Computer Graphics (UCS505)

Project on

ROTOR RUSH

(HELICOPTOR GAME)

Submitted By

Awantika Awasthi	102103015
Suvidha Srivastava	102103019

3CO1

B.E. Third Year – COE

Submitted To:

Ms. Anupam Garg

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147001

Table of Contents

Sr. No.	Description	Page No.
1.	Introduction	3
2.	Concepts used	4
3.	User Defined Functions	5
4.	Code	6
5.	Output Screenshots	12

INTRODUCTION

About Computer Graphics

Computer graphics revolutionizes our interaction with computers by leveraging our strong visual perception to process information effectively. Interactive computer graphics offers unmatched interactivity and flexibility, enabling real-time creation and manipulation of images. It allows us to generate not only lifelike depictions of physical objects but also abstract and synthetic constructs. By utilizing mathematical formulas, intricate shapes and surfaces can be fabricated, surpassing the limitations of traditional techniques. Additionally, graphics aids in visualizing and comprehending vast amounts of data in ways traditional tools cannot match. In summary, computer graphics fundamentally transforms how we engage with computers, offering interactive visual representations of both abstract and tangible concepts.

About OpenGL

OpenGL is a widely used standard specification that defines an API for developing cross-platform applications that produce 2D and 3D computer graphics. It was initially developed by Silicon Graphics Inc. (SGI) in 1992 and is utilized in a diverse range of applications, including CAD, virtual reality, scientific visualization, information visualization, and flight simulation. Additionally, it is extensively employed in the video game industry, where it competes with direct 3D on Microsoft Windows platforms.

OpenGL offers a uniform API to simplify interfacing with different 3D accelerators, thereby masking their complexities. Furthermore, it conceals the hardware platform differences by requiring that all implementations support the entire OpenGL feature set. This approach has influenced the development of 3D accelerators, promoting a base level of functionality that is now commonly found in consumer-level hardware.

OpenGL comprises over 250 different function calls that allow developers to draw complex 3D scenes from simple primitives. It provides support for rasterized points, lines, and polygons, a transform and lighting pipeline, Z buffering, texture mapping, and alpha blending, among other features.

About our Project

In our application, we'll leverage OpenGL through two libraries and c++ language that provide direct access to its functions.The primary GLUT library, or OpenGL, contains functions prefixed with "gl" and stored in a library typically referred to as GL (or OpenGL in Windows). The OpenGL Utility Library (GLUT) utilizes GL functions and includes code for simplified viewing and creating common objects.

Our project aims to showcase the computer graphics skills we've acquired by creating an helicopter game. The game features a helicopter controlled by the player using mouse or keyboard inputs ('w' for up, 's' for down). The objective is to avoid obstacles represented by rectangles while collecting points. The game ends when the helicopter collides with an obstacle.

COMPUTER GRAPHICS CONCEPT USED

To achieve 2D effects, we'll employ OpenGL software, which provides a graphical interface acting as an intermediary between application programs and graphics hardware. The advantages include:

- Streamlined design
- Hardware-independent interface, applicable across various hardware platforms
- Capability to draw a small set of geometric primitives such as points, lines, and polygons
- Provision of double buffering essential for providing transformations
- Event-driven software

Translation involves adding the required translation quantities to each point of the objects within the selected area. If P(x,y) represents a point and (tx, ty) represents translation quantities, the translated point is obtained by using the function glTranslatef(dx,dy,dz).

Rotation, a fundamental concept in computer graphics, is facilitated by OpenGL. Through transformation matrices and functions like glRotatef(), OpenGL enables the rotation of objects around specified axes. This capability is crucial for creating dynamic and interactive visualizations, allowing for the portrayal of movement and orientation within virtual environments. Rotation adds an additional dimension of realism and interactivity to computer-generated scenes, enhancing the overall user experience and facilitating the communication of complex ideas and concepts.

