Computer Graphics

(UCS05)

Project on

Bus Stop Simulation

Submitted By

Reena Arora	102027169
Akashdeep Singh Gill	102017155

B.E. Third Year – CSE

Submitted To:

Ms. Jasmine Kaur

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147004

Table of Contents

INTRODUCTION

Bus stop simulation is an open GL computer graphics project using open GL function it is a user interactive program where in the user can view the required display by making use of the input devices Keyboard and mouse.

The main aim of the transport Simulation Computer Graphics is to illustrate the concepts and usage of pre-built functions OpenGL. Simulation of a bus is being done using computer graphics. Bus Stop Simulation is a project where we can also change the color, name, of the bus and other graphics. We can also interchange the boy or girl picking and dropping by the bus.

This project depicts two scenes, one scene where the bus moves from left of the screen to the right near the house of the student whom it is picking up and in the second scene, we can see the college where the bus we will drop the student and once again moves from left to right of the screen.

It has additional feature of changing the gender of the student being picked by changing their clothes, picking and dropping is automatically done when the bus goes near the bus stop sign.

Creating Figures like Bus and the surrounding environment using inbuilt functions provided by the glut library. The environment is built in -ve Y-axis and translated to +ve Y-axis to make it look like the bus is moving In addition to the visual elements, there are also interactive elements to the animation.

Overall, the animation project is a highly immersive, visually stunning, and interactive experience that combines elements of Computer Graphics, interactivity, and engagement to create a highly engaging and entertaining animation for users to explore and enjoy.

Computer Graphics concepts used

1. POLYGON DRAWING

Polygons are used to make various shapes and structures. Polygon drawing is a critical concept in Bus Simulation as it allows designers and engineers to create complex shapes and structures that are essential for the overall performance and appearance.

2. COLOR FILLING

Color filling process involves filling the interior of a closed polygon with a color, creating a solid surface that is visible in the rendered image.

3. TRANSLATION

Translation is a fundamental concept in computer graphics that is used in the project to move and position the bus components within the virtual 3D environment Translation involves moving the vertices of the polygons that make up the bus to position them at the desired location within the 3D space. Movement of bus from one end to another is possible because of translation

4. SCALING

Used to reduce or enlarge an image to fit the desired size or proportions. Scaling functions can be used to ensure that an image is the correct size for a particular display or for printing. They can also be used to transform an image from one aspect ratio to another.

USER DEFINED FUNCTIONS

wheel1():

This function is used to make front two wheel tyres and front two wheels of bus. They are drawn using inbuilt functions glPushMatrix(), glTranslatef(),glutSolidTorus() and glPopMatrix().

wheel2():

This function is used to make back two tyres of bus and back two wheels. They are drawn using inbuilt functions glPushMatrix(), glTranslatef(),glutSolidTorus() and glPopMatrix().

polygon() and colorcube():

Here the overall structure of bus is drawn using inbuilt function glVertex3fv() where v is a pointer to an array of elements and wheel1() and wheel2() functions are called to complete the structure Similar to it two other user defined functions are defined polygond() and colorcubed() for the second scene.

bus_stop():

Here the scene of bus stop is drawn Ground and chairs are made using GL_POLYGON and GL_LINES by specifying the x,y and z coordinates.

road2():

The road of the bus stop is made using this function Stripes on the road are made after every 60m using two for loops one for x-direction and other for y-direction. Colour of the stripes can be changed by specifying the different values in inbuilt function glColor3f().

tree1():

Here the trees are made First the trunk of the tree is created using following inbuilt functions glPushMatrix(), glTranslated(), glScaled(), glutSolidCube(), glPopMatrix(). Then leaves are made using glutSolidCone() inbuilt function.

woman():

Here, the woman student is made. Face of the woman is created using inbuilt functions glPushMatrix(), glTranslatef(), glutSolidTorus(), glPopMatrix() whereas the other parts of the woman such as nose, eyebrow, eyes, ear, hair, skirt, shoes are created using inbuilt function glVertex2i() by specifying x and y coordinates.

man():

The male student is modelled similarly to the female student but with some additional features such as collar, belt, button, pants, beard using inbuilt function glVertex2i() by specifying x and y coordinates.

wheel1d():

This function is similar to function wheel1() Here the front wheels of bus are created for second part of the scene.

wheel2d():

This function is similar to function wheel2() Here the back wheels of bus are created for second part of the scene.

womand():

This function is similar to woman() Female student is drawn for the second part of the scene when bus drops off her at college.

road2d():

This function is similar to road2() Here roads are created for the second part of the scene.

gated():

This function draws college gates by using inbuilt function glVertex2i() specifying x and y coordinates.

stopd():

Utilizing the inbuilt functions glPushMatrix(), glTranslate(), glScale(), glutSolidCube(), glPopMatrix(), this function draws a pillar indicating where the bus has to stop.

walld():

Here the wall of college is created by joining small rectangular shapes, known as bricks. For loops are used to create those rectangular shapes.

buildingd():

The function is used to draw the building, windows and doors of the college using the inbuilt functions glPushMatrix(), glTranslatef(), glScalef(), glutSolidCube(), and glPopMatrix().

intro():

Here the rectangular introduction board for the college is drawn depicting the name of the college using inbuilt function glVertex2i() specifying the x and y coordinates.

mouse(int btn, int state, int x, int y):

In this function, the key is set on the basis of which bus will start when the left-clicked button is released and when the state is down, the bus will start moving.

bus_move():

In this function x-coordinated in calculated and updated when the bus moves and condition is checked if the value of x reaches beyond the certain value then the bus will stop and flag is set Other changes are also specified in this function that takes place when the bus moves and bus stop and some function calls are made.

Similarly another user defined function bus_moved() is defined for the second scene i.e. when bus drop student at college.

text(), text1(), text2(), text3(), text 4d(), text 5d(), texti():

These functions are used to display the text whereever it is required in the scene All these functions are implemented using the same set of codes by changing the values of string and their colors.

display(void):

This is the default display function. Here, the main user-defined busstop() function is called, and a few other function calls are also made to display the scene in which the bus arrives at the bus stop and picks up the students to take them to college. With reference to this the code is implemented using if else conditions For the display of any scene first condition is checked to determine which scene is displayed.

