Samger Li | **2012**

[Final Programming Project]

[Samger’s Tic-Tac-Toe Game]

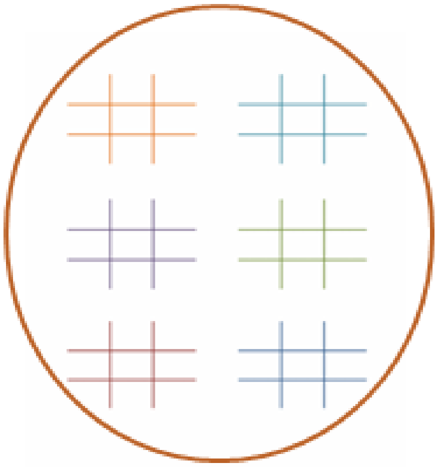


Table of Contents

Game Description

You would think that Tic-Tac-Toe is a straightforward game; in fact, you'll discover that this game isn't quite as simple as you think! Tic-Tac-Toe involves looking ahead and trying to figure out what the computer might do next. You must come up with a strategy to beat the computer. The program will consist of a two dimensional array with three rows and three columns to create the outline of the game. Each of the 9 squares will be given a number and the human player will enter a number when it is their turn to make their mark as X or O. The program will be capable of checking for moves and responding to them. For example, blocking a win by choosing a mark (as in X or O) when it sees that the player has made 2 in a row. The computer will also be able to win the game, if it sees that it only needs one more mark to win it. The program calculates the total amount of wins, losses, and ties for both the player and the computer. The program will also check after every turn if a player has won or tied. If there are no capable blocks or wins for the computer player, the program will determine the empty squares that are left after each human turn, with the list of empty squares the program will randomly choose one of them for the computer player. As the human player, you must figure out how to get three marks in a row, you also must figure out how to stop the computer from getting three marks' in a row. You must pay attention and look ahead in order to win games. Although you might not win, you will at least tie.

Rules:

1. The game is played against the computer on a 3 squares by 3 squares grid.

2. You have the option to make your mark as X or O.

3. Player X will always go first.

4. Each player will take turns putting his or her marks in an empty square.

5. The first player to get 3 of his/her marks in a row (up, down, diagonally, or across) is the winner.

6. When all 9 squares are full, the game is over. If no player has 3 marks in a row, the game ends in a tie.

Features:

1. Basic opponent artificial intelligence.

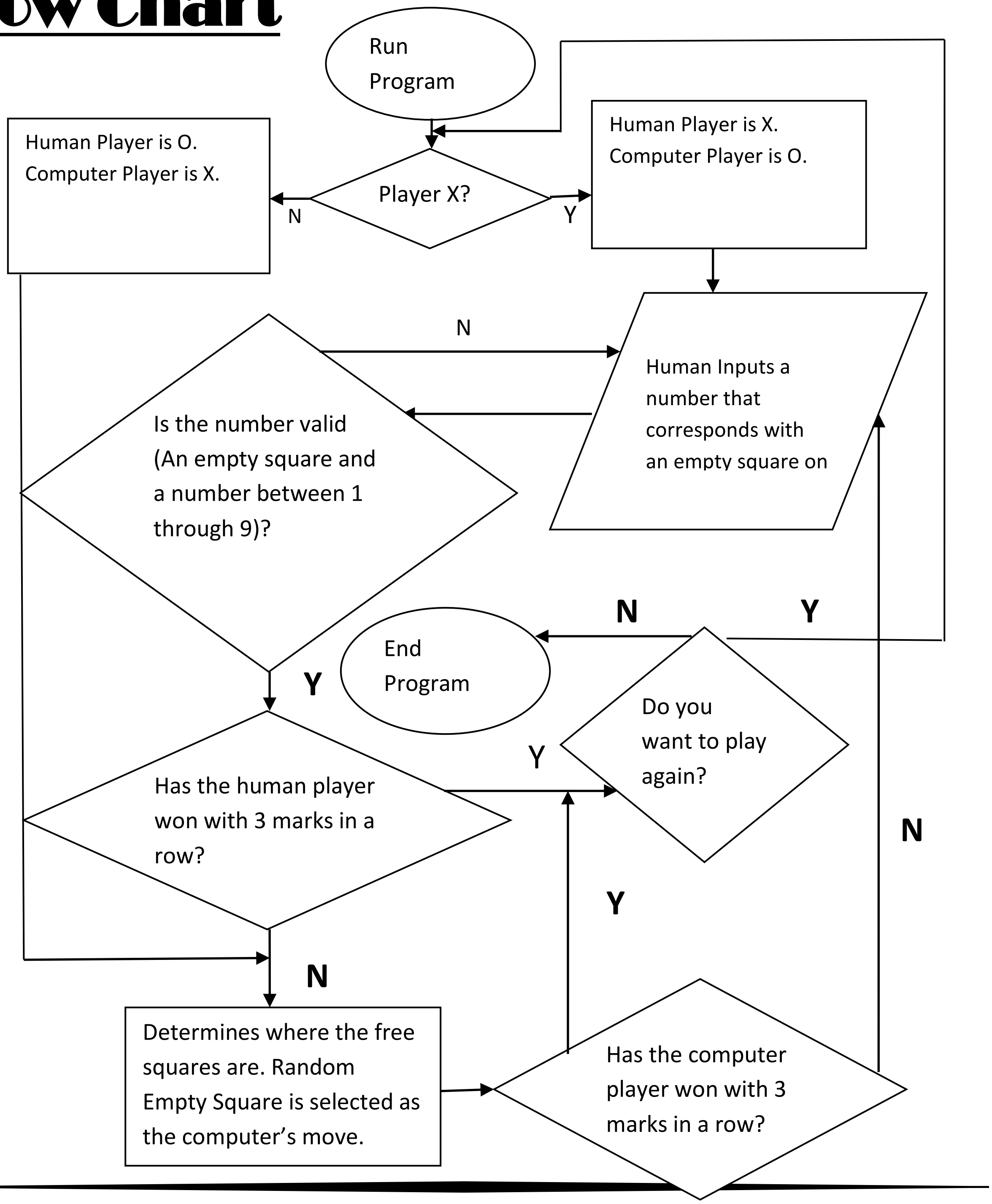
2. Simple Interface.

3. Nice Graphics.

Table 1.0: Program Input/ Processing/ Output Table

Input	Process	Output
-User chooses to be X or O	-If user is X, computer is O -If user if O, computer is X	-Player X will go first.
-User inputs their symbol in one of the squares in the 3 squares by 3 squares grid	-If square in the grid is not empty	-Prompt the player to choose a different square
	-If the square in the grid is empty	
	- If the player got 3 marks in a row (up, down, diagonally, or across)	-End game saying the player that just went has won the game
	-If there is no 3 marks in a row and there are still empty squares left	-Continue to prompt the player for a mark on an empty square in the grid
	-If there are no more squares for the player to mark and both players don't have a 3 in a row	-End game saying it was a tie game, ask the player if they want to play again
	-Number of squares on the grid decrease by 1	
-Prompt user if he/she wants to play again	-If yes, reset the game	-Prompt again if the player wants to be X or O
	-If no, end the game	-Display the results (number of wins, losses, ties)

Flow Chart


Run Program
Human Player is O.
Computer Player is X.
Player X?
N
Y
Human Player is X.
Computer Player is O.
N
Human Inputs a number that corresponds with an empty square on
Is the number valid (An empty square and a number between 1 through 9)?
N
Y
Y
End Program
Do you want to play again?
Y
Has the human player won with 3 marks in a row?
N
Y
N
Determines where the free squares are. Random Empty Square is selected as the computer's move.
Has the computer player won with 3 marks in a row?

Methods Header Descriptions

Main Method: Firstly, the main method will prompt the user for a symbol choice, either to be X or O. Whichever symbol the player chooses, the player will remain as this symbol throughout the game. It will also check to see if a player has won or tied. The main method will also prompt the user whether they would like to play again or not.

Player Method: This method will prompt the user to enter a number that corresponds with an empty square on the tic-tac-toe grid. This method will also check for if any player has won.