In our project, there are 3 different display screens

1. The frontsheet which is the game's introduction page
2. Game scenario with rotating helicopter and obstacles
3. Game over screen

USER DEFINED FUNCTIONS

- Write(float x, float y, float z, float scale, const char* s)
- frontsheet(void)
- drawcopter(void)
- renderBitmapString(float x, float y, float z, void* font, const char* string)
- display(void)
- moveHeliU(void)
- moveHeliD(void)
- mouse(int button, int state, int x, int y)
- keys(unsigned char key, int x, int y)
- keyboards(unsigned char key, int x4, int y4)

CODE

```
#include<stdlib.h>
#include<GL/glut.h>
#include<time.h>
#include<stdio.h>
#include<iostream>
#include<windows.h>
#include<string>
using namespace std;

int win1, win2;
bool fanRotation = true;
float fanAngle = 0.0;

void Write(float x, float y, float z, float scale, const char* s)
{
   int i, l = strlen(s);
   glPushMatrix();
   glTranslatef(x, y, z);
   glScalef(scale, scale, scale);
   for (i = 0; i < l; i++)
      glutStrokeCharacter(GLUT_STROKE_ROMAN, s[i]);
   glPopMatrix();
}

void frontsheet(void)
{
   glClearColor(0, 0, 0, 1);
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1.0, 1.0, 0.0);
   Write(-0.67, 0.9, 1, 0.0007, "Thapar Institute of Engineering");
   Write(-0.55, 0.8, 1, 0.0006, "  and Technology, Patiala");
   glColor3f(1.0, 0.0, 0.0);
   Write(-0.45, 0.6, 0.0, 0.0007, "Rotor Rush");