CODE

```
#include <stdlib.h>
#include <GL/glut.h>
#include<string.h>
#include<stdio.h>

int x = -150, o = 0, xd = -150;
int x1 = 0, z = 0;
float a = 0, a1 = 0, moving, angle = 0;
float z5 = 0, u = 0, flag12 = 0, begin;
float k = 0, y2 = 0;
int flag = 0, flag55 = 0, var = 0, flag1 = 0, flag551 = 0, vari = 0, vard = 0, varid = 0, then = 0;
float p = 0.0, q = 0.0;
#define maxx 10
#define maxy 12
#define dx 27.7
#define dy 12
GLfloat xangle = 0.0, yangle = 0.0, zangle = 0.0;   /* axis angles */
GLfloat vertices[][3] = { {160,390,-70},{425,390,-70},
                                        {425,510,-70}, {160,520,-70},

                                        {135,370,70}, {400,370,70},
                                        {400,490,70}, {135,500,70},

                                        {135,447,70},{400,447,70},
                                        {425,467,-70},{410,520,-70},

                                        {385,500,70}, {160,467,-70},
                                        {320,467,-70},{320,520,-70},

{380,520,-70},{380,390,-70},{320,390,-70} };

GLfloat colors[][3] = { {1.0,1.0,0.0},{0.0,0.6,0.7},{.3,.4,.5} };

GLfloat verticesd[][3] = { {160,390 - 175,-70},{425,390 - 175,-70},
                                        {425,510 - 175,-70}, {160,520 - 175,-70},

                                        {135,370 - 175,70}, {400,370 - 175,70},
                                        {400,490 - 175,70}, {135,500 - 175,70},

                                        {135,447 - 175,70},{400,447 - 175,70},
                                        {425,467 - 175,-70},{410,520 - 175,-70},

                                        {385,500 - 175,70}, {160,467 - 175,-70},
                                        {320,467 - 175,-70},{320,520 - 175,-70},

{380,520 - 175,-70},{380,390 - 175,-70},{320,390 - 175,-70} };

GLfloat colorsd[][3] = { {1.0,1.0,0.0},{0.0,0.6,0.7},{.3,.4,.5} };



void wheel1()
{
        glColor3f(0, 0, 0);

        glPushMatrix();
        glTranslatef(345, 377, -70);
        glutSolidTorus(5, 15, 100, 90);
        glPopMatrix();

        glPushMatrix();
        glTranslatef(190, 377, -70);
        glutSolidTorus(5, 15, 100, 90);//front two wheels tyre
        glPopMatrix();

        glColor3ub(100, 100, 100);

        glPushMatrix();
        glTranslatef(345, 377, -70);
        glutSolidTorus(5, 5, 10, 69);
        glPopMatrix();
```

```
	glPushMatrix();
	glTranslatef(190, 377, -70);
	glutSolidTorus(5, 5, 10, 69);
	glPopMatrix();// front two wheels
}

void wheel2()
{
	glColor3f(0, 0, 0);

	glPushMatrix();
	glTranslatef(180, 370, 70);
	glutSolidTorus(5, 15, 100, 90);
	glPopMatrix();

	glPushMatrix();
	glTranslatef(335, 370, 70);
	glutSolidTorus(5, 15, 100, 90);
	glPopMatrix();		//back two wheels tyre

	glColor3ub(100, 100, 100);

	glPushMatrix();
	glTranslatef(335, 370, 70);
	glutSolidTorus(5, 5, 10, 69);
	glPopMatrix();

	glPushMatrix();
	glTranslatef(180, 370, 70);
	glutSolidTorus(5, 5, 10, 69);
	glPopMatrix();	//back two wheels
}
void polygon(int a, int b, int c, int d, int E, int f)
{
	glBegin(GL_POLYGON);
```

```
        glColor3fv(colors[E]);
        glVertex3fv(vertices[a]);
        glVertex3fv(vertices[b]);
        glVertex3fv(vertices[c]);
        glVertex3fv(vertices[d]);
        if (f != 0)
                glVertex3fv(vertices[f]);
        glEnd();
}
void colorcube()
{
        int i;
        wheel1();
        polygon(0, 1, 5, 4, 0, 0);
        polygon(13, 14, 18, 0, 0, 0);
        polygon(15, 16, 17, 18, 2, 0);
        polygon(16, 11, 2, 1, 0, 17);
        polygon(0, 4, 8, 13, 0, 0);
        polygon(1, 10, 9, 5, 0, 0);
        polygon(9, 10, 2, 6, 1, 0);
        polygon(4, 5, 9, 8, 0, 0);
        polygon(8, 9, 6, 12, 1, 7);
        glColor3ub(100, 40, 50);
        for (i = 0; i <= 180; i += 45)
        {
                glBegin(GL_LINES);
                glVertex3f(180 + i, 447, 70);
                glVertex3f(180 + i, 500, 70);
                glEnd();
        }
        polygon(13, 8, 7, 3, 1, 0);
        polygon(3, 15, 14, 13, 1, 0);
        polygon(6, 2, 11, 12, 0, 0);
        polygon(11, 3, 7, 12, 0, 0);
        wheel2();
}
```

```
void bus_stop()
{
        glColor3ub(100, 100, 100);
        glBegin(GL_POLYGON);
        glVertex3i(340 - 200, 470, -110);
        glVertex3i(680 - 200, 470, -110);
        glVertex3i(710 - 200, 500, -240);
        glVertex3i(370 - 200, 500, -240);
        glEnd();
        glColor3ub(100, 100, 100);
        glBegin(GL_POLYGON);
        glVertex3i(340 - 200, 470, -110);
        glVertex3i(680 - 200, 470, -110);
        glVertex3i(680 - 200, 450, -110);
        glVertex3i(340 - 200, 450, -110);
        glEnd();
        glBegin(GL_POLYGON);
        glVertex3i(680 - 200, 470, -110);
        glVertex3i(710 - 200, 500, -240);
        glVertex3i(710 - 200, 480, -240);
        glVertex3i(680 - 200, 450, -110);
        glEnd();
        glBegin(GL_POLYGON);
        glVertex3i(710 - 200, 500, -240);
        glVertex3i(710 - 200, 480, -240);
        glVertex3i(370 - 200, 480, -240);
        glVertex3i(370 - 200, 500, -240);
        glEnd();
        glBegin(GL_POLYGON);
        glVertex3i(370 - 200, 480, -240);
        glVertex3i(370 - 200, 500, -240);
        glVertex3i(340 - 200, 470, -110);
        glVertex3i(340 - 200, 450, -110);
        glEnd();
```

```
glColor3f(1.0, 1.0, 1.0);
glBegin(GL_LINE_STRIP);
glVertex3i(340 - 200, 470, -110);
glVertex3i(680 - 200, 470, -110);
glVertex3i(710 - 200, 500, -240);
glEnd();

glColor3f(1.0, 1.0, 1.0);
glBegin(GL_LINE_STRIP);
glVertex3i(680 - 200, 470, -110);
glVertex3i(680 - 200, 450, -110);
glEnd();
glColor3ub(10, 50, 80);
glBegin(GL_POLYGON);
glVertex3i(370 - 200, 610, -140);
glVertex3i(400 - 200, 625, -200);
glVertex3i(400 - 200, 490, -200);
glVertex3i(370 - 200, 480, -140);
glEnd();
glColor3ub(10, 170, 80);
glBegin(GL_POLYGON);
glVertex3i(395 - 200, 580, -200);
glVertex3i(690 - 200, 580, -200);
glVertex3i(690 - 200, 520, -200);
glVertex3i(395 - 200, 520, -200);
glEnd();
glColor3f(0, 0, 0);
glBegin(GL_LINES);
glVertex3i(395 - 200, 580, -200);
glVertex3i(690 - 200, 580, -200);
glVertex3i(690 - 200, 520, -200);
glVertex3i(395 - 200, 520, -200);
glEnd();
glColor3ub(10, 50, 80);
glBegin(GL_POLYGON);
glVertex3i(690 - 200, 625, -200);
```

```
glVertex3i(670 - 200, 610, -140);
glVertex3i(670 - 200, 475, -140);
glVertex3i(690 - 200, 490, -200);
glEnd();
glColor3ub(0, 0, 0);
glBegin(GL_POLYGON);
glVertex3i(425 - 200, 530, -180);
glVertex3i(520 - 200, 530, -180);
glVertex3i(500 - 200, 515, -150);
glVertex3i(405 - 200, 515, -150);
glEnd();

glColor3ub(0, 0, 0);
glBegin(GL_LINES);
glVertex3i(425 - 200, 515, -160);
glVertex3i(425 - 200, 480, -160);
glVertex3i(437 - 200, 515, -170);
glVertex3i(437 - 200, 487, -170);
glEnd();

glColor3ub(0, 0, 0);
glBegin(GL_LINES);
glVertex3i(485 - 200, 515, -163);
glVertex3i(485 - 200, 480, -163);
glVertex3i(495 - 200, 515, -170);
glVertex3i(495 - 200, 487, -170);
glEnd();
glColor3ub(0, 10, 20);
glBegin(GL_POLYGON);
glVertex3i(560 - 200, 530, -180);
glVertex3i(655 - 200, 530, -180);
glVertex3i(635 - 200, 515, -150);
glVertex3i(540 - 200, 515, -150);
glEnd();

glColor3ub(0, 0, 0);
```

```
 glBegin(GL_LINES);
 glVertex3i(560 - 200, 515, -160);
 glVertex3i(560 - 200, 480, -160);
 glVertex3i(572 - 200, 515, -170);
 glVertex3i(572 - 200, 487, -170);
 glEnd();

 glColor3ub(0, 0, 0);
 glBegin(GL_LINES);
 glVertex3i(620 - 200, 515, -163);
 glVertex3i(620 - 200, 480, -163);
 glVertex3i(630 - 200, 515, -170);
 glVertex3i(630 - 200, 487, -170);
 glEnd();
 /********      upper      ********/
 glColor3ub(10, 50, 80);
 glBegin(GL_POLYGON);
 glVertex3i(350 - 200, 620, -120);
 glVertex3i(700 - 200, 620, -120);
 glVertex3i(700 - 200, 600, -120);
 glVertex3i(350 - 200, 600, -120);
 glEnd();
glBegin(GL_POLYGON);
 glVertex3i(350 - 200, 620, -120);
 glVertex3i(700 - 200, 620, -120);
 glVertex3i(720 - 200, 640, -240);
 glVertex3i(380 - 200, 640, -240);
 glEnd();

 glBegin(GL_POLYGON);
 glVertex3i(700 - 200, 620, -120);
 glVertex3i(720 - 200, 640, -240);
 glVertex3i(720 - 200, 620, -240);
 glVertex3i(700 - 200, 600, -120);
 glEnd();
 glBegin(GL_POLYGON);
```

```
        glVertex3i(350 - 200, 600, -120);
        glVertex3i(350 - 200, 620, -120);
        glVertex3i(380 - 200, 640, -240);
        glVertex3i(380 - 200, 620, -240);
        glEnd();
       glColor3f(1.0, 1.0, 1.0);
        glBegin(GL_LINES);
        glVertex3i(350 - 200, 620, -120);
        glVertex3i(700 - 200, 620, -120);
        glVertex3i(700 - 200, 620, -120);
        glVertex3i(720 - 200, 640, -240);
        glVertex3i(700 - 200, 620, -120);
        glVertex3i(700 - 200, 600, -120);
        glEnd();
}
void road2()
{
        /*************** left part of road  *********/
        int x, y;
        glColor3ub(7, 255, 13);
        glBegin(GL_POLYGON);
        glVertex2i(0, 650);
        glVertex2i(1000, 650);
        glVertex2i(1000, 0);
        glVertex2i(0, 0);
        glEnd();
