**Computer Graphics and Virtual Reality Course Project Report**

**Group:** SY 29

**Class:** CS-B

**CP Topic:** 2D-Helicopter Game

**Project Guide:** Prof. Archana Burujwale

**Group Members:**

|  |  |
| --- | --- |
| **Roll No.** | **Name** |
| 22 | Krushna Gore |
| 23 | Kartarsingh Gothwal |
| 25 | Tanvi Gunjal |
| 26 | Sejal Hage |

**Code:**

#include<stdlib.h>

#include<GL/glut.h>

#include<time.h>

#include<dos.h>

#include<stdio.h>

#include<conio.h>

#include<windows.h>

float bspd = 0.005;

char name[25];

float b1x = 50.0, b1y = 0;

float hm = 0.01;

int i = 0, sci = 1;float scf = 1;

char scs[20], slevel[20];

int level = 1, lflag = 1, wflag = 1;

float bladeAngle = 0.0;

float heliColorR = 0.7f;

float heliColorG = 1.0f;

float heliColorB = 1.0f;

typedef struct {

    float x;

    float y;

    float speed;

    int active;

} Bullet;

Bullet bullet = { 0, 0, 0.1, 0 };

void init(void)

{

    srand(time(0));

    b1y = (rand() % 45) + 10;

    glClearColor(0.0, 0.0, 0.0, 0.0);

    glShadeModel(GL\_SMOOTH);

    glLoadIdentity();

    glOrtho(0.0, 100.0, 0.0, 100.0, -1.0, .0);

}

void drawcopter()

{

    glColor3f(heliColorR, heliColorG, heliColorB);

    glRectf(10, 49.8, 19.8, 44.8); // body

    glRectf(2, 46, 10, 48);        // tail

    glRectf(2, 46, 4, 51);         // tail up

    glRectf(14, 49.8, 15.8, 52.2); // propeller stand

    glRectf(7, 53.6, 22.8, 52.2);  // propeller

    glPushMatrix();

    glTranslatef(15, 52.9, 0);

    glRotatef(bladeAngle, 0.0, 0.0, 1.0); // Rotate propeller

    glTranslatef(-15, -52.9, 0);

    glRectf(7, 53.6, 22.8, 52.2);

    glPopMatrix();

}

void changeHeliColor() {

    static int colorIndex = 0;

    float colors[][3] = {

        {0.7f, 1.0f, 1.0f}, // Cyan

        {1.0f, 0.0f, 0.0f}, // Red

        {0.0f, 1.0f, 0.0f}, // Green

        {0.0f, 0.0f, 1.0f}, // Blue

        {1.0f, 1.0f, 0.0f}, // Yellow

        {1.0f, 0.5f, 0.0f}, // Orange

        {0.5f, 0.0f, 0.5f}  // Purple

    };

    colorIndex = (colorIndex + 1) % 7; // Cycle through colors

    heliColorR = colors[colorIndex][0];

    heliColorG = colors[colorIndex][1];

    heliColorB = colors[colorIndex][2];

}

void resetGame() {

    b1x = 50.0;

    b1y = (rand() % 45) + 10;

    hm = 0.01;

    bspd = 0.005;

    bladeAngle = 0.0;

    sci = 1;

    scf = 1;

    level = 1;

    lflag = 1;

    wflag = 1;

}

void drawBullet() {

    if (bullet.active) {

        glColor3f(1.0, 1.0, 0.0);

        glRectf(bullet.x, bullet.y, bullet.x + 2, bullet.y + 4);

    }

}

void updateBullet() {

    if (bullet.active) {

        bullet.x += bullet.speed;

        if (bullet.x >= b1x && bullet.x <= b1x + 5 && bullet.y >= b1y && bullet.y <= b1y + 35) {

            bullet.active = 0;

            b1x = -10;

        }

        if (bullet.x > 100) {

            bullet.active = 0;

        }

    }

}

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

{

    char\* c;

    glRasterPos3f(x, y, z);

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

    {

        glutBitmapCharacter(font, \*c);

    }

}

void display(void)

{

    glClear(GL\_COLOR\_BUFFER\_BIT);

    if (

        (i == 2500 || i == -2500)

