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

GLfloat x1[500],x2[500],y11[5000],y2[5000],y3[5000],y4[5000];
GLfloat yb11=250,xb1=100;
int points[100]={0};
int games=1,flag=0,flag1=0,flag2=0;

void rect_box(GLfloat x11,GLfloat x22,GLfloat y111,GLfloat y22,GLfloat y33,GLfloat y44)
//drawing the two pipes with outerline
{
    GLfloat i,k=20.0,l=20.0,m=20.0,n=20.0;

      glColor3f(0.31,0.55,0.51);  //lower pipe
      glBegin(GL_POLYGON);
      glVertex2i(x11,y111);
      glVertex2i(x22,y111);
      glVertex2i(x22,y22);
      glVertex2i(x11,y22);
    glEnd();
```

```
    glColor3f(0.0,0.0,0.0);  //lower pipe outline
    glBegin(GL_LINE_LOOP);
    glVertex2i(x11,y111);
    glVertex2i(x22,y111);
    glVertex2i(x22,y22);
    glVertex2i(x11,y22);
glEnd();

glColor3f(0.31,0.55,0.51);  //upper pipe line
glBegin(GL_POLYGON);
    glVertex2i(x11,y33);
    glVertex2i(x22,y33);
    glVertex2i(x22,y44);
    glVertex2i(x11,y44);
glEnd();

glColor3f(0.0,0.0,0.0);  //upper pipe outline
glBegin(GL_LINE_LOOP);
    glVertex2i(x11,y33);
    glVertex2i(x22,y33);
    glVertex2i(x22,y44);
    glVertex2i(x11,y44);
```

```
    glEnd();


      for(i=y111;i<=y22-20.0; i=i+20.0)   //zigzag lines
       {
        glBegin(GL_LINES);
        glVertex2i(x11,i);
        glVertex2i(x22,y111+k);
        glEnd();
        k=k+20.0;
       }
      for(i=y33;i<=y44-20.0; i=i+20.0)
       {
        glBegin(GL_LINES);
        glVertex2i(x11,i);
        glVertex2i(x22,y33+m);
        glEnd();
        m=m+20.0;
       }
}



void circ()  // for drawing the ball
```

```
{
float yb1,xb2,yb2;
float angle;
double radius=25;

glColor4f(0.0,0.74,0.99,0.08);

glBegin(GL_TRIANGLE_FAN);
glVertex2f(xb1,yb11);

for (angle=1.0f;angle<361.0f;angle+=0.2)
{
   xb2 = xb1+sin(angle)*radius;
   yb2 = yb11+cos(angle)*radius;
   glVertex2f(xb2,yb2);
}
 glEnd();
}
```



```
void rect(void)  //assigning pipes coordintaes using random function
```

```
{
  GLfloat a,b,c,d;
  int i;
  for(i=0;i<500;i++)
  {
    x1[i]=500+(200*i);
    x2[i]=x1[i]+50;
        a=0;
        b=(rand()%150)+50;
        c=b+200;
      d=500;
    y11[i]=a;
    y2[i]=b;
    y3[i]=c;
    y4[i]=d;
   }
}

void pipes()  //for drawing pipes
{
 int i;
 for(i=0;i<500;i++)
```

```
  {
    if(x1[i]<500 && x2[i]>0)
     rect_box(x1[i],x2[i],y11[i],y2[i],y3[i],y4[i]);
  }
}



void bg()   //background
{

  glBegin(GL_POLYGON);    //for top blue background
  glColor3f(0.55,0.83,0.83);
  glVertex2f(0,130);
  glVertex2f(500,130);
  glVertex2f(500,500);
  glVertex2f(0,500);
glEnd();


glBegin(GL_POLYGON);  // for bottom brown background
  glColor3f(0.95,0.64,0.36);
  glVertex2f(0,0);
  glVertex2f(500,0);
  glVertex2f(500,130);
```

```
  glVertex2f(0,130);
glEnd();
}

void move_rect_n_ball()   //idle funtion which is responsible for movement of pipes and ball
{
if(flag1==1)
{
  int j;
  for(j=0;j<500;j++)   //moving pipes left to right
  {
     {
      if(x2[j]>0)
      {
    x1[j]=x1[j]-1;
    x2[j]=x2[j]-1;
       }
     }
  }

  if (x2[499]<100)        //level up
  rect();
```

```
  if(yb11<500)
     yb11=yb11+1;   //ball moving down
  glutPostRedisplay();
}

}


void scr()  //to calculate the score
{
int i;
for(i=0;i<500;i++)
   {
    if(x2[i]<70)
     {
       points[games]++;
      }
   }
}


void keystrk(unsigned char key,int x,int y)  //keyboard function
{
```

```
 if(key=='d')   //for moving the ball downn
 {
  if(yb11>0)
   {
  yb11=yb11-50;
  glutPostRedisplay();
   }
 }
}


void check()
{
  int i;
    for(i=0;i<500;i++)
      {
        if(x1[i]<125 && x1[i]>25)    //to see if the pipe is within the ball boundary
         {
           if((yb11+25)>y3[i] || (yb11-25)<y2[i])   //to see if the ball has touched the pipes
           {
             scr();    //calculate the score at last
             menu();
                games++;
```

```
            flag=0;
            flag2=0;
           }
         }
      }
}

void output(int x, int y, char *string)  //outputs the text on the window
{
        int len, i;
        glRasterPos2f(x, y);    //moves cursor to the point
        len = (int)strlen(string);
        for (i = 0; i < len; i++)