        glColor3ub(30, 40, 50);
        glBegin(GL_POLYGON);
        glVertex2i(0, 420);
        glVertex2i(1000, 420);
        glVertex2i(1000, 300);
        glVertex2i(0, 300);
        glEnd();
        glBegin(GL_POLYGON);
        glVertex2i(750, 650);
        glVertex2i(900, 650);
```

```
    glVertex2i(1000, 0);
    glVertex2i(650, 0);
    glEnd();
    glColor3f(1.0, 0.9, 0.0);
    for (x = 0; x < 1000; x = x + 60)
    {
        glBegin(GL_POLYGON);
        glVertex2f(x, 352.5 + 19);
        glVertex2f(x, 357.5 + 19);
        glVertex2f(x + 30, 357.5 + 19);
        glVertex2f(x + 30, 352.5 + 19);
        glEnd();
    }

    for (y = 650; y > 0; y = y - 60)
    {
        glBegin(GL_POLYGON);
        glVertex2f(822, y);
        glVertex2f(826, y);
        glVertex2f(826, y - 30);
        glVertex2f(822, y - 30);
        glEnd();
    }
}
void text()
{
    char string[] = "BUS STOP";
    char string1[] = "";
    void* font = GLUT_BITMAP_TIMES_ROMAN_24;
    int i, j;
    void* font1 = GLUT_BITMAP_TIMES_ROMAN_10;
    glColor3f(1.0, 1.0, 1.0);
   glRasterPos3f(280, 602, -120);
    for (i = 0; i < strlen(string); i++)
        glutBitmapCharacter(font, string[i]);
    glRasterPos3f(420, 602, -120);
```

```
        for (j = 0; j < strlen(string1); j++)
                glutBitmapCharacter(font1, string1[j]);
}

void text1()
{
        char string2[] = "TIET";
        void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
        int k;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(230 + p, 400, 70);
        for (k = 0; k < strlen(string2); k++)
                glutBitmapCharacter(font2, string2[k]);
}
void text2()
{
        glBegin(GL_POLYGON);
        glColor3f(1.0, 1.0, 1.0);
        glVertex2i(830 - 500, 120 + 150);
        glVertex2i(1000 - 500 + 40, 120 + 150);
        glVertex2i(1000 - 500 + 40, 35 + 150);
        glVertex2i(830 - 500, 35 + 150);
        glEnd();
        char string2[] = "Pick up the woman ";
        void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
        int k;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(830 - 500 + 20, 100 + 150, 70);
        for (k = 0; k < strlen(string2); k++)
                glutBitmapCharacter(font2, string2[k]);

        char string3[] = " at the bus stop";
        void* font3 = GLUT_BITMAP_TIMES_ROMAN_24;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(830 - 500 + 20, 80 + 150, 70);
        for (k = 0; k < strlen(string3); k++)
```

```
		glutBitmapCharacter(font3, string3[k]);

	char string4[] = " using the arrow ";
	void* font4 = GLUT_BITMAP_TIMES_ROMAN_24;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(830 - 500 + 20, 60 + 150, 70);
	for (k = 0; k < strlen(string4); k++)
		glutBitmapCharacter(font4, string4[k]);
	char string5[] = " keys ";
	void* font5 = GLUT_BITMAP_TIMES_ROMAN_24;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(830 - 500 + 20, 40 + 150, 70);
	for (k = 0; k < strlen(string5); k++)
		glutBitmapCharacter(font5, string5[k]);

}
void text3()
{
	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex2i(830 - 500, 120 + 150);
	glVertex2i(1020 - 500 + 40, 120 + 150);
	glVertex2i(1020 - 500 + 40, 35 + 150);
	glVertex2i(830 - 500, 35 + 150);
	glEnd();
	char string2[] = "YAY!Now get her ";
	void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
	int k;
	glColor3ub(240, 0, 0);
	glRasterPos3f(832 - 500 + 20, 100 + 150, 70);
	for (k = 0; k < strlen(string2); k++)
		glutBitmapCharacter(font2, string2[k]);
	char string3[] = "to her college";
	void* font3 = GLUT_BITMAP_TIMES_ROMAN_24;
	glColor3ub(240, 0, 0);
	glRasterPos3f(832 - 500 + 20, 100 + 130, 70);
```

```
    for (k = 0; k < strlen(string3); k++)
        glutBitmapCharacter(font3, string3[k]);
    char string4[] = "jus straight ahead.";
    void* font4 = GLUT_BITMAP_TIMES_ROMAN_24;
    glColor3ub(240, 0, 0);
    glRasterPos3f(834 - 500 + 20, 100 + 110, 70);
    for (k = 0; k < strlen(string4); k++)
        glutBitmapCharacter(font4, string4[k]);



}
void text4d()
{
    glBegin(GL_POLYGON);
    glColor3ub(150, 150, 250);
    glVertex2i(830 - 500, 120 - 50);
    glVertex2i(1020 - 500, 120 - 50);
    glVertex2i(1020 - 500, 35 - 50);
    glVertex2i(830 - 500, 35 - 50);
    glEnd();
    char string2[] = "Drop her at the  ";
    void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
    int k;
    glColor3ub(0, 0, 0);
    glRasterPos3f(832 - 500 + 7, 100 - 50, 70);
    for (k = 0; k < strlen(string2); k++)
        glutBitmapCharacter(font2, string2[k]);

    char string3[] = "stop sign";
    void* font3 = GLUT_BITMAP_TIMES_ROMAN_24;
    glColor3ub(0, 0, 0);
    glRasterPos3f(832 - 500 + 7, 100 - 70, 70);
    for (k = 0; k < strlen(string3); k++)
        glutBitmapCharacter(font3, string3[k]);
}
void text5d()
```

```
{
    glBegin(GL_POLYGON);
    glColor3ub(20, 3, 5);
    glVertex2i(830 - 500, 120 - 50);
    glVertex2i(1060 - 500, 120 - 50);
    glVertex2i(1060 - 500, 35 - 50);
    glVertex2i(830 - 500, 35 - 50);
    glEnd();

    char string2[] = "Mission Accomplished! ";
    void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
    int k;
    glColor3ub(255, 255, 255);
    glRasterPos3f(832 - 500, 100 - 50, 70);
    for (k = 0; k < strlen(string2); k++)
        glutBitmapCharacter(font2, string2[k]);

    char string3[] = "Parking is right ahead";
    void* font3 = GLUT_BITMAP_TIMES_ROMAN_24;
    glColor3ub(255, 255, 255);
    glRasterPos3f(832 - 500, 100 - 70, 70);
    for (k = 0; k < strlen(string3); k++)
        glutBitmapCharacter(font3, string3[k]);
}
void line()
{
    // lines on d front face
    glBegin(GL_POLYGON);
    glColor3ub(0, 0, 0);
    glVertex3i(400, 390, 70);
    glVertex3i(425, 410, -70);
    glVertex3i(425, 407, -70);
    glVertex3i(400, 387, 70);
    glVertex3i(393, 393, 70);
    glVertex3i(393, 390, 70);
    glEnd();
```

```
        glBegin(GL_LINES);
        glColor3ub(0, 0, 0);
        glVertex3f(408, 405, 20);
        glVertex3f(418, 412, -20);
        glVertex3f(405, 410, 40);
        glVertex3f(420, 420, -40);
        glVertex3f(402, 415, 60);
        glVertex3f(422, 429, -60);
        glEnd();
}
void tree12()
{
        //trunk1
        glColor3ub(95, 6, 5);
        double len = 100;
        double thick = 20;
        glPushMatrix();
        glTranslated(100 + 450, 150 + 330, 0.0);
        glScaled(thick, len, thick);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
       glTranslated(100 + 450, 230 + 310, 0.0);
        glutSolidCone(70, 140, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100 + 450, 260 + 310, 0.0);
        glutSolidCone(60, 120, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100 + 450, 290 + 310, 0);
        glutSolidCone(50, 100, 3, 2);
        glPopMatrix();
}void tree1()
{
        //trunk1
        glColor3ub(95, 6, 5);
        double len = 100;
        double thick = 20;
        glPushMatrix();
        glTranslated(100, 150 - 48, 0.0);
        glScaled(thick, len, thick);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 235 - 48, 0.0);
        glutSolidCone(70, 140, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 270 - 48, 0.0);
        glutSolidCone(60, 120, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 300 - 48, 0);
        glutSolidCone(50, 100, 3, 2);
        glPopMatrix();
}
void woman()
{
        //face
```

```
glColor3ub(0, 0, 0);
glPushMatrix();
glTranslatef(540, 495, 0);
glutSolidTorus(1, 10, 100, 90);
glPopMatrix();
glColor3ub(255, 191, 128);
glPushMatrix();
glTranslatef(540, 494, 0);
glutSolidTorus(7, 7, 100, 90);
glPopMatrix();
glColor3ub(0, 0, 0);
glBegin(GL_LINES);
glVertex2i(540, 494);
glVertex2i(540, 490); //nose
glVertex2i(531, 498);
glVertex2i(532, 499);
glVertex2i(532, 499);
glVertex2i(537, 498);//eyebrow
glVertex2i(549, 498);
glVertex2i(548, 499);
glVertex2i(548, 499);
glVertex2i(543, 498);//eyebrow
glEnd();
//ear right
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(540 - 14, 494 + 1);
glVertex2i(540 - 14, 490 + 1);
glVertex2i(538 - 14, 489 + 1);
glVertex2i(538 - 14, 495 + 1);
glEnd();
//ear left
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(554, 495);
glVertex2i(556, 496);
```

```
glVertex2i(556, 491);
glVertex2i(554, 490);
glEnd();
//ear ring right
glBegin(GL_POLYGON);
glColor3ub(255, 85, 90);
glVertex2i(539 - 14, 492);
glVertex2i(542 - 14, 485);
glVertex2i(536 - 14, 485);
glEnd();
//ear ring left
glBegin(GL_POLYGON);
glColor3ub(255, 85, 90);
glVertex2i(551, 485);
glVertex2i(555, 492);
glVertex2i(558, 485);
glEnd();
//hair
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(525, 499);
glVertex2i(549, 509);
glVertex2i(540, 512);
glVertex2i(528, 507);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(540, 507);
glVertex2i(549, 509);
glVertex2i(552, 507);
glVertex2i(555, 499);
glEnd();
// eyes
glBegin(GL_POLYGON);
glVertex2i(533, 496);
glVertex2i(535, 496);
```

```
glVertex2i(535, 494);
glVertex2i(533, 494);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(545, 496);
