Test Plan and Test Output

Tic Tac Toe game – C programming Mini project

**Contents**

[INTRODUCTION 2](#_Toc56362911)

[OBJECTIVES 3](#_Toc56362912)

[Features to be tested 3](#_Toc56362913)

[Test Plan 4](#_Toc56362914)

[Test Cases 5](#_Toc56362915)

[TEST OUTPUT 9](#_Toc56362916)

# INTRODUCTION

The project name is Tic-Tac-Toe game. Tic tac toe is one of the classic games that can only be played by two players. This game is very popular and is simple by itself. It is a two-player game. There is a board with n x n squares. In my project, I have considered 3 x 3 squares. The two players take turns filling in different marks (usually a cross and a circle) in a 3 x 3 grid. The goal of Tic-Tac- Toe is to be one of the players to get three same symbols in a row - horizontally, vertically or diagonally - on a 3 x 3 grid. The player who gets the same symbol in same row – horizontally, vertically or diagonally will be the winner of the game. Suppose if neither of them gets their respective symbol in same row, then it will be a tie. A player can choose between two symbols with his opponent, usual games use “X” and “O”. If first player chooses “X” then the second player have to play with “O” and vice versa. A player marks any of the 3x3 squares with his symbol (may be “X” or “O”) and his aim is to create a straight line horizontally or vertically or diagonally with two intensions:

a) Create a straight line before his opponent to win the game.

b) Restrict his opponent from creating a straight line first.

In case logically no one can create a straight line with his own symbol, the game results a tie. Hence there are only three possible results – Player 1 Wins, Player 2 Wins or it’s a tie.

# OBJECTIVES

This document describes the plan for testing the developed Tic Tac Toe Game. The objective of this test plan is to identify the information that should be tested and to describe the testing strategy. The test plan applies to unit testing.

Unity Test framework is used for unit testing. In unity the input and the output should be given by us so that the tool can know how to check the result and notify whether the unit is passed or not.

### Features to be tested:

1. Display Tic Tac Toe game board.
2. Take input from the user asking to make a move.
3. Check if the move made by the player is valid or not.
4. Check if there is a winner.
5. Check if there is a horizontal win or vertical win or diagonal win.
6. Check if the game is a tie.
7. Display the rules of the game.

### Test Plan:

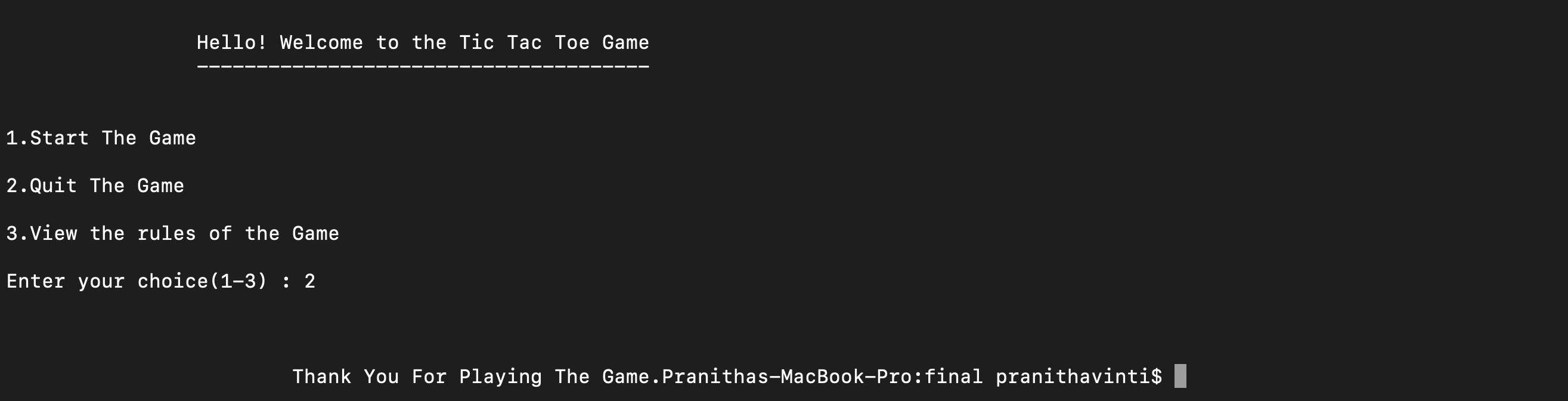
|  |  |
| --- | --- |
| S.No | Objective |
| 1. | To check if the Tic Tac Toe board game is displayed with all the vacant positions with the respective number. |
| 2. | To verify whether the user asked to enter a position where he/she wants to make a move and if the input is taken from the user. |
| 3. | To check if the input given by the user is a valid input or not. |
| 4. | In case the user gives an invalid input, check if the user is asked to select from available spots or not. |
| 5. | To check if the player turn is flipped to player 2 in case if player 1 has given a valid input. |
| 6. | To check if the player turn is flipped to player 2 in case if player 1 has given a valid input. |
| 7. | To verify if the rows are checked for a win. |
| 8. | To verify if the columns are checked for a win. |
| 9. | To verify if the diagonals are checked for a win. |
| 10. | To verify if any of the player is declared as winner in case of a win. |
| 11. | T*o* check whether in case of no winner, if the code is telling that the game is a tie. |

