

```

#include <stdio.h>
#include <math.h>
#include <GL/glut.h>
#define pi 3.142
static GLfloat angle = 0;

static int submenu;
static int mainmenu;
static int value = -1;

void init()
{
    gluOrtho2D(-1000, 1000, -1000, 1000);
}

void circle1(float rad)
{
    glBegin(GL_POINTS);
    glColor3f(1, 0, 0);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(rad * cos(i), rad * sin(i));
    }
    glEnd();
}

void circle2(float rad)
{
    glBegin(GL_POINTS);
    glColor3f(1, 1, 0);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(rad * cos(i), rad * sin(i));
    }
    glEnd();
}

void circle3(float rad)
{
    glBegin(GL_POINTS);
    glColor3f(1, 0, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(rad * cos(i), rad * sin(i));
    }
    glEnd();
}

void drawString(float x, float y, float z, char *string)
{
    glColor3f(1,1,1);
    glRasterPos3f(x, y, z);

    for (char *c = string; *c != '\0'; c++)
    {
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, *c);
    }
}

void drawhead(float x, float y, float z, char *string)
{
    glColor3f(1, 1, 1);
    glRasterPos3f(x, y, z);

    for (char *c = string; *c != '\0'; c++)
    {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *c);
    }
}

void drawsubhead(float x, float y, float z, char *string)

```

```

{
    glColor3f(1, 1, 1);
    glRasterPos3f(x, y, z);

    for (char *c = string; *c != '\0'; c++)
    {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *c);
    }
}

void nuc(float rad)
{
    // Set the fill color
    //glColor3f(0.5f, 0.5f, 0.5f); // Gray color

    // Draw the filled nucleus
    glBegin(GL_POLYGON);
    glVertex2f(0.0f, 0.0f); // Center of the nucleus

    // Add vertices to form a circle
    int numSegments = 100; // Number of line segments to approximate a circle
    // Radius of the nucleus
    for (int i = 0; i <= numSegments; ++i)
    {
        float theta = 2.0f * 3.1415926f * static_cast<float>(i) / static_cast<float>(numSegments);
        float x = rad * cos(theta);
        float y = rad * sin(theta);
        glVertex2f(x, y);
    }

    glEnd();

    // Set the wireframe color
    glColor3f(0.0f, 0.0f, 0.0f); // Black color

    // Draw the wireframe nucleus
    glBegin(GL_LINE_LOOP);
    for (int i = 0; i <= numSegments; ++i)
    {
        float theta = 2.0f * 3.1415926f * static_cast<float>(i) / static_cast<float>(numSegments);
        float x = rad * cos(theta);
        float y = rad * sin(theta);
        glVertex2f(x, y);
    }
    glEnd();
    char n[] = "NUCLEUS";
    drawString(-65, 20, 0, n);
    char d[] = "(NEUTRON + PROTON)";
    drawString(-140, -30, 0, d);

    //glFlush();
}

void eleright(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(rad + 20 * cos(i), 20 * sin(i));
    }
    glEnd();
}

void eleleft(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);

```

```

    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(-(rad + 20 * cos(i)), 20 * sin(i));
    }
    glEnd();
}

void eletop(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(20 * cos(i), rad + 20 * sin(i));
    }
    glEnd();
}

void eledown(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(20 * cos(i), -(rad + 20 * sin(i)));
    }
    glEnd();
}

void eletr(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i((rad - 175) + 20 * cos(i), ((rad - 175) + 20 * sin(i)));
    }
    glEnd();
}

void eletl(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(-(rad - 175) + 20 * cos(i), ((rad - 175) + 20 * sin(i)));
    }
    glEnd();
}

void eledl(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i(-(rad - 175) + 20 * cos(i), -((rad - 175) + 20 * sin(i)));
    }
    glEnd();
}

void eledr(float rad)
{
    glBegin(GL_POLYGON);
    glColor3f(1, 1, 1);
    for (float i = 0; i < (2 * pi); i += 0.00001)
    {
        glVertex2i((rad - 175) + 20 * cos(i), -((rad - 175) + 20 * sin(i)));
    }
    glEnd();
}