glVertex2i(547, 496);
glVertex2i(547, 494);
glVertex2i(545, 494);
glEnd();
//mouth
glBegin(GL_POLYGON);
glColor3ub(150, 50, 50);
glVertex2i(534, 487);
glVertex2i(540, 484);
glVertex2i(546, 487);
glVertex2i(540, 485);
glEnd();
//shirt
glBegin(GL_POLYGON);
glColor3ub(160, 150, 250);
glVertex2i(529, 480);
glVertex2i(551, 480);
glVertex2i(566, 469);
glVertex2i(561, 460);
glVertex2i(556, 465);
glVertex2i(556, 445);
glVertex2i(524, 445);
glVertex2i(524, 465);
glVertex2i(519, 460);
glVertex2i(514, 469);
glEnd();
//neck
glBegin(GL_POLYGON);
glColor3ub(255, 190, 128);
glVertex2i(533, 480);
glVertex2i(547, 480);
```

```
glVertex2i(545, 471);
glVertex2i(535, 471);
glEnd();
//hands
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(565, 468);
glVertex2i(575, 453);
glVertex2i(567, 454);
glVertex2i(562, 462);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(575, 453);
glVertex2i(556, 438);
glVertex2i(556, 445);
glVertex2i(567, 454);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(515, 468);
glVertex2i(505, 453);
glVertex2i(513, 454);
glVertex2i(518, 462);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(505, 453);
glVertex2i(524, 438);
glVertex2i(524, 445);
glVertex2i(513, 454);
glEnd();
// belt
glBegin(GL_POLYGON);
glColor3ub(10, 120, 130);
glVertex2i(556, 445);
glVertex2i(524, 445);
glVertex2i(524, 440);
glVertex2i(556, 440);
```

```
glEnd();
/// leg
glBegin(GL_POLYGON);
glColor3ub(255, 190, 128);
glVertex2i(555, 440);
glVertex2i(525, 440);
glVertex2i(520, 405);
glVertex2i(530, 405);
glVertex2i(533, 438);
glVertex2i(550, 405);
glVertex2i(560, 405);
glEnd();
//skirt
glBegin(GL_POLYGON);
glColor3ub(180, 80, 90);
glVertex2i(524, 440);
glVertex2i(556, 440);
glVertex2i(566, 410);
glVertex2i(514, 410);
glEnd();
//shoe left
glBegin(GL_POLYGON);
glColor3ub(180, 0, 0);
glVertex2i(530, 405);
glVertex2i(530, 396);
glVertex2i(528, 396);
glVertex2i(528, 404);
glVertex2i(522, 396);
glVertex2i(512, 396);
glVertex2i(520, 405);
glEnd();
//shoe right
glBegin(GL_POLYGON);
glColor3ub(180, 0, 0);
glVertex2i(550, 405);
glVertex2i(550, 396);
```

```
		glVertex2i(552, 396);
		glVertex2i(552, 404);
		glVertex2i(558, 396);
		glVertex2i(568, 396);
		glVertex2i(560, 405);
		glEnd();
}
void man()
{
		glColor3ub(0, 0, 0);
		glPushMatrix();
		glTranslatef(540 - 220, 495 + 76, 0);
		glutSolidTorus(1, 10, 100, 90);
		glPopMatrix();
		glColor3ub(255, 191, 128);
		glPushMatrix();
		glTranslatef(540 - 220, 495 + 76, 0);
		glutSolidTorus(7, 7, 100, 90);
		glPopMatrix();
		glColor3ub(0, 0, 0);
		glBegin(GL_LINES);
		glVertex2i(540 - 220, 495 + 76);
		glVertex2i(540 - 220, 490 + 76); //nose
		glVertex2i(531 - 220, 500 + 76);
		glVertex2i(537 - 220, 500 + 76);//eyebrow
		glVertex2i(543 - 220, 500 + 76);
		glVertex2i(549 - 220, 500 + 76);//eyebrow
		glEnd();
		//ear right
		glBegin(GL_POLYGON);
		glColor3ub(255, 191, 128);
		glVertex2i(540 - 14 - 220, 494 + 1 + 76);
		glVertex2i(540 - 14 - 220, 490 + 1 + 76);
		glVertex2i(538 - 14 - 220, 489 + 1 + 76);
		glVertex2i(538 - 14 - 220, 495 + 1 + 76);
		glEnd();
```

```
//ear left
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(554 - 220, 495 + 76);
glVertex2i(556 - 220, 496 + 76);
glVertex2i(556 - 220, 491 + 76);
glVertex2i(554 - 220, 490 + 76);
glEnd();
//hair
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(527 - 220, 503 + 76);
glVertex2i(553 - 220, 503 + 76);
glVertex2i(547 - 220, 509 + 76);
glVertex2i(533 - 220, 509 + 76);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(533 - 220, 498 + 76);
glVertex2i(535 - 220, 498 + 76);
glVertex2i(535 - 220, 496 + 76);
glVertex2i(533 - 220, 496 + 76);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(545 - 220, 498 + 76);
glVertex2i(547 - 220, 498 + 76);
glVertex2i(547 - 220, 496 + 76);
glVertex2i(545 - 220, 496 + 76);
glEnd();
// mouth
glBegin(GL_POLYGON);
glVertex2i(535 - 220, 487 + 76);
glVertex2i(540 - 220, 485 + 76);
glVertex2i(545 - 220, 487 + 76);
glVertex2i(540 - 220, 487 + 76);
glEnd();
//beard
```

```
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(538 - 220, 480 + 76);
glVertex2i(542 - 220, 480 + 76);
glVertex2i(542 - 220, 484 + 76);
glVertex2i(538 - 220, 484 + 76);
glEnd();
//shirt
glBegin(GL_POLYGON);
glColor3ub(55, 50, 70);
glVertex2i(529 - 220, 480 + 76);
glVertex2i(551 - 220, 480 + 76);
glVertex2i(566 - 220, 469 + 76);
glVertex2i(561 - 220, 461 + 76);
glVertex2i(556 - 220, 465 + 76);
glVertex2i(556 - 220, 445 + 76);
glVertex2i(524 - 220, 445 + 76);
glVertex2i(524 - 220, 465 + 76);
glVertex2i(519 - 220, 460 + 76);
glVertex2i(514 - 220, 469 + 76);
glEnd();
//hands
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(565 - 220, 468 + 76);
glVertex2i(575 - 220, 453 + 76);
glVertex2i(567 - 220, 454 + 76);
glVertex2i(562 - 220, 462 + 76);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(575 - 220, 453 + 76);
glVertex2i(556 - 220, 438 + 76);
glVertex2i(556 - 220, 445 + 76);
glVertex2i(567 - 220, 454 + 76);
glEnd();
glBegin(GL_POLYGON);
```

```
glVertex2i(515 - 220, 468 + 76);
glVertex2i(505 - 220, 453 + 76);
glVertex2i(513 - 220, 454 + 76);
glVertex2i(518 - 220, 462 + 76);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(505 - 220, 453 + 76);
glVertex2i(524 - 220, 438 + 76);
glVertex2i(524 - 220, 445 + 76);
glVertex2i(513 - 220, 454 + 76);
glEnd();
// belt
glBegin(GL_POLYGON);
glColor3ub(150, 12, 30);
glVertex2i(556 - 220, 445 + 76);
glVertex2i(524 - 220, 445 + 76);
glVertex2i(524 - 220, 440 + 76);
glVertex2i(524 - 220, 440 + 76);
glVertex2i(556 - 220, 440 + 76);
glEnd();
// collar
glBegin(GL_POLYGON);
glColor3ub(200, 140, 110 + 76);
glVertex2i(529 - 220, 480 + 76);
glVertex2i(551 - 220, 480 + 76);
glVertex2i(546 - 220, 470 + 76);
glVertex2i(534 - 220, 470 + 76);
glEnd();
glBegin(GL_TRIANGLES);
glColor3ub(20, 140, 110);
glVertex2i(540 - 220, 477 + 76);
glVertex2i(545 - 220, 470 + 76);
glVertex2i(535 - 220, 470 + 76);
glEnd();
// buttons
glColor3ub(0, 0, 0);
```

```
glPushMatrix();
glTranslatef(540 - 220, 465 + 76, 0);
glutSolidTorus(1, 1, 100, 90);
glPopMatrix();
glPushMatrix();
glTranslatef(540 - 220, 458 + 76, 0);
glutSolidTorus(1, 1, 100, 90);
glPopMatrix();
glPushMatrix();
glTranslatef(540 - 220, 451 + 76, 0);
glutSolidTorus(1, 1, 100, 90);
glPopMatrix();
/// pant
glBegin(GL_POLYGON);
glColor3ub(80, 80, 230);
glVertex2i(555 - 220, 440 + 76);
glVertex2i(525 - 220, 440 + 76);
glVertex2i(520 - 220, 405 + 76);
glVertex2i(530 - 220, 405 + 76);
glVertex2i(533 - 220, 438 + 76);
glVertex2i(550 - 220, 405 + 76);
glVertex2i(560 - 220, 405 + 76);
glEnd();
//shoe left
glBegin(GL_POLYGON);
glColor3ub(100, 10, 10);
glVertex2i(530 - 220, 405 + 76);
glVertex2i(530 - 220, 396 + 76);
glVertex2i(512 - 220, 396 + 76);
glVertex2i(520 - 220, 405 + 76);
glEnd();
//shoe right
glBegin(GL_POLYGON);
glColor3ub(100, 10, 10);
glVertex2i(550 - 220, 405 + 76);
glVertex2i(550 - 220, 396 + 76);
```

```
        glVertex2i(568 - 220, 396 + 76);
        glVertex2i(560 - 220, 405 + 76);
        glEnd();
}
void wheel1d()
{
        glColor3f(0, 0, 0);
        glPushMatrix();
        glTranslatef(345, 377 - 175, -70);
        glutSolidTorus(5, 15, 100, 90);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(190, 377 - 175, -70);
        glutSolidTorus(5, 15, 100, 90);
        glPopMatrix();
        glColor3ub(100, 100, 100);
        glPushMatrix();
        glTranslatef(345, 377 - 175, -70);
        glutSolidTorus(5, 5, 10, 69);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(190, 377 - 175, -70);
        glutSolidTorus(5, 5, 10, 69);
        glPopMatrix();
}
void wheel2d()
{
        glColor3f(0, 0, 0);

        glPushMatrix();
        glTranslatef(180, 370 - 175, 70);
        glutSolidTorus(5, 15, 100, 90);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(335, 370 - 175, 70);
        glutSolidTorus(5, 15, 100, 90);
```

```
		glPopMatrix();
		glColor3ub(100, 100, 100);
		glPushMatrix();
		glTranslatef(335, 370 - 175, 70);
		glutSolidTorus(5, 5, 10, 69);
		glPopMatrix();
		glPushMatrix();
		glTranslatef(180, 370 - 175, 70);
		glutSolidTorus(5, 5, 10, 69);
		glPopMatrix();
}

void polygond(int a, int b, int c, int d, int E, int f)
{
		glBegin(GL_POLYGON);
		glColor3fv(colorsd[E]);
		glVertex3fv(verticesd[a]);
		glVertex3fv(verticesd[b]);
		glVertex3fv(verticesd[c]);
		glVertex3fv(verticesd[d]);
		if (f != 0)
				glVertex3fv(verticesd[f]);
		glEnd();
}

void colorcubed()
{
		int i;
		wheel1d();
		polygond(0, 1, 5, 4, 0, 0);
		polygond(13, 14, 18, 0, 0, 0);
		polygond(15, 16, 17, 18, 2, 0);
		polygond(16, 11, 2, 1, 0, 17);
		polygond(0, 4, 8, 13, 0, 0);
		polygond(1, 10, 9, 5, 0, 0);
		polygond(9, 10, 2, 6, 1, 0);
```

```
    polygond(4, 5, 9, 8, 0, 0);
    polygond(8, 9, 6, 12, 1, 7);
    glColor3ub(100, 40, 50);
    for (i = 0; i <= 180; i += 45)
    {
        glBegin(GL_LINES);
        glVertex3f(180 + i, 447 - 175, 70);
        glVertex3f(180 + i, 500 - 175, 70);
        glEnd();
    }
    polygond(13, 8, 7, 3, 1, 0);
    polygond(3, 15, 14, 13, 1, 0);
    polygond(6, 2, 11, 12, 0, 0);
