



Daffodil
International
University

Project Report

Course Title: Computer Graphics Lab

Course code: CSE 422

Submitted To:

Md. Atik Asif Khan Akash

Lecturer

Department of CSE

Daffodil International University

Submitted By:

Umme Hani Sheba (ID : 201-15-3137 D2)

Muhtasim Hossain (ID : 201-15-3444 D1)

Towhid Mohammed Alam (ID : 201-15-3655 D1)

Nayem Al Mahmud (ID : 201-15- 3110 D1)

Section : PC-D

Department of CSE

Project Code

```
#include<windows.h>
#include <GL/glut.h>
#include <stdio.h>
#include <GL/gl.h>
#include<math.h>
#define PI

float angle1 = 0.0f;

float ska=0.5,skb=1.0,skc=1.0,
wa=0.0,wb=0.8,wc=0.9,sma=1.0,smb=1.0,smc=0.0,smra=1.000, smrb=0.388,
smrc=0.278,ca=1.0,cb=1.,cc=1.0,sa=0.5,sb=1.0,sc=1.0;

float xc1Pos,yc1Pos, xc2Pos ,yc2Pos , xc3Pos ,yc3Pos , xc4Pos ,yc4Pos = 0.0;
float cloud1boundaryRight = 1.0, cloud1boundaryLeft = -0.2, cloud3boundaryRight = 0.5,
cloud3boundaryLeft = -0.8, cloud2boundaryRight = 0.9, cloud2boundaryLeft = -0.4,
cloud4boundaryRight = 0.2, cloud4boundaryLeft = -1.0;

float xbird1Pos, ybird1Pos, xbird2Pos, ybird2Pos=0.0;
float bird1boundaryRight = 1.0, bird1boundaryLeft = -0.85, bird2boundaryRight = 0.4,
bird2boundaryLeft = -1.0;

float xb1Pos, yb1Pos, xb2Pos, yb2Pos, xb3Pos, yb3Pos = 0.0;
float boat1boundaryRight = 0.68, boat1boundaryLeft = -1.0, boat2boundaryRight = 0.4,
boat2boundaryLeft = -1.0, boat3boundaryRight = 1.0, boat3boundaryLeft = -0.2;

float xv1Pos, yv1Pos, xv2Pos, yv2Pos = 0.0;
float vehicle1boundaryRight = 1.0, vehicle1boundaryLeft = -0.5, vehicle2boundaryRight =
1.0, vehicle2boundaryLeft = -0.5;

void init(void)
{
    glClearColor(0.0,0.0,0.0,0.0); //GLfloat red,green,blue,alpha initial value 0 alpha values
    used by glclear to clear the color buffers
    glMatrixMode(GL_PROJECTION); // To specify which matrix is the current matrix &
    projection applies subsequent matrix to projection matrix stack
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
    //gluOrtho2D(0.0,300.0,0.0,300.0); // Orthographic representation; multiply the current
    matrix by an orthographic matrix 2D= left right,bottom,top equivalent near=-1,far=1
}
```

```

void Draw()
{
    glClear(GL_COLOR_BUFFER_BIT);
    //Write your code here

    glLoadIdentity();
    glBegin(GL_QUADS); //sky
    glColor3d(ska,skb,skc);
    glVertex2d(0.0,0.6875);
    glVertex2d(0.0,1.0);
    glVertex2d(1.0,1.0);
    glVertex2d(1.0,0.6875);
    glEnd();

    glLoadIdentity();
    glTranslatef(0.05,0.95,0);
    glRotatef(angle1, 0.0f, 0.0f, 1.0f);
    glTranslatef(-0.05,-0.95,0);

    glColor3f(smra,s mrb,smrc); //sun ray
    glBegin(GL_TRIANGLES);
    glVertex2d(0.0, 0.92);
    glVertex2d(0.05, 1.01);
    glVertex2d(0.1, 0.92);
    glEnd();

    glLoadIdentity();
    glTranslatef(0.05,0.95,0);
    glRotatef(angle1, 0.0f, 0.0f, 1.0f);
    glTranslatef(-0.05,-0.95,0);
    glColor3d(smra,s mrb,smrc); //sun ray 2
    glBegin(GL_TRIANGLES);
    glVertex2d(0.0, 0.98);
    glVertex2d(0.05, 0.89);
    glVertex2d(0.1, 0.98);
    glEnd();

    glLoadIdentity();

    int i;
    GLfloat x=0.05, y=0.95, radius =0.04 ;
    int triangleAmount = 50; //# of triangles used to draw circle
    GLfloat twicePi = 2.0 * M_PI;
    glBegin(GL_TRIANGLE_FAN);

```

```

glColor3f(sma,smb,smc); // set color to red
glVertex2f(x, y); // center of circle
for(i = 0; i <= triangleAmount; i++)
{
    glVertex2f(
        x + (radius * cos(i * twicePi / triangleAmount)),
        y + (radius * sin(i * twicePi / triangleAmount))
    );
}
glEnd();

```

```

glLoadIdentity();
glTranslatef(xc1Pos, yc1Pos, 0.0);
glBegin(GL_POLYGON); // Cloud 1
glColor3d(ca,cb,cc);
glVertex2d(0.1,0.925);
glVertex2d(0.1,0.9375);
glVertex2d(0.11,0.95);
glVertex2d(0.125,0.95);
glVertex2d(0.135,0.95625);
glVertex2d(0.16,0.95625);
glVertex2d(0.17,0.95);
glVertex2d(0.185,0.95);
glVertex2d(0.195,0.9375);
glVertex2d(0.195,0.925);
glEnd();
glLoadIdentity();

```

```

glTranslatef(xc2Pos, yc2Pos, 0.0);
glBegin(GL_POLYGON); // Cloud 2
glColor3d(ca,cb,cc);
glVertex2d(0.3,0.925);
glVertex2d(0.3,0.9375);
glVertex2d(0.31,0.95);
glVertex2d(0.325,0.95);
glVertex2d(0.335,0.95625);
glVertex2d(0.36,0.95625);
glVertex2d(0.37,0.95);
glVertex2d(0.385,0.95);
glVertex2d(0.395,0.9375);
glVertex2d(0.395,0.925);
glEnd();
glLoadIdentity();

