```
#include <windows.h>
#include <iostream>
#include <gl/glut.h>
#include<math.h>
float i=0;
int flag=0,till=0;
void init()
   {
          glClearColor(1,1,1,0);
          gluOrtho2D(0,600,0,600);
        glMatrixMode(GL_MODELVIEW);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
   }
void move()
{
    if(i<=380&&flag==0)
    {
        i+=0.1;

    }
    else
        flag=1;
    if(i>=0&&flag==1)
    {
        i-=0.1;

    }
    else
        flag=0;

}
void text()
{
    int b;
    char name[30]="CGV MINI PROJECT";
    char m[10]="AIET";
    char p[40]="OS LOGO'S";
    glRasterPos2f(i,500);
    for(b=0;name[b]!='\0';b++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24,name[b]);
    glRasterPos2f(236,440);
    for(b=0;p[b]!='\0';b++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24,p[b]);
    glRasterPos2f(274,105);
    for(b=0;m[b]!='\0';b++)
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24,m[b]);

}
void ano(int x,int y,int r)
{
    float f;
    glBegin(GL_POLYGON);
    for( int angle = 0; angle < 360; angle ++ ){
        f=angle*3.142/180;
       glVertex2f( x+r*cos(f),y+r*sin(f));}
    glEnd();
}
```

```
void Logo()
{
    glScalef(0.75,0.75,0);
    glColor3f(1,0,0);
    ano(300,170,100);
    glPushMatrix();
    glColor3f(1,1,0);
    glScalef(1,0.5,0);
    ano(300,230,40);
    glColor3f(1,0,0);
    ano(300,250,30);

    glColor3f(1,1,0);
    glScalef(1,0.75,0);
    ano(300,270,20);

    glColor3f(1,0,0);
    glBegin(GL_TRIANGLES);
    glVertex2d(280,280);
    glVertex2d(320,280);
    glVertex2d(300,400);
    glEnd();
    glPopMatrix();

    glColor3f(0,0,0);
    glRecti(300,100,303,245);
    glRecti(270,100,273,249);
    glRecti(330,100,333,249);
    glRecti(240,89,243,250);
    glRecti(358,90,361,247);
    glRecti(385,120,388,220);
    glRecti(210,122,213,215);

    glColor3f(1,1,1);
    glRecti(315,100,318,247);
    glRecti(285,100,288,249);
    glRecti(255,100,258,249);
    glRecti(344,100,347,249);
    glRecti(225,90,228,247);
    glRecti(373,90,376,247);


    glColor3f(0,1,1);
    ano(260,260,10);
    ano(340,260,10);
    glColor3f(1,1,1);
    ano(300,140,60);
    glColor3f(0,0,0);
    ano(300,140,50);
    glColor3f(1,1,0);
    ano(300,140,30);
    glColor3f(1,0,0);
    ano(300,140,20);
    glColor3f(1,1,0);
    ano(300,140,10);
    glColor3f(1,1,1);
    glRecti(180,65,430,85);
}


void window(void)
{
       glClear(GL_COLOR_BUFFER_BIT);

       glPushMatrix();// box no:1 (top right)
         glScalef(6,6,1);
         glTranslatef(1,1,0);
         glColor3f(0.117647, 0.564706, 1);      //(red, green, blue)
       glBegin(GL_POLYGON);
       //A
       glVertex2f(0.0, 12.0);
       //B
       glVertex2f(12.0, 15.0);
       //C
       glVertex2f(12.0, 0.5);
       //D
       glVertex2f(0.0, 0.5);
       glEnd();
       glFlush();
       // box No: 2 (top left)
       glColor3f(0.117647, 0.564706, 1);      //(red, green, blue)
       glBegin(GL_POLYGON);
       //A
       glVertex2f(-10.0, 10.0);
       //B
       glVertex2f(-0.5, 12.0);
       //C
       glVertex2f(-0.5, 0.5);
       //D
       glVertex2f(-10.0, 0.5);
       glEnd();
       glFlush();
       // box No: 3 (bottom left)
       glColor3f(0.117647, 0.564706, 1);      //(red, green, blue)
       glBegin(GL_POLYGON);
       //A
       glVertex3f(-10.0, -0.5, 0.0);
       //B
       glVertex3f(-0.5, -0.5, 0.0);
       //C
       glVertex3f(-0.5, -12.0, 0.0);
       //D
       glVertex3f(-10.0, -10.0, 0.0);
       glEnd();
       glFlush();
       // box No: 4 (bottom right)
