000,304);
glVertex2i(0,307);
glVertex2i(1000,307);
glEnd();
}
```

```
void car2()
{
   glPushMatrix();
   glTranslatef(-car1_speed, 0.0, 0.0);

   glPushMatrix();
   glTranslatef(923,295, 0);
   glBegin(GL_POLYGON);
   glColor3ub(255,0,0);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
```

```
  glEnd();
  glPopMatrix();
  glPushMatrix();
   glTranslatef(941,295, 0);
   glBegin(GL_POLYGON);
   glColor3ub(255,0,0);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
  glEnd();
  glPopMatrix();

   glColor3f(255,0,0);
   glBegin(GL_QUADS);
   glVertex2i(10+900, 317-25);
   glVertex2i(10+40+900, 317-25);
   glVertex2i(10+40+900, 325-25);
   glVertex2i(10+900, 325-25);
   glVertex2i(10+10+900, 325-25);
   glVertex2i(10+40-5+900, 325-25);
   glVertex2i(10+40-8+900, 317+20-25);
   glVertex2i(10+15+900, 317+20-25);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_LINES);
   glVertex2i(30-20+50-50+900, 317+4-25);
   glVertex2i(30+40-20+50-50+900, 317+4-25);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_QUADS);
   glVertex2i(10+10+3+900, 325-25);
   glVertex2i(10+40-5-12+900, 325-25);
   glVertex2i(10+40-5-12+900, 317+20-25-4);
   glVertex2i(10+15+1+900, 317+20-25-4);
   glVertex2i(10+40-5-12+2+900, 325-25);
   glVertex2i(10+40-5-12+10+900, 325-25);
   glVertex2i(10+40-5-12+10-2+900, 317+20-25-
4);
   glVertex2i(10+40-5-12+2+900, 317+20-25-4);
```

```
   glEnd();
   glPopMatrix();
///
   glPushMatrix();
   glTranslatef(-car2_speed, 0.0, 0.0);

   glPushMatrix();
   glTranslatef(1003,295, 0);
   glBegin(GL_POLYGON);
   glColor3ub(0,0,255);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y);
}
  glEnd();
  glPopMatrix();
  glPushMatrix();
   glTranslatef(1003+18,295, 0);
   glBegin(GL_POLYGON);
   glColor3ub(0,0,255);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y);
}
  glEnd();
  glPopMatrix();

   glColor3f(0,0,255);
   glBegin(GL_QUADS);
   glVertex2i(10+900+80, 317-25);
   glVertex2i(10+40+900+80, 317-25);
   glVertex2i(10+40+900+80, 325-25);
   glVertex2i(10+900+80, 325-25);
   glVertex2i(10+10+900+80, 325-25);
   glVertex2i(10+40-5+900+80, 325-25);
```

	glVertex2i(10+40-8+900+80, 317+20-25);
	glVertex2i(10+15+900+80, 317+20-25);
	glEnd();
	glColor3f(0,255,0);
	glBegin(GL_LINES);
	glVertex2i(30-20+50-50+900+80, 317+4-25);
	glVertex2i(30+40-20+50-50+900+80, 317+4-25);
	glEnd();
	glColor3f(0,255,0);
	glBegin(GL_QUADS);
	glVertex2i(10+10+3+900+80, 325-25);
	glVertex2i(10+40-5-12+900+80, 325-25);
	glVertex2i(10+40-5-12+900+80, 317+20-25-4);
	glVertex2i(10+15+1+900+80, 317+20-25-4);
	glVertex2i(10+40-5-12+2+900+80, 325-25);
	glVertex2i(10+40-5-12+10+900+80, 325-25);
	glVertex2i(10+40-5-12+10-2+900+80, 317+20-25-4);
	glVertex2i(10+40-5-12+2+900+80, 317+20-25-4);
	glEnd();
	glPopMatrix();
	}

	void house4()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(220, 470);
	glVertex2i(260, 470);
	glVertex2i(240, 490);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(228, 470);
	glVertex2i(252, 470);
	glVertex2i(252, 440);
	glVertex2i(228, 440);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(224, 440);
	glVertex2i(256, 440);
	glVertex2i(256, 435);
	glVertex2i(224, 435);

```
glEnd();
///Door
   glColor3ub(1, 51, 102);
glBegin(GL_QUADS);
   glVertex2i(236, 460);
glVertex2i(244, 460);
glVertex2i(244, 440);
   glVertex2i(236, 440);
glEnd();
}
```

```
void HillTree()
{
   float a;
   for(a=0;a<1000;a+=100)
   {
      float b=a+2;
      glColor3ub(0, 0, 0);
      glBegin(GL_QUADS);
      glVertex2i(a, 545);
      glVertex2i(b, 545);
      glVertex2i(b, 560);
      glVertex2i(a, 560);
      glEnd();
      glColor3ub(102, 204, 0);
      glBegin(GL_POLYGON);
      glVertex2i(a-5, 560);
      glVertex2i(b+5, 560);
      glVertex2i(a+1,570);
      glEnd();
      glColor3ub(102, 204, 0);
      glBegin(GL_POLYGON);
      glVertex2i(a-5, 560+4);
      glVertex2i(b+5, 560+4);
      glVertex2i(a+1,570+4);
      glEnd();
   }
}
```

