# COMPUTER GRAPGHICS

# UCS505

## Mini Project Report

## On

## “Bus Stop Simulation”


**Submitted by:**

**101803230 Dhwani Malaviya**

**101803391 Abhiraj Singh**

**101803392 Amritpal Singh**

**BE Third Year- COE**

**Group- COE18**


**Submitted to:**

**Dr. Shatrughan Modi**

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


**Computer Science and Engineering Department Thapar**

**Institute of Engineering and Technology, Patiala**


**May-June 2022**

# 1. INTRODUCTION

## 1.1 Computer Graphics

Computer Graphics involves technology to access. The Process transforms and presents information in a visual form. The role of computer graphics insensible. In today life, computer graphics has now become a common element in user interfaces, T.V. commercial motion pictures.

Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering.

In computer graphics, two or three-dimensional pictures can be created that are used for research. Many hardware devices algorithm has been developing for improving the speed of picture generation with the passes of time. It includes the creation storage of models and image of objects. These models for various fields like engineering, mathematical and so on.

## 1.2 OpenGL

OpenGL (Open Graphics Library) is a software interface to graphics hardware. The interface consists of over 250 different function calls which can be used to draw complex two and three-dimensional scenes from simple geometric primitives such as points, lines, and polygons. There are also routines for rendering the scenes with control over lighting, object surface properties, transparency, anti-aliasing and texture mapping. OpenGL was developed by Silicon Graphics Inc. in 1992 to be a more ‘open’ (portable) version of their early SGI GL Graphics Library. OpenGL is designed as a streamlined, hardware-independent interface to be implemented on many different graphics

hardware platforms.

## 1.3 Bus Stop Simulation

Bus stop simulation is a graphical representation of bus and its journey to various bus stops. It involves various interactive components like bus, passenger and environment. Objective here is to graphically illustrate the simulation of bus stop and the journey of bus carrying passenger from bus stop to their destination.

# 2. PROBLEM STATEMENT

- To design a moving graphical visualization of the scenario of a bus picking up the passenger from bus stop and dropping to their destination.
- Background should be stable.
- Bus should be the moving character.
- Pick up passenger from bus stop.
- Drop passenger at their destination (apartments).

# 3. DESIGN & IMPLEMENTATION

## 3.1 User defined functions

Following are the functions which are used to create surroundings and objects interacting with surroundings.

```
void busstop();
void busstop_top();
void busstop_left();
void lamp();
void bus();
void bus_scal();
void bus_final();
void river();
void drawDisk(double radius);
void small_car_moving();
void boat_moving();
void drawWheel();
void tree();
void building_scal();
void building2_abd_bottom();
void building3_scal();
void building2_translate();
void building_abd_traslate();
void building_abd();
void building2_abd();
void building_abd_scal();
void building_abd_traslate_again_bottom();
void grass();
void building();
GLvoid Tire(GLdouble y);
GLvoid tree(GLdouble x);
GLvoid window1(GLdouble x);
GLvoid window2(GLdouble y);
GLvoid window3(GLdouble w);
GLvoid window4(GLdouble v);
GLvoid window_bus(GLdouble x);
```

For creating **Trees:** we used 3 triangles and 1 rectangle to form a tree.

```
void tree(){
    glColor3ub(200,80,10);
    glBegin(GL_POLYGON); // rectangle
    glVertex2f(50,0);
    glVertex2f(70,0);
    glVertex2f(70,80);
    glVertex2f(50,80);
    glEnd();

     glColor3ub(0,51,0);
    glBegin(GL_POLYGON); // triangle 1
    glVertex2f(60,230);
    glVertex2f(120,80);
    glVertex2f(0,80);
    glEnd();

     glBegin(GL_POLYGON); // triangle 2
    glVertex2f(60,270);
    glVertex2f(110,120);
    glVertex2f(10,120);
    glEnd();

     glBegin(GL_POLYGON); // triangle 3
    glVertex2f(60,280);
    glVertex2f(100,160);
    glVertex2f(20,160);
    glEnd();
}
```

For creating **Bus**, following functions were used:

*Drawing disks and wheels of bus:*

```
void drawDisk(double radius) {
	int d;
	glBegin(GL_POLYGON);
	for (d = 0; d < 32; d++) {
		double angle = 2*PI/32 * d;
		glVertex2d( radius*cos(angle), radius*sin(angle));
	}
	glEnd();
}
void drawWheel() {

	glColor3f(0,0,0);
	drawDisk(1);
	glColor3f(0.75f, 0.75f, 0.75f);
	drawDisk(0.8);
	glColor3f(0,0,0);
	drawDisk(0.2);
```

```
}
```