```

```

glTranslatef(xc3Pos, yc3Pos, 0.0);
glBegin(GL_POLYGON); // Cloud 3

```

```
glColor3d(ca,cb,cc);
glVertex2d(0.6,0.925);
glVertex2d(0.6,0.9375);
glVertex2d(0.61,0.95);
glVertex2d(0.625,0.95);
glVertex2d(0.635,0.95625);
glVertex2d(0.66,0.95625);
glVertex2d(0.67,0.95);
glVertex2d(0.685,0.95);
glVertex2d(0.695,0.9375);
glVertex2d(0.695,0.925);
glEnd();
glLoadIdentity();
```

```
glTranslatef(xc4Pos, yc4Pos, 0.0);
glBegin(GL_POLYGON); // Cloud 4
glColor3d(ca,cb,cc);
glVertex2d(0.8,0.925);
glVertex2d(0.8,0.9375);
glVertex2d(0.81,0.95);
glVertex2d(0.825,0.95);
glVertex2d(0.835,0.95625);
glVertex2d(0.86,0.95625);
glVertex2d(0.87,0.95);
glVertex2d(0.885,0.95);
glVertex2d(0.895,0.9375);
glVertex2d(0.895,0.925);
glEnd();
glLoadIdentity();
```

```
glBegin(GL_TRIANGLES); //mountain 1
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.0,0.687);
glVertex2d(0.16,0.8125);
glVertex2d(0.75,0.6875);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 2
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.17,0.675);
glVertex2d(0.3,0.8125);
glVertex2d(0.595,0.68125);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 3
```

```
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.345,0.78);
glVertex2d(0.4,0.9);
glVertex2d(0.7,0.7625);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 4
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.255,0.675);
glVertex2d(0.5,0.9375);
glVertex2d(0.75,0.68125);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 5
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.5,0.85);
glVertex2d(0.64,0.875);
glVertex2d(0.9,0.5);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 6
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.235,0.675);
glVertex2d(0.75,0.8125);
glVertex2d(0.95,0.68125);
glEnd();
```

```
glLoadIdentity();
glBegin(GL_TRIANGLES); //mountain 7
glColor3d(0.000, 0.502, 0.502);
glVertex2d(0.78,0.7625);
glVertex2d(0.965,0.79375);
glVertex2d(1.2,0.25);
glEnd();
```

```
glLoadIdentity();
glTranslatef(xbird1Pos, ybird1Pos, 0.0);
glBegin(GL_LINES); //bird1 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.13,0.875);
glVertex2d(0.14,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird1 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.14,0.8875);
glVertex2d(0.145,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird1 3
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.145,0.875);
glVertex2d(0.15,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird1 4
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.15,0.8875);
glVertex2d(0.155,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird2 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.16,0.8875);
glVertex2d(0.17,0.9);
glEnd();
```

```
glBegin(GL_LINES); //bird2 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.17,0.9);
glVertex2d(0.175,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird2 3
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.175,0.8875);
glVertex2d(0.18,0.9);
glEnd();
```

```
glBegin(GL_LINES); //bird2 4
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.18,0.9);
glVertex2d(0.185,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird3 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.16,0.8625);
glVertex2d(0.17,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird3 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.17,0.875);
glVertex2d(0.175,0.8625);
glEnd();
```

```
glBegin(GL_LINES); //bird3 3
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.175,0.8625);
glVertex2d(0.18,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird3 4
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.18,0.875);
glVertex2d(0.185,0.8625);
glEnd();
```

```
glLoadIdentity();
glTranslatef(xbird2Pos, ybird2Pos, 0.0);
glBegin(GL_LINES); //bird4 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.78,0.875);
glVertex2d(0.79,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird4 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.79,0.8875);
glVertex2d(0.795,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird4 3
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.795,0.875);
glVertex2d(0.8,0.8875);
glEnd();
```



```
glBegin(GL_LINES); //bird4 4
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.80,0.8875);
glVertex2d(0.805,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird5 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.81,0.8625);
glVertex2d(0.82,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird5 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.82,0.875);
glVertex2d(0.825,0.8625);
glEnd();
```

```
glBegin(GL_LINES); //bird5 3
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.825,0.8625);
glVertex2d(0.83,0.875);
glEnd();
```

```
glBegin(GL_LINES); //bird5 4
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.83,0.875);
glVertex2d(0.835,0.8625);
glEnd();
```

```
glBegin(GL_LINES); //bird6 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.81,0.8875);
glVertex2d(0.82,0.9);
glEnd();
```

```
glBegin(GL_LINES); //bird6 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.82,0.9);
glVertex2d(0.825,0.8875);
glEnd();
```

```
glBegin(GL_LINES); //bird6 3
glColor3d(0.0, 0.0, 0.0);
```

```
glVertex2d(0.825,0.8875);  
glVertex2d(0.83,0.9);  
glEnd();
```

```
glBegin(GL_LINES); //bird6 4  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.83,0.9);  
glVertex2d(0.835,0.8875);  
glEnd();
```

```
glLoadIdentity();
```

```
glBegin(GL_QUADS); //Greenery background  
glColor3d(0.0, 0.6, 0.0);  
glVertex2d(0.0,0.525);  
glVertex2d(0.0,0.6875);  
glVertex2d(1.0,0.6875);  
glVertex2d(1.0,0.525);  
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //apatment road shadow  
glColor3d(0.5, 0.0, 0.0);  
glVertex2d(0.0,0.525);  
glVertex2d(0.0,0.5375);  
glVertex2d(0.25,0.5375);  
glVertex2d(0.28,0.525);  
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //apatment road  
glColor3d(0.8, 0.0, 0.0);
```

```
glVertex2d(0.0,0.5375);  
glVertex2d(0.0,0.575);  
glVertex2d(0.27,0.575);  
glVertex2d(0.25,0.5375);  
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //apatment road slide  
glColor3d(0.8, 0.0, 0.0);  
glVertex2d(0.25,0.5375);  
glVertex2d(0.27,0.575);  
glVertex2d(0.3,0.5625);  
glVertex2d(0.28,0.525);
```

```
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //School road shadow  
glColor3d(0.5, 0.0, 0.0);  
glVertex2d(1.0,0.525);  
glVertex2d(1.0,0.5375);  
glVertex2d(0.75,0.5375);  
glVertex2d(0.7,0.525);  
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //school road  
glColor3d(0.8, 0.0, 0.0);
```

```
glVertex2d(1.0,0.5375);  
glVertex2d(1.0,0.575);  
glVertex2d(0.73,0.575);  
glVertex2d(0.75,0.5375);  
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //school road slide  
glColor3d(0.8, 0.0, 0.0);  
glVertex2d(0.75,0.5375);  
glVertex2d(0.73,0.575);  
glVertex2d(0.7,0.5625);  
glVertex2d(0.72,0.525);
```

```
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //hospital road front  
glColor3d(0.5, 0.5, 0.5);
```

```
glVertex2d(0.28,0.525);  
glVertex2d(0.3,0.5625);  
glVertex2d(0.7,0.5625);  
glVertex2d(0.72,0.525);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital stare 1  
glColor3d(0.0,0.0,0.7);
```

```
glVertex2d(0.3,0.5625);  
glVertex2d(0.3,0.580);  
glVertex2d(0.7,0.580);  
glVertex2d(0.7,0.5625);
```

```
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //hospital stare 2  
glColor3d(0.0,0.0,0.7);
```

```
glVertex2d(0.33,0.580);  
glVertex2d(0.33,0.595);  
glVertex2d(0.67,0.595);  
glVertex2d(0.67,0.580);
```

```
glEnd();
```

```
glLoadIdentity();  
glBegin(GL_QUADS); //Hospital Pillar 1  
glColor3d(1.0, 1.0, 1.0);
```

```
glVertex2d(0.335,0.595);  
glVertex2d(0.335,0.75);  
glVertex2d(0.345,0.75);  
glVertex2d(0.345,0.595);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //Hospital Pillar 2  
glColor3d(1.0, 1.0, 1.0);
```

```
glVertex2d(0.655,0.595);  
glVertex2d(0.655,0.75);  
glVertex2d(0.665,0.75);  
glVertex2d(0.665,0.595);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital roof
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.3,0.75);
glVertex2d(0.3,0.7625);
glVertex2d(0.7,0.7625);