```
void tree2()
{
  glPushMatrix();
  glColor3ub(0, 0, 0);
   glBegin(GL_QUADS);
   glVertex2i(169, 390);
   glVertex2i(171, 390);
```

	glVertex2i(171, 400+20);
	glVertex2i(169, 400+20);
	glEnd();
	glPopMatrix();
	
	glPushMatrix();
	glTranslatef(170,430, 0);
	glBegin(GL_POLYGON);
	glColor3ub(50,205,50);
	for(int i=0;i<200;i++)
	{
	float pi=3.1416;
	float A=(i*2*pi)/200;
	float r=8;
	float x = r * cos(A);
	float y = r * sin(A);
	glVertex2f(x,y);
	}
	glEnd();
	glPopMatrix();
	
	glPushMatrix();
	glTranslatef(170,420, 0);
	glBegin(GL_POLYGON);
	glColor3ub(50,205,50);
	for(int i=0;i<200;i++)
	{
	float pi=3.1416;
	float A=(i*2*pi)/200;
	float r=8;
	float x = r * cos(A);
	float y = r * sin(A);
	glVertex2f(x,y);
	}
	glEnd();
	glPopMatrix();
	}

	void moon()
	{
	glPushMatrix();
	glTranslatef(500,850, 0);
	glPushMatrix();
	glBegin(GL_POLYGON);
	glColor3f(1.0, 1.0, 1.0);
	for(int i=0;i<200;i++)

	{
	float pi=3.1416;
	float A=(i*2*pi)/200;
	float r=50;
	float x = r * cos(A);
	float y = r * sin(A);
	glVertex2f(x,y);
	}
	glEnd();
	glPopMatrix();
	glPopMatrix();
	
	}

	void RoadLeftLamp()
	{
	
	float a;
	for(a=0; a<=1000; a+=70)
	{
	float b=a+3;
	glColor3ub(66, 66, 49);
	glBegin(GL_QUADS);
	glVertex2i(a,332);
	glVertex2i(b,332);
	glVertex2i(b,350);
	glVertex2i(a,350);
	glEnd();
	glColor3ub(255, 255, 255);
	glBegin(GL_QUADS);
	glVertex2i(a,350);
	glVertex2i(b,350);
	glVertex2i(b,355);
	glVertex2i(a,355);
	glEnd();
	}
	}

	void boat()
	{
	glPushMatrix();
	glTranslatef(boat_speed,0.0f, 0.0f);
	glColor3ub(0, 0,128);
	glBegin(GL_POLYGON);

```
    glVertex2i(35,0);
    glVertex2i(95,0);
    glVertex2i(130,20);
    glVertex2i(0,20);
    glEnd();
  //glColor3ub(255,51,113);*/
    glBegin(GL_POLYGON);
    glVertex2i(25,20);
    glVertex2i(105,20);
    glVertex2i(105,40);
    glVertex2i(25,40);
    glEnd();

    glBegin(GL_POLYGON);
    glVertex2i(35,40);
    glVertex2i(95,40);
    glVertex2i(95,60);
    glVertex2i(35,60);
glEnd();
     glBegin(GL_POLYGON);
    glVertex2i(45,60);
    glVertex2i(85,60);
    glVertex2i(85,80);
    glVertex2i(45,80);
    glEnd();
     glBegin(GL_POLYGON);
    glVertex2i(35,80);
    glVertex2i(95,80);
    glVertex2i(90,90);
    glVertex2i(40,90);
    glEnd();

    //window11
    glColor3ub(204, 255, 255);
      glBegin(GL_POLYGON);
    glVertex2i(31.25,25);
    glVertex2i(46.25,25);
    glVertex2i(46.25,35);
    glVertex2i(31.25,35);
    glEnd();

    //window12
    glBegin(GL_POLYGON);
    glVertex2f(48.75,25);
     glVertex2f(63.75,25);
```

```
      glVertex2f(63.75,35);
       glVertex2f(48.75,35);
    glEnd();

    //window13
      glBegin(GL_POLYGON);
    glVertex2f(66.25,25);
     glVertex2f(81.25,25);
      glVertex2f(81.25,35);
       glVertex2f(66.25,35);
    glEnd();

    //w14
      glBegin(GL_POLYGON);
    glVertex2f(83.75,25);
     glVertex2f(98.75,25);
      glVertex2f(98.75,35);
       glVertex2f(83.75,35);
    glEnd();
//w21
  glBegin(GL_POLYGON);
    glVertex2f(40,45);
     glVertex2f(55,45);
      glVertex2f(55,55);
       glVertex2f(40,55);
    glEnd();
//w22
    glBegin(GL_POLYGON);
    glVertex2f(57.5,45);
     glVertex2f(72,45);
      glVertex2f(72.5,55);
       glVertex2f(57,55);
    glEnd();
//w23
glBegin(GL_POLYGON);
    glVertex2f(75,45);
     glVertex2f(90,45);
      glVertex2f(90,55);
       glVertex2f(75,55);
    glEnd();

    //w31
    glBegin(GL_POLYGON);
    glVertex2f(50,65);
     glVertex2f(62.5,65);
```

```
      glVertex2f(62.5,75);
       glVertex2f(50,75);
    glEnd();

    //w32
    glBegin(GL_POLYGON);
    glVertex2f(67.5,65);
     glVertex2f(80,65);
      glVertex2f(80,75);
       glVertex2f(67.5,75);
    glEnd();
glPopMatrix();

glFlush();


