Computer Graphics (UCS505)

Project on: Tic Tac Toe Game

Submitted By

Arpit Bansal	101903376
Piyush Bawa	101903392
Guradesh Singh	101903387

Group-2

B.E. Third Year – COE

Submitted To:

Dr. Amrita Kaur

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147004

Table of Contents

1. Introduction to Project:

Tic-tac-toe is a simple, two-player game that, if played optimally by both players, will always result in a tie. The game is also called noughts and crosses or Xs and Os.

The "X" player goes first. And, players alternate placing Xs and Os on the board until a row is completed horizontally, vertically, or diagonally, or all 9 squares are filled. If all 9 squares are filled and neither player has 3 in a row, the game is a draw.

2. Computer Graphics concepts used:

i) Glut Mouse Down

ii) Keyboard Input

iii) Raster Position

iv) Line Drawing

3. User Defined Functions:

```
 i) void initgame()
ii) void KeyPress(unsigned char key, int x, int y)
iii) void click(int button, int state, int x, int y)
iv) void drawstring(void* font, char* s, float x, float y)
v) void drawlines()
vi) void drawxo()
vii) bool checkifwin()
viii) bool checkifdraw()
ix) void view(int x, int y)
```

4. Code:

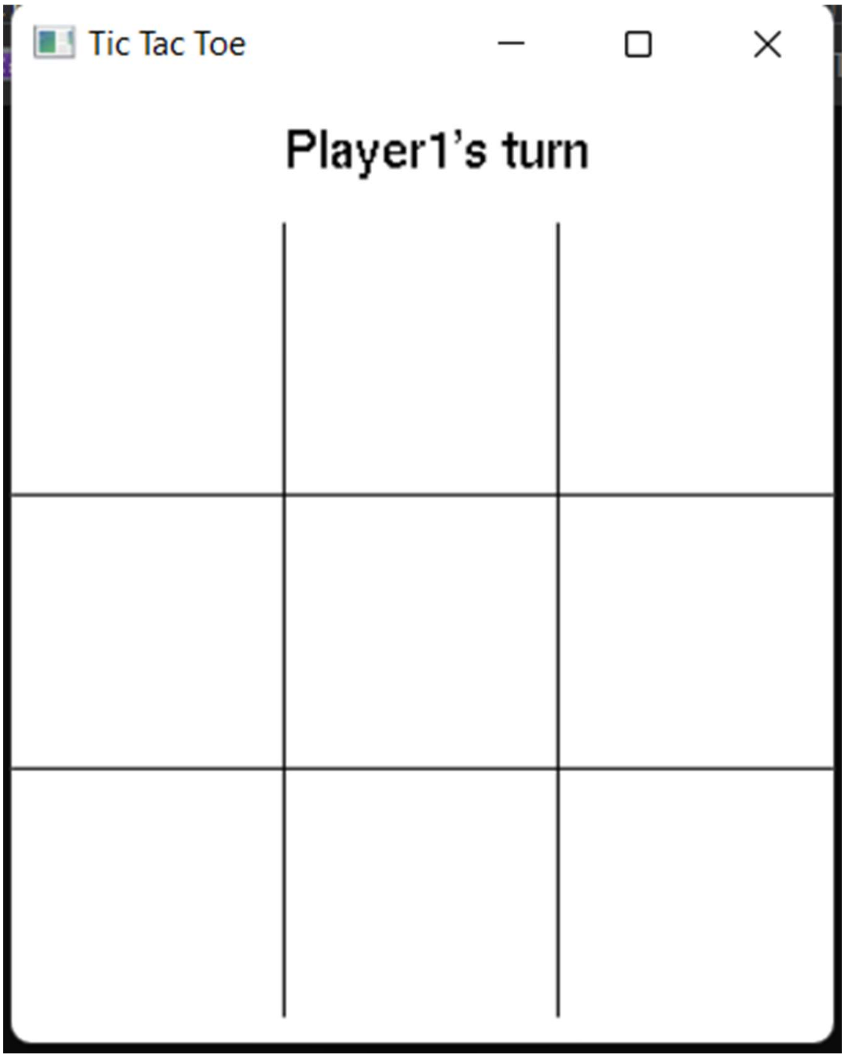
```
#include<iostream>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<string.h>
#include<GL/glut.h>

int matrix[3][3];
int playerturn;
int result;
bool gameover;

void initgame()
{
        playerturn = 1; //x starts first

        for (int i = 0; i <= 2; i++)
        {
                for (int j = 0; j <= 2; j++)
                {
                        matrix[i][j] = 0;
                }
        }
}


void KeyPress(unsigned char key, int x, int y) {
        switch (key) {
        case 'y':
                if (gameover == true)
                {
                        gameover = false;
                        initgame();
                }
                break;
        case 'n':
                if (gameover == true)
                        exit(0);
                break;
        case 27:
```

```
        exit(0);

    }
}

void click(int button, int state, int x, int y)
{
    if (gameover == false && button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        if (playerturn == 1)
        {
            if (matrix[(y - 50) / 100][x / 100] == 0)
            {
                matrix[(y - 50) / 100][x / 100] = 1;
                playerturn = 2;
            }
        }
        else
        {
            if (matrix[(y - 50) / 100][x / 100] == 0)
            {
                matrix[(y - 50) / 100][x / 100] = 2;
                playerturn = 1;
            }
        }
    }
}

void drawString(void* font, char* s, float x, float y) {
    unsigned int i;
    glRasterPos2f(x, y);
    for (i = 0; i < strlen(s); i++)
        glutBitmapCharacter(font, s[i]);
}

void drawlines()
{
    glBegin(GL_LINES);
    glColor3f(0, 0, 0);
    glVertex2f(100, 50);
    glVertex2f(100, 340);
    glVertex2f(200, 340);
    glVertex2f(200, 50);
    glVertex2f(0, 150);
    glVertex2f(300, 150);
    glVertex2f(0, 250);
    glVertex2f(300, 250);
    glEnd();
}

void drawxo()
{
    for (int i = 0; i <= 2; i++)
    {
        for (int j = 0; j <= 2; j++)
        {
            if (matrix[i][j] == 1) //if it is 1 then draw x
            {
                glBegin(GL_LINES);
                glVertex2f(50 + j * 100 - 25, 100 + i * 100 - 25);
                glVertex2f(50 + j * 100 + 25, 100 + i * 100 + 25);
                glVertex2f(50 + j * 100 - 25, 100 + i * 100 + 25);
                glVertex2f(50 + j * 100 + 25, 100 + i * 100 - 25);
                glEnd();
            }
            else if (matrix[i][j] == 2) //if it is 2 then draw o
            {
                glBegin(GL_LINE_LOOP);