    polygond(11, 3, 7, 12, 0, 0);
    wheel2d();
}
void womand()
{
    //face
    glColor3ub(0, 0, 0);
    glPushMatrix();
    glTranslatef(540, 495 - 175, 0);
    glutSolidTorus(1, 10, 100, 90);
    glPopMatrix();
    glColor3ub(255, 191, 128);
    glPushMatrix();
    glTranslatef(540, 494 - 175, 0);
    glutSolidTorus(7, 7, 100, 90);
    glPopMatrix();
    glColor3ub(0, 0, 0);
    glBegin(GL_LINES);
    glVertex2i(540, 494 - 175);
    glVertex2i(540, 490 - 175); //nose
    glVertex2i(531, 498 - 175);
    glVertex2i(532, 499 - 175);
    glVertex2i(532, 499 - 175);
```

```
glVertex2i(537, 498 - 175);//eyebrow
glVertex2i(549, 498 - 175);
glVertex2i(548, 499 - 175);
glVertex2i(548, 499 - 175);
glVertex2i(543, 498 - 175);//eyebrow
glEnd();
//ear right
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(540 - 14, 494 + 1 - 175);
glVertex2i(540 - 14, 490 + 1 - 175);
glVertex2i(538 - 14, 489 + 1 - 175);
glVertex2i(538 - 14, 495 + 1 - 175);
glEnd();
//ear left
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(554, 495 - 175);
glVertex2i(556, 496 - 175);
glVertex2i(556, 491 - 175);
glVertex2i(554, 490 - 175);
glEnd();
//ear ring right
glBegin(GL_POLYGON);
glColor3ub(255, 85, 90);
glVertex2i(539 - 14, 492 - 175);
glVertex2i(542 - 14, 485 - 175);
glVertex2i(536 - 14, 485 - 175);
glEnd();
//ear ring left
glBegin(GL_POLYGON);
glColor3ub(255, 85, 90);
glVertex2i(551, 485 - 175);
glVertex2i(555, 492 - 175);
glVertex2i(558, 485 - 175);
glEnd();
```

```
//hair
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(525, 499 - 175);
glVertex2i(549, 509 - 175);
glVertex2i(540, 512 - 175);
glVertex2i(528, 507 - 175);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(540, 507 - 175);
glVertex2i(549, 509 - 175);
glVertex2i(552, 507 - 175);
glVertex2i(555, 499 - 175);
glEnd();
// eyes
glBegin(GL_POLYGON);
glVertex2i(533, 496 - 175);
glVertex2i(535, 496 - 175);
glVertex2i(535, 494 - 175);
glVertex2i(533, 494 - 175);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(545, 496 - 175);
glVertex2i(547, 496 - 175);
glVertex2i(547, 496 - 175);
glVertex2i(547, 494 - 175);
glVertex2i(545, 494 - 175);
glEnd();
//mouth
glBegin(GL_POLYGON);
glColor3ub(150, 50, 50);
glVertex2i(534, 487 - 175);
glVertex2i(540, 484 - 175);
glVertex2i(546, 487 - 175);
glVertex2i(540, 485 - 175);
```

```
glEnd();
//shirt
glBegin(GL_POLYGON);
glColor3ub(160, 150, 250);
glVertex2i(529, 480 - 175);
glVertex2i(551, 480 - 175);
glVertex2i(566, 469 - 175);
glVertex2i(561, 460 - 175);
glVertex2i(556, 465 - 175);
glVertex2i(556, 445 - 175);
glVertex2i(524, 445 - 175);
glVertex2i(524, 465 - 175);
glVertex2i(519, 460 - 175);
glVertex2i(514, 469 - 175);
glEnd();
//neck
glBegin(GL_POLYGON);
glColor3ub(255, 190, 128);
glVertex2i(533, 480 - 175);
glVertex2i(547, 480 - 175);
glVertex2i(545, 471 - 175);
glVertex2i(535, 471 - 175);
glEnd();
//hands
glBegin(GL_POLYGON);
glColor3ub(255, 191, 128);
glVertex2i(565, 468 - 175);
glVertex2i(575, 453 - 175);
glVertex2i(567, 454 - 175);
glVertex2i(562, 462 - 175);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(575, 453 - 175);
glVertex2i(556, 438 - 175);
glVertex2i(556, 445 - 175);
glVertex2i(567, 454 - 175);
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2i(515, 468 - 175);
glVertex2i(505, 453 - 175);
glVertex2i(513, 454 - 175);
glVertex2i(518, 462 - 175);
glEnd();
glBegin(GL_POLYGON);
glVertex2i(505, 453 - 175);
glVertex2i(524, 438 - 175);
glVertex2i(524, 445 - 175);
glVertex2i(513, 454 - 175);
glEnd();
// belt
glBegin(GL_POLYGON);
glColor3ub(10, 120, 130);
glVertex2i(556, 445 - 175);
glVertex2i(524, 445 - 175);
glVertex2i(524, 440 - 175);
glVertex2i(556, 440 - 175);
glEnd();
/// leg
glBegin(GL_POLYGON);
glColor3ub(255, 190, 128);
glVertex2i(555, 440 - 175);
glVertex2i(525, 440 - 175);
glVertex2i(520, 405 - 175);
glVertex2i(530, 405 - 175);
glVertex2i(533, 438 - 175);
glVertex2i(550, 405 - 175);
glVertex2i(560, 405 - 175);
glEnd();
//skirt
glBegin(GL_POLYGON);
glColor3ub(180, 80, 90);
glVertex2i(524, 440 - 175);
```

```
	glVertex2i(556, 440 - 175);
	glVertex2i(566, 410 - 175);
	glVertex2i(514, 410 - 175);
	glEnd();
	//shoe left
	glBegin(GL_POLYGON);
	glColor3ub(180, 0, 0);
	glVertex2i(530, 405 - 175);
	glVertex2i(530, 396 - 175);
	glVertex2i(528, 396 - 175);
	glVertex2i(528, 404 - 175);
	glVertex2i(522, 396 - 175);
	glVertex2i(512, 396 - 175);
	glVertex2i(520, 405 - 175);
	glEnd();
	//shoe right
	glBegin(GL_POLYGON);
	glColor3ub(180, 0, 0);
	glVertex2i(550, 405 - 175);
	glVertex2i(550, 396 - 175);
	glVertex2i(552, 396 - 175);
	glVertex2i(552, 404 - 175);
	glVertex2i(558, 396 - 175);
	glVertex2i(568, 396 - 175);
	glVertex2i(560, 405 - 175);
	glEnd();
}
void road2d()
{
	int x;
	glColor3ub(7, 255, 130);
	glBegin(GL_POLYGON);
	glVertex2i(0, 650);
	glVertex2i(1000, 650);
	glVertex2i(1000, 0);
	glVertex2i(0, 0);
	glEnd();
	glColor3ub(30, 40, 50);
	glBegin(GL_POLYGON);
	glVertex2i(0, 420 - 175);
	glVertex2i(1000, 420 - 175);
	glVertex2i(1000, 300 - 175);
	glVertex2i(0, 300 - 175);
	glEnd();
	glColor3f(1.0, 0.9, 0.0);
	for (x = 0; x < 1000; x = x + 60)
	{
		glBegin(GL_POLYGON);
		glVertex2f(x, 352.5 + 19 - 175);
		glVertex2f(x, 357.5 + 19 - 175);
		glVertex2f(x + 30, 357.5 + 19 - 175);
		glVertex2f(x + 30, 352.5 + 19 - 175);
		glEnd();
	}
}
void textd()
{
	char string1[] = "";
	void* font1 = GLUT_BITMAP_TIMES_ROMAN_24;

	int j;
	glRasterPos3f(420, 602 - 175, -120);
	for (j = 0; j < strlen(string1); j++)
		glutBitmapCharacter(font1, string1[j]);
}
void text1d()
{
	char string2[] = "TIET";
	void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
	int k;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(230 + p, 400 - 175, 70);
```

```
        for (k = 0; k < strlen(string2); k++)
                glutBitmapCharacter(font2, string2[k]);
}
void text2d()
{
        char string2[] = "TIET";
        void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
        int k;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(295, 400, 70);
        for (k = 0; k < strlen(string2); k++)
                glutBitmapCharacter(font2, string2[k]);
        char string3[] = "Engineering";
        void* font3 = GLUT_BITMAP_TIMES_ROMAN_24;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(270, 380, 70);
        for (k = 0; k < strlen(string3); k++)
                glutBitmapCharacter(font3, string3[k]);
        char string4[] = "College";
        void* font4 = GLUT_BITMAP_TIMES_ROMAN_24;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(300, 360, 70);
        for (k = 0; k < strlen(string4); k++)
                glutBitmapCharacter(font4, string4[k]);
        char string5[] = "Patiala";
        void* font5 = GLUT_BITMAP_TIMES_ROMAN_24;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(270, 340, 70);
        for (k = 0; k < strlen(string5); k++)
                glutBitmapCharacter(font5, string5[k]);
        char string6[] = "Punjab";
        void* font6 = GLUT_BITMAP_TIMES_ROMAN_24;
        glColor3f(0.0, 0.0, 0.0);
        glRasterPos3f(265 + 6, 315, 70);
        for (k = 0; k < strlen(string6); k++)
                glutBitmapCharacter(font6, string6[k]);
```

```
}
void text3d()
{
	char string[] = "STOP";
	void* font = GLUT_BITMAP_TIMES_ROMAN_24;
	int k;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(230 + 140, 400 + 65, 70);
	for (k = 0; k < strlen(string); k++)
		glutBitmapCharacter(font, string[k]);
}
void buildingd()
{
	//buliding
	glColor3ub(255, 70, 20);
	double len = 300;
	double thick = 380;
	glPushMatrix();
	glTranslatef(650 + 55, 520, 70.0);
	glScalef(thick, len, thick);
	glutSolidCube(1.0);
	glPopMatrix();
	//door
	glColor3f(0.0, 0.6, 0.7);
	double len1 = 50;
	double thick1 = 80;
	glPushMatrix();
	glTranslatef(650 + 55, 520 - 125, 70.0);
	glScalef(thick1, len1, thick1);
	glutSolidCube(1.0);
	glPopMatrix();
	glColor3ub(0, 0, 0);
	glBegin(GL_LINE_LOOP);
	glVertex2i(550 + 115, 550 - 130);
	glVertex2i(630 + 115, 550 - 130);
	glVertex2i(630 + 115, 520 - 150);
```

```
glVertex2i(550 + 115, 520 - 150);
glEnd();
glBegin(GL_LINES);
glVertex2i(704, 550 - 130);
glVertex2i(704, 520 - 150);
glEnd();
glColor3f(0.0, 0.6, 0.7);
double len2 = 30;
double thick2 = 30;
glPushMatrix();
glTranslatef(650 - 100, 520, 70.0);
glScalef(thick2, len2, thick2);
glutSolidCube(1.0);
glPopMatrix();
glColor3f(0.0, 0.6, 0.7);
double len3 = 30;
double thick3 = 30;
glPushMatrix();
glTranslatef(650, 520, 70.0);
glScalef(thick3, len3, thick3);
glutSolidCube(1.0);
glPopMatrix();
glColor3f(0.0, 0.6, 0.7);
double len4 = 30;
double thick4 = 30;
glPushMatrix();
glTranslatef(650 + 100, 520, 70.0);
glScalef(thick4, len4, thick4);
glutSolidCube(1.0);
glPopMatrix();
glColor3f(0.0, 0.6, 0.7);
double len5 = 30;
double thick5 = 30;
glPushMatrix();
glTranslatef(650 + 200, 520, 70.0);
glScalef(thick5, len5, thick5);
```

```
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.6, 0.7);
        double len6 = 30;
        double thick6 = 30;
        glPushMatrix();