        //top and bottom checking

        ||

        (((int)b1x == 10 || (int)b1x == 7 || (int)b1x == 4 || (int)b1x == 1) && (int)b1y < 53 + (int)hm && (int)b1y + 35>53 + (int)hm)

        // propeller front checking

        ||

        (((int)b1x == 9 || (int)b1x == 3 || (int)b1x == 6) && (int)b1y < 45 + (int)hm && (int)b1y + 35>45 + (int)hm)

        //lower body checking

        ||

        (((int)b1x == 0) && (int)b1y < 46 + (int)hm && (int)b1y + 35>46 + (int)hm))

        // lower tail checking

    {

        wflag = 2;

        bspd = 0;

        hm = 0.0;

        glColor3f(0.0, 0.0, 1.0);

        glRectf(0.0, 0.0, 100.0, 100.0);

        glColor3f(1.0, 0.0, 0.0);

        renderBitmapString(40, 70, 0, GLUT\_BITMAP\_HELVETICA\_18, "GAME OVER!!!");

        glColor3f(1.0, 1.0, 1.0);

        renderBitmapString(25, 58, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "You");

        renderBitmapString(45, 58, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "scored:");

        renderBitmapString(70, 58, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, scs);

        renderBitmapString(30, 38, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Press 'r' to restart or 'Esc' to Exit:");

        glutSwapBuffers();

        glFlush();

        printf("\nGAME OVER\n\n");

        printf("%s You scored  %s", name, scs);

        printf("\n\nClose the console window to exit...\n");

        getchar();

    }

    updateBullet();

    drawBullet();

    if (wflag == 2) {

        renderBitmapString(40, 50, 0, GLUT\_BITMAP\_HELVETICA\_18, "Press 'r' to Restart or 'Esc' to Exit");

        return;

    }

    else if (wflag == 1)//Welcome Screen

    {

        wflag = 0;

        glColor3f(0.0, 0.0, 0.0);

        glRectf(0.0, 0.0, 100.0, 10.0);//ceil

        glRectf(0.0, 100.0, 100.0, 90.0);//floor

        glColor3f(1.0, 1.0, 1.0);

        renderBitmapString(35, 85, 0, GLUT\_BITMAP\_HELVETICA\_18, "Vishwakarma Institute of Technology");

        renderBitmapString(41, 80, 0, GLUT\_BITMAP\_HELVETICA\_12, "Pune, Maharashtra- 411 037");

        glColor3f(1.0, 1.0, 0.0);

        renderBitmapString(20, 65, 0, GLUT\_BITMAP\_8\_BY\_13, "a mini project for Computer Graphics & Virtual  Reality");

        renderBitmapString(45.5, 70, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Helicopter");

        glColor3f(1.0, 0.0, 0.0);

        renderBitmapString(40, 45, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Welcome");

        renderBitmapString(53, 45, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, name);

        renderBitmapString(43, 30, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Click To Start");

        renderBitmapString(17, 24, 0, GLUT\_BITMAP\_9\_BY\_15, "CLICK AND HOLD LEFT MOUSE BUTTON TO GO UP RELEASE TO GO DOWN");

        glColor3f(0.0, 0.0, 0.0);

        drawcopter();

        glutSwapBuffers();

        glFlush();

    }

    else

    {

        //on every increase by 50 in score in each level

        if (sci % 50 == 0 && lflag == 1)

        {

            lflag = 0; //make level\_flag=0

            level++;//increase level by 1

            bspd += 0.0001;//increase block\_dx\_speed by 0.01

        }

        //within every level make level\_flag=1

        else if (sci % 50 != 0 && lflag != 1)

        {

            lflag = 1;

        }

        glPushMatrix();

        glColor3f(0.0, 0.0, 0.0);

        glRectf(0.0, 0.0, 100.0, 10.0);

        glRectf(0.0, 100.0, 100.0, 90.0);

        glColor3f(0.0, 0.0, 0.0);

        renderBitmapString(1, 3, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Distance:");

        sprintf(slevel, "%d", level);

        renderBitmapString(80, 3, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, "Level:");

        renderBitmapString(93, 3, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, slevel);

        scf += 0.015;

        sci = (int)scf;

        sprintf(scs, "%d", sci);

        renderBitmapString(20, 3, 0, GLUT\_BITMAP\_TIMES\_ROMAN\_24, scs);

        glTranslatef(0.0, hm, 0.0);

        drawcopter();

        if (b1x < -10)