glVertex2d(0.7,0.75);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital
glColor3d(0.8, 0.8, 0.8);
glVertex2d(0.35,0.595);
glVertex2d(0.35,0.75);
glVertex2d(0.648,0.75);
glVertex2d(0.648,0.595);
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital vertical
glColor3d(0.9, 0.9, 0.9);
glVertex2d(0.44,0.595);
glVertex2d(0.44,0.7875);
glVertex2d(0.56,0.7875);
glVertex2d(0.56,0.595);
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital small head
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.43,0.7875);
glVertex2d(0.43,0.8);
glVertex2d(0.57,0.8);
glVertex2d(0.57,0.7875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital large head
glColor3d(0.9, 0.9, 0.9);
glVertex2d(0.42,0.8);
glVertex2d(0.42,0.8125);
glVertex2d(0.58,0.8125);
glVertex2d(0.58,0.8);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital Plus box
```

```
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.45,0.6875);
glVertex2d(0.45,0.775);
glVertex2d(0.55,0.775);
glVertex2d(0.55,0.6875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital Plus 1
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.495,0.69375);
glVertex2d(0.495,0.76);
glVertex2d(0.505,0.76);
glVertex2d(0.505,0.69375);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //main bg hospital Plus 2
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.47,0.71875);
glVertex2d(0.47,0.73125);
glVertex2d(0.53,0.73125);
glVertex2d(0.53,0.71875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital window white
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.355,0.66875);
glVertex2d(0.355,0.74);
glVertex2d(0.435,0.74);
glVertex2d(0.435,0.66875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital 1st window line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.394,0.66875);
glVertex2d(0.394,0.74);
glVertex2d(0.396,0.74);
glVertex2d(0.396,0.66875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital window 2nd white
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.355,0.66);
glVertex2d(0.355,0.59475);
glVertex2d(0.435,0.59475);
glVertex2d(0.435,0.66);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital 2nd window line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.394,0.66);
glVertex2d(0.394,0.59475);
glVertex2d(0.396,0.59475);
glVertex2d(0.396,0.66);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital window 3rd white
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.565,0.66875);
glVertex2d(0.565,0.74);
glVertex2d(0.647,0.74);
glVertex2d(0.647,0.66875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); // hospital 3rd window line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.602,0.66875);
glVertex2d(0.602,0.74);
glVertex2d(0.604,0.74);
glVertex2d(0.604,0.66875);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital window 4th white
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.565,0.66);
glVertex2d(0.565,0.59475);
glVertex2d(0.647,0.59475);
glVertex2d(0.647,0.66);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital window 4th window line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.602,0.66);
glVertex2d(0.602,0.59475);
glVertex2d(0.604,0.59475);
glVertex2d(0.604,0.66);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital divider
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.45,0.6625);
glVertex2d(0.45,0.675);
glVertex2d(0.55,0.675);
glVertex2d(0.55,0.6625);
```

```
glEnd();
```

```
glBegin(GL_QUADS); // hospital door roof
glColor3d(0.7, 0.7, 0.7);
glVertex2d(0.46,0.6425);
glVertex2d(0.46,0.655);
glVertex2d(0.54,0.655);
glVertex2d(0.54,0.6425);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital door 1st pillar
glColor3d(0.7, 0.7, 0.7);
glVertex2d(0.465,0.595);
glVertex2d(0.465,0.6475);
glVertex2d(0.475,0.6475);
glVertex2d(0.475,0.595);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital door 2nd pillar
glColor3d(0.7, 0.7, 0.7);
glVertex2d(0.527,0.595);
glVertex2d(0.527,0.6475);
glVertex2d(0.537,0.6475);
glVertex2d(0.537,0.595);
```

```
glEnd();
```



```
glBegin(GL_QUADS); //hospital door
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.478,0.595);
glVertex2d(0.478,0.6375);
glVertex2d(0.522,0.6375);
glVertex2d(0.522,0.595);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //hospital door line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.498,0.59);
glVertex2d(0.498,0.637);
glVertex2d(0.5,0.637);
glVertex2d(0.5,0.59);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main
glColor3d(1.0, 1.0, 0.0);
glVertex2d(0.02,0.575);
glVertex2d(0.02,0.8375);
glVertex2d(0.16,0.8375);
glVertex2d(0.16,0.575);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main roof
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.02,0.825);
glVertex2d(0.02,0.8375);
glVertex2d(0.16,0.8375);
glVertex2d(0.16,0.825);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main 1st L window
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.03,0.7625);
glVertex2d(0.03,0.8);
glVertex2d(0.15,0.8);
glVertex2d(0.15,0.7625);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main 2nd L window
```

```
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.03,0.7125);
glVertex2d(0.03,0.75);
glVertex2d(0.15,0.75);
glVertex2d(0.15,0.7125);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main divider
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.02,0.68125);
glVertex2d(0.02,0.69375);
glVertex2d(0.16,0.69375);
glVertex2d(0.16,0.68125);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment main 1st S window
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.03,0.5875);
glVertex2d(0.03,0.6625);
glVertex2d(0.05,0.6625);
glVertex2d(0.05,0.5875);
```

```
glEnd();
```

```
glBegin(GL_LINES); //apartment main 1st S window Line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.03,0.625);
glVertex2d(0.05,0.625);
glEnd();
```

```
glBegin(GL_QUADS); //apartment main 2nd S window
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.13,0.5875);
glVertex2d(0.13,0.6625);
glVertex2d(0.15,0.6625);
glVertex2d(0.15,0.5875);
glEnd();
```

```
glBegin(GL_LINES); //apartment main 2nd S window Line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.13,0.625);
glVertex2d(0.15,0.625);
glEnd();
```

```
glBegin(GL_QUADS); //apartment main door
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.075,0.575);
glVertex2d(0.075,0.6375);
glVertex2d(0.115,0.6375);
glVertex2d(0.115,0.575);
glEnd();
```

```
glBegin(GL_LINES); //apartment main door line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.095,0.575);
glVertex2d(0.095,0.6375);
glEnd();
```

```
glBegin(GL_QUADS); //apartment secondary
glColor3d(0.95, 0.95, 0.0);
```

```
glVertex2d(0.16,0.575);
glVertex2d(0.16,0.7375);
glVertex2d(0.24,0.7375);
glVertex2d(0.24,0.575);
```

```
glEnd();
```

```
glBegin(GL_QUADS); //apartment secondary small roof
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.16,0.7375);
glVertex2d(0.16,0.75);
glVertex2d(0.24,0.75);
glVertex2d(0.24,0.7375);
glEnd();
```

```
glBegin(GL_QUADS); //apartment secondary 1st L window
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.17,0.7);
glVertex2d(0.17,0.725);
glVertex2d(0.23,0.725);
glVertex2d(0.23,0.7);
glEnd();
```

```
glBegin(GL_QUADS); //apartment secondary 2nd L window
glColor3d(0.0, 0.8, 0.8);
```

```
glVertex2d(0.17,0.65);  
glVertex2d(0.17,0.675);  
glVertex2d(0.23,0.675);  
glVertex2d(0.23,0.65);  
glEnd();
```

```
glBegin(GL_QUADS); //apartment secondary 3rd L window  
glColor3d(0.0, 0.8, 0.8);  
glVertex2d(0.17,0.6);  
glVertex2d(0.17,0.625);  
glVertex2d(0.23,0.625);  
glVertex2d(0.23,0.6);  
glEnd();
```

```
glBegin(GL_QUADS); //School main  
glColor3d(1.0, 0.8, 0.0);  
glVertex2d(0.73,0.575);  
glVertex2d(0.73,0.7125);  
glVertex2d(0.91,0.7125);  
glVertex2d(0.91,0.575);  
glEnd();
```

```
glBegin(GL_QUADS); //School main roof  
glColor3d(0.7, 0.0, 0.0);  
glVertex2d(0.70,0.7125);  
glVertex2d(0.75,0.7625);  
glVertex2d(0.93,0.7625);  
glVertex2d(0.93,0.7125);  
glEnd();
```

```
glBegin(GL_QUADS); //School main divider  
glColor3d(1.0, 1.0, 0.0);  
glVertex2d(0.73,0.64375);  
glVertex2d(0.73,0.65625);  
glVertex2d(0.91,0.65625);  
glVertex2d(0.91,0.64375);  
glEnd();
```

```
glBegin(GL_QUADS); //School main window 1  
glColor3d(0.0, 0.8, 0.8);  