*Drawing windows of bus:*

```
 GLvoid  window_bus(GLdouble x)
{       glBegin(GL_QUADS);
    glColor3ub(255, 255, 255);
        glVertex2f(x+12,20);
        glVertex2f(x+17,20);

        glVertex2f(x+17,25);
        glVertex2f(x+12,25);
    glEnd();
    glFlush();
}
```

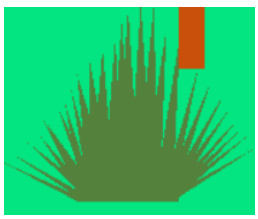
```
void bus(){
     glPushMatrix();
    glTranslatef(30,15,0);
    glScalef(3,3,0);
    drawWheel();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(16,15,0);
    glScalef(3,3,0);
    drawWheel();
       glPopMatrix();

    glBegin(GL_POLYGON); // creating
body of bus
    glColor3ub(255, 155, 21);
    glVertex2f(10,15);
    glVertex2f(10,28);
    glVertex2f(34.5,28);
    glVertex2f(36,23);
    glVertex2f(36,15);
    glEnd();


    window_bus(0);
    window_bus(8);

    glEnd();

}
```

*Drawing body of bus:* simple polygon is used.

```
glBegin(GL_POLYGON);
    glColor3ub(255, 155, 21);
    glVertex2f(10,15);
    glVertex2f(10,28);
    glVertex2f(34.5,28);
    glVertex2f(36,23);
    glVertex2f(36,15);
    glEnd();
```

For creating **Bushes**,

multiple triangles were used.



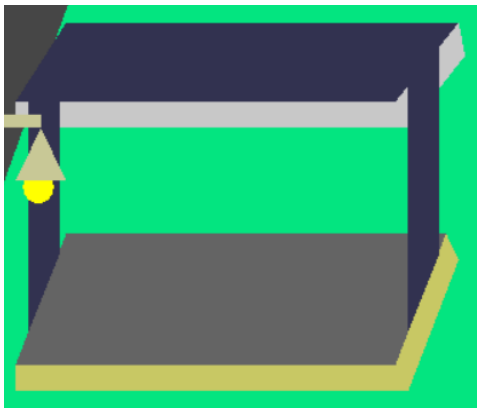
```
void grass()
{
    glColor3ub(84,130,61);
    glBegin(GL_QUADS);
    glVertex2f(150,150);
    glVertex2f(150,170);
    glVertex2f(230,170);
    glVertex2f(230,150);
    glEnd();

    glBegin(GL_POLYGON);
    glVertex2f(190,150);
    glVertex2f(210,150);
    glVertex2f(200,300);
```

***NOTE*****:** *As plenty of triangles were used, so only some part of code is shown here to save space.*

For creating **Lamp-post:** we used 2 rectangles ( to form "T" shape ), 1 triangle and 1 circle to form each light hanging from lamp-post.

```
void lamp()
{
    glColor3ub(200,200,150);
    glBegin(GL_POLYGON);
    glVertex2f(120,600);
    glVertex2f(140,600);
    glVertex2f(140,800);
    glVertex2f(120,800);
    glEnd();

    glBegin(GL_POLYGON);
    glVertex2f(70,800);
    glVertex2f(70,810);
    glVertex2f(190,810);
    glVertex2f(190,800);
    glEnd();

    glPushMatrix();
    glTranslatef(88,695,0);
    glScalef(4,4,0);
    Tire(10);
    glPopMatrix();

     glPushMatrix();
    glTranslatef(-32,695,0);
    glScalef(4,4,0);
    Tire(10);
    glPopMatrix();

    glColor3ub(200,200,150);
    glBegin(GL_POLYGON);
    glVertex2f(70,800);
    glVertex2f(50,760);
    glVertex2f(90,760);
    glEnd();


    glBegin(GL_POLYGON);
    glVertex2f(190,800);
    glVertex2f(170,760);
    glVertex2f(210,760);
    glEnd();

}
```