        {

            b1x = 50;

            b1y = (rand() % 25) + 20;

        }

        else

            b1x -= bspd;

        glTranslatef(b1x, -hm, 0.0);

        glColor3f(1.0, 0.0, 0.0);

        glRectf(b1x, b1y, b1x + 5, b1y + 35);//block 1

        glPopMatrix();

        glutSwapBuffers();

        glFlush();

    }

}

void moveHeliU(void)

{

    hm += 0.025;

    i++;

    glutPostRedisplay();

}

void moveHeliD()

{

    hm -= 0.025;

    i--;

    glutPostRedisplay();

}

void mouse(int button, int state, int x, int y)

{

    switch (button)

    {

    case GLUT\_LEFT\_BUTTON:

        if (state == GLUT\_DOWN)

        {

            glutIdleFunc(moveHeliU);

            bladeAngle += 30.0;

            if (bladeAngle >= 360.0)

                bladeAngle -= 360.0;

        }

        else if (state == GLUT\_UP)

        {

            glutIdleFunc(moveHeliD);

        }

        break;

    default:

        break;

    }

    glutPostRedisplay();  // Ensure the screen updates with the rotation

}

void keys(unsigned char key, int x, int y)

{

    switch (key)

    {

    case 'w':

        glutIdleFunc(moveHeliU);

        break;

    case 's':

        glutIdleFunc(moveHeliD);

        break;

    case '7': // Change helicopter color

        changeHeliColor();

        break;

    case 'r':  // Restart the game

        resetGame();

        glutPostRedisplay();

        break;

    case 27:  // Escape key for exit

        exit(0);

    case ' ':  // Space bar for firing bullet

        if (!bullet.active) {

            bullet.active = 1;

            bullet.x = 20;  // Start from the nose of the copter

            bullet.y = 50 + hm;  // Adjust to copter's vertical position

        }

        break;

    }

    glutPostRedisplay();

}

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

{

    printf("enter your name to play: ");

    scanf("%s", name);

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

    glutInitWindowSize(800, 600);

    glutInitWindowPosition(200, 200);

    glutCreateWindow("2D Copter Game");

    init();

    glutDisplayFunc(display);

    glutMouseFunc(mouse);

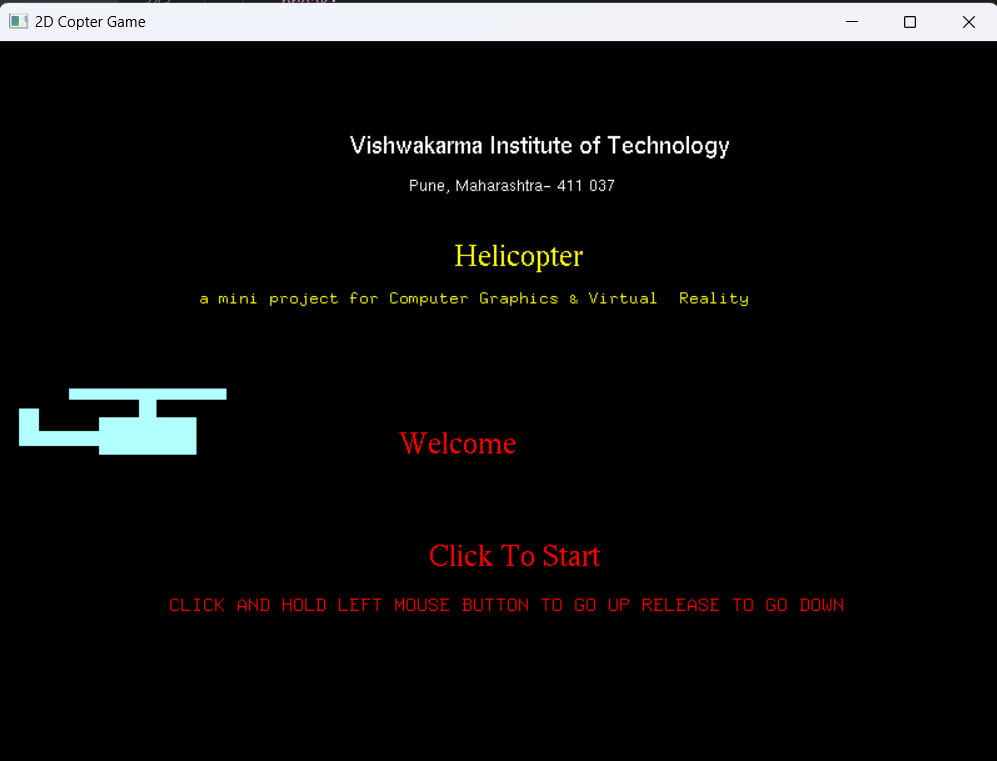
    glutKeyboardFunc(keys);