### Test Cases:

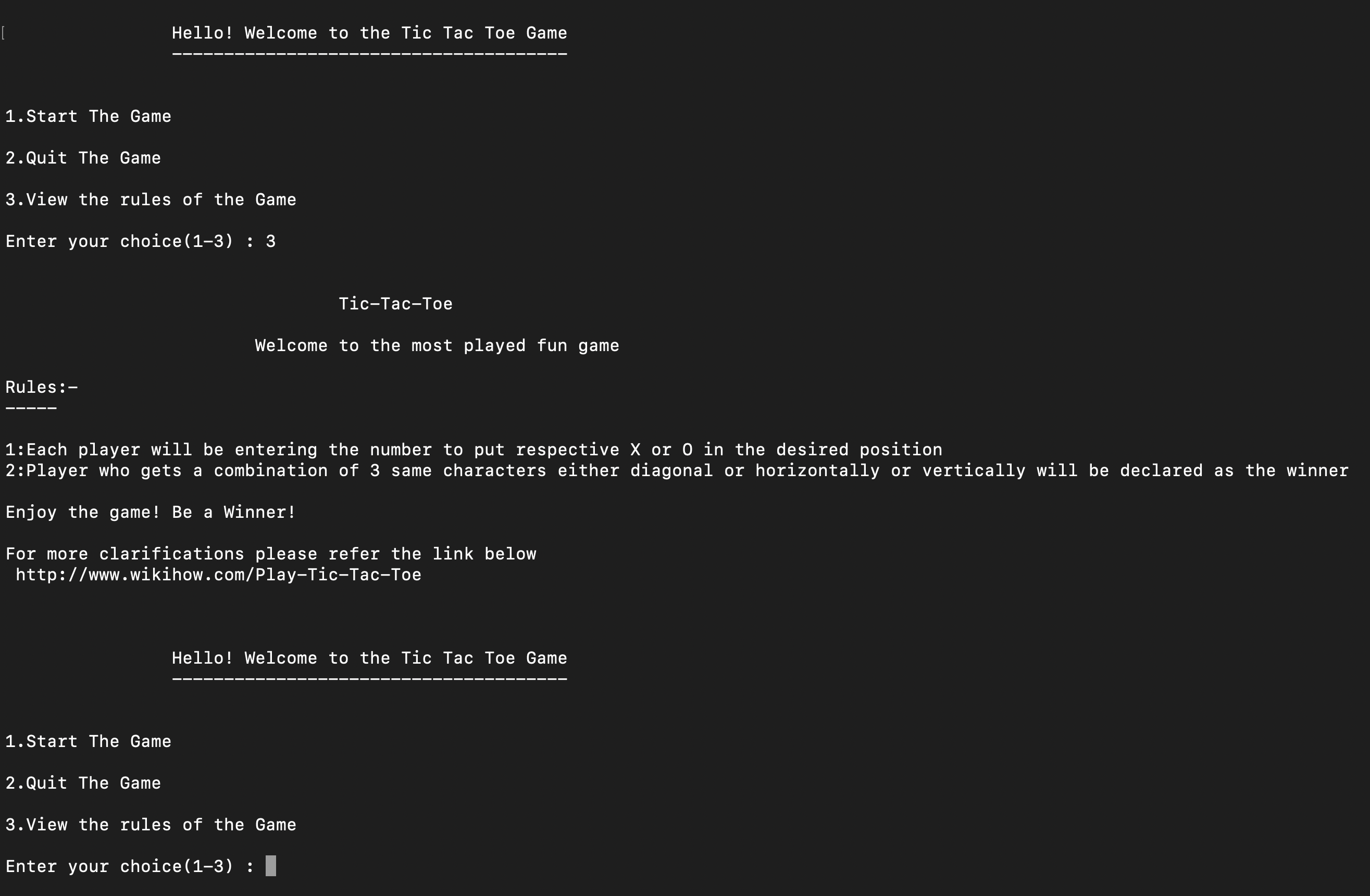
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Test case scenario | Test  Data | Expected  Result | Test result |
| 1. | To check if the Tic Tac Toe board game is displayed with all the vacant positions with the respective number. | Run the program | Should Display a heading “TIC TAC TOE” and game board with each block indicated by its respective block  number | Tic Tac Toe Game board with designated block numbers is displayed after running the program. |
| 2. | To verify whether the user asked to enter a position where he/she wants to make a move and if the input is taken from the user | Run the program | After displaying  the board, it should ask the player 1 to enter a position number. | A message asking the user to enter a position number and input from Player 1 is taken. |
| 3. | To check if the input given by the user is a valid input or not. | 3 | If block  indicated by  number 3 is vacant, Player 1’S symbol (I.e., X) should be put in place of 3 in the board and the board should be  displayed. | Since block 3 is vacant, X symbol is put in place of 3 and the board is displayed with the move made by player 1. |
| 4. | In case the user gives an invalid input, check if the user is asked to make a selection from available spots or not. | 10 | Should show a message that it is an invalid move and display the board again and ask player 1 to enter a number  again. | As 10 is not in the range, it shows invalid move and Board is displayed again and player 1 is asked to enter a number again. |
| 5. | To check if the player turn is flipped to player 2 in case if player 1 has given a valid input. | 3 | The board with move made should be displayed and now Player 2’s input should be  asked. | Board showing player 1’S move is displayed, and Player 2 is asked to enter a number. |