}
```

```
void boat3()
{
   glPushMatrix();
   glTranslatef(boat3_speed,0.0f, 0.0f);
   glColor3ub(128, 0, 0);

      glBegin(GL_POLYGON);
    glVertex2i(1120,50);
    glVertex2i(1180,50);
    glVertex2i(1200,80);
    glVertex2i(1100,80);
    glEnd();

       glBegin(GL_POLYGON);
    glVertex2i(1115,80);
    glVertex2i(1185,80);
    glVertex2i(1185,100);
    glVertex2i(1115,100);
    glEnd();

       glBegin(GL_POLYGON);
    glVertex2i(1115,100);
    glVertex2i(1185,100);
    glVertex2i(1175,110);
    glVertex2i(1125,110);
    glEnd();

       glBegin(GL_POLYGON);
```

	glVertex2i(1148,110);
	glVertex2i(1152,110);
	glVertex2i(1152,128);
	glVertex2i(1148,120);
	glEnd();
	
	glColor3ub(255, 255, 153);
	
	glBegin(GL_LINES);
	glVertex2i(1115,100);
	glVertex2i(1185,100);
	glVertex2i(1116,100);
	glVertex2i(1116,80);
	glVertex2i(1184,100);
	glVertex2i(1184,80);
	glVertex2i(1116,80);
	glVertex2i(1184,80);
	glEnd();
	
	//w1
	glColor3ub(204, 255, 255);
	glBegin(GL_POLYGON);
	glVertex2i(1120,85);
	glVertex2i(1135,85);
	glVertex2i(1135,95);
	glVertex2i(1120,95);
	glEnd();
	//w2
	glBegin(GL_POLYGON);
	glVertex2i(1180,85);
	glVertex2i(1165,85);
	glVertex2i(1165,95);
	glVertex2i(1180,95);
	glEnd();
	
	glPopMatrix();
	//glFlush();
	
	}

	void boat2()

```
{
   glPushMatrix();
   glTranslatef(boat2_speed,0.0f, 0.0f);
   glColor3ub(128, 128, 0);
    glBegin(GL_POLYGON);
    glVertex2i(20,30);
    glVertex2i(80,30);
    glVertex2i(97,50);
    glVertex2i(5,50);
    glEnd();

     glBegin(GL_POLYGON);
    glVertex2i(10,50);
    glVertex2i(90,50);
    glVertex2i(75,70);
    glVertex2i(25,70);
    glEnd();

      glBegin(GL_POLYGON);
    glVertex2i(25,70);
    glVertex2i(75,70);
    glVertex2i(60,90);
    glVertex2i(40,90);
    glEnd();
     glBegin(GL_TRIANGLES);
    glVertex2i(50,104);
    glVertex2i(60,90);
    glVertex2i(40,90);
    glEnd();

   glBegin(GL_POLYGON);
    glVertex2i(50,104);
    glVertex2i(51,104);
    glVertex2i(51,124);
    glVertex2i(50,124);
    glEnd();

     glColor3ub(255, 0, 0);
     glBegin(GL_TRIANGLES);
    glVertex2f(57,113);
    glVertex2f(51,118);
    glVertex2f(51,109);
    glEnd();

    glColor3ub(255, 255, 204);
```

```
      glBegin(GL_LINES);
      glVertex2i(97,50);
      glVertex2i(5,50);
      glVertex2i(10,50);
      glVertex2i(90,50);
      glVertex2i(25,70);
      glVertex2i(75,70);
      glVertex2i(40,90);
      glVertex2i(60,90);
      glEnd();

      //w11
       glColor3ub(204, 255, 255);
      glBegin(GL_POLYGON);
    glVertex2i(17,55);
    glVertex2i(40,55);
    glVertex2i(40,65);
    glVertex2i(25,65);
    glEnd();

    //w12
     glBegin(GL_POLYGON);
    glVertex2f(42.5,55);
    glVertex2f(57.5,55);
    glVertex2f(57.5,65);
    glVertex2f(42.5,65);
    glEnd();

  //w13
    glBegin(GL_POLYGON);
    glVertex2i(60,55);
    glVertex2i(83,55);
    glVertex2i(75,65);
    glVertex2i(60,65);
    glEnd();
    //w21

  glBegin(GL_POLYGON);
    glVertex2f(31,75);
    glVertex2f(48.5,75);
    glVertex2f(48.5,85);
    glVertex2f(40,85);
    glEnd();