    glutMainLoop();

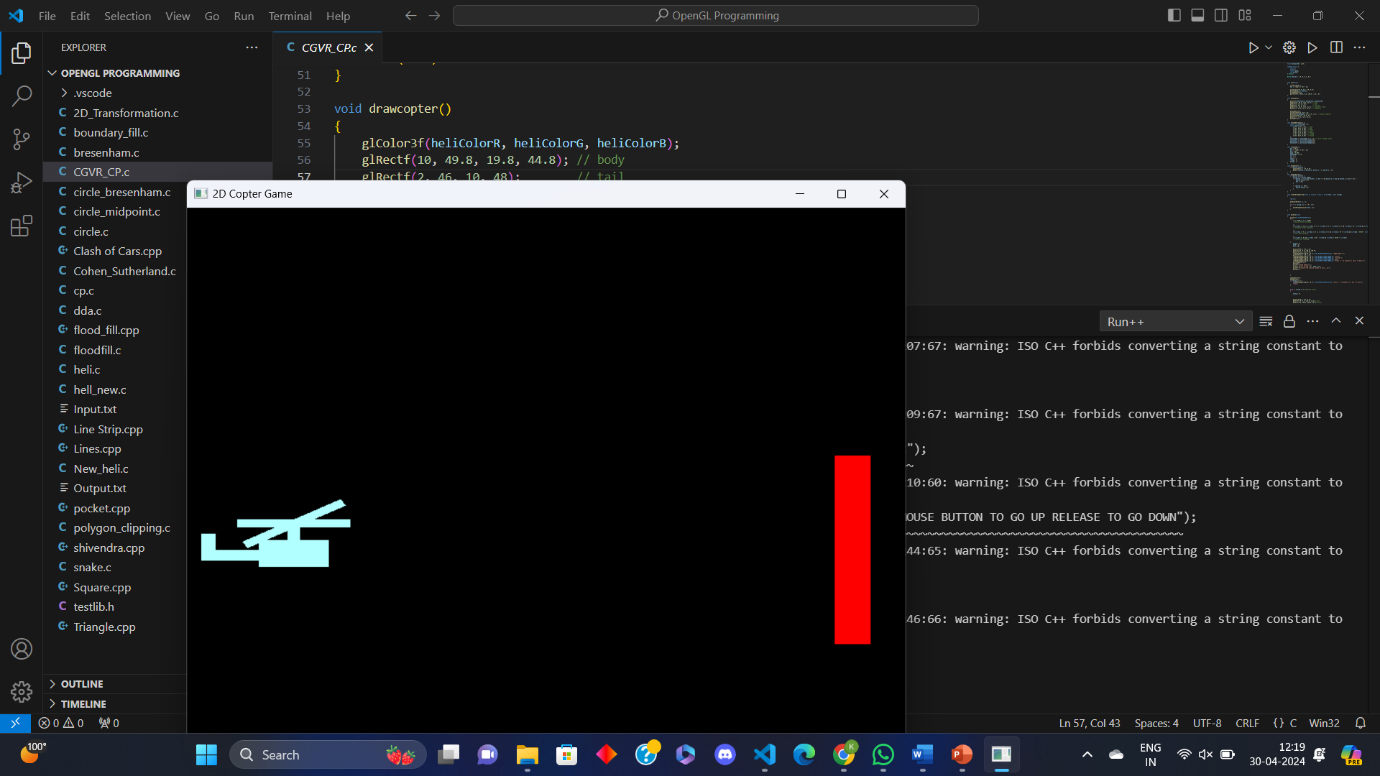
    return 0;

}

**Output:**

****

Welcome Window



Game Over Window