Computer Method: This method checks if the center spot is already occupied, if not the computer will choose to position its mark in the center. If it is occupied, this method will lead on to the sum of lines and offensive move methods.

Sum of Lines Method: Calculates the sum of each horizontal line from top to bottom, each vertical line from left to right, and both diagonal lines after each turn, which will become useful in the offensive move method, defensive move method, and special move method.

Offensive Move Method: Basically in this method, the sum of lines method has calculated for the sum of each vertical, horizontal, and diagonal line. First off, the method checks the top horizontal line, there will be a for loop that will start at 0 and ends at less than 3 and have an increment of 1, the loop will continue until an empty square is found for the computer. The method continues by checking the middle horizontal line, where the loop starts at 3 and ends at less than 6. With the pattern, this method will also check the bottom horizontal line, all three vertical lines and both diagonal lines. If the computer does not find an offensive move, it will lead to the defensive move method.

Defensive Move Method: This method is put to use if the offensive move method is not valid. This method checks if the opponent has made 2 of their symbols in a single line. If this is true, the computer will choose to block the two symbols from the player from getting their third in a row to win.

Special Move Method: This method is put to use when the human player occupies the center square. This method allows the computer to choose one of the corners.

Random Move Method: This method determines all the empty squares available for the computer. A random square will be selected from this list for the computer.

Print Board Method: This method creates the game board outline.

Check Win Method: This method checks if any player has won.

Reset Game Method: This method will reset values of the variables.

Information Method: This method displays the program's information.

Winner Line Method: This method will be used when the game is complete. A line will be drawn through the 3 symbols in a row on the grid.

Pseudo code

Declaring variables and constants:

Declare variables: symbolChoice (X or O), playAgain (Y or N), intSquare, strSquare, compOptions, valid, numPlayed, enterNum, I, choice, compNum, compChoice, emptySquares, squareCounter, winner, gamesWon, gamesLost, gamesPlayed, horizontalTop, horizontalMid, horizontalBot, verticalLeft, verticalMid, verticalRight, diagonalRight, diagonalLeft

Declare constants: player, square, and computer.

```
Main Method:
Prompt for symbol choice.
Process for winner: user, computer or tie.
Prompt user to play again.
    If user wants to play again, reset the game.
    If user doesn't want to play again, display the results and end program.
End Main Method
```

```
Player Method:
Prompt player for a number that corresponds with an empty square.
Process to see if the number entered is a valid number.
    If invalid number is entered, prompt for a valid number.
Process to see if the player has occupied square's 1, 5, and 9 on the grid.
End Player Method
```

```
Computer Method:
Determines if the center square is occupied
    If not occupied, computer takes the square.
    If occupied, move onto offensive move method.
End Computer Method
```

```
Offensive Move Method:
For loop is used to find an empty square from each line.
    If offensive move not found, defensive method is put to use.
End Offensive Move Method
```

```
Defensive Move Method:
For loop is used to determine if the opponent has already occupied two of their symbols in a single line.
    Else, Special move method is in use.
End Defensive Move Method
```

```
Special Move Method:
If human player occupies the center square, computer player will choose one of the corners.
        Else move on to random move method.
End Special Move Method

Random Move Method:
Determine the empty squares of the tic-tac-toe grid.
Random square is selected from the list of empty squares.
End Random Move Method

Line Sums Method:
Obtain the sum of each horizontal, vertical, and diagonal line of the tic-tac-toe grid.
Data is used in the offensive, defensive and special move methods.
End Line Sums Method

Check Win Method:
Process to see if any player has occupied three in a row
        If no one has won, program is continued
        If a player has won, method leads to information method
End Check Win Method

Winner Line Method:
Process until game is complete.
Identifies if the either player or the computer has won
Board game is drawn and a line goes through the 3 symbols in a row.
End Winner Line Method

Print Board Method:
Creates a 3 squares by 3 squares game board
End Print Board Method

Information Method:
Information about the program is displayed at the end of the game.
End Information Method

Reset Game Method:
Variables are reset.
End Reset Game Method
```