    //w22
```

	glBegin(GL_POLYGON);
	glVertex2f(51.5,75);
	glVertex2f(69,75);
	glVertex2f(60.5,85);
	glVertex2f(51.5,85);
	glEnd();
	
	glBegin(GL_TRIANGLES);
	glVertex2f(50,100);
	glVertex2f(45,95);
	glVertex2f(55,95);
	glEnd();
	
	glPopMatrix();
	
	//glFlush();
	
	}

	void Rain(int value){
	
	if(rainday){
	
	_rain += 0.01f;
	
	glBegin(GL_POINTS);
	for(int i=1;i<=1000;i++)
	{
	int x=rand(),y=rand();
	x%=1000; y%=1000;
	glBegin(GL_LINES);
	glColor3f(1.0, 1.0, 1.0);
	glVertex2d(x,y);
	glVertex2d(x+3,y+12);
	glEnd();
	}
	
	glutPostRedisplay();
	glutTimerFunc(5, Rain, 0);
	
	glFlush();
	
	}

	}

	void RoadRightLamp()
	{
	float a;
	for(a=0; a<=1000; a+=70)
	{
	float b=a+3;
	glColor3ub(66, 66, 49);
	glBegin(GL_QUADS);
	glVertex2i(a,278);
	glVertex2i(b,278);
	glVertex2i(b,296);
	glVertex2i(a,296);
	glEnd();
	glColor3ub(255, 255, 255);
	glBegin(GL_QUADS);
	glVertex2i(a,296);
	glVertex2i(b,296);
	glVertex2i(b,301);
	glVertex2i(a,301);
	glEnd();
	}
	}

	void LeftRoadTree()
	{
	///1st
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(282, 345);
	glVertex2i(285, 345);
	glVertex2i(285, 370);
	glVertex2i(282, 370);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9, 370);
	glVertex2i(285+9, 370);
	glVertex2i(284, 390);
	glEnd();
	glColor3ub(0, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9, 370+6);
	glVertex2i(285+9, 370+6);
	glVertex2i(284, 390+6);
	glEnd();

	///2nd
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(275, 345+50);
	glVertex2i(278, 345+50);
	glVertex2i(278, 370+50);
	glVertex2i(275, 370+50);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7, 370+50);
	glVertex2i(285+9-7, 370+50);
	glVertex2i(284-7, 390+50);
	glEnd();
	glColor3ub(0, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7, 370+50+6);
	glVertex2i(285+9-7, 370+50+6);
	glVertex2i(284-7, 390+50+6);
	glEnd();
	///3rd
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(275-7, 345+50+50);
	glVertex2i(278-7, 345+50+50);
	glVertex2i(278-7, 370+50+50);
	glVertex2i(275-7, 370+50+50);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7-7, 370+50+50);
	glVertex2i(285+9-7-7, 370+50+50);
	glVertex2i(284-7-7, 390+50+50);
	glEnd();
	glColor3ub(0, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7-7, 370+50+50+6);
	glVertex2i(285+9-7-7, 370+50+50+6);
	glVertex2i(284-7-7, 390+50+50+6);
	glEnd();
	///4th
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(275-7-7, 345+50+50+50);

	glVertex2i(278-7-7, 345+50+50+50);
	glVertex2i(278-7-7, 370+50+50+50);
	glVertex2i(275-7-7, 370+50+50+50);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7-7-7, 370+50+50+50);
	glVertex2i(285+9-7-7-7, 370+50+50+50);
	glVertex2i(284-7-7-7, 390+50+50+50);
	glEnd();
	glColor3ub(0, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(282-9-7-7-7, 370+50+50+50+6);
	glVertex2i(285+9-7-7-7, 370+50+50+50+6);
	glVertex2i(284-7-7-7, 390+50+50+50+6);
	glEnd();
	}

	void star5()
	{
	glColor3f(1.0f,1.0f,1.0f) ;
	glBegin(GL_POLYGON);
	
	glVertex2i (970,880);
	glVertex2i (970,885);
	
	glVertex2i(970,885);
	glVertex2i(975,885);
	
	glVertex2i(975,885);
	glVertex2i(975,880);
	
	glVertex2i(975,880);
	glVertex2i(970,880);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(970,880);
	glVertex2i(972,875);
	
	glVertex2i(975,880);
	glVertex2i(972,875);

```
glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(975,880);
glVertex2i(980,883);

glVertex2i(975,885);
glVertex2i(980,883);
glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(975,885);
glVertex2i(972,890);

glVertex2i(970,885);
glVertex2i(972,890);
glEnd();


glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(970,885);
glVertex2i(965,883);


glVertex2i(970,880);
glVertex2i(965,883);
glEnd();
}
```

```
void star6()
{
   glColor3f(1.0f,1.0f,1.0f) ;
glBegin(GL_POLYGON);

glVertex2i (470,680);
glVertex2i (470,685);

glVertex2i(470,685);
glVertex2i(475,685);

glVertex2i(475,685);
glVertex2i(475,680);
```

```
glVertex2i(475,680);
glVertex2i(470,680);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(470,680);
glVertex2i(472,675);

glVertex2i(475,680);
glVertex2i(472,675);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(475,680);
glVertex2i(480,683);

glVertex2i(475,685);
glVertex2i(480,683);
glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(475,685);
glVertex2i(472,690);

glVertex2i(470,685);
glVertex2i(472,690);
glEnd();


glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(470,685);
glVertex2i(465,683);


glVertex2i(470,680);
glVertex2i(465,683);
glEnd();
```

	}

	void star4()
	{
	glColor3f(1.0f,1.0f,1.0f) ;
	glBegin(GL_POLYGON);
	
	glVertex2i (570,880);
	glVertex2i (570,885);
	
	glVertex2i(570,885);
	glVertex2i(575,885);
	
	glVertex2i(575,885);
	glVertex2i(575,880);
	
	glVertex2i(575,880);
	glVertex2i(570,880);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(570,880);
	glVertex2i(572,875);
	
	glVertex2i(575,880);
	glVertex2i(572,875);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(575,880);
	glVertex2i(580,883);
	
	glVertex2i(575,885);
	glVertex2i(580,883);
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(575,885);
	glVertex2i(572,890);