   glColor3f(1.0, 1.0, 0.5);
   Write(-0.4, -0.8, 0.0, 0.0006, "Press 'C' to continue");
   glColor3f(1, 1, 0.0);
   Write(-1.0, 0.1, 0.0, 0.0007, " Submitted BY:");
   glColor3f(1.0, 1.0, 1.0);
   Write(-1.0, -0.03, 0.0, 0.0006, "1. Awantika Awasthi    : 102103015");
   Write(-1.0, -0.13, 0.0, 0.0006, "2. Suvidha Srivastava  : 102103019");

   glColor3f(1, 1, 0.0);
   Write(-1.0, -0.4, 0.0, 0.0007, " Submitted to: ");
   glColor3f(1.0, 1.0, 1.0);
   Write(0.15, -0.415, 0.0, 0.0006, "Ms. Anupam Garg");
   glFlush();
}
```

```
float bspd = 0.005;
float b1x = 50.0, b1y = 0;
float hm = 0.0;
int i = 0, sci = 1; float scf = 1;
char scs[20], slevel[20];
int level = 1, lflag = 1, wflag = 1;

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()
{
   // Body
   glColor3f(0.5, 1.0, 0.3);
   glRectf(10, 49.8, 19.8, 44.8);

   // Tail
   glColor3f(0.5, 1.0, 0.3);
   glRectf(2, 46, 10, 48); // tail
   glRectf(2, 46, 4, 51); // upper tail

   // Fan
   glPushMatrix();
   glTranslatef(14, 51, 0); // position fan
   glRotatef(fanAngle, 0, 0, 1); // rotate fan
   glColor3f(1.0, 0.0, 0.0);
   glBegin(GL_TRIANGLES);
   glVertex2f(0, 0);
   glVertex2f(5, 0);
   glVertex2f(2.5, 5);
   glEnd();
   glPopMatrix();

   // Main Propeller
   glRectf(14, 49.8, 15.8, 52.2);

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

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

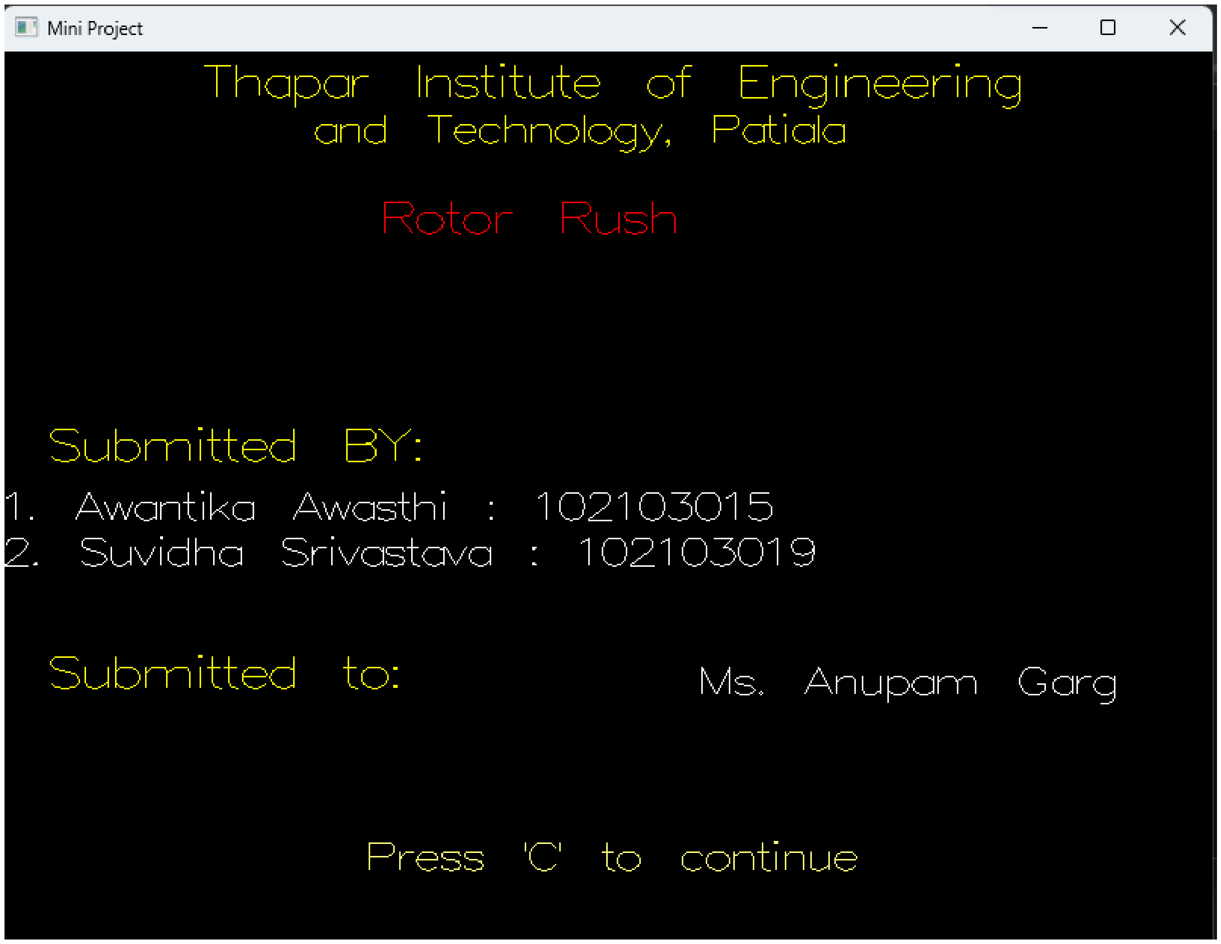
```
{


