Neuron Series

M.Rizwan

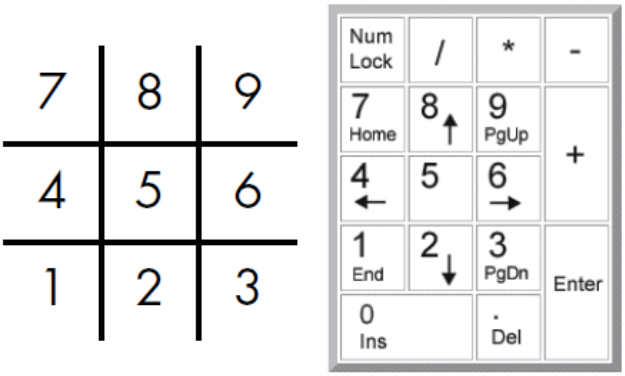
https://twitter.com/mrizwanse

Tic-Tac-Toe AI Game Project

Tic-Tac-Toe is normally played with two people. One player is **X** and the other player is **O**. Players take turns placing their **X** or **O**. If a player gets three of their marks on the board in a row, column, or diagonal, they win. When the game board fills up with neither player winning, the game ends in a draw.

An **AI** (artificial intelligence) is a computer program that can intelligently respond to the player's moves. The **AI** that plays Tic-Tac-Toe isn't complicated; it's really just a few lines of code.

To help us remember which index in the list is for which space, I'll number the board like a *keyboard's number pad*, as shown in Figure.



The game board is numbered like the keyboard's number pad.

Sample Run of Tic-Tac-Toe AI Game Project

Here's what the user sees when they run the Tic-Tac-Toe AI game project.

```
Welcome to Tic-Tac-Toe Game!
Do you want to be X or O?
 O
The computer will go first.
   |   | X
---+---+---
   |   |
---+---+---
   |   |
```

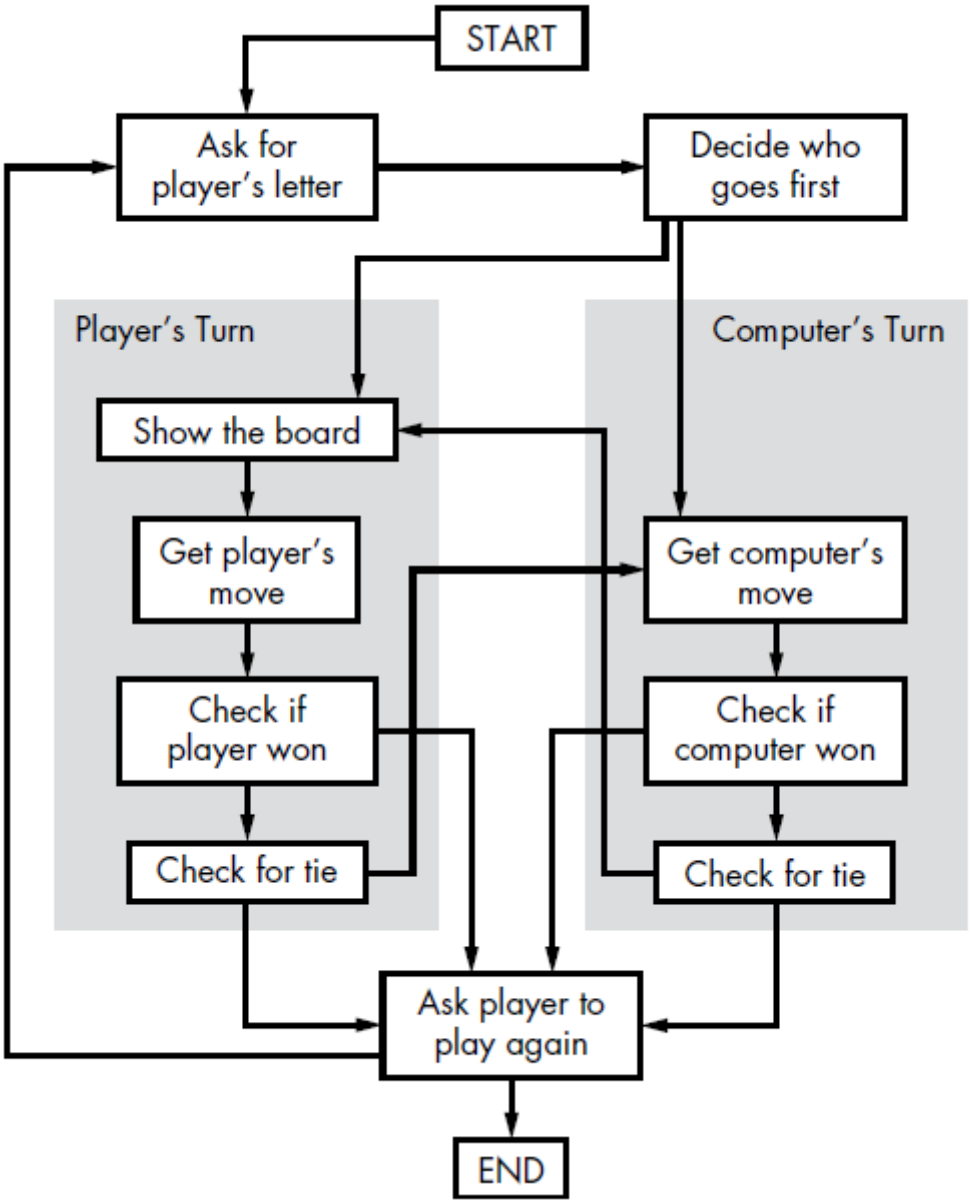
```
What is your next move? (1-9)
 3
 X |   | X
---+---+---
   |   |
---+---+---
   |   | O
What is your next move? (1-9)
 8
 X | O | X
---+---+---
   |   |
---+---+---
 X |   | O

What is your next move? (1-9)
 4
 X | O | X
---+---+---
 O | X |
---+---+---
 X |   | O
The computer has beaten you! You lose.
Do you want to play again? (yes or no)
 yes
Do you want to be X or O?
 X
The player will go first.
   |   |
---+---+---
   |   |
---+---+---
   |   |
```

```
What is your next move? (1-9)
 7
 X |   |
---+---+---
   |   |
---+---+---
 O |   |
What is your next move? (1-9)
 9
 X | O | X
---+---+---
   |   |
---+---+---
 O |   |
```

```
What is your next move? (1-9)
 3
 X | O | X
---+---+---
   | O |
---+---+---
 O |   | X
What is your next move? (1-9)
 6
 X | O | X
---+---+---
   | O | X
---+---+---
 O |   | X
Hooray! You have won the game!
Do you want to play again? (yes or no)
 no
```