```
glVertex2i(570,885);
glVertex2i(572,890);
glEnd();


glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(570,885);
glVertex2i(565,883);


glVertex2i(570,880);
glVertex2i(565,883);
glEnd();
}
```

```
void star5()
{
   glColor3f(1.0f,1.0f,1.0f) ;
glBegin(GL_POLYGON);

glVertex2i (970,880);
glVertex2i (970,885);

glVertex2i(970,885);
glVertex2i(975,885);

glVertex2i(975,885);
glVertex2i(975,880);
```

```
glVertex2i(975,880);
glVertex2i(970,880);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(970,880);
glVertex2i(972,875);

glVertex2i(975,880);
glVertex2i(972,875);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(975,880);
glVertex2i(980,883);

glVertex2i(975,885);
glVertex2i(980,883);
glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(975,885);
glVertex2i(972,890);

glVertex2i(970,885);
glVertex2i(972,890);
glEnd();


glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(970,885);
glVertex2i(965,883);


glVertex2i(970,880);
glVertex2i(965,883);
glEnd();
```

	}

	void star3()
	{
	glColor3f(1.0f,1.0f,1.0f) ;
	glBegin(GL_POLYGON);
	
	glVertex2i (70,980);
	glVertex2i (70,985);
	
	glVertex2i(70,985);
	glVertex2i(75,985);
	
	glVertex2i(75,985);
	glVertex2i(75,980);
	
	glVertex2i(75,980);
	glVertex2i(70,980);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(70,980);
	glVertex2i(72,975);
	
	glVertex2i(75,980);
	glVertex2i(72,975);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(75,980);
	glVertex2i(80,983);
	
	glVertex2i(75,985);
	glVertex2i(80,983);
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(75,985);
	glVertex2i(72,990);

	glVertex2i(70,985);
	glVertex2i(72,990);
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(70,985);
	glVertex2i(65,983);
	
	glVertex2i(70,980);
	glVertex2i(65,983);
	glEnd();
	}

	void star2()
	{
	glColor3f(1.0f,1.0f,1.0f) ;
	glBegin(GL_POLYGON);
	
	glVertex2i (170,880);
	glVertex2i (170,885);
	
	glVertex2i(170,885);
	glVertex2i(175,885);
	
	glVertex2i(175,885);
	glVertex2i(175,880);
	
	glVertex2i(175,880);
	glVertex2i(170,880);
	
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(170,880);
	glVertex2i(172,875);
	
	glVertex2i(175,880);
	glVertex2i(172,875);
	
	glEnd();

	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(175,880);
	glVertex2i(180,883);
	
	glVertex2i(175,885);
	glVertex2i(180,883);
	glEnd();
	
	glColor3f(11.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	
	glVertex2i(175,885);
	glVertex2i(172,890);
	
	glVertex2i(170,885);
	glVertex2i(172,890);
	glEnd();
	
	glColor3f(1.0f,1.0f,1.0f);
	glBegin(GL_POLYGON);
	glVertex2i(170,885);
	glVertex2i(165,883);
	
	glVertex2i(170,880);
	glVertex2i(165,883);
	glEnd();
	}

	void star1()
	{
	glColor3f(1.0f,1.0f,1.0f) ;
	glBegin(GL_POLYGON);
	
	glVertex2i (70,880);
	glVertex2i (70,885);
	
	glVertex2i(70,885);
	glVertex2i(75,885);
	
	glVertex2i(75,885);
	glVertex2i(75,880);

```
glVertex2i(75,880);
glVertex2i(70,880);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(70,880);
glVertex2i(72,875);

glVertex2i(75,880);
glVertex2i(72,875);

glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(75,880);
glVertex2i(80,883);

glVertex2i(75,885);
glVertex2i(80,883);
glEnd();

glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);

glVertex2i(75,885);
glVertex2i(72,890);

glVertex2i(70,885);
glVertex2i(72,890);
glEnd();


glColor3f(1.0f,1.0f,1.0f);
glBegin(GL_POLYGON);
glVertex2i(70,885);
glVertex2i(65,883);


