TIC TAC TOE

AICP Internship Task

DONE BY WAREESHA NAVEED

DESCRIPTION:

1. **INCLUDE LIBRARIES:**

```
#include <iostream>
using namespace std;
```

- **#include<iostream>**: This line shows the input/output stream library, which uses cout and cin for printing.
- **using namespace std;** : This line shows the standard library name without the std::prefix.

2. **CLASS DEFINITION:**

```
class TTT
{
    int Grid[3][3];

public:
    TTT()
    {
        for (int i = 0; i < 3; ++i)
        {
            for (int j = 0; j < 3; ++j)
            {
                Grid[i][j] = 0;
            }
        }
    }
```

- **Class TTT:** This defines a class named TTT for the tic tac toe program.
- **private:** which is following; it is a 3*3 array to represent the game grid.
- **public:** These are functions that can be called outside the class.

3. PRINTING GRID FUNCTION:

```
void print_grid()
{
    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            if (Grid[i][j] == 1)
            {
                cout << "O";
            } else if (Grid[i][j] == 2)
            {
                cout << "X";
            } else {
                cout << ".";
            }
        }
        cout << endl;
    }
}
```

- **print_grid():** Prints the current state of the game grid. O represents player 1, X represents player 2, and . represents an empty spot.

4. VALID MOVE FUNCTION:

```
bool valid_move(int row, int col)
{
    return (row >= 0 && row < 3 && col >= 0 && col < 3 && Grid[row][col] == 0);
}
```

- **valid_move():** Checks if the move is within the grid bounds and if the chosen spot is empty.

5. MAKE MOVE FUNCTION:

```
void make_move(int player, int row, int col)
{
    if (valid_move(row, col))
    {
        Grid[row][col] = player;
    }
    else
    {
        cout << "Invalid move. Try again." << endl;
    }
}
```

- make_move(): Places the player's mark (1 for O, 2 for X) on the grid if the move is valid.

6. GAME STATUS FUNCTION:

```
int game_status()
{
    for (int i = 0; i < 3; ++i)
    {
        if (Grid[i][0] == Grid[i][1] && Grid[i][1] == Grid[i][2] && Grid[i][0] != 0)
        {
            return Grid[i][0];
        }
        if (Grid[0][i] == Grid[1][i] && Grid[1][i] == Grid[2][i] && Grid[0][i] != 0)
        {
            return Grid[0][i];
        }
    }

    if (Grid[0][0] == Grid[1][1] && Grid[1][1] == Grid[2][2] && Grid[0][0] != 0)
    {
        return Grid[0][0];
    }
    if (Grid[0][2] == Grid[1][1] && Grid[1][1] == Grid[2][0] && Grid[0][2] != 0)
    {
        return Grid[0][2];
    }

    for (int i = 0; i < 3; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            if (Grid[i][j] == 0)
            {
                return 0;
            }
        }
    }
    return 3;
}
};
```

- **status_function():** Checks if there is winner or if the game is a draw. Returns 1 if player O wins, 1 if player X wins, 3 if it's a draw, and 0 if the game is still ongoing.

7. MAIN FUNCTION:

```
int main()
{
    TTT game;
    int player = 1;
    int row, col;
    int status;

    while (true)
    {
        game.print_grid();
        cout << "Player " << (player == 1 ? "O" : "X") << ", Enter your move (row and column): ";
        cin >>row >> col;

        if (game.valid_move(row, col))
        {
            game.make_move(player, row, col);
            status = game.game_status();
            if (status == 1)
            {
                game.print_grid();
                cout << "Player O wins!" << endl;
                break;
            }
            else if (status == 2)
            {
                game.print_grid();
                cout << "Player X wins!" << endl;
                break;
            }
            else if (status == 3)
            {
                game.print_grid();
                cout << "The game is a draw!" << endl;
                break;
            }
            player = (player == 1) ? 2 : 1;
        }
        else
        {
            cout << "Invalid move. Try again." << endl;
        }
    }

    return 0;
}
```

- **main():** The main function where the game loop runs.