glVertex2d(0.75,0.66875);  
glVertex2d(0.75,0.6875);  
glVertex2d(0.77,0.6875);  
glVertex2d(0.77,0.66875);  
glEnd();
```

```
glBegin(GL_QUADS); //School main window 2
```

```
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.79,0.66875);
glVertex2d(0.79,0.6875);
glVertex2d(0.81,0.6875);
glVertex2d(0.81,0.66875);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 3
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.83,0.66875);
glVertex2d(0.83,0.6875);
glVertex2d(0.85,0.6875);
glVertex2d(0.85,0.66875);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 4
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.87,0.66875);
glVertex2d(0.87,0.6875);
glVertex2d(0.89,0.6875);
glVertex2d(0.89,0.66875);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 5
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.75,0.60625);
glVertex2d(0.75,0.625);
glVertex2d(0.77,0.625);
glVertex2d(0.77,0.60625);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 6
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.79,0.60625);
glVertex2d(0.79,0.625);
glVertex2d(0.81,0.625);
glVertex2d(0.81,0.60625);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 7
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.83,0.60625);
glVertex2d(0.83,0.625);
glVertex2d(0.85,0.625);
glVertex2d(0.85,0.60625);
glEnd();
```

```
glBegin(GL_QUADS); //School main window 8
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.87,0.60625);
glVertex2d(0.87,0.625);
glVertex2d(0.89,0.625);
glVertex2d(0.89,0.60625);
glEnd();
```

```
glBegin(GL_QUADS); //School secondary
glColor3d(1.0, 0.5, 0.0);
glVertex2d(0.91,0.575);
glVertex2d(0.91,0.725);
glVertex2d(0.98,0.725);
glVertex2d(0.98,0.575);
glEnd();
```

```
glBegin(GL_TRIANGLES); //School secondary Large roof
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.89,0.725);
glVertex2d(0.945,0.785);
glVertex2d(1.0,0.725);
glEnd();
```

```
glBegin(GL_TRIANGLES); //School secondary small roof
glColor3d(1.0, 0.5, 0.0);
glVertex2d(0.91,0.725);
glVertex2d(0.945,0.7625);
glVertex2d(0.98,0.725);
glEnd();
```

```
glBegin(GL_QUADS); //School secondary small window
glColor3d(0.0, 0.8, 0.8);
glVertex2d(0.925,0.6875);
glVertex2d(0.925,0.7125);
glVertex2d(0.965,0.7125);
glVertex2d(0.965,0.6875);
glEnd();
```

```
glBegin(GL_LINES); //School secondary small window 1st line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.945,0.6875);
glVertex2d(0.945,0.7125);
glEnd();
```

```
glBegin(GL_LINES); //School secondary small window 2nd line
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.925,0.7);
```

```
glVertex2d(0.965,0.7);  
glEnd();
```

```
glBegin(GL_QUADS); //School secondary divider  
glColor3d(1.0, 1.0, 0.0);  
glVertex2d(0.91,0.6625);  
glVertex2d(0.91,0.675);  
glVertex2d(0.98,0.675);  
glVertex2d(0.98,0.6625);  
glEnd();
```

```
glBegin(GL_QUADS); //School secondary door  
glColor3d(0.0, 0.8, 0.8);  
glVertex2d(0.925,0.575);  
glVertex2d(0.925,0.6375);  
glVertex2d(0.965,0.6375);  
glVertex2d(0.965,0.575);  
glEnd();
```

```
glBegin(GL_LINES); //School secondary door line  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.945,0.575);  
glVertex2d(0.945,0.6375);  
glEnd();
```

```
glBegin(GL_QUADS); //grass pathway left  
glColor3d(0.0, 0.6, 0.0);  
glVertex2d(0.0,0.5);  
glVertex2d(0.0,0.525);  
glVertex2d(1.0,0.525);  
glVertex2d(1.0,0.5);  
glEnd();
```

```
glBegin(GL_QUADS); //road  
glColor3d(0.5, 0.5, 0.5);  
glVertex2d(0.0,0.375);  
glVertex2d(0.0,0.5);  
glVertex2d(1.0,0.5);
```

```
glVertex2d(1.0,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 1  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.0,0.4375);  
glVertex2d(0.05,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 2  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.10,0.4375);  
glVertex2d(0.15,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 3  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.2,0.4375);  
glVertex2d(0.25,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 4  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.3,0.4375);  
glVertex2d(0.35,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 5  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.4,0.4375);  
glVertex2d(0.45,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 6  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.5,0.4375);  
glVertex2d(0.55,0.4375);  
glEnd();
```

```
glBegin(GL_LINES); //road line 7  
glColor3d(1.0, 1.0, 1.0);  
glVertex2d(0.6,0.4375);  
glVertex2d(0.65,0.4375);  
glEnd();
```



```
glBegin(GL_LINES); //road line 8
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.7,0.4375);
glVertex2d(0.75,0.4375);
glEnd();
```

```
glBegin(GL_LINES); //road line 9
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.8,0.4375);
glVertex2d(0.85,0.4375);
glEnd();
```

```
glBegin(GL_LINES); //road line 10
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.9,0.4375);
glVertex2d(0.95,0.4375);
glEnd();
```

```
glBegin(GL_LINES); //road line 11
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.98,0.4375);
glVertex2d(1.0,0.4375);
glEnd();
```

```
glLoadIdentity();
glTranslatef(xv1Pos, yv1Pos, 0.0);
```

```
glBegin(GL_POLYGON); //car1
glColor3d(1.0, 1.0, 1.0);
glVertex2d(0.39,0.3875);
glVertex2d(0.39,0.40625);
glVertex2d(0.425,0.40625);
glVertex2d(0.435,0.425);
glVertex2d(0.485,0.425);
glVertex2d(0.5,0.40625);
glVertex2d(0.53,0.40625);
glVertex2d(0.53,0.3875);
glEnd();
```

```
glBegin(GL_QUADS); //car1 w1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.42,0.40625);
glVertex2d(0.435,0.41875);
glVertex2d(0.455,0.41875);
glVertex2d(0.455,0.40625);
glEnd();
```

```
glBegin(GL_QUADS); //car1 w2
glColor3d(0.0, 0.0, 0.0);
```

```
glVertex2d(0.46,0.40625);
glVertex2d(0.46,0.41875);
glVertex2d(0.485,0.41875);
glVertex2d(0.495,0.40625);
```

```
glEnd();
```

```
int w1;
GLfloat xw1=0.42, yw1=0.3875, radiusw1 =0.01 ;
int triangleAmountw1 = 50; // # of triangles used to draw circle
GLfloat twicePiw1 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(0.0, 0.0, 0.0); // set color to red
glVertex2f(xw1, yw1); // center of circle
for(w1 = 0; w1 <= triangleAmountw1; w1++)
{
    glVertex2f(
        xw1 + (radiusw1 * cos(w1 * twicePiw1 / triangleAmountw1)),
        yw1 + (radiusw1 * sin(w1 * twicePiw1 / triangleAmountw1))
    );
}
glEnd();
```

```
int w11;
GLfloat xw11=0.42, yw11=0.3875, radiusw11 =0.005 ;
int triangleAmountw11 = 50; // # of triangles used to draw circle
GLfloat twicePiw11 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(1.0, 1.0, 1.0); // set color to red
glVertex2f(xw11, yw11); // center of circle
for(w11 = 0; w11 <= triangleAmountw11; w11++)
{
    glVertex2f(
        xw11 + (radiusw11 * cos(w11 * twicePiw11 / triangleAmountw11)),
        yw11 + (radiusw11 * sin(w11 * twicePiw11 / triangleAmountw11))
    );
}
glEnd();
```

```
int w2;
GLfloat xw2=0.495, yw2=0.3875, radiusw2 =0.01 ;
int triangleAmountw2 = 50; // # of triangles used to draw circle
```

```

GLfloat twicePiw2 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(0.0, 0.0, 0.0); // set color to red
glVertex2f(xw2, yw2); // center of circle
for(w2 = 0; w2 <= triangleAmountw2; w2++)
{
    glVertex2f(
        xw2 + (radiusw2 * cos(w2 * twicePiw2 / triangleAmountw1)),
        yw2 + (radiusw2 * sin(w2 * twicePiw2 / triangleAmountw1))
    );
}
glEnd();

int w22;
GLfloat xw22=0.495, yw22=0.3875, radiusw22 =0.005 ;
int triangleAmountw22 = 50; // # of triangles used to draw circle
GLfloat twicePiw22 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(1.0, 1.0, 1.0); // set color to red
glVertex2f(xw22, yw22); // center of circle
for(w22 = 0; w22 <= triangleAmountw22; w22++)
{
    glVertex2f(
        xw22 + (radiusw22 * cos(w22 * twicePiw22 / triangleAmountw22)),
        yw22 + (radiusw22 * sin(w22 * twicePiw22 / triangleAmountw22))
    );
}
glEnd();

glLoadIdentity();
glTranslatef(xv2Pos, yv2Pos, 0.0);

glBegin(GL_POLYGON); //car2
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.38,0.455);
glVertex2d(0.38,0.47375);
glVertex2d(0.425,0.47375);
glVertex2d(0.435,0.4925);
glVertex2d(0.485,0.4925);
glVertex2d(0.5,0.47375);
glVertex2d(0.53,0.47375);
glVertex2d(0.53,0.455);

glEnd();