| 6. | To check if after Player 2’s valid input, if the player is again flipped to Player 1 or not. | 4 | The board showing moves made till now by both players should  be displayed and player 1’S input  should be asked | In block 3, X is present, and block 4 O is present and now again Player 1 is asked to enter a number. |
| 7. | To verify if the rows are checked for a win. | X X X  O O X  X O O  in top row | Message telling that Player 1 has won the game must be shown and should exit the  program. | “Congratulations Player 1 is the winner” message is displayed, and program exits |
| 8. | To verify if the columns are checked for a win. | X O O  X O X  X X O  in first column | Message telling that Player 1 has won the game must be shown and should exit the program. | “Congratulations Player 1 is the winner” message is displayed, and program exits |
| 9. | To verify if the diagonals are checked for a win. | X O O  O X X  O O X | Message telling that Player 1 has won the game must be shown and should exit the  program | “Congratulations Player 1 is the winner” message is displayed, and program exits |
| 10. | To verify if any of the player is declared as winner in case of a win. | X O X  X O O  X | Player 1 is the winner. So, game must end and should display that Player 1 is the  winner. | “Congratulations Player 1 is the winner” message is displayed, and program exits |
| 11. | To check whether in case of no winner, if the code is telling that the game is a tie. | X O X  O X X  X O O | No winner is found horizontally, vertically or  diagonally. So, message telling that the game is a tie must be  displayed. | “Game is a tie” message is displayed, and program is exited. |

# TEST OUTPUT

Output when user selects option 2



Output when user selects option 3



Output when user selects option 1