        glTranslatef(650 - 100, 520 + 100, 70.0);
        glScalef(thick6, len6, thick6);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.6, 0.7);
        double len7 = 30;
        double thick7 = 30;
        glPushMatrix();
        glTranslatef(650, 520 + 100, 70.0);
        glScalef(thick7, len7, thick7);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.6, 0.7);
        double len8 = 30;
        double thick8 = 30;
        glPushMatrix();
        glTranslatef(650 + 100, 520 + 100, 70.0);
        glScalef(thick8, len8, thick8);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.6, 0.7);
        double len9 = 30;
        double thick9 = 30;
        glPushMatrix();
        glTranslatef(650 + 200, 520 + 100, 70.0);
        glScalef(thick9, len9, thick9);
        glutSolidCube(1.0);
        glPopMatrix();

    }
```

```
void lined()
{
	// lines on d front face
	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex3i(400, 390 - 175, 70);
	glVertex3i(425, 410 - 175, -70);
	glVertex3i(425, 407 - 175, -70);
	glVertex3i(400, 387 - 175, 70);
	glVertex3i(393, 393 - 175, 70);
	glVertex3i(393, 390 - 175, 70);
	glEnd();
	glBegin(GL_LINES);
	glColor3ub(0, 0, 0);
	glVertex3f(408, 405 - 175, 20);
	glVertex3f(418, 412 - 175, -20);
	glVertex3f(405, 410 - 175, 40);
	glVertex3f(420, 420 - 175, -40);
	glVertex3f(402, 415 - 175, 60);
	glVertex3f(422, 429 - 175, -60);
	glEnd();
}
void walld()
{
	int i, j;
	float x0 = { 750.0 }, y01 = { 300.0 };
	float x[maxx] = { 40.0 }, y[maxy] = { 20.0 };
	float xc = { 0.0 }, yc = { 300.0 };
	glColor3ub(200, 50, 50);
	glBegin(GL_POLYGON);
	glVertex2i(600 + 150, 433);
	glVertex2i(900 + 150, 433);
	glVertex2i(900 + 150, 300);
	glVertex2i(600 + 150, 300);
	glEnd();
	for (i = 0; i < maxx; i++)
```

```
                x[i] = x0 + i * dx;
        for (j = 0; j < maxy; j++)
                y[j] = y01 + j * dy;
        for (i = 0; i < maxx - 1; i++)
                for (j = 0; j < maxy - 1; j++)
                {
                        glColor3f(0.0, 0.0, 0.0);
                        glBegin(GL_LINE_LOOP);
                        glVertex2f(x[i], y[j]);
                        glVertex2f(x[i + 1], y[j]);
                        glVertex2f(x[i + 1], y[j + 1]);
                        glVertex2f(x[i], y[j + 1]);
                        glEnd();
                }
        glColor3ub(200, 50, 50);
        glBegin(GL_POLYGON);
        glVertex2i(0 - 50, 433);
        glVertex2i(300 - 50, 433);
        glVertex2i(300 - 50, 300);
        glVertex2i(0 - 50, 300);
        glEnd();
        for (i = 0; i < maxx; i++)
                x[i] = xc + i * dx;
        for (j = 0; j < maxy; j++)
                y[j] = yc + j * dy;

        for (i = 0; i < maxx - 1; i++)
                for (j = 0; j < maxy - 1; j++)
                {
                        glColor3f(0.0, 0.0, 0.0);
                        glBegin(GL_LINE_LOOP);
                        glVertex2f(x[i], y[j]);
                        glVertex2f(x[i + 1], y[j]);
                        glVertex2f(x[i + 1], y[j + 1]);
                        glVertex2f(x[i], y[j + 1]);
                        glEnd();
                }
        glColor3ub(250, 220, 220);
        glBegin(GL_POLYGON);
        glVertex2i(0 + 250, 433);
        glVertex2i(300 + 80, 433);
        glVertex2i(300 + 80, 300);
        glVertex2i(0 + 250, 300);
        glEnd();
        glColor3ub(255, 200, 200);
        glBegin(GL_POLYGON);
        glVertex2i(0 + 260, 423);
        glVertex2i(300 + 70, 423);
        glVertex2i(300 + 70, 310);
        glVertex2i(0 + 260, 310);
        glEnd();
}

void gated()
{
        glBegin(GL_POLYGON);
        glColor3ub(0, 0, 0);
        glVertex2i(750, 300);
        glVertex2i(600, 300);
        glVertex2i(600, 303);
        glVertex2i(750, 303);
        glEnd();
        glBegin(GL_POLYGON);
        glColor3ub(0, 0, 0);
        glVertex2i(600, 300);
        glVertex2i(600, 450);
        glVertex2i(597, 450);
        glVertex2i(597, 303);
        glEnd();
        glBegin(GL_POLYGON);
        glColor3ub(0, 0, 0);
        glVertex2i(600, 450);
```

```
glVertex2i(750, 433);
glVertex2i(750, 430);
glVertex2i(600, 447);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(650, 300);
glVertex2i(650, 442);
glVertex2i(653, 442);
glVertex2i(653, 300);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(711, 300);
glVertex2i(711, 437);
glVertex2i(714, 437);
glVertex2i(714, 300);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(600, 350);
glVertex2i(750, 350);
glVertex2i(750, 345);
glVertex2i(600, 345);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(380, 300);
glVertex2i(500, 340);
glVertex2i(500, 343);
glVertex2i(380, 303);
glEnd();
glBegin(GL_POLYGON);
glColor3ub(0, 0, 0);
glVertex2i(380, 433);
glVertex2i(500, 473);
```

```
	glVertex2i(500, 476);
	glVertex2i(380, 436);
	glEnd();
	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex2i(500 - 1, 340);
	glVertex2i(500 - 1, 473);
	glVertex2i(503 - 1, 476);
	glVertex2i(503 - 1, 343);
	glEnd();
	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex2i(500 - 41, 340 - 15);
	glVertex2i(500 - 41, 473 - 15);
	glVertex2i(503 - 41, 476 - 15);
	glVertex2i(503 - 41, 343 - 15);
	glEnd();

	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex2i(500 - 81, 340 - 25);
	glVertex2i(500 - 81, 473 - 25);
	glVertex2i(503 - 81, 476 - 25);
	glVertex2i(503 - 81, 343 - 25);
	glEnd();
	glBegin(GL_POLYGON);
	glColor3ub(0, 0, 0);
	glVertex2i(380, 433 - 90);
	glVertex2i(500, 473 - 90);
	glVertex2i(500, 478 - 90);
	glVertex2i(380, 438 - 90);
	glEnd();
}
void treed()
{
	glColor3ub(95, 6, 5);
	double len = 80;
	double thick = 15;
	glPushMatrix();
	glTranslated(100 + 850, 150 + 330, 0.0);
	glScaled(thick, len, thick);
	glutSolidCube(1.0);
	glPopMatrix();
	glColor3f(0.0, 0.2, 0.0);
	glPushMatrix();
	glLoadIdentity();
	glTranslated(100 + 850, 230 + 290, 0.0);
	glutSolidCone(60, 120, 3, 2);
	glPopMatrix();
	glColor3f(0.0, 0.2, 0.0);
	glPushMatrix();
	glLoadIdentity();
	glTranslated(100 + 850, 260 + 290, 0.0);
	glutSolidCone(50, 100, 3, 2);
	glPopMatrix();
	glColor3f(0.0, 0.2, 0.0);
	glPushMatrix();
	glLoadIdentity();
	glTranslated(100 + 850, 290 + 290, 0);
	glutSolidCone(40, 800, 3, 2);
	glPopMatrix();
}
void tree1d()
{
	glColor3ub(95, 6, 5);
	double len = 80;
	double thick = 15;
	glPushMatrix();
	glTranslated(100, 150 + 330, 0.0);
	glScaled(thick, len, thick);
	glutSolidCube(1.0);
	glPopMatrix();
```

```
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 230 + 290, 0.0);
        glutSolidCone(60, 120, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 260 + 290, 0.0);
        glutSolidCone(50, 100, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(100, 290 + 290, 0);
        glutSolidCone(40, 800, 3, 2);
        glPopMatrix();
}
void tree2d()
{
        //trunk1
        glColor3ub(95, 6, 5);
        double len = 80;
        double thick = 15;
        glPushMatrix();
        glTranslated(200, 150 + 330, 0.0);
        glScaled(thick, len, thick);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(200, 230 + 290, 0.0);
        glutSolidCone(60, 120, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(200, 260 + 290, 0.0);
        glutSolidCone(50, 100, 3, 2);
        glPopMatrix();
        glColor3f(0.0, 0.2, 0.0);
        glPushMatrix();
        glLoadIdentity();
        glTranslated(200, 290 + 290, 0);
        glutSolidCone(40, 800, 3, 2);
        glPopMatrix();
}
void stopd()
{
        glColor3ub(100, 100, 100);
        double len = 180;
        double thick = 10;
        glPushMatrix();
        glTranslatef(650 - 250, 520 - 180, 70.0);
        glScalef(thick, len, thick);
        glutSolidCube(1.0);
        glPopMatrix();
        glColor3ub(190, 0, 0);
        glPushMatrix();
        glTranslatef(540 - 140, 495 - 25, 0);
        glutSolidTorus(8, 32, 100, 90);
        glPopMatrix();
        glColor3ub(255, 255, 255);
        glPushMatrix();
        glLoadIdentity();
        glTranslatef(400, 495 - 25, 0);
        glutSolidSphere(32, 20, 20);
        glPopMatrix();

}
```

```
void intro()
{
        glColor3ub(247, 185, 183);
        glBegin(GL_POLYGON);
        glVertex2i(0, 650);
        glVertex2i(600, 650);
        glVertex2i(800, 250);
        glVertex2i(0, 250);
        glEnd();
        glColor3ub(247, 185, 183);
        glBegin(GL_POLYGON);
        glVertex2i(600, 650);
        glVertex2i(1000, 650);
        glVertex2i(1000, 250);
        glVertex2i(600, 250);
        glEnd();
        glColor3ub(173, 216, 230);
        glBegin(GL_POLYGON);
        glVertex2i(600, 450);
        glVertex2i(1000, 450);
        glVertex2i(1000, 0);
        glVertex2i(600, 0);
        glEnd();
        glColor3ub(173, 216, 230);
        glBegin(GL_POLYGON);
        glVertex2i(0, 450);
        glVertex2i(800, 450);
        glVertex2i(800, 0);
        glVertex2i(0, 0);
        glEnd();
}
void texti()
{

        char string[] = "INTRODUCTION";
        void* font = GLUT_BITMAP_TIMES_ROMAN_24;
```