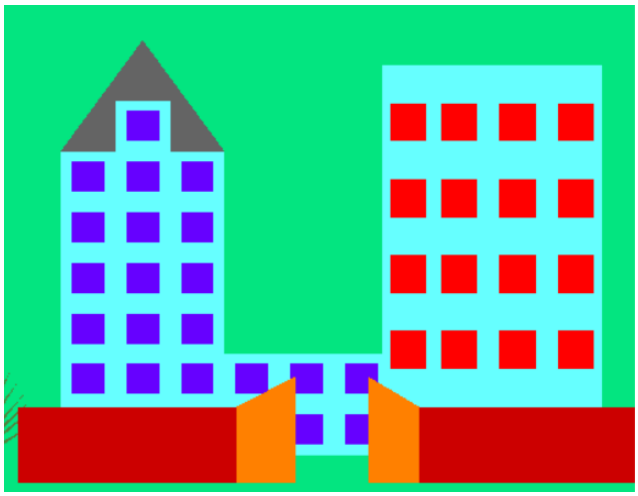
For creating **Bus-Stop**, following functions were used:



```
void busstop_top()
{
      //bus stop
    glColor3ub(50,50,80);
    glBegin(GL_POLYGON);
    glVertex2f(380,650);
    glVertex2f(420,710);
    glVertex2f(730,710);
    glVertex2f(700,650);
    glEnd();

    glColor3ub(200,200,200);
    glBegin(GL_POLYGON);
    glVertex2f(380,650);
    glVertex2f(380,630);
    glVertex2f(690,630);
    glVertex2f(680,650);
    glEnd();

    glColor3ub(200,200,200);
    glBegin(GL_POLYGON);
    glVertex2f(690,630);
    glVertex2f(680,650);
    glVertex2f(730,710);
    glVertex2f(735,685);
    glEnd();
}

void busstop_left()
{
    glColor3ub(50,50,80);
    glBegin(GL_POLYGON);
    glVertex2f(390,675);
    glVertex2f(390,865);
    glVertex2f(415,895);
    glVertex2f(415,735);
    glEnd();
}
```

```
void busstop()
{
      //bus stop
    glColor3ub(100,100,100);
    glBegin(GL_POLYGON);
    glVertex2f(380,650);
    glVertex2f(420,750);
    glVertex2f(720,750);
    glVertex2f(680,650);
    glEnd();

    glColor3ub(200,200,100);
    glBegin(GL_POLYGON);
    glVertex2f(380,650);
    glVertex2f(380,630);
    glVertex2f(690,630);
    glVertex2f(680,650);
    glEnd();

    glColor3ub(200,200,100);
    glBegin(GL_POLYGON);
    glVertex2f(690,630);
    glVertex2f(680,650);
    glVertex2f(720,750);
    glVertex2f(730,730);
    glEnd();


    glPushMatrix();
    glColor3ub(50,50,80);
    glTranslatef(0,200,0);
    busstop_top();
       glPopMatrix();

       busstop_left();

       //busstop right
       glPushMatrix();
    glTranslatef(300,0,0);
    busstop_left();
       glPopMatrix();
}
```

For creating **Apartment building**:



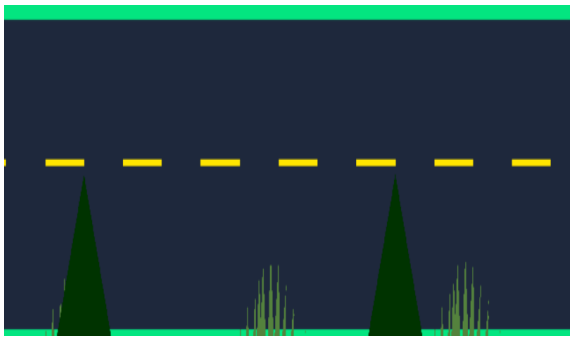
For creating **Different types of windows**:

```
GLvoid window1(GLdouble y)
{
    glBegin(GL_QUADS);
    glColor3ub(255, 0, 0);
        glVertex2f(y+70,50);
        glVertex2f(y+75,50);
        glVertex2f(y+75,45);
        glVertex2f(y+70,45);

        glEnd();
        glFlush();
}
GLvoid window2(GLdouble z)
{
    glBegin(GL_QUADS);
    glColor3ub(255, 0, 0);
        glVertex2f(z+70,60);
        glVertex2f(z+75,60);
        glVertex2f(z+75,55);
        glVertex2f(z+70,55);

        glEnd();
        glFlush();
}

GLvoid window3(GLdouble w)
{
    glBegin(GL_QUADS);
    glColor3ub(255, 0, 0);
        glVertex2f(w+70,70);
        glVertex2f(w+75,70);
        glVertex2f(w+75,65);
        glVertex2f(w+70,65);

        glEnd();
        glFlush();

}

GLvoid window4 (GLdouble v)
```

```
void building() {
  glBegin(GL_QUADS);
    glColor3ub(102, 255, 255);
        glVertex2f(70,85);
        glVertex2f(100,85);
        glVertex2f(100,30);
        glVertex2f(70,30);

  glEnd();

glBegin(GL_QUADS); //gate1
    glColor3ub(255, 128, 0);
        glVertex2f(50,30);
        glVertex2f(50,40);
        glVertex2f(58,44);
        glVertex2f(58,30);
  glEnd();

  glBegin(GL_QUADS); //gate2
    glColor3ub(255, 128, 0);
        glVertex2f(68,30);
        glVertex2f(75,30);
        glVertex2f(75,40);
        glVertex2f(68,44);
  glEnd();

window1(1);
window1(8);
window1(16);
window1(24);

window2(1);
window2(8);
window2(16);
window2(24);

window3(1);
window3(8);
window3(16);
window3(24);

window4(1);
window4(8);
window4(16);
window4(24);

glBegin(GL_QUADS); //wall-right
    glColor3ub(204, 0, 0);
        glVertex2f(75,40);
        glVertex2f(120,40);
        glVertex2f(120,30);
        glVertex2f(75,30);
  glEnd();

 glBegin(GL_QUADS);//wall-left
    glColor3ub(204, 0, 0);
        glVertex2f(20,30);
        glVertex2f(20,40);
        glVertex2f(50,40);
        glVertex2f(50,30);
  glEnd();
}
```

```
{
    glBegin(GL_QUADS);
    glColor3ub(255, 0, 0);
        glVertex2f(v+70,80);
        glVertex2f(v+75,80);
        glVertex2f(v+75,75);
        glVertex2f(v+70,75);

        glEnd();
        glFlush();
}
```