Designing the Game Project

Below figure shows a **flowchart** of the Tic-Tac-Toe game project. The project starts by asking the player to choose their letter, **X** or **O**. Who takes the first turn is randomly chosen. Then the player and computer take turns making moves.



Flowchart for Tic-Tac-Toe game project

The boxes on the left side of the flowchart show what happens during the player's turn, and the ones on the right side show what happens during the computer's turn. After the player or computer makes a move, the project checks whether they won or caused a tie, and then the game switches turns. After the game is over, the project asks the player if they want to play again.

Representing the Game Board as Data

First, you must figure out how to represent the board as data in a variable. On paper, the Tic-Tac-Toe game board is drawn as a pair of horizontal lines and a pair of vertical lines, with an **X**, **O**, or empty space in each of the nine spaces.

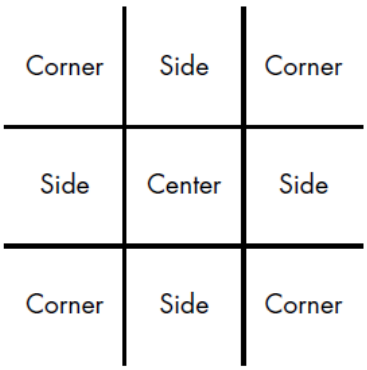
In the project, the Tic-Tac-Toe game board is represented as a list of strings. Each string represents one of the nine spaces on the game board.

The strings are either 'X' for the **X** player, 'O' for the **O** player, or a single space ' ' for a blank space.

Remember that we're laying out our game board like a number pad on a keyboard. So if a list with 10 strings was stored in a variable named board, then board[7] would be the top-left space on the board, board[8] would be the top-middle space, board[9] would be the top-right space, and so on. The project ignores the string at index 0 in the list. The player will enter a number from 1 to 9 to tell the game which space they want to move on.

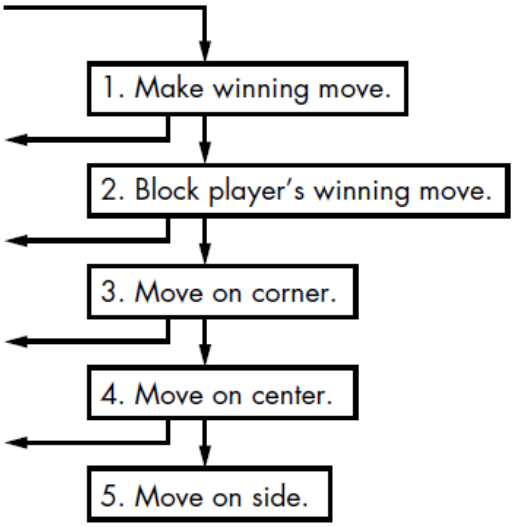
AI Strategy for playing Tic-Tac-Toe Game

The AI needs to be able to look at the game board and decide which types of spaces it will move on. To be clear, we will label three types of spaces on the Tic-Tac-Toe board: ***corners****,* ***sides****, and the* ***center****. The chart in Figure shows what each space is.*



Locations of the side, corner, and center spaces

The AI's strategy for playing Tic-Tac-Toe will follow a simple ***algorithm*** *— a finite series of instructions to compute a result. A single program can make use of several different algorithms. An algorithm can be represented with a flowchart. The Tic-Tac-Toe AI's algorithm will compute the best move to make, as shown in Figure.*



The boxes represent the five steps of the "Get computer's move" algorithm. The arrows pointing to the left go to the "Check if computer won" box.

The AI's algorithm has the following steps:

- See if there's a move the computer can make that will win the game. If there is, make that move. Otherwise, go to step 2.
- See if there's a move the player can make that will cause the computer to lose the game. If there is, move there to block the player. Otherwise, go to step 3.
- Check if any of the corner spaces (spaces 1, 3, 7, or 9) are free. If so, move there. If no corner space is free, go to step 4.
- Check if the center is free. If so, move there. If it isn't, go to step 5.
- Move on any of the side spaces (spaces 2, 4, 6, or 8). There are no more steps because the side spaces are all that's left if the execution reaches step 5.

This all takes place in the Get computer's move box on the flowchart. You could add this information to the flowchart with the boxes in above Figure.

This algorithm is implemented in getComputerMove() and the other functions that getComputerMove() calls.

Start the Source Code for Tic-Tac-Toe AI Game Project

Make sure you're using **Python 3.x**, not Python 2.x!