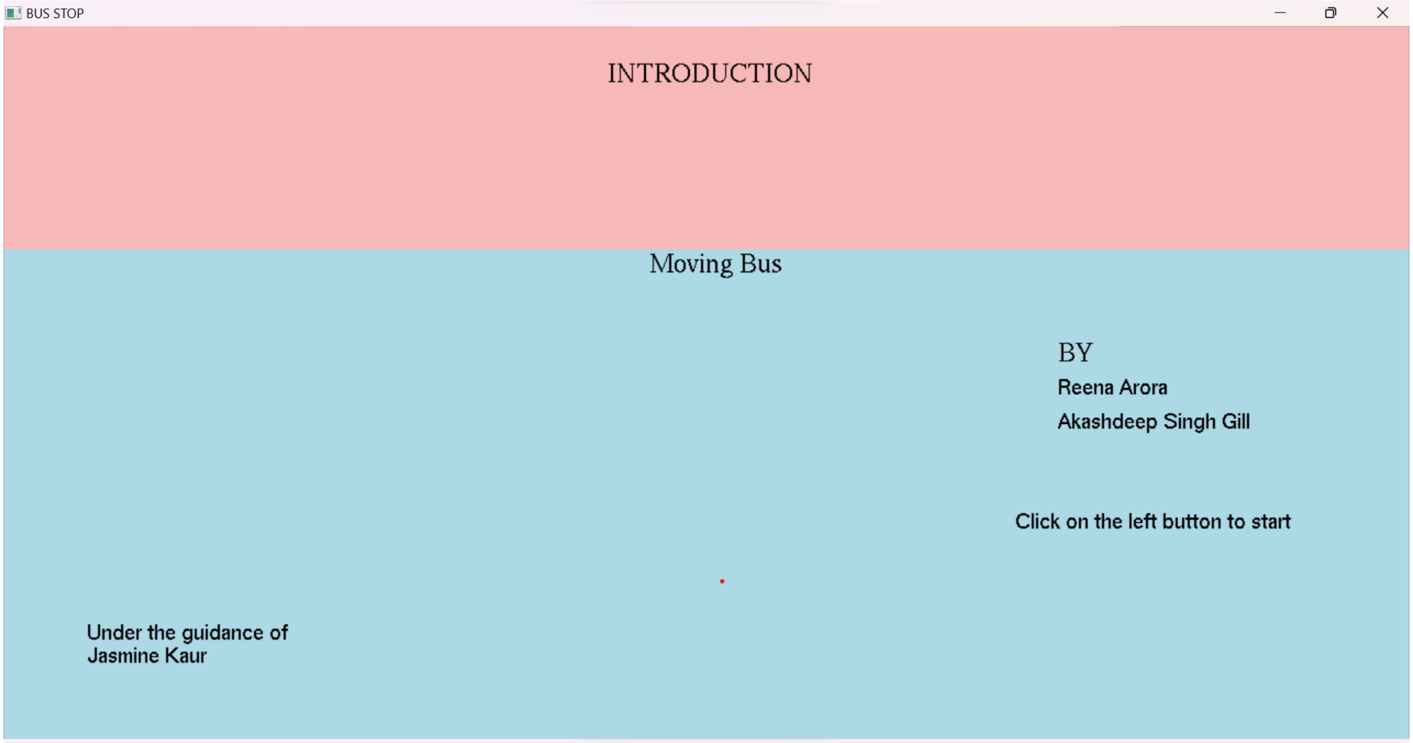
```
int k;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(430, 600, 70);
for (k = 0; k < strlen(string); k++)
	glutBitmapCharacter(font, string[k]);
char string1[] = "Moving Bus";
void* font1 = GLUT_BITMAP_TIMES_ROMAN_24;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(460, 430, 0);
for (k = 0; k < strlen(string1); k++)
	glutBitmapCharacter(font1, string1[k]);
char string2[] = "BY";
void* font2 = GLUT_BITMAP_TIMES_ROMAN_24;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(750, 250 + 100, 0);
for (k = 0; k < strlen(string2); k++)
	glutBitmapCharacter(font2, string2[k]);
char string3[] = "Reena Arora";
void* font3 = GLUT_BITMAP_HELVETICA_18;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(750, 220 + 100, 0);
for (k = 0; k < strlen(string3); k++)
	glutBitmapCharacter(font3, string3[k]);
char string4[] = "Akashdeep Singh Gill";
void* font4 = GLUT_BITMAP_HELVETICA_18;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(750, 220 + 70, 0);

for (k = 0; k < strlen(string4); k++)
	glutBitmapCharacter(font4, string4[k]);
char string5[] = "Under the guidance of ";
void* font5 = GLUT_BITMAP_HELVETICA_18;
glColor3f(0.0, 0.0, 0.0);
glRasterPos3f(60, 100, 0);
for (k = 0; k < strlen(string5); k++)
	glutBitmapCharacter(font5, string5[k]);
	char string6[] = "Jasmine Kaur";
	void* font6 = GLUT_BITMAP_HELVETICA_18;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(60, 80, 0);
	for (k = 0; k < strlen(string6); k++)
		glutBitmapCharacter(font6, string6[k]);
	char string7[] = "";
	void* font7 = GLUT_BITMAP_HELVETICA_18;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(60, 60, 0);
	for (k = 0; k < strlen(string7); k++)
		glutBitmapCharacter(font7, string7[k]);
	char string8[] = "Click on the left button to start";
	void* font8 = GLUT_BITMAP_HELVETICA_18;
	glColor3f(0.0, 0.0, 0.0);
	glRasterPos3f(720, 200, 0);
	for (k = 0; k < strlen(string8); k++)
		glutBitmapCharacter(font8, string8[k]);



}
void mouse(int btn, int state, int x, int y)
{
	if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
	{
		then = 1;
		glutPostRedisplay();
	}
}

void bus_move()
{
	if (x < 50)
	{
		x += 3;
```

```
                glPushMatrix();
                glTranslatef(-100, 0, -90);
                woman();
                glPopMatrix();
                glPushMatrix();
                glTranslatef(x, 0, 0);
                wheel1();
                colorcube();
                wheel2();
                line();
                text1();
                glPopMatrix();
        }
        if (x >= 50)
                vari = 1;


        if (flag55 == 1)
        {
                x += 6;
                glPushMatrix();
                glTranslatef(x, 0, 0);
                wheel1();
                colorcube();
                wheel2();
                line();
                text1();
                glPopMatrix();
        }
        if (x >= 865)
                var = 1;
}
void bus_moved()
{
        if (xd > 50)
        {
                xd += 3;
                glPushMatrix();
                glTranslatef(-100, 0, -90);
                womand();
                glPopMatrix();
                glPushMatrix();
                glTranslatef(xd, 0, 0);
                wheel1d();
                colorcubed();
                wheel2d();
                lined();
                text1d();
                glPopMatrix();
        }
        if (flag551 == 1)
        {
                xd += 5;
                glPushMatrix();
                glTranslatef(xd, 0, 0);
                wheel1d();
                colorcubed();
                wheel2d();
                lined();
                text1d();
                glPopMatrix();
        }
        if (xd > 50)
                varid = 1;

}
static void SpecialKeyFunc(int Key, int x, int y)
{
        switch (Key)
        {
        case GLUT_KEY_UP:                       /*move to right */
        glutPostRedisplay();
        break;
    case GLUT_KEY_RIGHT:

        glutPostRedisplay();
        break;
    }
}
void display(void)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0, 1000, 10.0, 650, -2000, 1500);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(1, 1, 1, 1);
    glClear(GL_DEPTH_BUFFER_BIT | GL_COLOR_BUFFER_BIT);
    if (then == 0)
    {
        intro();
        texti();
    }
    if (then == 1)
    {
        if (flag)
        {
            glPushMatrix();
            glTranslatef(-1.0, 0.0, -3.5);
            glRotatef(xangle + 25, 1.0, 0.0, 0.0);
            glRotatef(yangle, 0.0, 1.0, 0.0);
            glRotatef(zangle, 0.0, 0.0, 1.0);
            road2();
            glPushMatrix();
            glTranslatef(0, 00, -50);
            bus_stop();
            glPopMatrix();
            tree1();
            tree12();
            man();
            text2();
            bus_move();
            glPopMatrix();
        }
        else
        {
            road2();
            bus_stop();
            text();
            tree1();
            tree12();
            man();
            text2();
            bus_move();
            flag55 = 1;
        }
        if (vari == 1)
        {
            text3();
            if (x == 865)
                vari = 0;
        }
        if (var == 1)
        {
            if (flag1)
            {
                glPushMatrix();
                glTranslatef(-1.0, 0.0, -3.5);
                glRotatef(xangle + 25, 1.0, 0.0, 0.0);
                glRotatef(yangle, 0.0, 1.0, 0.0);
                glRotatef(zangle, 0.0, 0.0, 1.0);
                road2d();
                buildingd();
                walld();
```

```
                        text2d();
                        gated();
                        treed();
                        tree1d();
                        tree2d();
                        stopd();
                        text3d();
                        text4d();
                        bus_moved();
                        glPopMatrix();
                }
                else
                {

                        road2d();
                        textd();
                        buildingd();
                        walld();
                        text2d();
                        gated();
                        treed();
                        tree1d();
                        tree2d();
                        stopd();
                        text3d();
                        text4d();
                        bus_moved();
                        flag551 = 1;
                }
                if (varid == 1)
                        text5d();
        }


}
glFlush();
```

```
        glutSwapBuffers();
}
void myreshape(int w, int h)
{
        glViewport(0, 0, w, h);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        if (w <= h)
                glOrtho(-2.0, 2.0, -2.0 * (GLfloat)h / (GLfloat)w, 2.0 * (GLfloat)h / (GLfloat)w, -10.0, 10.0);
        else
                glOrtho(-2.0 * (GLfloat)w / (GLfloat)h, 2.0 * (GLfloat)w / (GLfloat)h, -2.0, 2.0, -10.0, 10.0);
        glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
        glutInitWindowSize(1000, 650);
        glutInitWindowPosition(0, 0);
        glutCreateWindow("BUS STOP");
        glutDisplayFunc(display);
        glutMouseFunc(mouse);
        glutSpecialFunc(SpecialKeyFunc);
        glutReshapeFunc(myreshape);

        glutMainLoop();
        return 1;
}
```