```

```

glBegin(GL_QUADS); //car2 w1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.415,0.47375);
glVertex2d(0.435,0.488);
glVertex2d(0.455,0.488);
glVertex2d(0.455,0.47375);
glEnd();

```

```

glBegin(GL_QUADS); //car2 w2
glColor3d(0.0, 0.0, 0.0);

```

```

glVertex2d(0.46,0.47375);
glVertex2d(0.46,0.488);
glVertex2d(0.485,0.488);
glVertex2d(0.495,0.47375);

```

```

glEnd();

```

```

int ww1; //second car wheel means ww
GLfloat xww1=0.42, yww1=0.455, radiusww1 =0.01 ;
int triangleAmountww1 = 50; // # of triangles used to draw circle
GLfloat twicePiww1 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(0.0, 0.0, 0.0); // set color to black
glVertex2f(xww1, yww1); // center of circle
for(ww1 = 0; ww1 <= triangleAmountww1; ww1++)
{
    glVertex2f(
        xww1 + (radiusww1 * cos(ww1 * twicePiww1 / triangleAmountww1)),
        yww1 + (radiusww1 * sin(ww1 * twicePiww1 / triangleAmountww1))
    );
}
glEnd();

```

```

int ww11;
GLfloat xww11=0.42, yww11=0.455, radiusww11 =0.005 ;
int triangleAmountww11 = 50; // # of triangles used to draw circle
GLfloat twicePiww11 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(1.0, 1.0, 1.0); // set color to white
glVertex2f(xww11, yww11); // center of circle
for(ww11 = 0; ww11 <= triangleAmountww11; ww11++)
{
    glVertex2f(

```

```

        xww11 + (radiusww11 * cos(ww11 * twicePiww11 / triangleAmountww11)),
        yww11 + (radiusww11 * sin(ww11 * twicePiww11 / triangleAmountww11))
    );
}
glEnd();