                glVertex2f(50 + j * 100 - 25, 100 + i * 100 - 25);
                glVertex2f(50 + j * 100 - 25, 100 + i * 100 + 25);
                glVertex2f(50 + j * 100 + 25, 100 + i * 100 + 25);
                glVertex2f(50 + j * 100 + 25, 100 + i * 100 - 25);
                glEnd();
            }
        }
    }
}

bool checkifwin()
{
    int i, j;

    //check if there are horizontal win
    for (i = 0; i <= 2; i++)
    {
        for (j = 1; j <= 2; j++)
        {
            if (matrix[i][0] != 0 && matrix[i][0] == matrix[i][j])
            {
                if (j == 2)
                    return true;
            }
            else
                break;
        }
    }

    //check if there are vertical win
    for (i = 0; i <= 2; i++)
    {
        for (j = 1; j <= 2; j++)
        {
            if (matrix[0][i] != 0 && matrix[0][i] == matrix[j][i])
            {
                if (j == 2)
                    return true;
            }
            else
                break;
        }
    }

    //check if there is any diagonal win
    for (i = 1; i <= 2; i++)
    {
        if (matrix[0][0] != 0 && matrix[0][0] == matrix[i][i])
        {
            if (i == 2)
                return true;
        }
        else
            break;
    }
    for (i = 1; i <= 2; i++)
    {
        if (matrix[2][0] != 0 && matrix[2][0] == matrix[2 - i][i])
        {
            if (i == 2)
                return true;
        }
        else
            break;
    }
}

bool checkifdraw()
{
    int i, j;
    bool draw;

    for (i = 0; i <= 2; i++)
    {
        for (j = 0; j <= 2; j++)
        {
            if (matrix[i][j] == 0)
                return false;
        }
    }
    return true;
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(1, 1, 1, 1);
    glColor3f(0, 0, 0);
    if (playerturn == 1)
        drawString(GLUT_BITMAP_HELVETICA_18,(char*)"Player1's turn", 100, 30);
    else
        drawString(GLUT_BITMAP_HELVETICA_18, (char*)"Player2's turn", 100, 30);

    drawlines();
    drawxo();

    if (checkifwin() == true)
    {
        if (playerturn == 1)
        {
            gameover = true;
            result = 2; //player2 wins
        }
        else
        {
```

```
                        gameover = true;
                        result = 1; //player1 wins
                }
        }
        else if (checkifdraw() == true)
        {
                gameover = true;
                result = 0;
        }

        if (gameover == true)
        {
                drawString(GLUT_BITMAP_HELVETICA_18, (char*)("Game Over"), 100, 160);

                if (result == 0)
                        drawString(GLUT_BITMAP_HELVETICA_18, (char*)("Its a draw"), 110,
185);

                if (result == 1)
                        drawString(GLUT_BITMAP_HELVETICA_18, (char*)"Player1 wins", 95,
185);

                if (result == 2)
                        drawString(GLUT_BITMAP_HELVETICA_18, (char*)"Player2 wins", 95,
185);

                drawString(GLUT_BITMAP_HELVETICA_18, (char*)"Do you want to continue
(y/n)", 40, 210);
        }

        glutSwapBuffers();
}

void view(int x, int y)
{
        glViewport(0, 0, x, y);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glOrtho(0, x, y, 0, 0, 1);
        glMatrixMode(GL_MODELVIEW);
}

//main function of the program
int main(int argc, char** argv)
{
        initgame();
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
        glutInitWindowPosition(100, 100);
        glutInitWindowSize(300, 350);
        glutCreateWindow("Tic Tac Toe");
        glutReshapeFunc(view);
        glutDisplayFunc(display);
        glutKeyboardFunc(KeyPress);
        glutMouseFunc(click);
        glutIdleFunc(display);
        glutMainLoop();
}
```