For creating **Road & Road-Strips**:



```
void road_strip()
{

     glColor3ub(255,229,0);
    int j =185;
    for(int i=535;i<=1000;i+=80){
        glBegin(GL_POLYGON);
        glColor3ub(255,229,0);
        glVertex2f(j,i);
        glVertex2f(j+5,i);
        glVertex2f(j+15,i+50);
        glVertex2f(j+20,i+50);
        glEnd();
    j+=35;
    }

    for(int i=10;i<=1800;i+=100){
        glBegin(GL_POLYGON);
        glColor3ub(255,229,0);
        glVertex2f(i,505);
        glVertex2f(i,510);
        glVertex2f(i+50,510);
        glVertex2f(i+50,505);
        glEnd();
    }
}
```

For creating **River**:

```
void river()
{
     glColor3ub(93,70,69);
    glBegin(GL_POLYGON);
    glVertex2f(0,0);
    glVertex2f(0,240);
    glVertex2f(200,260);
    glVertex2f(400,240);
    glVertex2f(700,250);
    glVertex2f(900,260);
    glVertex2f(1000,250);
    glVertex2f(1200,270);
    glVertex2f(1500,240);
    glVertex2f(1600,250);
    glVertex2f(1800,240);
    glVertex2f(1800,0);
    glEnd();
```

```
 glColor3ub(58,100,200);
    glBegin(GL_POLYGON);
    glVertex2f(0,0);
    glVertex2f(0,210);
    glVertex2f(200,230);
    glVertex2f(400,210);
    glVertex2f(700,220);
    glVertex2f(900,230);
    glVertex2f(1000,210);
    glVertex2f(1200,240);
    glVertex2f(1500,210);
    glVertex2f(1600,210);
    glVertex2f(1800,230);
    glVertex2f(1800,0);
    glEnd();
}
```

For creating **Girl (Passenger):**

## Drawing hands:

```
GLvoid hand(void)
{
    glLineWidth(4);
    glBegin(GL_LINE_STRIP);
    glColor3d(1,.8,.5);
    glVertex2f(.2,.05);
    glVertex2f(.3,-.2);
    glVertex2f(.2,-.4);
    glEnd();

    glLineWidth(4);
    glBegin(GL_LINE_STRIP);
    glColor3d(1,.8,.5);
    glVertex2f(-.2,.05);
    glVertex2f(-.3,-.2);
    glVertex2f(-.2,-.4);
    glEnd();
}
```

## Drawing legs:

```
GLvoid gleg(void)
{
    glLineWidth(6);
    glBegin(GL_LINE_STRIP);
    glColor3d(1,.8,.5);
    glVertex2f(-.2,-0.8);
    glVertex2f(-.4,-1.4);
    glEnd();

    glLineWidth(6);
    glBegin(GL_LINE_STRIP);
    glColor3d(1,.8,.5);
    glVertex2f(.2,-0.8);
    glVertex2f(.4,-1.4);
    glEnd();
}
```