glVertex2i(70,880);
glVertex2i(65,883);
glEnd();
```

	}

	void tree1()
	{
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(20, 380);
	glVertex2i(25, 380);
	glVertex2i(25, 420);
	glVertex2i(20, 420);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(20, 420);
	glVertex2i(20-15, 410);
	glVertex2i(20, 450);
	glEnd();
	
	glBegin(GL_QUADS);
	glVertex2i(20, 420);
	glVertex2i(25, 420);
	glVertex2i(25, 450);
	glVertex2i(20, 450);
	glEnd();
	
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(25, 420);
	glVertex2i(20+15+5, 410);
	glVertex2i(25, 450);
	glEnd();
	}

	void HillTree3()
	{
	float a;
	for(a=10;a<1000;a+=160)
	{
	float b=a+2;
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(a, 545);
	glVertex2i(b, 545);
	glVertex2i(b, 580);
	glVertex2i(a, 580);
	glEnd();
	glColor3ub(102, 204, 0);

	glBegin(GL_POLYGON);
	glVertex2i(a-4, 560-2);
	glVertex2i(b+4, 560-2);
	glVertex2i(a+1,590+4);
	glEnd();
	}
	}

	void HillTree2()
	{
	float a;
	for(a=10;a<1000;a+=120)
	{
	float b=a+2;
	glColor3ub(0, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(a, 545);
	glVertex2i(b, 545);
	glVertex2i(b, 580);
	glVertex2i(a, 580);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(a-8, 580);
	glVertex2i(b+8, 580);
	glVertex2i(a+1,590);
	glEnd();
	glColor3ub(102, 204, 0);
	glBegin(GL_POLYGON);
	glVertex2i(a-8, 580+5);
	glVertex2i(b+8, 580+5);
	glVertex2i(a+1,590+10);
	glEnd();
	
	}
	}

	void house10()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(220, 470-70);
	glVertex2i(260, 470-70);
	glVertex2i(240, 490-70);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);

	glVertex2i(220+8, 470-70);
	glVertex2i(260-8, 470-70);
	glVertex2i(260-8, 440-70);
	glVertex2i(220+8, 440-70);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(220+4, 440-70);
	glVertex2i(260-4, 440-70);
	glVertex2i(260-4, 435-70);
	glVertex2i(220+4, 435-70);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(220+8+8, 460-70);
	glVertex2i(260-8-8, 460-70);
	glVertex2i(260-8-8, 440-70);
	glVertex2i(220+8+8, 440-70);
	glEnd();
	}

	void house9()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(220-40, 470-20);
	glVertex2i(260-40, 470-20);
	glVertex2i(240-40, 490-20);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(220+8-40, 470-20);
	glVertex2i(260-8-40, 470-20);
	glVertex2i(260-8-40, 440-20);
	glVertex2i(220+8-40, 440-20);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(220+4-40, 440-20);
	glVertex2i(260-4-40, 440-20);
	glVertex2i(260-4-40, 435-20);
	glVertex2i(220+4-40, 435-20);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);

	glBegin(GL_QUADS);
	glVertex2i(220+8+8-40, 460-20);
	glVertex2i(260-8-8-40, 460-20);
	glVertex2i(260-8-8-40, 440-20);
	glVertex2i(220+8+8-40, 440-20);
	glEnd();
	
	}

	void house8()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i((20-20)*2+60, (545-40)+10);
	glVertex2i((40-20)*2+60, (545-40)+10);
	glVertex2i((30-20)*2+60, (560-40)+10);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i((25-20)*2+60, (545-40)+10);
	glVertex2i((35-20)*2+60, (545-40)+10);
	glVertex2i((35-20)*2+60, (530-40)+10);
	glVertex2i((25-20)*2+60, (530-40)+10);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i((22-20)*2+60, (530-40)+10);
	glVertex2i((40-2-20)*2+60, (530-40)+10);
	glVertex2i((40-2-20)*2+60, (525-40)+10);
	glVertex2i((22-20)*2+60, (525-40)+10);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i((25+3-20)*2+60, (540-40)+10);
	glVertex2i((35-3-20)*2+60, (540-40)+10);
	glVertex2i((35-3-20)*2+60, (530-40)+10);
	glVertex2i((25+3-20)*2+60, (530-40)+10);
	glEnd();
	}

	void house7()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(0, 505);

	glVertex2i(20, 505);
	glVertex2i(10, 520);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(5, 505);
	glVertex2i(15, 505);
	glVertex2i(15, 490);
	glVertex2i(5, 490);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(2, 490);
	glVertex2i(18, 490);
	glVertex2i(18, 485);
	glVertex2i(2, 485);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(8, 500);
	glVertex2i(12, 500);
	glVertex2i(12, 490);
	glVertex2i(8, 490);
	glEnd();
	}

	void house6()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(45, 545);
	glVertex2i(65, 545);
	glVertex2i(55, 560);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(50, 545);
	glVertex2i(60, 545);
	glVertex2i(60, 530);
	glVertex2i(50, 530);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(47, 530);
	glVertex2i(63, 530);

	glVertex2i(63, 525);
	glVertex2i(47, 525);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(53, 540);
	glVertex2i(57, 540);
	glVertex2i(57, 530);
	glVertex2i(53, 530);
	glEnd();
	}

	void house5()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(20, 545);
	glVertex2i(40, 545);
	glVertex2i(30, 560);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(25, 545);
	glVertex2i(35, 545);
	glVertex2i(35, 530);
	glVertex2i(25, 530);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(22, 530);
	glVertex2i(38, 530);
	glVertex2i(38, 525);
	glVertex2i(22, 525);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(28, 540);
	glVertex2i(32, 540);
	glVertex2i(32, 530);
	glVertex2i(28, 530);
	glEnd();
	}

	void house3()

	{
	glColor3ub(102, 0, 0);
	glBegin(GL_POLYGON);
	glVertex2i(190, 520);
	glVertex2i(230, 520);
	glVertex2i(210, 540);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(198, 520);
	glVertex2i(222, 520);
	glVertex2i(222, 490);
	glVertex2i(198, 490);
	glEnd();
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(194, 490);
	glVertex2i(226, 490);
	glVertex2i(226, 485);
	glVertex2i(194, 485);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(206, 490);
	glVertex2i(214 ,490);
	glVertex2i(214, 510);
	glVertex2i(206, 510);
	glE

	void house1()
	{
	glColor3ub(102, 0, 0);
	glBegin(GL_QUADS);
	glVertex2i(100, 520);
	glVertex2i(180, 520);
	glVertex2i(170, 530);
	glVertex2i(110, 530);
	glEnd();
	glColor3ub(128, 128, 128);
	glBegin(GL_QUADS);
	glVertex2i(115, 520);
	glVertex2i(165, 520);
	glVertex2i(165, 480);
	glVertex2i(115, 480);
	glEnd();

	glColor3ub(66, 66, 49);
	glBegin(GL_QUADS);
	glVertex2i(105, 480);
	glVertex2i(175, 480);
	glVertex2i(175, 475);
	glVertex2i(105, 475);
	glEnd();
	///Door
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(135, 480);
	glVertex2i(145, 480);
	glVertex2i(145, 510);
	glVertex2i(135, 510);
	glEnd();
	///window
	glColor3ub(1, 51, 102);
	glBegin(GL_QUADS);
	glVertex2i(120, 495);
	glVertex2i(128, 495);
	glVertex2i(128, 513);
	glVertex2i(120, 513);
	glEnd();
	glBegin(GL_QUADS);
	glVertex2i(120, 495);
	glVertex2i(128, 495);
	glVertex2i(128, 513);
	glVertex2i(120, 513);
	glEnd();
	glBegin(GL_QUADS);
	glVertex2i(150, 495);
	glVertex2i(158, 495);
	glVertex2i(158, 513);
	glVertex2i(150, 513);
	glEnd();
	///HomeSiteTree
	glColor3ub(204, 0, 0);
	glBegin(GL_LINES);
	
	glEnd();
	}

	void Windmill_One(){
	glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	glTranslatef(0,-10,0);

	Windmill();
	glPopMatrix();
	
	}

	void Windmill_Two(){
	glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	glTranslatef(208,-20,0);
	Windmill();
	glPopMatrix();
	
	}

	void Windmill_Three(){
	glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	glTranslatef(108,-90,0);
	Windmill();
	glPopMatrix();
	}

	void Windmill_four(){
	glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	glTranslatef(508,-70,0);
	Windmill();
	glPopMatrix();
	}

	void Windmill_five()
	{ glColor3f(0.1, 0.1, 0.1);
	glPushMatrix();
	glTranslatef(308,-70,0);
	Windmill();
	glPopMatrix();
	
	}

	void Windmill_Blade(){
	
	///Blade_One
	glPushMatrix();
	glRotatef(spin,0,0,90);
	glBegin(GL_POLYGON);
	glVertex2i(-5, 0);
	glVertex2i(-85, -36);
	glVertex2i(-83, -37);