   const 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 == 3600 || i == -3600)
      ||
      (((int)b1x == 10 || (int)b1x == 7 || (int)b1x == 4 || (int)b1x == 1) && (int)b1y < 53 + (int)hm &&
(int)b1y + 35>53 + (int)hm)
      ||
      (((int)b1x == 9 || (int)b1x == 3 || (int)b1x == 6) && (int)b1y < 45 + (int)hm && (int)b1y + 35>45 +
(int)hm)
      ||
      (((int)b1x == 0) && (int)b1y < 46 + (int)hm && (int)b1y + 35>46 + (int)hm))
      {
         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(30, 58, 0, GLUT_BITMAP_TIMES_ROMAN_24, "THANKS FOR PLAYING
THE GAME!!");
         renderBitmapString(50, 48, 0, GLUT_BITMAP_TIMES_ROMAN_24, scs);
         glutSwapBuffers();
         glFlush();
         printf("\nGAME OVER\n\n");
         system("pause");
         printf("\n\nClose the console window to exit...\n");
         exit(0);
   }
   else
   {
      if (sci % 50 == 0 && lflag == 1)
      {
         lflag = 0;
         level++;
         bspd += 0.0001;
      }
      else if (sci % 50 != 0 && lflag != 1)
      {
         lflag = 1;
      }
```

```
        glPushMatrix();


        glColor3f(0.0, 0.5, 0.7);
        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);

        sprintf_s(slevel, "Score: %d", level);
        renderBitmapString(80, 3, 0, GLUT_BITMAP_TIMES_ROMAN_24, slevel);

        scf += 0.01;
        sci = (int)scf;
        printf(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);

        glPopMatrix();

        glutSwapBuffers();
        glFlush();
    }
}

void moveHeliU(void)
{

    hm += 0.01;
    i++;
    if (fanRotation)
        fanAngle += 10;
    glutPostRedisplay();

}

void moveHeliD()
```

```
{

   hm -= 0.01;
   i--;
   if (fanRotation)
      fanAngle -= 10;
   glutPostRedisplay();

}
void mouse(int button, int state, int x, int y)
{
   switch (button)
   {
   case GLUT_LEFT_BUTTON:

      if (state == GLUT_DOWN)
         glutIdleFunc(moveHeliU);

      else if (state == GLUT_UP)
         glutIdleFunc(moveHeliD);
      break;
   default: break;
   }
}
void keys(unsigned char key, int x, int y)
{
   if (key == 'w') glutIdleFunc(moveHeliU);
   if (key == 's') glutIdleFunc(moveHeliD);
}
void keyboards(unsigned char key, int x4, int y4)
{
   if (key == 'c' || key == 'C')
   {
      glutDestroyWindow(win1);
      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
      win2 = glutCreateWindow("Helicopter Game");
      glClearColor(0.0, 0.0, 0.0, 0.0);
      glFlush();
      glutDisplayFunc(display);
      gluOrtho2D(-1000, 1000, 0, 1000);
      init();
      glutMouseFunc(mouse);
      glutKeyboardFunc(keys);
   }
}

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
   glutInitWindowSize(800, 600);
```

```
    glutInitWindowPosition(200, 20);
    win1 = glutCreateWindow("Mini Project");
    glFlush();
    glutDisplayFunc(frontsheet);
    glutKeyboardFunc(keyboards);
    glutMainLoop();
    return 0;