## Drawing dress:

```
GLvoid girldress(void)
{
    gleg();
    glBegin(GL_POLYGON);
    glColor3d(.2,0,.2);
    glVertex2f(-.2,.05);
    glVertex2f(.2,.05);
```

```
GLvoid girl(void)
{

    glPushMatrix();
    glTranslated(-4,3.4,0);
    girldress();
    glPopMatrix();

    glPushMatrix();
    glTranslated(-4,4,0);
    glColor3d(1,.8,.5);
    circle_for_girl(.25);
    glColor3d(0,0,0);
    glRotated(65,0,0,1);
    glTranslated(.1,.08,0);
    semi_circle_for_girl(.25);
    glRotated(65,0,0,1);
    semi_circle_for_girl(.25);
    glTranslated(-.15,0,0);
    circle_for_girl(.03);
    glTranslated(.11,.1,0);
    circle_for_girl(.03);
    glTranslated(-.13,.01,0);
    circle_for_girl(.0185);
    glColor3d(1,0,0);
    glTranslated(-.09,.07,0);
    glColor3d(1,0,0.2);
    circle_for_girl(.033);
    glColor3d(1,.8,.5);
    glTranslated(-.05,.07,0);
    circle_for_girl(.05);
    glPopMatrix();

}
```

```
//circle for girl face and other thing
GLvoid circle_for_girl(GLdouble rad)
{
       GLint points = 500;
       GLdouble delTheta = (2.0 * PI) /
(GLdouble)points;
       GLdouble theta = 0;

GLint i=0;
       glBegin(GL_POLYGON);
       {
              for(i = 0; i <=500; i++, theta +=
delTheta )
              {

                     glVertex2f(rad * cos(theta),rad
* sin(theta));
              }
       }
       glEnd();
}
```

```
    glVertex2f(.3,-1);
    glVertex2f(-.3,-1);
    glEnd();
    glPushMatrix();
    glColor3d(.8,0,.4);
    glRotated(90,0,0,1);
    semi_circle_for_girl(.33);
    glPopMatrix();

    glLineWidth(3);
    glBegin(GL_LINE_STRIP);
    glVertex2f(0,0);
    glVertex2f(0,-.8);
    glVertex2f(0,0);
    glVertex2f(-.1,-.7);
    glVertex2f(0,0);
    glVertex2f(.1,-.7);
    glEnd();

    glPushMatrix();
    glColor3d(0,.2,0);
    glTranslated(-.4,-1.35,0);
    circle_for_girl(.09);
    glTranslated(.8,0,0);
    circle_for_girl(.09);
    glPopMatrix();

    ghand();

    glEnd();

}
```

```
//not proper circle for girl hair;
GLvoid semi_circle_for_girl(GLdouble rad)
{
       GLint points = 110;
       GLdouble delTheta = (PI) / (GLdouble)points;
       GLdouble theta = 30;

    GLint i=0;
       glBegin(GL_POLYGON);
       {

              for(i = 0; i <=100; i++, theta +=
delTheta )
              {

            glVertex2d(rad *cos(theta),rad *
sin(theta));

              }
       }
       glEnd();
}
```

## First Window

Bus stop is drawn in First Window.

Road, river, road-strips, trees, Lamps and Buildings are also drawn in this window to create surrounding environment.

```
void first_window()
{


    road();
    road_strip();
    busstop();

       //tree
       glPushMatrix();
    glTranslatef(130,760,0);
    tree();
       glPopMatrix();


       //printing grass
       for(int i=-100;i<=1800;i+=150){
        glPushMatrix();
       // glTranslatef(i,150,0);
       // grass();
        glTranslatef(i, 250, 0);
        tree();
        glPopMatrix();
       }

       glPushMatrix();
```

Only if the bus is in-front of the bus-stop then only passenger girl can get in the bus. previously we declared global variable “girl_in” which is false initially to check if the girl is inside the bus or not.
If girl gets inside the bus by pressing “I” key on the keyboard then girl_in will be made true,

```
 glTranslatef(750,650,0);
    tree();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(900,650,0);
    tree();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(1050,650,0);
    tree();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(1700,650,0);
    tree();
       glPopMatrix();


       glPushMatrix();
    glTranslatef(1600,465,0);
    grass();
    glPopMatrix();

       glPushMatrix();
    glTranslatef(210,30,0);
    lamp();
       glPopMatrix();

    glPushMatrix();
    glTranslatef(1000, 30, 0);
    lamp();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(1150, 630, 0);
    building2_translate();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(1400,630,0);
     building2_translate();
       glPopMatrix();

       river();


       // if bus infront of bus_stop
        if(!girl_in){
         glPushMatrix();
        glTranslatef(740,520, 0);
        glScalef(43,45,0);
        girl();
        glPopMatrix();
        }
}
```