```

```

int ww2;
GLfloat xww2=0.495, yww2=0.455, radiusww2 =0.01 ;
int triangleAmountww2 = 50; //# of triangles used to draw circle
GLfloat twicePiww2 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(0.0, 0.0, 0.0); // set color to red
glVertex2f(xww2, yww2); // center of circle
for(ww2 = 0; ww2 <= triangleAmountww2; ww2++)
{
    glVertex2f(
        xww2 + (radiusww2 * cos(ww2 * twicePiww2 / triangleAmountww2)),
        yww2 + (radiusww2 * sin(ww2 * twicePiww2 / triangleAmountww2))
    );
}
glEnd();

```

```

int ww22;
GLfloat xww22=0.495, yww22=0.455, radiusww22 =0.005 ;
int triangleAmountww22 = 50; //# of triangles used to draw circle
GLfloat twicePiww22 = 2.0 * M_PI;
glBegin(GL_TRIANGLE_FAN);
glColor3f(1.0, 1.0, 1.0); // set color to red
glVertex2f(xww22, yww22); // center of circle
for(ww22 = 0; ww22 <= triangleAmountww22; ww22++)
{
    glVertex2f(
        xww22 + (radiusww22 * cos(ww22 * twicePiww22 / triangleAmountww22)),
        yww22 + (radiusww22 * sin(ww22 * twicePiww22 / triangleAmountww22))
    );
}
glEnd();

```

```

glLoadIdentity();

```

```

glBegin(GL_QUADS); //grass pathway right
glColor3d(0.0, 0.6, 0.0);

