#include<windows.h>

#include <iostream>

#include <math.h>

#include <GL/glut.h>

#include <dos.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 circle(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 drawString(float x, float y, float z, char \*string)

{

//text colour

glColor3f(0, 0, 0);

//glColor3f(1, 1, 1);

glRasterPos3f(x, y, z);

for (char \*c = string; \*c != '\0'; c++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_10, \*c);

}

}

void drawhead(float x, float y, float z, char \*string)

{

//text color

glColor3f(0, 0, 0);

//glColor3f(1, 1, 1);

glRasterPos3f(x, y, z);

for (char \*c = string; \*c != '\0'; c++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, \*c);

}

}

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

{

//text color

glColor3f(0, 0, 0);

//glColor3f(1, 1, 1);

glRasterPos3f(x, y, z);

for (char \*c = string; \*c != '\0'; c++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, \*c);

}

}

void nuc(float rad)

{

glBegin(GL\_POLYGON);

//polygon colour of nucli

glColor3f(0.8, 0.4, 0.5);

//glColor3f(0.8, 0.4, 0.5);

for (float i = 0; i < (2 \* pi); i = i + 0.00001)

{

glVertex2f(rad \* cos(i), rad \* sin(i));

}

glEnd();

}

void eleright(float rad)

{

glBegin(GL\_POLYGON);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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);

//electron colour

glColor3f(0, 0, 1);

//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(1, 1, 0.9, 1);

glClear(GL\_COLOR\_BUFFER\_BIT);

if (value == -1)

{

char cn[] = "MIT ACADEMY OF ENGINEERING";

drawhead(-650, 900, 0, cn);

char pn[] = "ALANDI ,PUNE -412105";

drawsubhead(-250, 850, 0, pn);

char dn[] = "DEPARTMENT OF COMPUTER ENGINEERING";

drawhead(-690, 650, 0, dn);

char prn[] = "A Mini Project On";

drawsubhead(-150, 450, 0, prn);

char pro[] = "ATOM SIMULATION";

drawhead(-250, 350, 0, pro);

char pb[] = "PROJECT BY: ";

drawhead(-690, -150, 0, pb);

char p1[] = "SAKSHI BHINGARKAR";

drawhead(-690, -250, 0, p1);

char p1u[] = "202301040260";

drawsubhead(-690, -300, 0, p1u);

char p2[] = "VAISHNAVI THORAVE";

drawhead(-690, -400, 0, p2);

char p2u[] = "202301040261";

drawsubhead(-690, -450, 0, p2u);

char g2[] = "SAMADHAN MANE";

drawhead(-690, -550, 0, g2);

char d2[] = "202301040262";

drawsubhead(-690, -600, 0, d2);

char gb[] = "GUIDED BY: ";

drawhead(290, -150, 0, gb);

char g1[] = "MRS. CHETNA NEMADE";

drawhead(290, -250, 0, g1);

char d1[] = "";

drawsubhead(290, -300, 0, d1);

char in[] = "Press enter to Continue";

drawhead(-250, -700, 0, in);

glutDetachMenu(GLUT\_RIGHT\_BUTTON);

glutSwapBuffers();

}

if (value != -1)

{

nuc(250);

char n[] = "NUCLEUS";

drawString(-90, 20, 0, n);

char d[] = "(NEUTRON + PROTON)";

drawString(-170, -30, 0, d);

if (value == 0)

{

char nu[] = "SELECT THE ELEMENT USING MENU";

drawhead(-490, 900, 0, nu);

glutSwapBuffers();

}

}

if (value == 1)

{

char n[] = "HYDROGEN";

drawhead(-100, 900, 0, n);

circle(400);

char o[] = "ORBIT";

drawString(410, 0, 0, o);

glPushMatrix();

glRotatef(angle, 0, 0, 1);

eleright(400);

char e[] = "ELECTRON";

drawString(420, 0, 0, e);

glPopMatrix();

glutSwapBuffers();

}

if (value == 2)

{

char n[] = "HELIUM";

drawhead(-100, 900, 0, n);

circle(400);

char o[] = "ORBIT";

drawString(410, 0, 0, o);

glPushMatrix();

glRotatef(angle, 0, 0, 1);

eleright(400);

eleleft(400);

char e[] = "ELECTRON";

drawString(420, 0, 0, e);

glPopMatrix();

glutSwapBuffers();

}

if (value == 3)

{

char n[] = "LITHIUM";

drawhead(-100, 900, 0, n);

circle(400);

circle(600);

char o[] = "ORBIT";

drawString(610, 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)

{

char n[] = "BERYLLIUM";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "BORON";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "CARBON";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "NITROGEN";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "OXYGEN";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "FLUORINE";

drawhead(-100, 900, 0, n);

circle(400);

circle(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)

{

char n[] = "NEON";

drawhead(-100, 900, 0, n);

circle(400);

circle(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();

}

//glutSwapBuffers();

}

void rotate()

{

angle = angle + 30.0;

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 == 'b' || key == 'B')

{

value=-1;

}

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 == 13)

{

exit(0);

}

else if (option == 11)

{

glutIdleFunc(rotate);

}

else if (option == 12)

{

glutIdleFunc(NULL);

}

else if(option==14){

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);

mainmenu = glutCreateMenu(menu);

glutAddSubMenu("SELECT THE ELEMENT", submenu);

glutAddMenuEntry("START SIMULATION", 11);

glutAddMenuEntry("STOP SIMULATION", 12);

glutAddMenuEntry("GOTO HOME SCREEN",14);

glutAddMenuEntry("EXIT", 13);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

}

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

{

glutInit(&argc, argv);

glutInitWindowPosition(300, 100);

glutInitWindowSize(700, 700);

glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE);

glutCreateWindow("ATOM SIMULATION");

init();

glutDisplayFunc(display);

glutMouseFunc(mouseControl);

glutKeyboardFunc(keyboard);

glutSpecialFunc(fkey);

createMenu();

glutMainLoop();

return 0;

}