## Second Window

Similarly second window is created where destination of girl is situated.

```
void second_window()
{
    //road
    glBegin(GL_POLYGON);
    glVertex2f(0,400);
    glVertex2f(1800,400);
    glVertex2f(1800,600);
    glVertex2f(0,600);
    glEnd();

    glColor3ub(255,229,0);


    //second window road strip
    for(int i=10;i<=1800;i+=100){
        glBegin(GL_POLYGON);
        glColor3ub(255,229,0);
        glVertex2f(i,505);
        glVertex2f(i,510);
        glVertex2f(i+50,510);
        glVertex2f(i+50,505);
        glEnd();
    }


    // printing grass
    for(int i=-
50;i<=1450;i+=250){
        glPushMatrix();
        glTranslatef(i,150,0);
        grass();
        glPopMatrix();
    }


    //tree
       glPushMatrix();
    glTranslatef(130,720,0);
    tree();
       glPopMatrix();


       glPushMatrix();
    glTranslatef(870,720,0);
    tree();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(950,470,0);
    grass();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(-50,470,0);
    grass();
       glPopMatrix();
```

```
glPushMatrix();
    glTranslatef(450,470,0);
    grass();
       glPopMatrix();

       glPushMatrix();
    glTranslatef(700,470,0);
    grass();
       glPopMatrix();


       //printing trees

       for(int i=200;i<=1400;i+=400){

        glPushMatrix();
        glTranslatef(i,220,0);
        tree();
        glPopMatrix();
       }


    glPushMatrix();
    glTranslatef(310,480,0);
     building_abd_scal();
       glPopMatrix();

    glPushMatrix();
    glTranslatef(1100,430,0);
    building_scal();
       glPopMatrix();


       glPushMatrix();
    glTranslatef(200,635,0);
     building3_scal();
       glPopMatrix();
    glPushMatrix();
    glTranslatef(460, 635, 0);
    building3_scal();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(720, 635, 0);
    building3_scal();
    glPopMatrix();

       //if bus infront of apartment
       if(girl_out){
        glPushMatrix();
        glTranslatef(1640,520, 0);
        glScalef(43,45,0);
        girl();
        glPopMatrix();
        girl_in = true;
        }
```

```
        glPushMatrix();
    glTranslatef(100,640, 0);

        glScalef(30,30,0);
        glTranslatef(var_car2,0,0);
    drawCart();
    glPopMatrix();

        river();



    glPushMatrix();
        glTranslatef(500,110, 0);
        glScalef(.7,.7,0);
        glTranslatef(var_boat2,0, 0);
    boat();
        glPopMatrix();

}
```

## Display and Main Function

```
//display
void display()
{

    glClear(GL_COLOR_BUFFER_BIT);
    glColor3ub(30,40,60);

    if(window==1){
        glClear(GL_COLOR_BUFFER_BIT);
        glutIdleFunc(NULL);
        second_window();
        glutIdleFunc(boat_moving2);

        glutPostRedisplay();
        if(tx<=0){
            window=0;
            tx=1750;
        }
        else if(tx>1800){
            window=0;
            tx=0;
        }
    }
    else{
        glClear(GL_COLOR_BUFFER_BIT);
        first_window();

        glutPostRedisplay();
        if(tx>=1790){
             glutIdleFunc(NULL);
            window=1;
            glutPostRedisplay();
            tx=10;
        }
    }
```

```
    glPushMatrix();
        glTranslatef(tx,ty,0);
    bus_final();
        glPopMatrix();


    glFlush();
    glutSwapBuffers();

}


//main
int main()
{

    glutInitDisplayMode(GLUT_DOUBLE |
GLUT_RGB);
    glutInitWindowSize(1800,1000);
    glutInitWindowPosition(0,0);
    glutCreateWindow("COE18 Project
'The Bus Simulator'");
    init();
    glutDisplayFunc(display);
    glutSpecialFunc(spe_key);
    glutKeyboardFunc(my_keyboard);
    glutMainLoop();
    return 0;
}
```

# Functions to make interface interactive

**Keys on keyboard to trigger some functions:**

**PRESS d/D** for changing night to day

**PRESS n/N** for changing day to night

If bus is in-front of the Bus stop:
**PRESS i/I** to get inside the bus

If bus is outside of the Apartment buildings:
**PRESS o/O** to get outside the bus

```
void my_keyboard(unsigned char key, int x, int
y)
{

       switch (key) {
          //for day night

             case 'd':
                 glClearColor(.01,.8,.5,0);
                    break;

             case 'n':

glClearColor(.04,.09,.02,0);
                    break;

             case 'N':
                  glClearColor(.04,.09,.02,0);
                     break;
         case 'D':
              glClearColor(.01,.8,.5,0);
         break;

         //get in the bus from bus stop
         case 'I':
             if(tx>=480 && tx<=650){
                girl_in =true;
                glutPostRedisplay();
             }
             else if(tx<480 || tx>650){
                 girl_in = false;
                glutPostRedisplay();
             }

         break;

          case 'i':
             if(tx>=480 && tx<=650){
                girl_in =true;
                glutPostRedisplay();
             }
             else if(tx<480 || tx>650){
                 girl_in = false;
                glutPostRedisplay();
             }

         break;
```