}
```

OUTPUT SCREENSHOTS



Thapar Institute of Engineering and Technology, Patiala

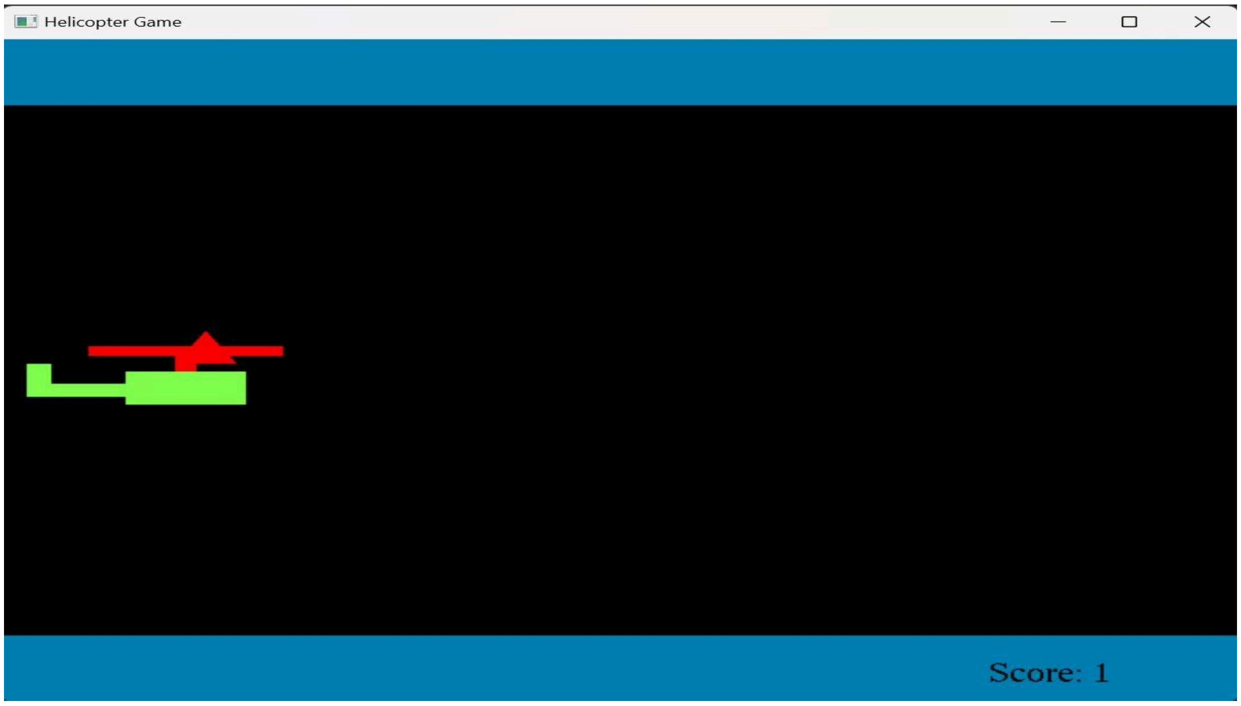
Rotor Rush

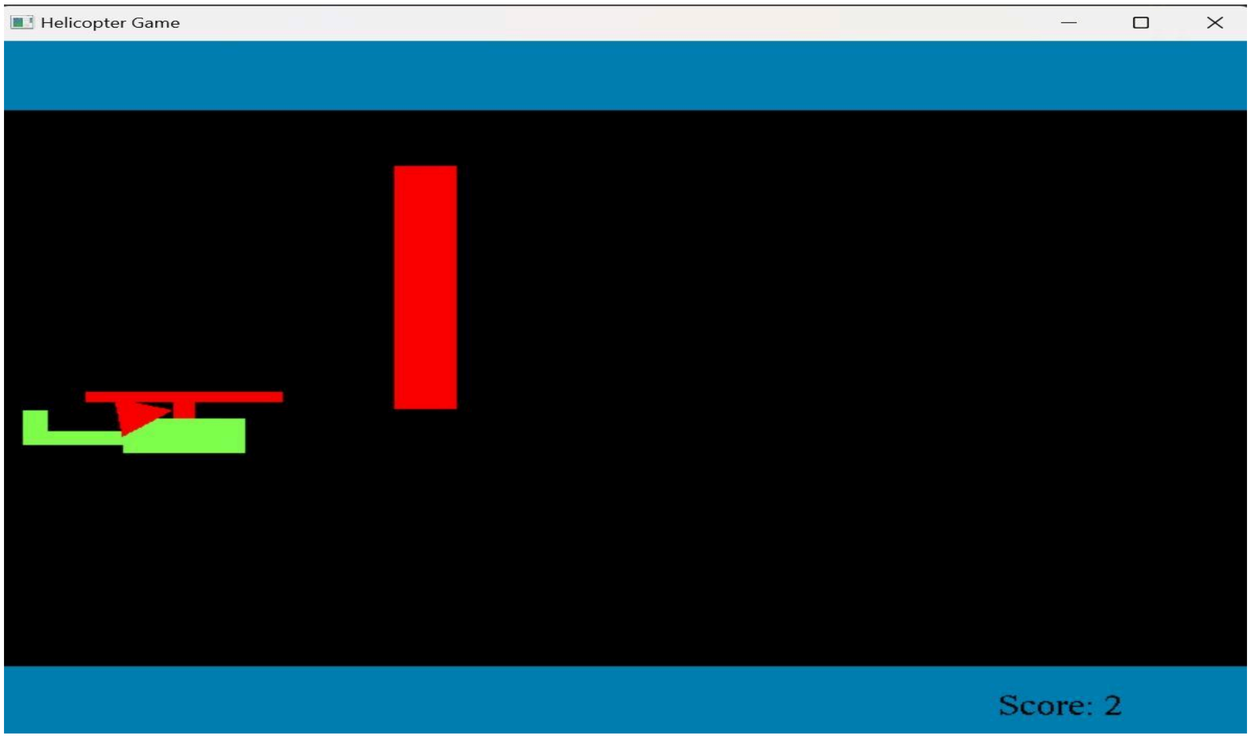