```

```
glVertex2d(0.0,0.3125);  
glVertex2d(0.0,0.375);  
glVertex2d(1.0,0.375);  
glVertex2d(1.0,0.3125);  
glEnd();
```

```
glBegin(GL_LINES); //small tree  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.05,0.325);  
glVertex2d(0.05,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.04,0.35);  
glVertex2d(0.05,0.375);  
glVertex2d(0.06,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.04,0.3625);  
glVertex2d(0.05,0.3875);  
glVertex2d(0.06,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.04,0.375);  
glVertex2d(0.05,0.3975);  
glVertex2d(0.06,0.375);
```

```
glEnd();
```

```
glBegin(GL_LINES); //small tree 2  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.15,0.325);  
glVertex2d(0.15,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 2 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.14,0.35);  
glVertex2d(0.15,0.375);
```

```
glVertex2d(0.16,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 2 leaf 2
```

```
glColor3d(0.0, 0.8, 0.0);
```

```
glVertex2d(0.14,0.3625);
```

```
glVertex2d(0.15,0.3875);
```

```
glVertex2d(0.16,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 2 leaf 3
```

```
glColor3d(0.0, 0.8, 0.0);
```

```
glVertex2d(0.14,0.375);
```

```
glVertex2d(0.15,0.3975);
```

```
glVertex2d(0.16,0.375);
```

```
glEnd();
```

```
glBegin(GL_LINES); //small tree 3 L
```

```
glColor3d(0.0, 0.0, 0.0);
```

```
glVertex2d(0.25,0.325);
```

```
glVertex2d(0.25,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 3 leaf 1
```

```
glColor3d(0.0, 0.8, 0.0);
```

```
glVertex2d(0.24,0.35);
```

```
glVertex2d(0.25,0.375);
```

```
glVertex2d(0.26,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 3 leaf 2
```

```
glColor3d(0.0, 0.8, 0.0);
```

```
glVertex2d(0.24,0.3625);
```

```
glVertex2d(0.25,0.3875);
```

```
glVertex2d(0.26,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 3 leaf 3
```

```
glColor3d(0.0, 0.8, 0.0);
```

```
glVertex2d(0.24,0.375);
```

```
glVertex2d(0.25,0.3975);
```

```
glVertex2d(0.26,0.375);
```

```
glEnd();
```

```
glBegin(GL_LINES); //small tree 4 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.35,0.325);  
glVertex2d(0.35,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 4 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.34,0.35);  
glVertex2d(0.35,0.375);  
glVertex2d(0.36,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 4 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.34,0.3625);  
glVertex2d(0.35,0.3875);  
glVertex2d(0.36,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 4 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.34,0.375);  
glVertex2d(0.35,0.3975);  
glVertex2d(0.36,0.375);  
glEnd();
```

```
glBegin(GL_LINES); //small tree 5 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.45,0.325);  
glVertex2d(0.45,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 5 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.44,0.35);  
glVertex2d(0.45,0.375);  
glVertex2d(0.46,0.35);
```



```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 5 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.44,0.3625);  
glVertex2d(0.45,0.3875);  
glVertex2d(0.46,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 5 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.44,0.375);  
glVertex2d(0.45,0.3975);  
glVertex2d(0.46,0.375);  
glEnd();  
glLoadIdentity();
```

```
glBegin(GL_LINES); //small tree 6 l  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.55,0.325);  
glVertex2d(0.55,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 6 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.54,0.35);  
glVertex2d(0.55,0.375);  
glVertex2d(0.56,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 6 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.54,0.3625);  
glVertex2d(0.55,0.3875);  
glVertex2d(0.56,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 6 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.54,0.375);  
glVertex2d(0.55,0.3975);  
glVertex2d(0.56,0.375);
```

```
glEnd();
```

```
glBegin(GL_LINES); //small tree 7 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.65,0.325);  
glVertex2d(0.65,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 7 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.64,0.35);  
glVertex2d(0.65,0.375);  
glVertex2d(0.66,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 7 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.64,0.3625);  
glVertex2d(0.65,0.3875);  
glVertex2d(0.66,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 7 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.64,0.375);  
glVertex2d(0.65,0.3975);  
glVertex2d(0.66,0.375);  
glEnd();
```

```
glBegin(GL_LINES); //small tree 8 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.75,0.325);  
glVertex2d(0.75,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 8 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.74,0.35);
```

```
glVertex2d(0.75,0.375);  
glVertex2d(0.76,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 8 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.74,0.3625);  
glVertex2d(0.75,0.3875);  
glVertex2d(0.76,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 8 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.74,0.375);  
glVertex2d(0.75,0.3975);  
glVertex2d(0.76,0.375);  
glEnd();
```

```
glBegin(GL_LINES); //small tree 9 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.85,0.325);  
glVertex2d(0.85,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 9 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.84,0.35);  
glVertex2d(0.85,0.375);  
glVertex2d(0.86,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 9 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.84,0.3625);  
glVertex2d(0.85,0.3875);  
glVertex2d(0.86,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 9 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.84,0.375);  
glVertex2d(0.85,0.3975);
```

```
glVertex2d(0.86,0.375);  
glEnd();
```

```
glBegin(GL_LINES); //small tree 10 1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.95,0.325);  
glVertex2d(0.95,0.35);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 10 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.94,0.35);  
glVertex2d(0.95,0.375);  
glVertex2d(0.96,0.35);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 10 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.94,0.3625);  
glVertex2d(0.95,0.3875);  
glVertex2d(0.96,0.3625);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //small tree 10 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.94,0.375);  
glVertex2d(0.95,0.3975);  
glVertex2d(0.96,0.375);  
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree L1  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.05,0.50625);  
glVertex2d(0.05,0.53125);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree leaf 1
```

```
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.04,0.53125);
glVertex2d(0.05,0.55);
glVertex2d(0.06,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.04,0.54375);
glVertex2d(0.05,0.5625);
glVertex2d(0.06,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.04,0.55625);
glVertex2d(0.05,0.575);
glVertex2d(0.06,0.55625);
```

```
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 2 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.15,0.50625);
glVertex2d(0.15,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 2 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.14,0.53125);
glVertex2d(0.15,0.55);
glVertex2d(0.16,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 2 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.14,0.54375);
glVertex2d(0.15,0.5625);
glVertex2d(0.16,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 2 leaf 3
```

```
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.14,0.55625);
glVertex2d(0.15,0.575);
glVertex2d(0.16,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 3 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.25,0.50625);
glVertex2d(0.25,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 3 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.24,0.53125);
glVertex2d(0.25,0.55);
glVertex2d(0.26,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 3 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.24,0.54375);
glVertex2d(0.25,0.5625);
glVertex2d(0.26,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 3 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.24,0.55625);
glVertex2d(0.25,0.575);
glVertex2d(0.26,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 4 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.35,0.50625);
glVertex2d(0.35,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 4 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.34,0.53125);
```

```
glVertex2d(0.35,0.55);  
glVertex2d(0.36,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 4 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.34,0.54375);  
glVertex2d(0.35,0.5625);  
glVertex2d(0.36,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 4 leaf 3  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.34,0.55625);  
glVertex2d(0.35,0.575);  
glVertex2d(0.36,0.55625);  
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 5 L  
glColor3d(0.0, 0.0, 0.0);  
glVertex2d(0.45,0.50625);  
glVertex2d(0.45,0.53125);  
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 5 leaf 1  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.44,0.53125);  
glVertex2d(0.45,0.55);  
glVertex2d(0.46,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 5 leaf 2  
glColor3d(0.0, 0.8, 0.0);  
glVertex2d(0.44,0.54375);  
glVertex2d(0.45,0.5625);  
glVertex2d(0.46,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 5 leaf 3
```

```
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.44,0.55625);
glVertex2d(0.45,0.575);
glVertex2d(0.46,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 6 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.55,0.50625);
glVertex2d(0.55,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 6 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.54,0.53125);
glVertex2d(0.55,0.55);
glVertex2d(0.56,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 6 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.54,0.54375);
glVertex2d(0.55,0.5625);
glVertex2d(0.56,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 6 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.54,0.55625);
glVertex2d(0.55,0.575);
glVertex2d(0.56,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 7 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.65,0.50625);
glVertex2d(0.65,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 7 leaf 1
```



```
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.64,0.53125);
glVertex2d(0.65,0.55);
glVertex2d(0.66,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 7 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.64,0.54375);
glVertex2d(0.65,0.5625);
glVertex2d(0.66,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 7 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.64,0.55625);
glVertex2d(0.65,0.575);
glVertex2d(0.66,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 8 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.75,0.50625);
glVertex2d(0.75,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 8 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.74,0.53125);
glVertex2d(0.75,0.55);
glVertex2d(0.76,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 8 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.74,0.54375);
glVertex2d(0.75,0.5625);
glVertex2d(0.76,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 8 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.74,0.55625);
glVertex2d(0.75,0.575);
glVertex2d(0.76,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 9 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.85,0.50625);
glVertex2d(0.85,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 9 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.84,0.53125);
glVertex2d(0.85,0.55);
glVertex2d(0.86,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 9 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.84,0.54375);
glVertex2d(0.85,0.5625);
glVertex2d(0.86,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 9 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.84,0.55625);
glVertex2d(0.85,0.575);
glVertex2d(0.86,0.55625);
glEnd();
```

```
glBegin(GL_LINES); //v2 small tree 10 L
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.95,0.50625);
glVertex2d(0.95,0.53125);
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 10 leaf 1
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.94,0.53125);
glVertex2d(0.95,0.55);
glVertex2d(0.96,0.53125);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 10 leaf 2
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.94,0.54375);
glVertex2d(0.95,0.5625);
glVertex2d(0.96,0.54375);
```

```
glEnd();
```

```
glBegin(GL_TRIANGLES); //v2 small tree 10 leaf 3
glColor3d(0.0, 0.8, 0.0);
glVertex2d(0.94,0.55625);
glVertex2d(0.95,0.575);
glVertex2d(0.96,0.55625);
glEnd();
```

```
glBegin(GL_QUADS); //River Shore
glColor3d(1.0, 0.4, 0.0);
glVertex2d(0.0,0.25);
glVertex2d(0.0,0.3125);
glVertex2d(1.0,0.3125);
glVertex2d(1.0,0.25);
glEnd();
```

```
glBegin(GL_QUADS); //River
glColor3d(wa, wb, wc);
glVertex2d(0.0,0.0);
glVertex2d(0.0,0.25);
glVertex2d(1.0,0.25);
glVertex2d(1.0,0.0);
glEnd();
```

```
glLoadIdentity();
glTranslatef(xb1Pos, yb1Pos, 0.0);
```

```
glBegin(GL_QUADS); //big boat
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.33,0.1125);
```

```
glVertex2d(0.3,0.1625);
glVertex2d(0.53,0.1625);
glVertex2d(0.46,0.1125);
glEnd();
```

```
glBegin(GL_QUADS); //big boat cover
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.35,0.1625);
glVertex2d(0.35,0.1875);
glVertex2d(0.4,0.1875);
glVertex2d(0.4,0.1625);
glEnd();
```

```
glBegin(GL_LINES); //big boat line
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.41,0.1625);
glVertex2d(0.41,0.2195121951);
glEnd();
```

```
glBegin(GL_TRIANGLES); //big boat sail
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.46,0.1625);
glVertex2d(0.41,0.2195121951);
glVertex2d(0.52,0.1625);
glEnd();
glLoadIdentity();
```

```
glTranslatef(xb2Pos, yb2Pos, 0.0);
glBegin(GL_QUADS); //small boat 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.72,0.1);
glVertex2d(0.7,0.125);
glVertex2d(0.8,0.125);
glVertex2d(0.78,0.1);
glEnd();
```

```
glBegin(GL_LINES); //small boat 2 line 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.74,0.125);
glVertex2d(0.74,0.1625);
glEnd();
```

```
glBegin(GL_LINES); //small boat 2 line 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.745,0.125);
glVertex2d(0.745,0.1625);
glEnd();
```

```
glBegin(GL_TRIANGLES); //small boat 2 sail
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.75,0.125);
glVertex2d(0.75,0.1625);
glVertex2d(0.81,0.125);
glEnd();
```

```
glLoadIdentity();
glTranslatef(xb3Pos, yb3Pos, 0.0);
glBegin(GL_QUADS); //small boat 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.12,0.0375);
glVertex2d(0.1,0.0625);
glVertex2d(0.2,0.0625);
glVertex2d(0.18,0.0375);
glEnd();
```

```
glBegin(GL_LINES); //small boat 1 line 1
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.155,0.0625);
glVertex2d(0.155,0.1);
glEnd();
```

```
glBegin(GL_LINES); //small boat 1 line 2
glColor3d(0.0, 0.0, 0.0);
glVertex2d(0.16,0.0625);
glVertex2d(0.16,0.1);
glEnd();
```

```
glBegin(GL_TRIANGLES); //small boat 1 sail
glColor3d(1.0, 0.0, 0.0);
glVertex2d(0.09,0.0625);
glVertex2d(0.15,0.1);
glVertex2d(0.15,0.0625);
glEnd();
```

```
glutSwapBuffers();
}
```

```

void timercloud1(int value)
{
    xc1Pos += 0.001; // Move the first cloud forward to the right
    if (xc1Pos > cloud1boundaryRight)
        xc1Pos = cloud1boundaryLeft;

    glutPostRedisplay();
    glutTimerFunc(100, timercloud1, 0);
}

void timercloud2(int value)
{
    xc2Pos -= 0.001; // Move the second cloud backward to the left
    if (xc2Pos < cloud2boundaryLeft)
        xc2Pos = cloud2boundaryRight;

    glutPostRedisplay();
    glutTimerFunc(100, timercloud2, 0);
}

void timercloud3(int value)
{
    xc3Pos += 0.001; // Move the third cloud backward to the left
    if (xc3Pos > cloud3boundaryRight)
        xc3Pos = cloud3boundaryLeft; // Reset x position to the left boundary

    glutPostRedisplay();
    glutTimerFunc(100, timercloud3, 0); // Set the timer to trigger every 100 milliseconds
}

void timercloud4(int value)
{
    xc4Pos -= 0.001; // Move the fourth cloud backward to the left
    if (xc4Pos < cloud4boundaryLeft)
        xc4Pos = cloud4boundaryRight; // Reset x position to the left boundary

    glutPostRedisplay();
    glutTimerFunc(100, timercloud4, 0); // Set the timer to trigger every 100 milliseconds
}

void timerbird1(int value)
{
    xbird1Pos += 0.008; // Move 1st pair of birds to right

