PROJECT REPORT

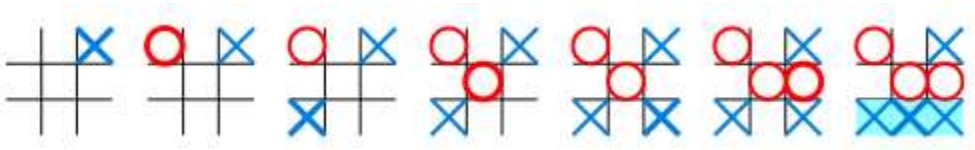
Tic-Tac-Toe Game

Mizanur Rahman - 201014061

22.05.2021

Project Overview:

My project name is Tic-Tac-Toe game. This game is very popular and is fairly simple by itself. It's actually a two players game. In this game, there is a board with n*n squares. In our game, it is 3x3 squares. The goal of Tic-Tac-Toe is to be one of the players to get three symbols in arrow-horizontally, vertically, or diagonally –on 3x3 grid. Tic Tac Toe game – is a simple console application without graphics. The game can be start by any one of the two players. The main feature of our project that we can play any level of Tic-Tac-Toe with the robotic arm. I will see the result after we finish the game and the result after playing remain save in a particular file. The most skilled Tic-Tac-Toe competitors put the first "X" in a corner when they get to compete first. This gives the challenger the most the possibilities to make a mistake. If your competitor responds by placing an O anywhere apart from the center, you can guarantee a win. To win this game, if the opponent has completed two steps toward wins, the third step should be blocked by the other player.



Object-Oriented Programming Concepts

I have used class for getting user define data type and objects for using as an constructor, inheritance for allowing to create a new class (derived class) from an existing class (base class) and polymorphism which call to a member function will cause a different function to be executed depending on the type of object that invokes the function and also file handling to provide mechanism to store the output of a program in a file and to perform various operations on it.

1. At first, I will create the basic structure of C++ program. After that I will develop the code for getting proper output of tic tac toe game.

```
1   #include<iostream>
2   using namespace std;
3
4   int main()
5   {
6
7
8     return 0;
9   }
```

2. From the main function I called displayBoard, checkWin, playerTurn.

```
int main() {
  displayBoard dis;
  playerTurn play;
  checkWin check;
  play.playTurn();
  check.checkwin();
}
```

3. I have created a class named displayBoard which contains the main board of tic tac toe game (3X3).

```
void disBoard() {
    cout << "\n\n\tTic Tac Toe\n\n";
    cout << "Player 1 (X)  -  Player 2 (O)" << 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;
  }
};
```

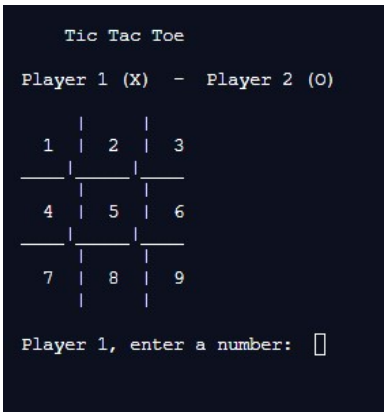
4. I also created another class named checkWin which inherits displayBoard and check the input and in this class displayBoard.

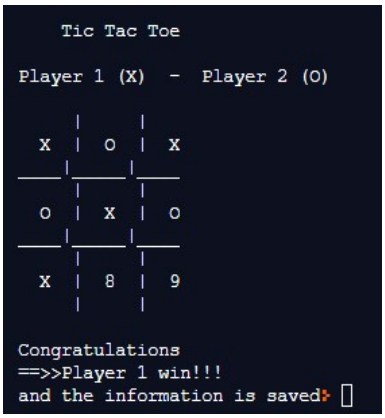
```
class checkWin : public displayBoard {
public:
    int checkwin() {
        if (square[1] == square[2] && square[2] == square[3])

            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;
        displayBoard::disBoard();
    }
};
```

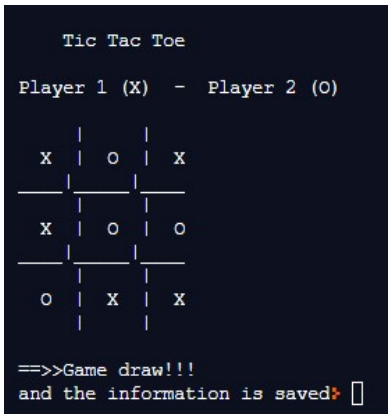
5. After that I used playerTurn class. This function inherits checkWin class and check the input valid or invalid. If it is valid it will take and input in the required position the symbol of the player and show again for another player to input. This process will continue until the full fill of displayBoard and will check the game is draw or any player win in this part I also use file handling part which will store information players win or draw every time for file handling I have used #include<fstream> hearder function.

```
class playerTurn : public checkWin {
public:
    // int player = 1,i,choice;
    int playTurn() {
        int player = 1, i, choice;

        char mark;
        while (checkwin() == -1) {
            displayBoard::disBoard();
            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 ";

            if (checkwin() == 1) {
                cout << "Congratulations" << endl;
                cout << "==>>Player " << --player << " win!!! " << endl;
                ofstream file;
                file.open("File.txt", ios::out | ios::app);
                file << "==>>Player " << player << " win!!! " << endl;
                cout << "and the information is saved";
                file.close();
            } else {
                cout << "==>>Game draw!!!" << endl;
                ofstream file;
                file.open("File.txt", ios::out | ios::app);
                file << "==>>Game draw " << endl;
                cout << "and the information is saved";
                file.close();
            }
            // cin.ignore();
            // cin.get();
            return 0;
        }
    };
```

6. If any player can take three input horizontally, vertically, or diagonally, will win the game other wise the game will draw and will show the result .







7. After playing the game user can see the stored information means the game is drawn of any player and how many time paly players play and get the result.

File.txt

```
1
2   ==>>Player 1 win!!!
3   ==>>Player 2 win!!!
4   ==>>Game draw
5
```

Repl Link: https://replit.com/join/ngeqoqlz-mdmizanurrahman

Detailed Source Code:

```
#include <iostream>
#include <fstream>
using namespace std;
/************************************************************************
   FUNCTION TO DRAW BOARD OF TIC TAC TOE WITH PLAYERS MARK
************************************************************************/
class displayBoard {
public:
  char square[10] = {'o', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
  void disBoard() {
    cout << "\n\n\tTic Tac Toe\n\n";
    cout << "Player 1 (X)  -  Player 2 (O)" << 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;
  }
};
class checkWin : public displayBoard {
public:
  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;
    displayBoard::disBoard();
  }
};
class playerTurn : public checkWin {
public:
  // int player = 1,i,choice;
  int playTurn() {
    int player = 1, i, choice;

    char mark;
    while (checkwin() == -1) {
      displayBoard::disBoard();
      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();
      }

      player++;
    }
    disBoard();
    if (checkwin() == 1) {
      cout << "Congratulations" << endl;
      cout << "==>>Player " << --player << " win!!! " << endl;
      ofstream file;
      file.open("File.txt", ios::out | ios::app);
      file << "==>>Player " << player << " win!!! " << endl;
      cout << "and the information is saved";
      file.close();
    } else {
      cout << "==>>Game draw!!!" << endl;
      ofstream file;
      file.open("File.txt", ios::out | ios::app);
      file << "==>>Game draw " << endl;
      cout << "and the information is saved";
      file.close();
    }
    // cin.ignore();
    // cin.get();
    return 0;
  }
};
int main() {
  displayBoard dis;
```

```
  playerTurn play;
  checkWin check;
  play.playTurn();
  check.checkwin();
}
```

The End