Report On

TIC-TAC-TOE- THE GAME.

Submitted in partial fulfillment of the requirements of the Course project in
Semester III of Second Year Artificial Intelligence and Data Science

by
Dnyanesh Baburao Panchal(34)
Krithik Devendra Pandey(35)
Shubham Sanjay Mohanty(30)

Supervisor
Mrs. Sejal D'Mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “TIC-TAC-TOE-THE GAME” is a bonafide work of "Dnyanesh Panchal(34),Krithik Pandey(35),Shubham Mohanty(30)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Artificial Intelligence and Data Science engineering.

Supervisor

Mrs. Sejal D’Mello

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Pg. No

OVERVIEW

Title: Tic-Tac-Toe Game (C)

Overview:
The Computer Graphics Tic-Tac-Toe Game is a project implemented in the C programming language, designed to provide an engaging and interactive gaming experience. This project leverages computer graphics techniques to create a visually appealing and user-friendly version of the classic Tic-Tac-Toe game.

Key Features:
1. Graphical User Interface (GUI): The game features a graphical user interface designed using C graphics libraries, offering a visually appealing and intuitive game board.

2. User Interactivity: Players can take turns by clicking on the grid to place their X or O marks, providing a seamless and interactive gaming experience.

3. Real-time Graphics: The game updates the display in real-time, showing the state of the game board and highlighting the winning moves or a tie when the game ends.

4. Winning Detection: The program includes logic to detect when a player has won the game, ensuring that the game concludes when a player achieves a winning combination.

5. Player Feedback: The game provides feedback to players, such as highlighting the winning line and displaying a message when the game ends.

6. Restart and Quit Options: Players can choose to restart the game or quit, returning to the main menu for a new match or to exit the game.

7. Game Sound Effects: Optional sound effects can be added to enhance the gaming experience, with audio cues for player moves, wins, and ties.

HOW TO PLAY:-

1)Press enter to start the game

2) Press Space-bar to enter ‘x’ or ‘o’

3) Play till the game ends.