```

```

void display()
{
    glClearColor(0, 0, 0.1, 0.9);

    if (value == -1)
    {
        char cn[] = "JSSATE";
        drawhead(-130, 900, 0, cn);
        char pn[] = " Bangalore- 560060";
        drawsubhead(-210, 850, 0, pn);

        char dn[] = "DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING";
        drawhead(-450, 650, 0, dn);

        char prn[] = "A Mini Project On";
        drawsubhead(-180, 450, 0, prn);
        char pro[] = "ATOM SIMULATION";
        drawhead(-200, 350, 0, pro);

        char pb[] = "PROJECT BY: ";
        drawhead(-690, -150, 0, pb);

        char p1[] = "Shamitha A S";
        drawhead(-690, -250, 0, p1);
        char plu[] = "1JS20CS146";
        drawsubhead(-690, -300, 0, plu);

        char p2[] = "Varshini Jakati";
        drawhead(-690, -400, 0, p2);
        char p2u[] = "1JS20CS179";
        drawsubhead(-690, -450, 0, p2u);

        char gb[] = "GUIDED BY: ";
        drawhead(290, -150, 0, gb);

        char g1[] = "Dr Sharana Basavana Gowda";
        drawhead(290, -250, 0, g1);
        char d1[] = "Associate Professor, Dept. Of CSE, JSSATE";
        drawsubhead(290, -300, 0, d1);

        char g2[] = "Mrs. Rashmi B N";
        drawhead(290, -400, 0, g2);
        char d2[] = "Assistant Professor, Dept. of CSE, JSSATE";
        drawsubhead(290, -450, 0, d2);

        char in[] = "Press enter to Continue";
        drawhead(-250, -700, 0, in);

        //glutSwapBuffers();
        glFlush();

        glutDetachMenu(GLUT_RIGHT_BUTTON);
    }

    if (value != -1)
    {
        nuc(250);

        if (value == 0)
        {
            char nu[] = "SELECT THE ELEMENT USING MENU";
            drawhead(-280, 800, 0, nu);
        }
    }
}

```

```

if (value == 1)
{
    glColor3f(0.0f,1.0f,0.0f);
    nuc(250);

    char n[] = "HYDROGEN";
    drawhead(-100, 900, 0, n);
    circle1(400);
    glColor3f(1, 1, 1);
    char o[] = "ORBIT";
    drawString(410, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    char e[] = "ELECTRON";
    drawString(420, 10, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 2)
{
    glColor3f(0.5f, 0.5f, 0.5f);
    nuc(250);
    char n[] = "HELIUM";
    drawhead(-100, 900, 0, n);
    circle1(400);
    char o[] = "ORBIT";
    drawString(410, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    char e[] = "ELECTRON";
    drawString(420, 10, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 3)
{
    glColor4f(1.0f,0.0f,0.0f,0.0f);
    nuc(250);
    char n[] = "LITHIUM";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);

    char o[] = "ORBIT";
    drawString(900, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 4)
{
    glColor3f(1.0f,0.5f,0.0f);
    nuc(250);
    char n[] = "BERYLLIUM";
    drawhead(-100, 900, 0, n);

```

```

circle1(400);
circle1(402);
circle1(404);
circle2(600);
char o[] = "ORBIT";
drawString(610, 0, 0, o);
glPushMatrix();
glRotatef(angle, 0, 0, 1);
eleright(400);
eleleft(400);
eletop(600);
eledown(600);
char e[] = "ELECTRON";
drawString(0, 620, 0, e);
glPopMatrix();
glutSwapBuffers();
}
if (value == 5)
{
    glColor3f(0.0f,0.5f,0.5f);
    nuc(250);
    char n[] = "BORON";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 6)
{
    glColor4f(0.0f,1.0f,1.0f,1.0f);
    nuc(250);
    char n[] = "CARBON";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    eledl(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}

```

```

if (value == 7)
{
    glColor4f(0.0f,1.0f,1.0f,1.0f);
    nuc(250);
    char n[] = "NITROGEN";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    eledl(600);
    eletl(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 8)
{
    glColor4f(0.0f,1.0f,1.0f,1.0f);
    nuc(250);
    char n[] = "OXYGEN";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    eledl(600);
    eletl(600);
    eledr(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 9)
{
    glColor3f(0.0f,1.0f,0.0f);
    nuc(250);
    char n[] = "FLUORINE";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();

```

```

    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    eledl(600);
    eletl(600);
    eledr(600);
    eleleft(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 10)
{
    glColor3f(0.5f, 0.5f, 0.5f);
    nuc(250);
    char n[] = "NEON";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle2(600);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);
    eletr(600);
    eledl(600);
    eletl(600);
    eledr(600);
    eleleft(600);
    eleright(600);
    char e[] = "ELECTRON";
    drawString(0, 620, 0, e);
    glPopMatrix();
    glutSwapBuffers();
}
if (value == 11)
{
    glColor4f(1.0f,0.0f,0.0f,0.0f);
    nuc(250);
    char n[] = "SODIUM";
    drawhead(-100, 900, 0, n);
    circle1(400);
    circle1(402);
    circle1(404);
    circle1(406);
    circle2(600);
    circle2(602);
    circle2(604);
    circle3(800);
    char o[] = "ORBIT";
    drawString(610, 0, 0, o);
    glPushMatrix();
    glRotatef(angle, 0, 0, 1);
    eleright(400);
    eleleft(400);
    eletop(600);
    eledown(600);