       glColor3f(0.117647, 0.564706, 1);      //(red, green, blue)
       glBegin(GL_POLYGON);
       //A
       glVertex3f(0.0, -0.5, 0.0);
       //B
       glVertex3f(12.0, -0.5, 0.0);
       //C
       glVertex3f(12.0, -15.0, 0.0);
       //D
       glVertex3f(0.0, -12.0, 0.0);
       glEnd();
         glPopMatrix();
       glFlush();
}

void bno(int x,int y,int r)
{
    float f;
    glBegin(GL_POLYGON);
    for( int angle = 0; angle < 360; angle ++ ){
        f=angle*3.142/180;
       glVertex2f( x+r*cos(f),y+r*sin(f));}
    glEnd();


}
void mac()
{

  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(0,0,0);
  glPushMatrix();
  glRotatef(30,0,1,0);
  bno(250,300,110);
  bno(350,300,110);
  glPopMatrix();
  bno(300,220,20);
  glPushMatrix();
  glRotatef(80,0.5,1,0);
  bno(150,540,50);
  glPopMatrix();
  glColor3f(1,1,1);
  bno(450,330,70);
  glFlush();
}

void ubuntu()
{
  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(1,0,0);
  ano(300,300,110);
  glColor3f(1,1,1);
  ano(300,300,75);
  glColor3f(1,0,0);
  ano(300,300,50);
  glColor3f(1,0,0);
  ano(375,260,19);
  ano(225,260,19);
  ano(300,385,19);
  glColor3f(1,1,1);
  ano(375,260,15);
  ano(225,260,15);
  ano(300,385,15);
  glColor3f(1,0,0);
  glRecti(295,225,305,250);
  glPushMatrix();

  glRecti(225,310,255,320);
  glPopMatrix();
  glRecti(345,310,375,320);
  glFlush();
}


void android()
{
  glClear(GL_COLOR_BUFFER_BIT);
  glLoadIdentity();
  glColor3f(0,1,0);
  glRecti(200,200,400,400);
  ano(300,400,100);
  glPushMatrix();
  glColor3f(1,1,1);
  ano(340,460,8);
  ano(260,460,8);
  glRecti(200,390,400,400);
  glPopMatrix();
  glColor3f(0,1,0);
  glRecti(410,250,440,360);
  ano(425,250,15);
  ano(425,360,15);
  glRecti(160,250,190,360);
  ano(175,250,15);
  ano(175,360,15);
  glRecti(250,110,280,200);
  ano(265,110,15);
  glRecti(320,110,350,200);
  ano(335,110,15);
  glColor3f(0,1,0);
  glPushMatrix();
 // glRotatef(30,0,0,1);
   glRecti(350,480,360,530);
   ano(355,530,5);
   glPopMatrix();
   glPushMatrix();
 //  glRotatef(-30,0,0,1);
   glRecti(245,480,255,530);
   ano(250,530,5);
   glPopMatrix();
   glFlush();
}

void mainMenuHandler(int choice) {
      switch(choice)
      {
      case 1:till=1;
            glutPostRedisplay();
            break;
      case 2:till=2;
            glutPostRedisplay();
            break;
      case 3:till=3;
            glutPostRedisplay();
```

```
          break;
      case 4:till=4;
          glutPostRedisplay();
          break;
      case 5:exit(0);
      default:break;
      }
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

    glLoadIdentity();
    glColor3f(0,0,0);

      if(till==0)
      {
          text();
          Logo();
          glutPostRedisplay();
      }
      if(till==1)
      {
          ubuntu();
      }
      if(till==2)
      {
          android();
      }
      if(till==3)
      {
          window();
      }
      if(till==4)
      {
          mac();
      }

      glFlush();
}
int main(int argc, char **argv) {
      glutInit(&argc, argv);
      glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
      glutInitWindowPosition(0,0);
      glutInitWindowSize(600, 600);
      glutCreateWindow("OpenGL OS Logo's");
      init();

      glutIdleFunc(move);

      glutCreateMenu(mainMenuHandler);
      glutAddMenuEntry("Ubuntu", 1);
      glutAddMenuEntry("Android", 2);
      glutAddMenuEntry("Windows", 3);
      glutAddMenuEntry("Mac", 4);
      glutAddMenuEntry("Exit", 5);
      glutAttachMenu(GLUT_RIGHT_BUTTON);
      glutDisplayFunc(display);
      glutMainLoop();
      return 0;
}
```