Rules for Tic Tac Toe

Planning

RULES

Tic tac toe is a simple game that involves two players taking it in turns to place naughts and crosses onto a nine by nine grid. The game continues until either all nine spots have been filled on the board or one player has successfully placed three of their counters in sequence horizontally, vertically or diagonally. E.g. three noughts in a row. Once placed on the board. Counters cannot be moved or removed. Commonly, players ill flip a coin to determine who goes first.

PLAN

**Variables and constants that I plan to use and their uses.**

|  |  |  |
| --- | --- | --- |
| **Variable or constant** | **Data type** | **Planned use** |
| playerOneTurn | Boolean | Used to determine if it is player one’s turn |
| playerTwoTurn | Boolean | Used to determine if it is player two’s turn |
| boardFilled | Boolean | Used to determine if the board is filled or not |
| gameWon | Boolean | Used to determine if the game has been won |
| spaceFree | Boolean | Used to determine if the space that the player is trying to place a counter has already been taken |
| boardLayout | int | This will be the grid that they players will fill counters with |
| maxRow | int | Used to determine the current amount of counters in a row |

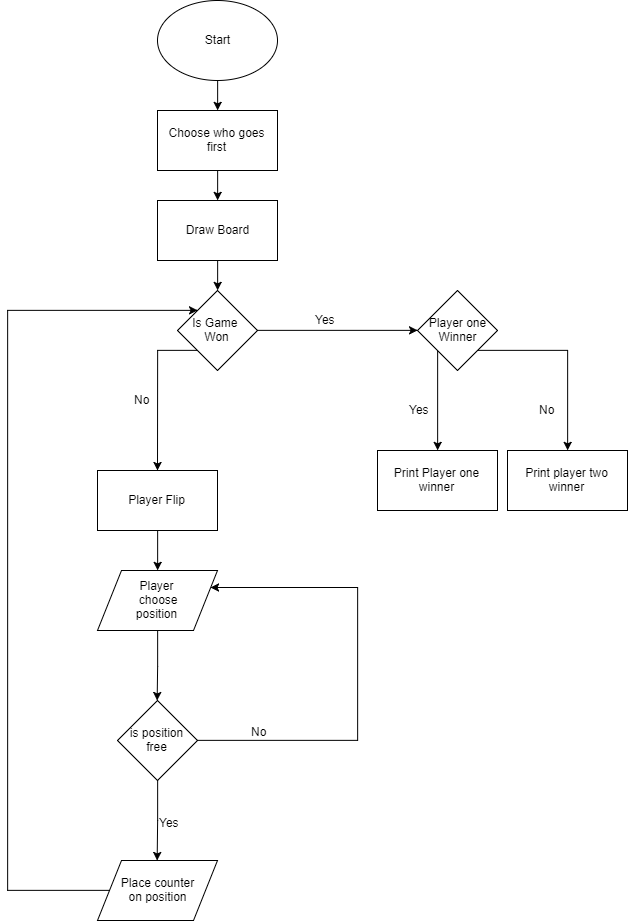
**Files needed and their uses**

|  |  |
| --- | --- |
| **File** | **Planned use** |
| Crossimage | An image file containing the cross sprite |
| Noughtimage | An image file containing the nought image |
| BoardImage | An image file containing the board layout |
| Stats | Current highest win streaks and player names are stored here |

**Data structures needed and their uses**

|  |  |
| --- | --- |
| **Data Structure** | **Planned use** |
| Free spots | Used to determine if a player can use a spot |
| Board layout | Used to place number values to each spot on the board |
| Three | Length of list used to determine winner |

**Game plan flowchart**

****

Test plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Number** | **Test Description** | **Test type** | **Data inputted** | **Expected outcome** | **Actual outcome** |
| 1 | Entering a value equal to an empty space on the grid | valid | 2,3 | A counter is placed on the correct space in the grid |  |
| 2 | Entering a value that corresponds to an already taken spot | Invalid | 2,2 | The program asks the user to enter a different value. |  |
| 3 | Entering a value that will win the game | boundary | 1,3 | The game will finish and a winning message will be displayed with the player who won. |  |
| 4 | Entering a string instead of a number | erroneous | banana | Game will ask the user to try another value that is an integer. |  |