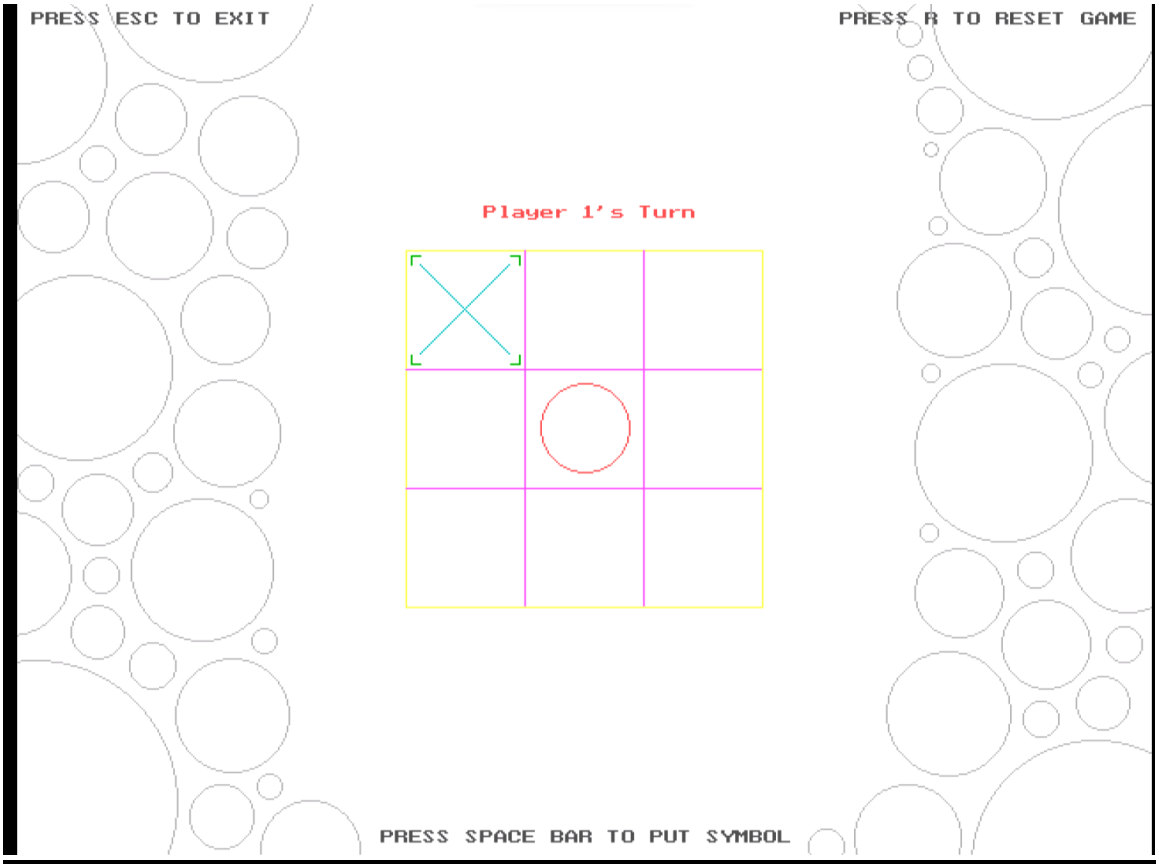
PROGRAM:-

```
#include<math.h>
#include<process.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>

void main()
{
    int
gd=DETECT,gm,i=0,j=0,ch,x=0,y=0,t=0,place[9]={0,0,0,0,0,0,0,0,0},mato[3][3
]={0,0,0,0,0,0,0,0,0},matx[3][3]={0,0,0,0,0,0,0,0,0},matchk=0;

    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    cleardevice();

    setbkcolor(15);  delay(300);
    settextstyle(1,HORIZ_DIR,7);setcolor(CYAN);
    outtextxy(getmaxx()/2-53,getmaxy()/2-100-10,"T");
    outtextxy(getmaxx()/2-53+45,getmaxy()/2-100-10,"I");
    outtextxy(getmaxx()/2-53+73,getmaxy()/2-100-10,"C");
    setcolor(LIGHTRED);delay(250);
    outtextxy(getmaxx()/2-53,getmaxy()/2-50-10,"TAC");
    setcolor(LIGHTMAGENTA);delay(290);
    outtextxy(getmaxx()/2-53,getmaxy()/2-10,"TOE");

    while(1)
    {
        settextstyle(1,HORIZ_DIR,7);setcolor(CYAN);
        outtextxy(getmaxx()/2-53,getmaxy()/2-100-10,"T");
        outtextxy(getmaxx()/2-53+45,getmaxy()/2-100-10,"I");
        outtextxy(getmaxx()/2-53+73,getmaxy()/2-100-10,"C");
        setcolor(LIGHTRED);
        outtextxy(getmaxx()/2-53,getmaxy()/2-50-10,"TAC");
        setcolor(LIGHTMAGENTA);
        outtextxy(getmaxx()/2-53,getmaxy()/2-10,"TOE");
        settextstyle(15,HORIZ_DIR,1);
        for(i=0;i<201;i++)
        { delay(4);
        }
        if(i>199)
        { setcolor(1);
        outtextxy(247,getmaxy()/2+100,"Press Enter to Play");
        }
        if(kbhit())
        {
            ch=getch();
```

```
            if(ch==27) exit(0);
            else break;
        }
    }
    loop:
    cleardevice();
    setcolor(7);
    circle(45,90,10); delay(15);              circle(550,100,30);
circle(647,116,60);
    circle(80,125,30);circle(130,80,28);  delay(15);  circle(573,143,11);c
ircle(586,180,20);
    delay(15);circle(117,178,25);circle(20,120,20);
delay(15);  circle(528,167,32);circle(580,0,65);
    delay(15);circle(35,205,52);circle(75,65,20);   delay(15);  circle(520
,60,13);circle(556,253,50);
    delay(15);circle(135,132,17);circle(0,40,50);   delay(15);  circle(605
,209,7);circle(620,189,7);
    delay(15);circle(108,-
16,60);circle(118,242,30);delay(15);  circle(652,233,39);circle(515,208,5)
;
    delay(15);circle(76,264,11);circle(45,285,20);  delay(15);  circle(514
,298,5);circle(531,332,25);
    delay(15);circle(10,270,10);circle(-
5,321,35);  delay(15);  circle(574,319,10);circle(525,400,35);
    delay(15);circle(104,319,40);circle(47,322,10);
delay(15);  circle(502,470,30);circle(456,475,10);
    delay(15);circle(45,354,15);circle(76,373,13);  delay(15);  circle(580
,361,25);circle(626,311,32);
    delay(15);circle(121,401,32);circle(139,359,7);
delay(15);  circle(624,361,10);circle(612,394,15);
    delay(15);circle(142,441,7);circle(115,458,18);
delay(15);  circle(608,485,70);circle(577,403,10);
    delay(15);circle(10,450,80);circle(165,477,28);
delay(15);  circle(509,36,7);circle(503,17,7);
    delay(15);circle(136,279,5);circle(515,82,4);   delay(15);  circle(490
,-9,18);circle(519,125,5);

    delay(50);setcolor(YELLOW);rectangle(getmaxx()/2-102+2,getmaxy()/2-
102+2,getmaxx()/2+102-1,getmaxy()/2+102-1);
    setcolor(13); delay(70);
    line(getmaxx()/2-100+65+2,getmaxy()/2-100,getmaxx()/2-
100+65+2,getmaxy()/2+100);
    line(getmaxx()/2+100-65-1,getmaxy()/2-100,getmaxx()/2+100-65-
1,getmaxy()/2+100);
    line(getmaxx()/2-100,getmaxy()/2-100+65+2,getmaxx()/2+100,getmaxy()/2-
100+65+2);
    line(getmaxx()/2-100,getmaxy()/2+100-65-
1,getmaxx()/2+100,getmaxy()/2+100-65-1);
    delay(25);
```

```
    while(1)
    {
        setcolor(7);circle(45,90,10);           circle(550,100,30);circle(
647,116,60);
        circle(80,125,30);circle(130,80,28);    circle(573,143,11);circle(
586,180,20);
        circle(117,178,25);circle(20,120,20);   circle(528,167,32);circle(
580,0,65);
        circle(35,205,52);circle(75,65,20);     circle(520,60,13);circle(5
56,253,50);
        circle(135,132,17);circle(0,40,50);     circle(605,209,7);circle(6
20,189,7);
        circle(108,-
16,60);circle(118,242,30);  circle(652,233,39);circle(515,208,5);
        circle(76,264,11);circle(45,285,20);    circle(514,298,5);circle(5
31,332,25);
        circle(10,270,10);circle(-
5,321,35);    circle(574,319,10);circle(525,400,35);
        circle(104,319,40);circle(47,322,10);   circle(502,470,30);circle(
456,475,10);
        circle(45,354,15);circle(76,373,13);    circle(580,361,25);circle(
626,311,32);
        circle(121,401,32);circle(139,359,7);   circle(624,361,10);circle(
612,394,15);
        circle(142,441,7);circle(115,458,18);   circle(608,485,70);circle(
577,403,10);
        circle(10,450,80);circle(165,477,28);   circle(509,36,7);circle(50
3,17,7);
        circle(136,279,5);circle(515,82,4);     circle(490,-
9,18);circle(519,125,5);
```