        {
                glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24 , string[i] );
        }       //displays text string with the specified format of text
}

darw_box(int xd1,int xd2,int yd1,int yd2)  //to draw boxes in menu
{
glColor3f(1.0,0.50,0.0);
glBegin(GL_POLYGON);
```

```
        glVertex2i(xd1,yd1);
        glVertex2i(xd2,yd1);
        glVertex2i(xd2,yd2);
        glVertex2i(xd1,yd2);
    glEnd();
}



draw_selbox(int xd1,int xd2,int yd1,int yd2)  //to draw sub boxes in menu
{
glColor3f(0.0,0.0,1.0);
glBegin(GL_POLYGON);
        glVertex2i(xd1,yd1);
        glVertex2i(xd2,yd1);
        glVertex2i(xd2,yd2);
        glVertex2i(xd1,yd2);
    glEnd();
}



menu()    // to create menu
{
int pts;
```

```
bg();
   darw_box(200,350,400,450);
  draw_selbox(200,225,400,450);
   glColor3f(0.0,0.50,1.0);
   output(220,425,"new game");  //to start menu

   darw_box(200,350,300,350);
  draw_selbox(240,265,300,350);
   glColor3f(0.0,0.50,1.0);
   output(220,325,"controls");  //to display controls

   darw_box(200,350,200,250);
  draw_selbox(280,305,200,250);
   glColor3f(0.0,0.50,1.0);
   output(220,225,"high scores");  // to display high score

   darw_box(200,350,120,170);
  draw_selbox(320,350,120,170);
    glColor3f(0.0,0.50,1.0);
   output(220,145,"exit");

   darw_box(200,450,30,80);
  draw_selbox(380,410,30,80);
```

```
    glColor3f(0.0,0.50,1.0);
   output(220,55,"creators information");  //to display the creators information



 darw_box(100,240,260,290);
    glColor3f(0.0,0.50,1.0);
   output(110,270,"score=");     // to display a box with high scores

          char buff[10];
  sprintf(buff,"%d",points[games-1]);  // to covert an integer to character
  output(180,270,buff);


}

bck()  //back box in submenus
{
  darw_box(50,150,350,400);
  glColor3f(0.0,0.50,1.0);
   output(70,375,"back");
}
```

```
functions()  //controls part
{
 bg();
 bck();
 glColor3f(0.0,0.0,1.0);

 output(100,300,"press D");
 output(100,250,"to move the ball");
 output(100,200,"down   or else");
 output(100,150,"the ball comes down automatically");
  output(100,100,"as the ball is floating in water");
}

info()  // display part about the cerators
{
bg();
bck();
 glColor3f(0.0,0.0,1.0);

 output(100,300,"this");
 output(100,250,"project");
 output(100,200,"is created by");
 output(100,150,"VtuCs.com");
```

```
  output(100,100,"From Bits to Bytes, all about Computer Science");
}

void scores()
{
bg();
bck();
   int i;
   char buf1[10];
  darw_box(200,300,450,100);
  glColor3f(0.0,0.0,1.0);
 for(i=1;i<games;i++)
    {
     sprintf(buf1,"%d",points[i]);
     output(220,440-(20*i),buf1);
    }
}

void mouse(int button, int state, int x, int y)
{
   if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
   {
```

```
    if(x>240 && x<265)// && y>300 && y<350)
    {
     //controls part
    flag=1;
    flag1=2;
    glutPostRedisplay();
   }
   if(x>280 && x<305 )//&& y>200 && y<250)
  {
    // high scores are selcted
    flag1=3;
    flag=1;
    glutPostRedisplay();
    }

    if(x>320 && x<350 )//&& y>100 && y<150)
    exit(1);    //exits from the menu

     if(x>380 && x<410 )
   {
   flag=1;
flag1=5;
  glutPostRedisplay();   //information is selected
```

```
  }
 if( x>200 && x<225 )//&& y>400 && y<450)
   {
    flag=1;
    flag1=1;   // starts new game
    rect();
   }
  if(x>50 && x<150)
    {
    flag=0;              //goes back to the main menu from sub menus
    flag1=0;
    glutPostRedisplay();
    }
}
}


void display(void)
{
  int a;
  glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0,1.0,0.0);
  if(!flag)
```

```
{
 menu();
}
if(flag==1)
{
if(flag1==2)
  functions();
 else if(flag1==3)
    scores();
  else if(flag1==5)
    info();
 else
     {
 flag2=1;

         bg();
         pipes();
         circ();
          glColor3f(0.0,0.0,1.0);
         output(200,475,"floating ball");
         check();  //for gameover checking
   }
  }
```

```
 glFlush();
}

init()
{
glClearColor(1.0,1.0,1.0,1.0);
glColor3f(1.0,0.0,0.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0,499.0,0.0,499.0);
}



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

   glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
   glutInitWindowSize(500, 500);
   glutCreateWindow("Floating Ball");
   glutDisplayFunc(display);
   glutIdleFunc(move_rect_n_ball);
```

```
    glutKeyboardFunc(keystrk);
    glutMouseFunc(mouse);
    glBlendFunc(GL_SRC_ALPHA,GL_ONE_MINUS_SRC_ALPHA);
    glEnable(GL_BLEND);
    init();
    glutMainLoop();

return 0;
}
```