Summer Training Project

**Tic Tac Toe Game using C++**

## School of Computer Science & Engineering

**LOVELY PROFESSIONAL UNIVERSITY**

July 2023



**Submitted by**

**Amit Kumar Dansena**

**12111643**

**DECLARATION**

I Amit Kumar Dansena, student of SEMESTER- IV of Lovely Professional University, Phagwara, Jalandhar, do hereby declare that the project is undertaken by me as a part of my University curriculum and skill development with industrial application.

Name of the student : Amit Kumar Dansena

Registration Number : 12111643

**ACKNOWLEDGEMENT**

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. Special thanks to my University and CipherSchools mentors whose help, stimulating suggestion and encouragement helped me in all time of fabrication process and writing this report. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

I would also like to acknowledge with much appreciation the crucial role of the mentor, who gave me guidance for the completion of the project.

**PREFACE**

This report is an introduction to the tic tac toe game using c++ programming language. Anyone who don’t know the basic can understand the and get the knowledge. The main purpose of this report is to develop the tic tac toe game in C++ language.

The report contain code which gives good idea of making tic tac toe game in C++ language.

The idea of making this game and report is given by the course coordinator.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | PREFACE………………………………………………………………... |  |
| 2 | ACKNOWLEDGEMENTS……………………...……………………….. |  |
| 3 | WHAT IS TIC TAC TOE?.……………………………………………...… |  |
| 4 | REQUIREMENT.…………………………………………………………. |  |
| 5 | PROGRAMMING FOR TIC TAC TOE .……………………………….... |  |
| 6 | Conclusion and Reference………………………………………………….. |  |

**CHAPTER 1**

**WHAT IS TIC TAC TOE?**

TIC-TAC-TOE is not a very challenging game for human beings. If you’re an enthusiast, you’ve probably moved from the basic game to some variant like three dimensional tic-tac-toe on a larger grid. If you sit downright now to play ordinary three-by-three tic-tac-toe with a friend, what will probably happen is that every game will come out a tie. Both you and your friend can probably play perfectly, never making a mistake that would allow your opponent to win. But can you describe how you know where to move each turn? Most of the time, you probably aren’t even aware of alternative possibilities; you just look at the board and instantly anywhere you want to move. That kind of instant knowledge is great for human beings, because it makes you a fast player. But it isn’t much help in writing a computer program. For that, you have to know very explicitly what your strategy is.

**REQUIREMENT**

Requirement is to develop a Tic-Tac-Toe game using C++ that can be played in the terminal. The game should provide an interactive experience for two players to take turns and play the classic Tic-Tac-Toe game. This project will help you practice C++ programming concepts, logic implementation, and user interaction in the terminal.

Requirements:

1. Game Board:

○ Implement a 3x3 grid to represent the Tic-Tac-Toe game board.

○ Use appropriate data structures (e.g., arrays, vectors) to store and

update the board state.

2. Player Input:

○ Allow two players to take turns and enter their moves.

○ Prompt the players to enter the row and column numbers to place

their respective symbols (e.g., 'X' or 'O') on the board.

○ Validate the input to ensure it corresponds to an empty cell on the

board.

3. Game Logic:

○ Implement the game logic to determine the winner or a draw.

○ Check for winning conditions such as three symbols in a row,

column, or diagonal.

○ Display the winner or a draw message at the end of the game.

4. Display:

○ Display the current state of the board after each move.

○ Use appropriate symbols ('X' and 'O') to represent the players'

moves on the board.

5. Input Validation:

○ Validate player inputs to ensure they are within the valid range of

the board.

○ Handle invalid inputs and prompt the players to enter valid moves. 6. Play Again:

○ After completing a game, provide an option to play again.

○ If players choose to play again, reset the board and allow them to

continue playing.

○ If players choose to exit, terminate the game.

**PROGRAM FOR TIC TAC TOE**

#include <iostream>

using namespace std;

char square[10] = {'o','1','2','3','4','5','6','7','8','9'};

int checkwin();

void board();

int main()

{

int player = 1,i,choice;

char mark;

do

{

board();

player=(player%2)?1:2;

cout << "Player " << player << ", enter a number: ";

cin >> choice;

mark=(player == 1) ? 'X' : 'O';

if (choice == 1 && square[1] == '1')

square[1] = mark;

else if (choice == 2 && square[2] == '2')

square[2] = mark;

else if (choice == 3 && square[3] == '3')

square[3] = mark;

else if (choice == 4 && square[4] == '4')

square[4] = mark;

else if (choice == 5 && square[5] == '5')

square[5] = mark;

else if (choice == 6 && square[6] == '6')

square[6] = mark;

else if (choice == 7 && square[7] == '7')

square[7] = mark;

else if (choice == 8 && square[8] == '8')

square[8] = mark;

else if (choice == 9 && square[9] == '9')

square[9] = mark;

else

{

cout<<"Invalid move ";

player--;

cin.ignore();

cin.get();

}

i=checkwin();

player++;

}while(i==-1);

board();

if(i==1)

cout<<"==>\aPlayer "<<--player<<" win "; //\a for alert bell

else

cout<<"==>\aGame draw";

cin.ignore();

cin.get();

return 0;

}

int checkwin()

{

if (square[1] == square[2] && square[2] == square[3])

return 1;

else if (square[4] == square[5] && square[5] == square[6])

return 1;

else if (square[7] == square[8] && square[8] == square[9])

return 1;

else if (square[1] == square[4] && square[4] == square[7])

return 1;

else if (square[2] == square[5] && square[5] == square[8])

return 1;

else if (square[3] == square[6] && square[6] == square[9])

return 1;

else if (square[1] == square[5] && square[5] == square[9])

return 1;

else if (square[3] == square[5] && square[5] == square[7])

return 1;

else if (square[1] != '1' && square[2] != '2' && square[3] != '3'

&& square[4] != '4' && square[5] != '5' && square[6] != '6'

&& square[7] != '7' && square[8] != '8' && square[9] != '9')

return 0;

else

return -1;

}

void board()

{

system("cls");

cout << "\n\n\tTic Tac Toe\n\n";

cout << "P 1 (X) - P 2 (O)" << endl << endl;

cout << endl;

cout << " | | " << endl;

cout << " " << square[1] << " | " << square[2] << " | " << square[3] << endl;

cout << "\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" << endl;

cout << " | | " << endl;

cout << " " << square[4] << " | " << square[5] << " | " << square[6] << endl;

cout << "\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" << endl;

cout << " | | " << endl;

cout << " " << square[7] << " | " << square[8] << " | " << square[9] << endl;

cout << " | | " << endl << endl;

}

**CONCLUSION**

The program should provide an interactive interface in the terminal. Upon running the code, the players should be able to take turns and play the Tic-Tac-Toe game.

1. Display the initial game board.

2. Prompt Player 1 to enter their move (row and column numbers).

3. Update the board with Player 1's move and display the updated

board.

4. Check if Player 1 has won or if the game has resulted in a draw.

5. If the game has a winner or ended in a draw, display the

corresponding message and provide an option to play again or exit.

6. If the game is not over, continue to the next turn with Player 2.

**REFERENCE**

1. **Google**
2. **Geek For Geek**
3. **Tutorial Points**