```

```

    if (xbird1Pos > bird1boundaryRight)
        xbird1Pos = bird1boundaryLeft;

    glutPostRedisplay();
    glutTimerFunc(100, timerbird1, 0);
}

void timerbird2(int value)
{
    xbird2Pos -= 0.008; // Move 2nd pair of birds to left
    if (xbird2Pos < bird2boundaryLeft)
        xbird2Pos = bird2boundaryRight;

    glutPostRedisplay();
    glutTimerFunc(100, timerbird2, 0);
}

void timercar1(int value)
{
    xv1Pos += 0.01; // Move the 1st car forward to the right
    if (xv1Pos > vehicle1boundaryRight)
        xv1Pos = vehicle1boundaryLeft;

    glutPostRedisplay();
    glutTimerFunc(50, timercar1, 0);
}

void timercar2(int value)
{
    xv2Pos -= 0.01; // // Move the 2nd car forward to the right
    if (xv2Pos < vehicle2boundaryLeft)
        xv2Pos = vehicle2boundaryRight;

    glutPostRedisplay();
    glutTimerFunc(50, timercar2, 0);
}

void timerboat1(int value)
{
    xb1Pos += 0.01; // Move the large boat forward to the right
    if (xb1Pos > boat1boundaryRight)

```

```

        xb1Pos = boat1boundaryLeft;

        glutPostRedisplay();
        glutTimerFunc(50, timerboat1, 0);
    }

void timerboat2(int value)
{
    xb2Pos += 0.01; // Move the rightfacing small boat forward to the right
    if (xb2Pos > boat1boundaryRight)
        xb2Pos = boat2boundaryLeft;

    glutPostRedisplay();
    glutTimerFunc(50, timerboat2, 0);
}

void timerboat3(int value)
{
    xb3Pos -= 0.01; // Move the leftfacing small boat forward to the right
    if (xb3Pos < boat3boundaryLeft)
        xb3Pos = boat3boundaryRight;

    glutPostRedisplay();
    glutTimerFunc(50, timerboat3, 0);
}

void update(int value)
{
    angle1 += 3.0f;
    glutPostRedisplay();
    glutTimerFunc(25, update, 0);
}

void handleMouse(int button, int state, int x, int y)
{
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        glutTimerFunc(50, timercar1, 0);
    }
    if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {
        glutTimerFunc(50, timercar2, 0);
    }
}

```



```

    }

}

void keyboard(unsigned char key, int cx, int cy)
{
    switch (key)
    {
        case 'd':
            ska=0.098,skb=0.098,skc=0.439,wa=0.275,wb=0.510,wc=
0.706,sma=1.0,smb=1.0,smc=1.0,ca=0.827,cb=0.827,cc=0.827,smra=0.098, smrb=0.098,
smrc=0.439 ;

            break;
        case 'a':
            ska=0.5,skb=1.0,skc=1.0,wa=0.0,wb=0.8,wc=0.9,sma=1.0,smb=1.0,smc=0.0,
ca=1.0,cb=1.0,cc=1.0,smra=1.000, smrb=0.388, smrc=0.278;
            break;

        case 27: // escape key
            exit(0);

        default:
            break;
    }
}

```

```

void handleSpecialKeypress(int key, int x, int y)
{
    switch(key)
    {
        case GLUT_KEY_LEFT:
            glutTimerFunc(50, timerboat3, 0);
            break;
        case GLUT_KEY_RIGHT:
            glutTimerFunc(50, timerboat1, 0);
            glutTimerFunc(50, timerboat2, 0);
            break;
    }
    glutPostRedisplay();
}

```

```

int main(int argc,char **argv)
{

```

```
glutInit(&argc,argv);
glutInitDisplayMode ( GLUT_RGB | GLUT_DOUBLE );
glutInitWindowPosition(0,0);
glutInitWindowSize(500,500);
glutCreateWindow("AAKA-LAB");
init();
glMatrixMode(GL_MODELVIEW);
glutDisplayFunc(Draw);

glutTimerFunc(100, timercloud1, 0);
glutTimerFunc(100, timercloud3, 0);
glutTimerFunc(100, timercloud2, 0);
glutTimerFunc(100, timercloud4, 0);

glutTimerFunc(100, timerbird1, 0);
glutTimerFunc(100, timerbird2, 0);

glutTimerFunc(50, update, 0);

glutKeyboardFunc(keyboard);
glutMouseFunc(handleMouse);
glutSpecialFunc(handleSpecialKeypress);

glutMainLoop();
return 0;
}
```

Screenshot