Submitted BY:

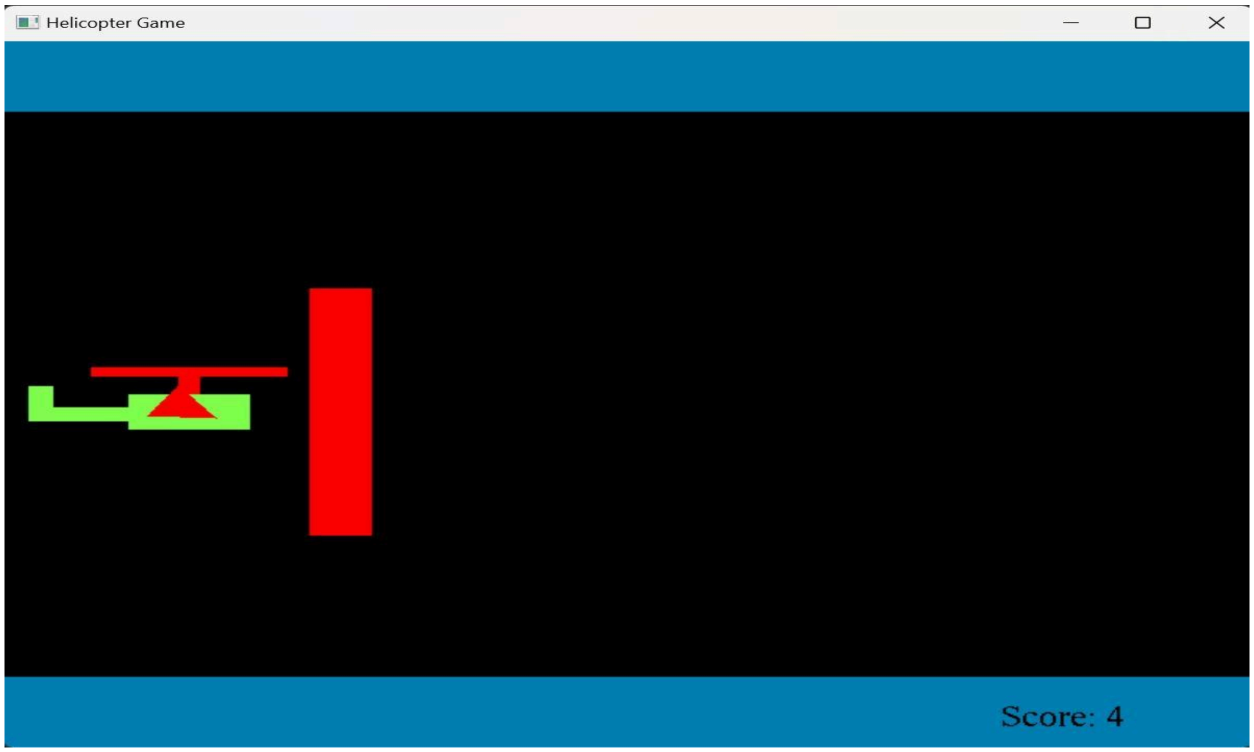
1. Awantika Awasthi : 102103015
2. Suvidha Srivastava : 102103019

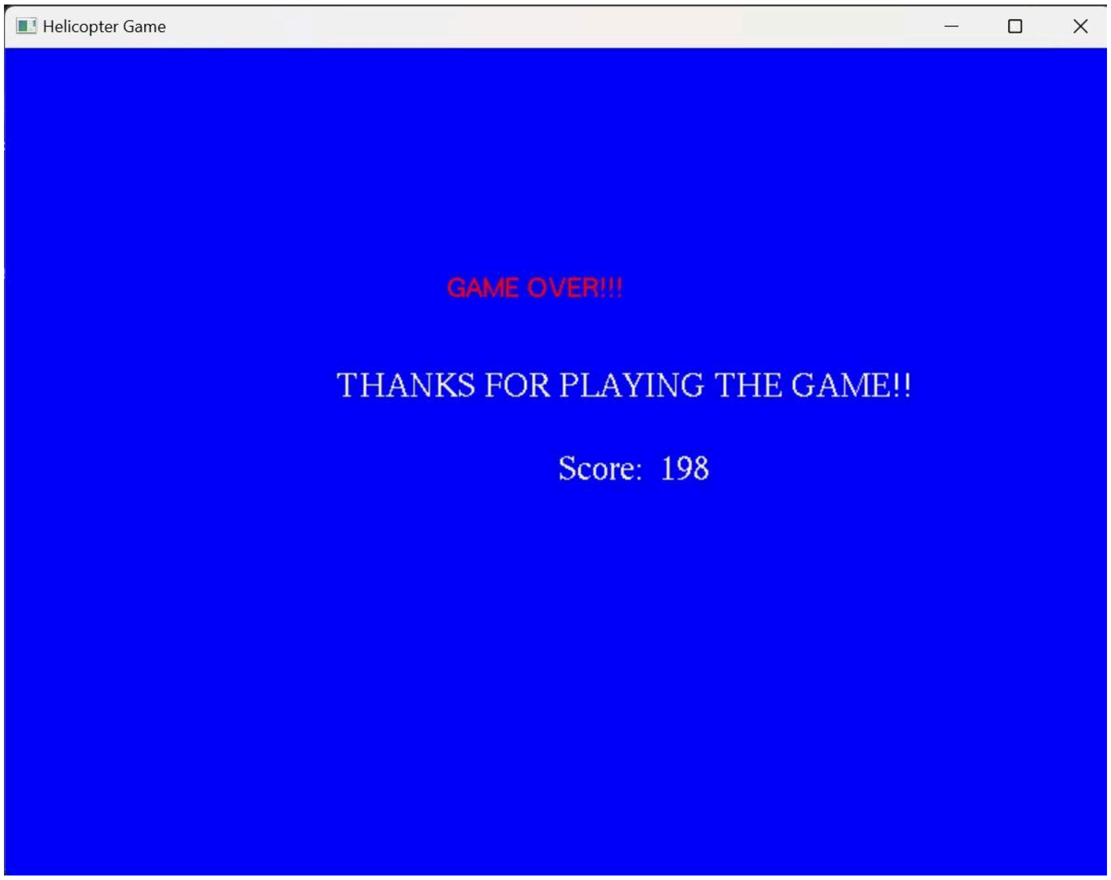
Submitted to: Ms. Anupam Garg

Press 'C' to continue




Helicopter Game
Score: 2


Helicopter Game
Score: 4



Helicopter Game

GAME OVER!!!

THANKS FOR PLAYING THE GAME!!

Score: 198