**PRESS right key** to make Bus move right

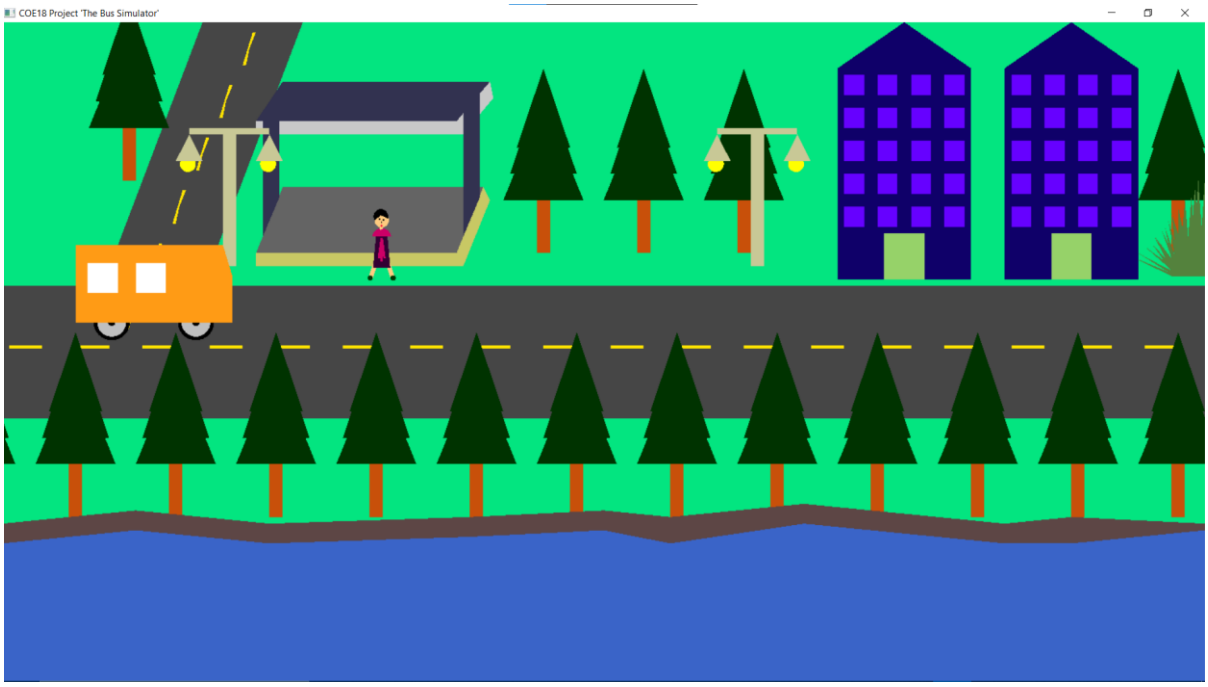
**PRESS left key** to make Bus move left

```
//get out from the bus

         case 'O':
             if(tx>=1330 && tx<=1650){
                girl_out =true;
                glutPostRedisplay();
             }
             else if(tx<1330 || tx>1650){
                 girl_out = false;
                glutPostRedisplay();
             }

         break;

          case 'o':
             if(tx>=1330 && tx<=1600){
                girl_out =true;
                glutPostRedisplay();
             }
             else if(tx<1330 || tx>1600){
                 girl_out = false;
                glutPostRedisplay();
             }

         break;


         default:
                     break;
       }
}

void spe_key(int key, int x, int y)
{

       switch (key) {

              case GLUT_KEY_LEFT:

                            tx -=10;

       glutPostRedisplay();
                            break;


              case GLUT_KEY_RIGHT:
                            tx +=10;

       glutPostRedisplay();

                            break;
         default:
                     break;
       }
}
```