```



```

        eletr(600);
        eledl(600);
        eletl(600);
        eledr(600);
        eleleft(600);
        eleright(600);
        eletop(800);
        char e[] = "ELECTRON";
        drawString(0, 620, 0, e);
        glPopMatrix();
        glutSwapBuffers();
    }
    if (value == 12)
    {
        glColor3f(1.0f,0.5f,0.0f);
        nuc(250);
        char n[] = "MAGNESIUM";
        drawhead(-100, 900, 0, n);
        circle1(400);
        circle1(402);
        circle1(404);
        circle1(406);
        circle2(600);
        circle2(602);
        circle2(604);
        circle3(800);
        char o[] = "ORBIT";
        drawString(610, 0, 0, o);
        glPushMatrix();
        glRotatef(angle, 0, 0, 1);
        eleright(400);
        eleleft(400);
        eletop(600);
        eledown(600);
        eletr(600);
        eledl(600);
        eletl(600);
        eledr(600);
        eleleft(600);
        eleright(600);
        eletop(800);
        eledown(800);
        char e[] = "ELECTRON";
        drawString(0, 620, 0, e);
        glPopMatrix();
        glutSwapBuffers();
    }

    if (value == 13)
    {
        glColor4f(0.0,0.0,1.0,0.0);
        nuc(250);
        char n[] = "ALUMINIUM";
        drawhead(-100, 900, 0, n);
        circle1(400);
        circle1(402);
        circle1(404);
        circle1(406);
        circle2(600);
        circle2(602);
        circle2(604);
        circle3(800);
        char o[] = "ORBIT";
        drawString(610, 0, 0, o);
        glPushMatrix();
        glRotatef(angle, 0, 0, 1);
    }

```

```

        eleright(400);
        eleleft(400);
        eletop(600);
        eledown(600);
        eletr(600);
        eledl(600);
        eletl(600);
        eledr(600);
        eleleft(600);
        eleright(600);
        eletop(800);
        eledown(800);
        eleleft(800);
        char e[] = "ELECTRON";
        drawString(0, 620, 0, e);
        glPopMatrix();
        glutSwapBuffers();
    }

    glutSwapBuffers();
}
void rotate()
{
    angle = angle + 1.0;
    if (angle > 360)
    {
        angle = angle - 360;
    }
    glClear(GL_COLOR_BUFFER_BIT);
    glutPostRedisplay();
}
void mouseControl(int button, int state, int x, int y)
{
    switch (button)
    {
        case GLUT_LEFT_BUTTON:
            if (state == GLUT_DOWN)
                glutIdleFunc(rotate);
            break;
        default:
            break;
    }
}

void keyboard(unsigned char key, int x, int y)
{
    if (key == 13)
    {
        value = 0;
        glClear(GL_COLOR_BUFFER_BIT);
        glutAttachMenu(GLUT_RIGHT_BUTTON);
        glutPostRedisplay();
    }
    else if (key == 's')
    {
        glutIdleFunc(NULL);
    }
    else if (key == 32)
    {
        glutIdleFunc(rotate);
    }
    else if (key == 'q' || key == 'Q')
    {

```

```

        exit(0);
    }
    else if (key == 27)
    {
        glutReshapeWindow(700, 700);
    }
}
void fkey(int key, int x, int y)
{

    if (key == GLUT_KEY_F10)
    {
        glutReshapeWindow(glutGet(GLUT_SCREEN_WIDTH), glutGet(GLUT_SCREEN_HEIGHT));
    }
}
void menu(int option)
{
    if (option == 16)
    {
        value=-1;
    }
    else if (option == 14)
    {
        glutIdleFunc(rotate);
    }
    else if (option == 15)
    {
        glutIdleFunc(NULL);
    }
    else if(option==17){
        value=-1;
    }
    else
    {
        value = option;
    }
    glClear(GL_COLOR_BUFFER_BIT);

    glutPostRedisplay();
}
void createMenu(void)
{
    submenu = glutCreateMenu(menu);
    glutAddMenuEntry("HYDROGEN", 1);
    glutAddMenuEntry("HELIUM", 2);
    glutAddMenuEntry("LITHIUM", 3);
    glutAddMenuEntry("BERILIUM", 4);
    glutAddMenuEntry("BORON", 5);
    glutAddMenuEntry("CARBON", 6);
    glutAddMenuEntry("NITROGEN", 7);
    glutAddMenuEntry("OXYGEN", 8);
    glutAddMenuEntry("FLUORINE", 9);
    glutAddMenuEntry("NEON", 10);
    glutAddMenuEntry("SODIUM", 11);
    glutAddMenuEntry("MAGNESIUM", 12);
    glutAddMenuEntry("ALUMINIUIUM", 13);
    mainmenu = glutCreateMenu(menu);
    glutAddSubMenu("SELECT THE ELEMENT", submenu);
    glutAddMenuEntry("START SIMULATION", 14);
    glutAddMenuEntry("STOP SIMULATION", 15);
    glutAddMenuEntry("GOTO HOME SCREEN",16);
    glutAddMenuEntry("EXIT", 17);

    glutAttachMenu(GLUT_RIGHT_BUTTON);
}
int main(int argc, char **argv)

```

```
{  
    glutInit(&argc, argv);  
    glutInitWindowPosition(0,0);  
    glutInitWindowSize(1500, 1500);  
    glutCreateWindow("ATOM SIMULATION");  
    init();  
    glutDisplayFunc(display);  
  
    glutMouseFunc(mouseControl);  
    glutKeyboardFunc(keyboard);  
    glutSpecialFunc(fkey);  
    createMenu();  
    glutMainLoop();  
    return 0;  
}
```