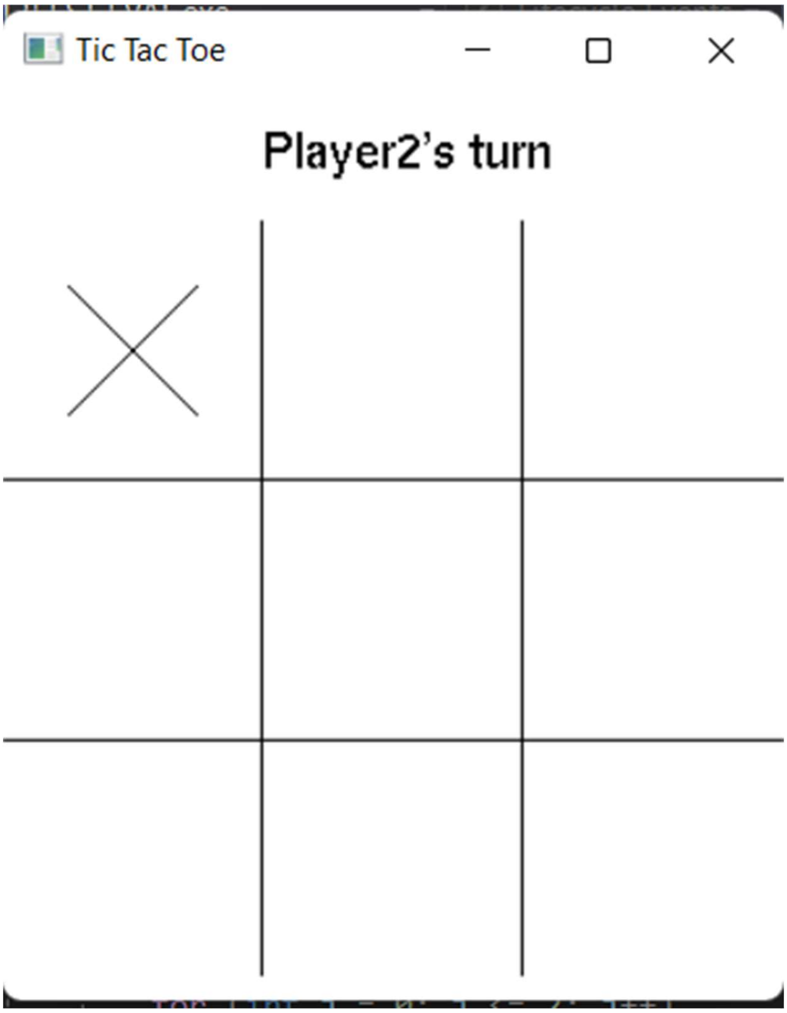
5. Output/Screenshots:

i) Starting a Game



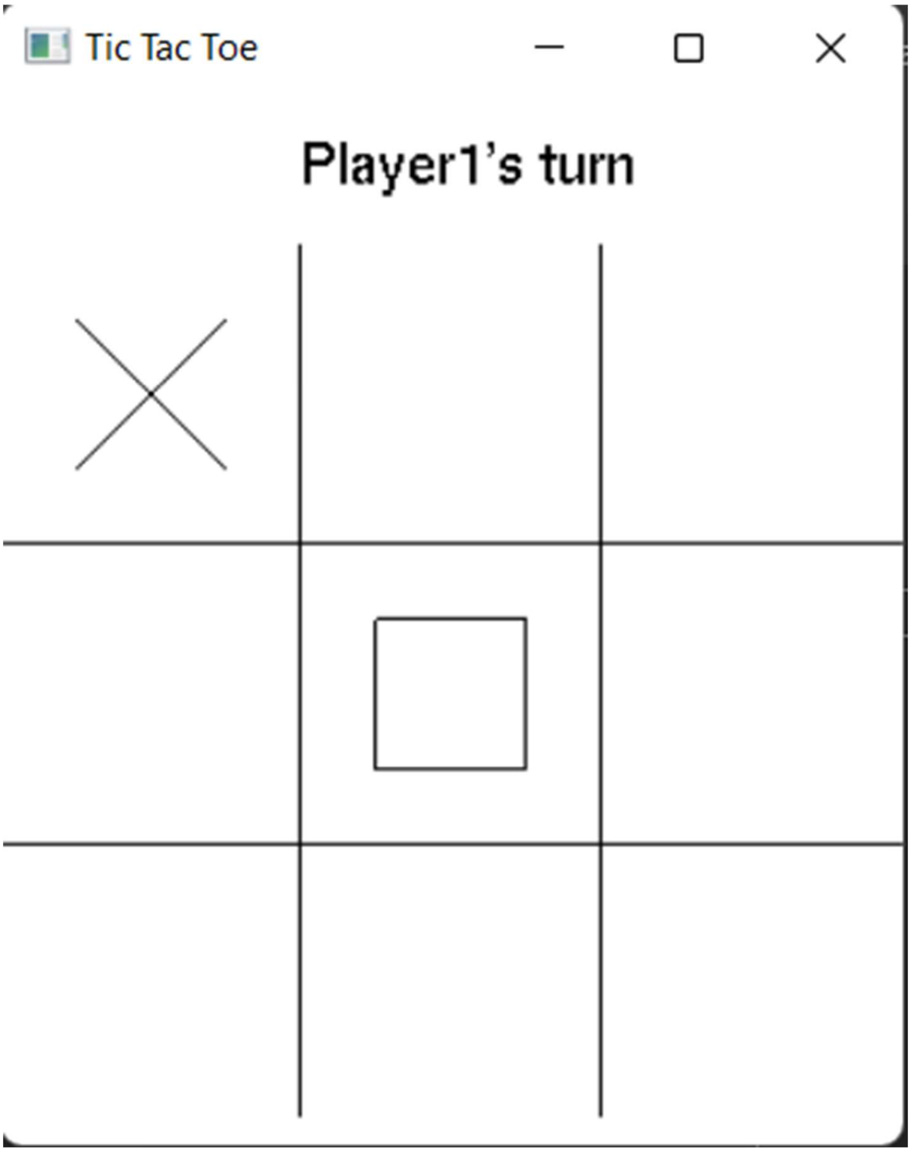
Player1’s turn

ii) Player 2’s turn



Player2’s turn

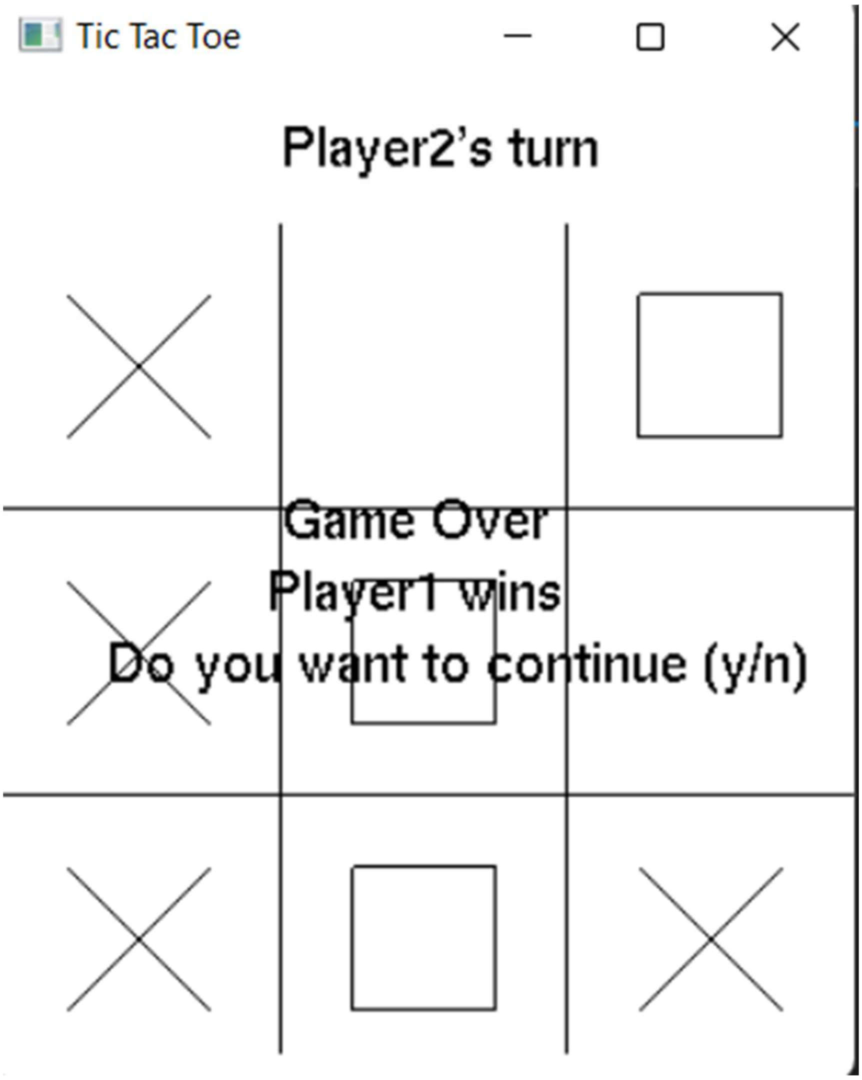
iii) Player 1’s turn



Tic Tac Toe

Player1’s turn

iv) Game Over



Tic Tac Toe

Player2's turn

Game Over

Player1 wins

Do you want to continue (y/n)