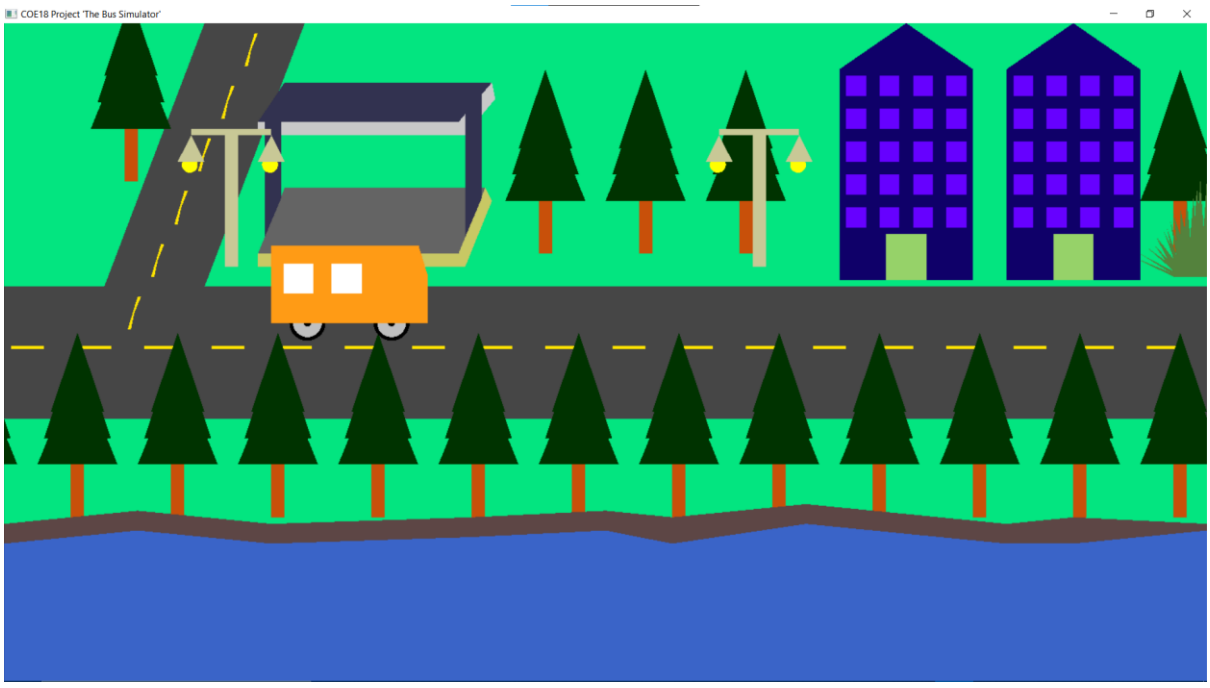
# 4. SCREENSHOTS

## First Window:

Press right key to move bus ahead.

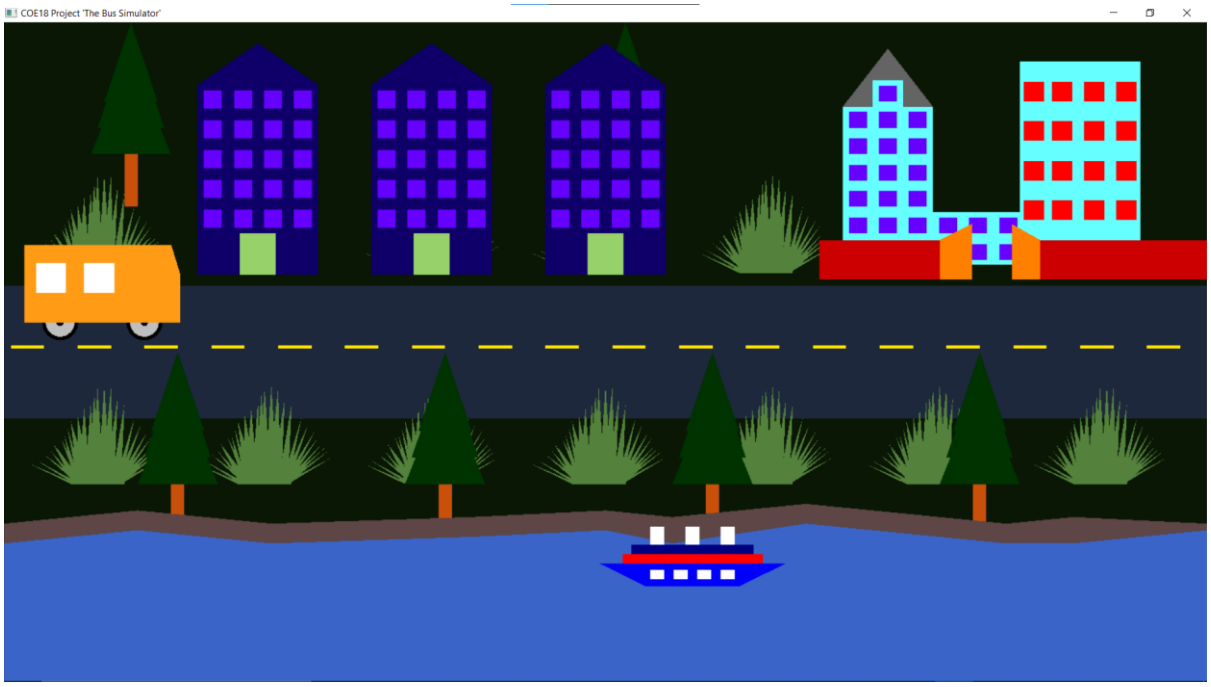



As bus is in-front of bus stop press i or I to get in the bus.



**Second Window:**

Press d or D key on keyboard to turn day into night.




Press o or O in-front of apartments to come out of the bus and reach destination.

# 5. REFERENCES


-> https://www.tutorialspoint.com/computer_graphics/index.htm

-> https://www.javatpoint.com/computer-graphics-tutorial

-> https://www.explainthatstuff.com/computer-graphics.html

-> https://www.tutorialandexample.com/computer-graphics-tutorial/