```
    glVertex2i(-3, -8);
    glEnd();
    glPopMatrix();

    ///Blade_Two
    glPushMatrix();
    glRotatef(spin,0,0,90);
    glBegin(GL_POLYGON);
    glVertex2i(0, 5);
    glVertex2i(45, 70);
    glVertex2i(50, 73);
    glVertex2i(5, 0);
    glEnd();
    glPopMatrix();

    ///Blade_Three
    glPushMatrix();
    glRotatef(spin,0,0,90);
    glBegin(GL_POLYGON);
    glVertex2i(68, -78);
    glVertex2i(0,0);
    glVertex2i(5, 5);
    glVertex2i(70, -77);
    glEnd();
    glPopMatrix();

}
```

```
void Windmill_Stand_Model(){

    glColor3f(0.1, 0.1, 0.1);
    glBegin(GL_POLYGON);
    glVertex2i(375, 500);
    glVertex2i(380, 640);
    glVertex2i(384, 640);
    glVertex2i(390, 500);
    glEnd();
}
```

```
void cloud_model_one(){

    glColor3f(1.25, 0.924, 0.930);

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
```

```
circle(15);
glPopMatrix();

///Top

glPushMatrix();
glTranslatef(340, 225, 0);
circle(16);
glPopMatrix();

///Right

glPushMatrix();
glTranslatef(360,210,0);
circle(16);
glPopMatrix();


///middle_Fill
glPushMatrix();
glTranslatef(355,210,0);
circle(16);
glPopMatrix();

glPushMatrix();
glTranslatef(350,210,0);
circle(16);
glPopMatrix();

glPushMatrix();
glTranslatef(345,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
glTranslatef(340,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
glTranslatef(335,204,0);
circle(10);
glPopMatrix();

glPushMatrix();
```

```
    glTranslatef(330,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(325,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(320,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(315,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(310,204,0);
    circle(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(305,204,0);
    circle(10);
    glPopMatrix();

    ///****Fill End****

}
```

```
void cloud_model_Two(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(305,205,0);
    circle(10);
    glPopMatrix();

    ///Top

    glPushMatrix();
    glTranslatef(320,210,0);
```

```
    circle(15);
    glPopMatrix();

    ///Right_Part
    glPushMatrix();
    glTranslatef(334,207,0);
    circle(10);
    glPopMatrix();

    ///Bottom_Part
    glPushMatrix();
    glTranslatef(320,207,0);
    circle(10);
    glPopMatrix();



}
```

```
void cloud_model_Three(){
    glColor3f(1.25, 0.924, 0.930);

    ///Left_Part
    glPushMatrix();
    glTranslatef(300,200,0);
    circle(15);
    glPopMatrix();

    ///Top_Left

    glPushMatrix();
    glTranslatef(320,210,0);
    circle(15);
    glPopMatrix();

    ///Top
    glPushMatrix();
    glTranslatef(340,220,0);
    circle(16);
    glPopMatrix();