OUTPUT/SCREENSHOTS



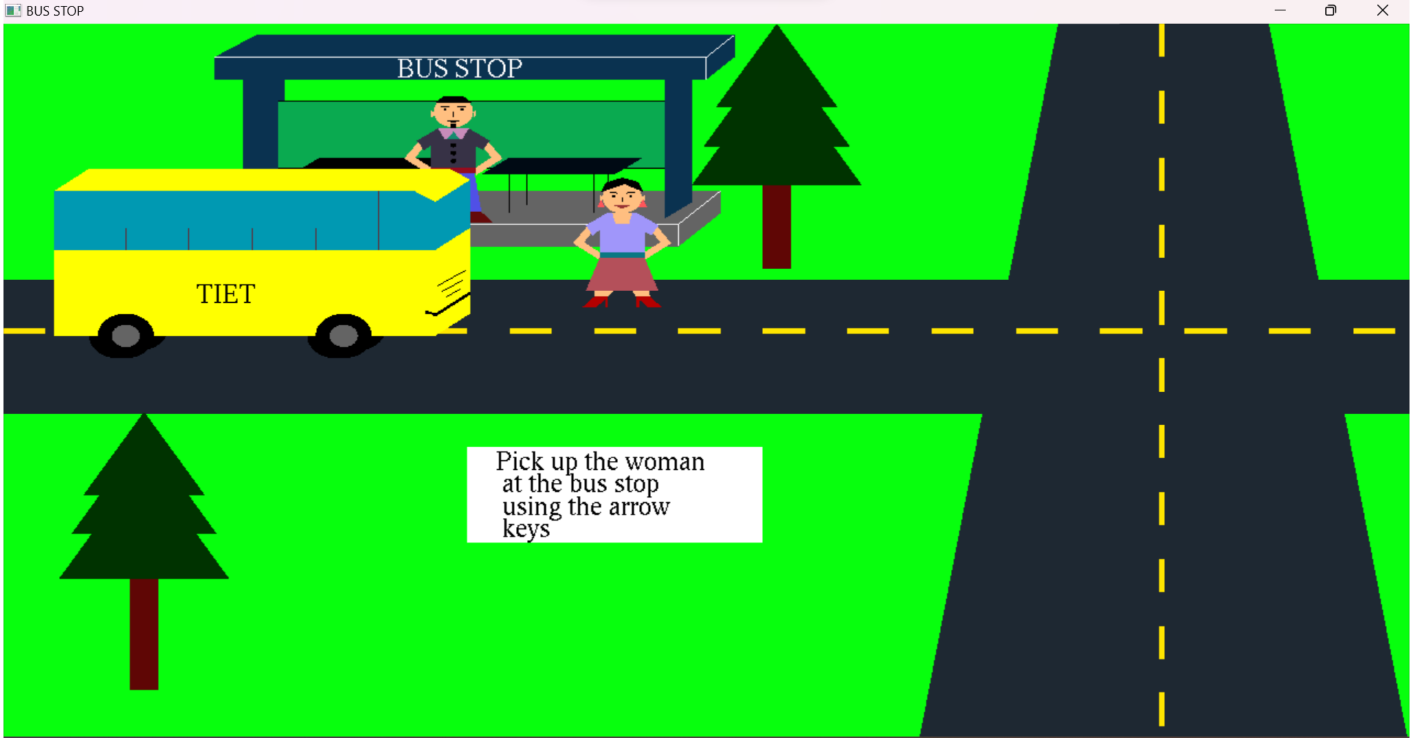
INTRODUCTION

Moving Bus

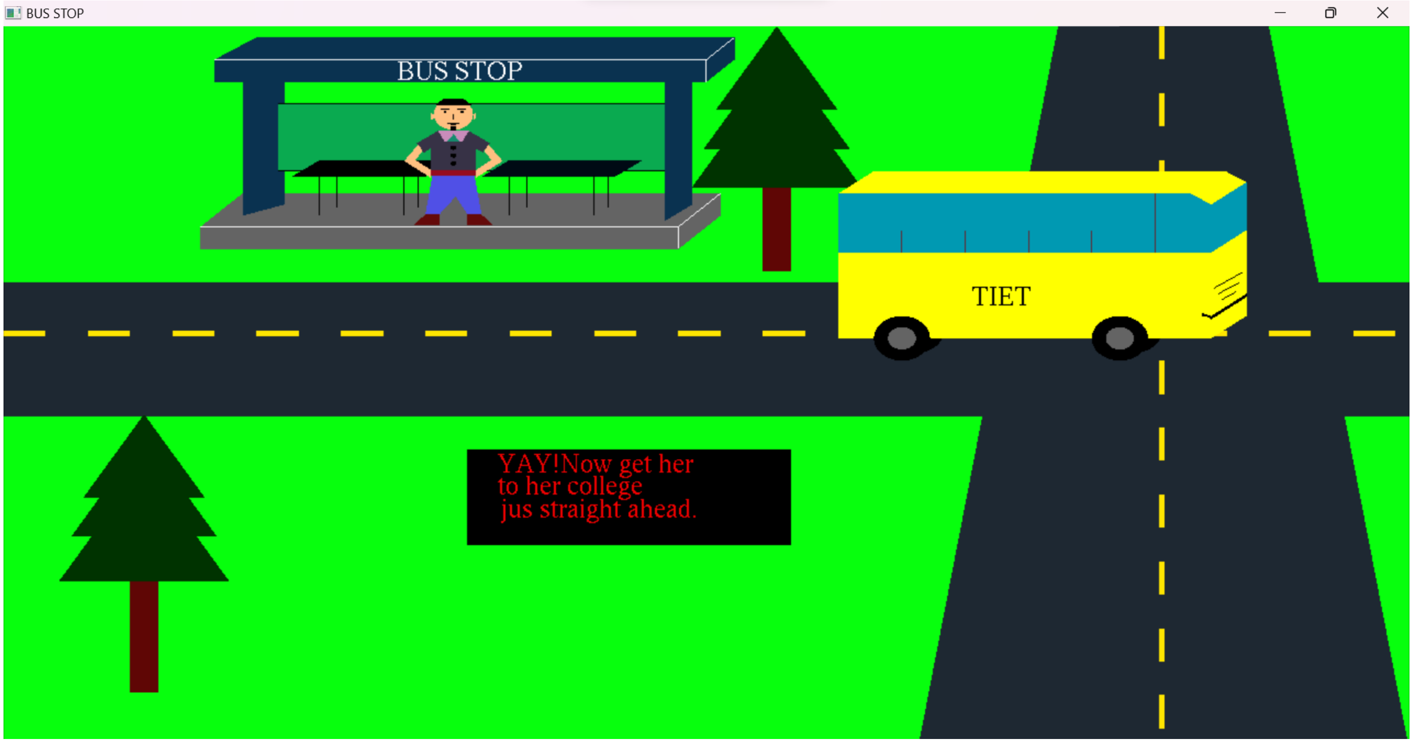
BY
Reena Arora
Akashdeep Singh Gill

Click on the left button to start

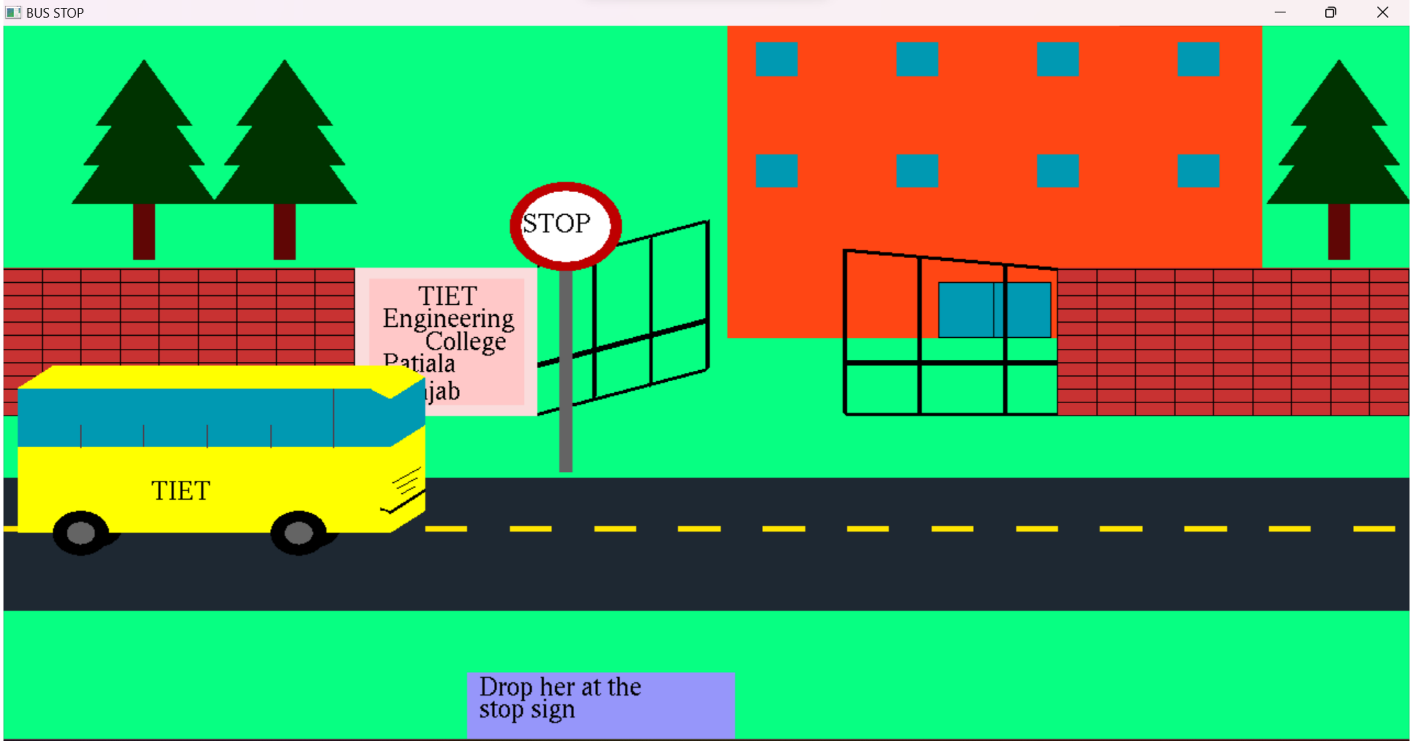
Under the guidance of
Jasmine Kaur



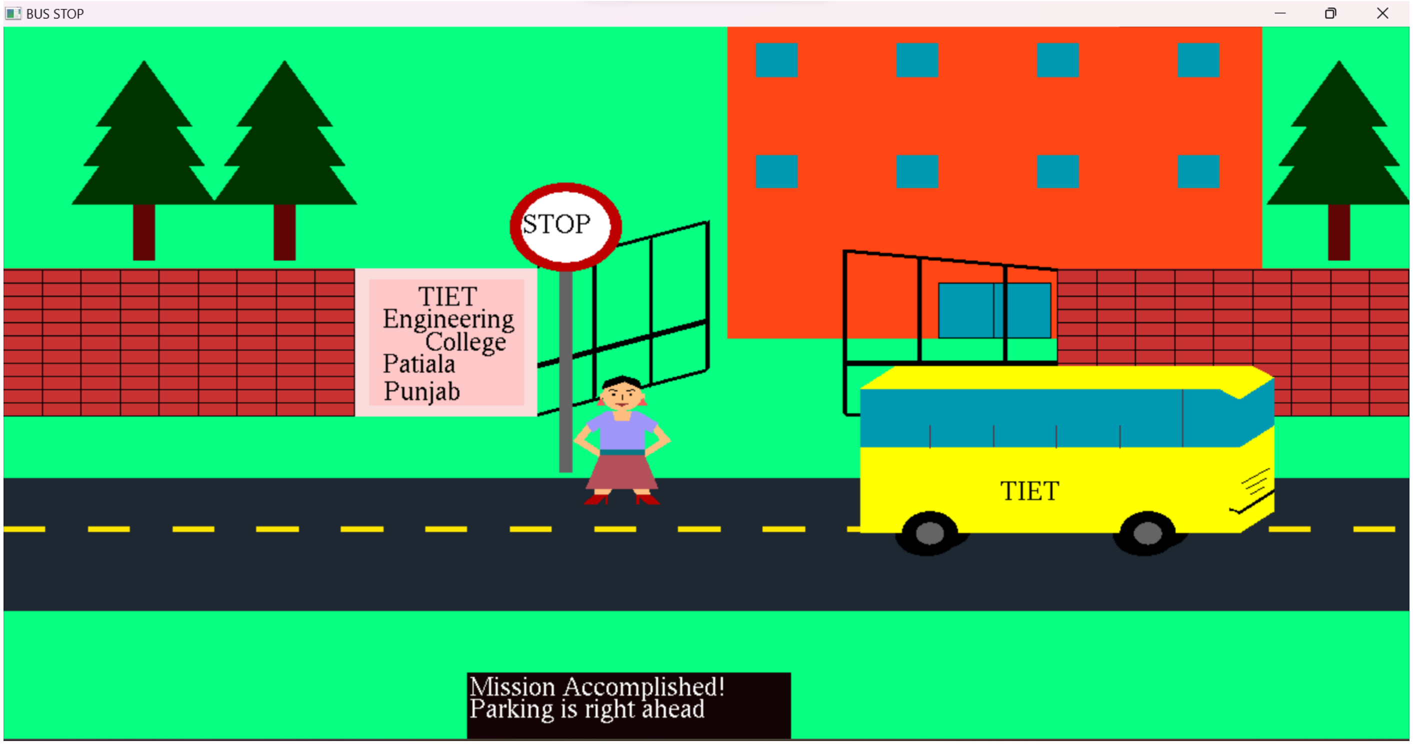
Pick up the woman
at the bus stop
using the arrow
keys



YAY!Now get her
to her college
jus straight ahead.



Drop her at the
stop sign



Mission Accomplished!
Parking is right ahead