SNEAK-PEEK:-


PRESS ESC TO EXIT
PRESS R TO RESET GAME
Player 1's Turn
PRESS SPACE BAR TO PUT SYMBOL

TECHNOLOGIES USED:-

Creating a Tic-Tac-Toe game in C for computer graphics involves several key technologies and concepts:

1. **Graphics Library (e.g., OpenGL)**: Most computer graphics in C are built upon graphics libraries like OpenGL or DirectX. These libraries provide functions for rendering graphics on the screen.

2. **2D Rendering**: Tic-Tac-Toe is typically a 2D game. You would use the graphics library to draw the game board, Xs, and Os on the screen.

3. **Game Logic**: Implement the rules of Tic-Tac-Toe. This includes checking for a winning condition, tracking player turns, and preventing illegal moves.

5. **Data Structures**: Use data structures like arrays or matrices to represent the game board. These data structures help you track the state of the game and make decisions based on it.

6. **Rendering Text**: Display text on the screen for game messages, like announcing the winner or indicating whose turn it is.

7. **Window Management**: Libraries like SDL or GLFW can be used to create a window for your game, manage screen resolution, and handle window events.

10. **Audio (optional)**: You can use audio libraries like OpenAL to add sound effects and background music to your game.

EXPLANATION:-

1. Graphics Library Integration: The project begins with integrating a graphics library (e.g., SDL) into the C program. The graphics library handles window creation, rendering graphics elements, and capturing user input. This integration allows for the creation of a graphical user interface for the game.

2. Game Logic: The game logic is responsible for managing the state of the game. This includes maintaining the Tic-Tac-Toe board, keeping track of player turns, and checking for game outcomes (win, draw, or continue). You'll need data structures to represent the game board, typically a 2D array to store X and O moves.

3. Graphics Elements: The graphics library is used to create graphical elements, such as the Tic-Tac-Toe grid, X and O symbols, and buttons for restarting the game. These elements are drawn on the screen and provide a visual representation of the game.

4. User Input Handling: The program must capture user input. This includes detecting mouse clicks on the game board and buttons. Keyboard input may also be used for actions like restarting the game. Input events trigger actions in the game logic, such as making a move or restarting the game.

5. Win Detection: A crucial part of the game logic is the function that checks for a win or a draw. It examines the positions of X and O symbols on the game board to determine the game's outcome. If a player has a winning combination (three in a row, column, or diagonal), the game should declare that player as the winner. If the board is filled with no winner, the game ends in a draw.

6. UI Updates: As the game progresses, the graphical user interface needs to reflect the current state of the game. This includes updating the board to display X and O moves, highlighting the winning combination, and displaying the game's outcome (win, draw, or continue).

7. Restart Option: The interface should include a "Play Again" button or similar feature to allow players to restart the game after it ends, providing a smoother gaming experience.

These components work together to create a fully functional Tic-Tac-Toe game with computer graphics in C. The graphics library helps create an

engaging and visually appealing user interface, while the game logic ensures that the rules of the game are followed and that the outcome is accurately determined.