    ///Top_Right
    glPushMatrix();
    glTranslatef(360,210,0);
    circle(15);
    glPopMatrix();
```

	///Right_Part
	glPushMatrix();
	glTranslatef(380,200,0);
	circle(15);
	glPopMatrix();
	
	///Bottom_Right
	glPushMatrix();
	glTranslatef(360,190,0);
	circle(20);
	glPopMatrix();
	
	///Bottom_Left
	glPushMatrix();
	glTranslatef(320,190,0);
	circle(20);
	glPopMatrix();
	
	///Bottom
	glPushMatrix();
	glTranslatef(340,190,0);
	circle(20);
	glPopMatrix();
	
	///****Fill End****
	
	}

	void cloud_one(){
	glPushMatrix();
	glTranslatef(cx,700,0);
	cloud_model_one();
	glPopMatrix();
	
	}

	void cloud_two(){
	glPushMatrix();
	glTranslatef(bx+-200,750,0);
	cloud_model_one();
	glPopMatrix();
	
	}

	void cloud_three(){
	glPushMatrix();
	glTranslatef(ax-80,670,0);
	cloud_model_Two();
	glPopMatrix();
	
	}

	void cloud_four(){
	glPushMatrix();
	glTranslatef(dx+300,675,0);
	cloud_model_Two();
	glPopMatrix();
	
	}

	void cloud_five(){
	
	glPushMatrix();
	glTranslatef(ax+-300,770,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void cloud_six(){
	
	glPushMatrix();
	glTranslatef(cx+-500,710,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void cloud_seven(){
	
	glPushMatrix();
	glTranslatef(cx+-200,780,0);
	cloud_model_Three();
	glPopMatrix();
	}

	void car1()
	{
	glPushMatrix();
	glTranslatef(car1_speed, 0.0, 0.0);
	
	glPushMatrix();
	glTranslatef(88,320, 0);
	glBegin(GL_POLYGON);
	glColor3ub(255,0,0);

```
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
  glEnd();
  glPopMatrix();
  glPushMatrix();
   glTranslatef(70,320, 0);
   glBegin(GL_POLYGON);
   glColor3ub(255,0,0);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
  glEnd();
  glPopMatrix();

   glColor3f(255,0,255);
   glBegin(GL_QUADS);
   glVertex2i(30-20+50, 317);
   glVertex2i(30+40-20+50, 317);
   glVertex2i(30+40-20+50, 325);
   glVertex2i(30-20+50, 325);
   glVertex2i(30+5-20+50, 325);
   glVertex2i(30+35-5-20+50, 325);
   glVertex2i(30+35-10-20+50, 317+20);
   glVertex2i(30+5+3-20+50, 317+20);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_LINES);
   glVertex2i(30-20+50, 317+4);
   glVertex2i(30+40-20+50, 317+4);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_QUADS);
```

```
    glVertex2i(30+5-20+3+50, 325);
    glVertex2i(30+35-5-20-14+50, 325);
    glVertex2i(30+35-5-20-14+50, 317+20-4);
    glVertex2i(30+5+3-20+2+50, 317+20-4);
    glVertex2i(30+35-5-20-14+2+50, 325);
    glVertex2i(30+35-5-20-14+11+50, 325);
    glVertex2i(30+35-5-20-14+10-2+50, 317+20-4);
    glVertex2i(30+35-5-20-14+2+50, 317+20-4);
    glEnd();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(car2_speed, 0.0, 0.0);


    glPushMatrix();
    glTranslatef(20,320, 0);
    glBegin(GL_POLYGON);
    glColor3ub(255,0,0);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
   glEnd();
   glPopMatrix();
   glPushMatrix();
    glTranslatef(38,320, 0);
    glBegin(GL_POLYGON);
    glColor3ub(255,0,0);
for(int i=0;i<200;i++)
{
float pi=3.1416;
float A=(i*2*pi)/200;
float r=5;
float x = r * cos(A);
float y = r * sin(A);
glVertex2f(x,y );
}
   glEnd();
   glPopMatrix();
```

```
   glColor3f(255,0,0);
   glBegin(GL_QUADS);
   glVertex2i(30-20, 317);
   glVertex2i(30+40-20, 317);
   glVertex2i(30+40-20, 325);
   glVertex2i(30-20, 325);
   glVertex2i(30+5-20, 325);
   glVertex2i(30+35-5-20, 325);
   glVertex2i(30+35-10-20, 317+20);
   glVertex2i(30+5+3-20, 317+20);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_LINES);
   glVertex2i(30-20, 317+4);
   glVertex2i(30+40-20, 317+4);
   glEnd();
   glColor3f(0,255,0);
   glBegin(GL_QUADS);
   glVertex2i(30+5-20+3, 325);
   glVertex2i(30+35-5-20-14, 325);
   glVertex2i(30+35-5-20-14, 317+20-4);
   glVertex2i(30+5+3-20+2, 317+20-4);
   glVertex2i(30+35-5-20-14+2, 325);
   glVertex2i(30+35-5-20-14+11, 325);
   glVertex2i(30+35-5-20-14+10-2, 317+20-4);
   glVertex2i(30+35-5-20-14+2, 317+20-4);
   glEnd();
   glPopMatrix();

}
```

Conclusion:

We have tried to highlight a modern city in our project. I have tried to present a picture of how electricity is created in a modern city and the way of life there. Trees play a vital role in the ecological balance. Without tress, cause natural hazards, drought, soil erosion, global warming, and many other bad things. We have tried to highlight tree plantations through the tree function. We have used many objects in our project, they are sky, hill, sun, river, car, house, land, cloud, boat, moon, hill tree, windmill, star, lamp, tree. We have tried to present our design using these objects. This pandemic situation has caused some problems in our project. Yet we are happy that we have been able to complete our project well.