

# **CSC 215-01: Artificial Intelligence (Spring 2020)**



## **Mini-Project 4: Tic Tac Toe ( Simple and Wild )** **Using Minimax search**

NAME  
Prerak Shah  
Sai Sri Meghana Dharmapuri

ID  
301614729  
220250135

## **Problem Statement :**

This project aims in building a simple and wild tic tac toe game using the Minimax search. It uses an artificial intelligence approach for playing the game without a database. Artificial intelligence (AI) is the intelligence of machines and the branch of computer science that aims to create it.

## **Methodology :**

There have been two types of implementation for the tic tac toe game, simple and wild tic tac toe. In both simple and wild tic tac toe, we implemented the minimax search algorithm which is widely used in games which consist of two players. It is used in decision making and game theory to find the optimal move for the player. The game has been implemented with two participants (AI system and human) competing against each other

## **Experimental Results and Analysis :**

### **[Part 1]**

The rules for the simple tic tac toe game are:

- The first player plays x and the second player plays o.
- The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.
- If all cells are used without someone getting a three in a row, the game is a draw.

Three functions were implemented along with minimax search,

- Optimal Vs Optimal:

The results are:

```
In [17]: print('\033[1m' + 'Optimal vs Optimal Output:' + '\033[0m')
print("Games x won: {}".format(x_won))
print("Games o won: {}".format(o_won))
print("Games Drawn: {}".format(draw))
```

**Optimal vs Optimal Output:**

Games x won: 0  
Games o won: 0  
Games Drawn: 100

- Random Vs Optimal:

The results are:

```
In [21]: print('\033[1m' + 'Random vs Optimal Output:' + '\033[0m')
print("Games x won: {}".format(x_won))
print("Games o won: {}".format(o_won))
print("Games Drawn: {}".format(draw))
```

**Random vs Optimal Output:**

Games x won: 0  
Games o won: 78  
Games Drawn: 22

- You Vs Optimal:

The results are:

On playing with the machine, we ended up getting a draw.

x :

```

  x | o | x |
  ---
  o | o | x |
  ---
  x | x | o |
  ---

```

Draw!

Out[22]: 'Draw'

## [Part 2]

The rules for the wild tic tac toe game are:

- On each turn a player can play either x or o.
- The player who gets the first three (x's or o's) in a row wins the game.
- If all cells are used without someone getting a three in a row, the game is a draw.

Three functions were implemented along with minimax search,

- Optimal Vs Optimal [ when the middle cell is chosen ] :

If both players play optimally, Ideally the first player can always win if the middle cell is chosen as his/her first move.

The result obtained:

```

In [5]: print('\033[1m' + 'Optimal vs Optimal Output with 1st input in middle cell:' + '\033[0m')
print("Games P1 won: {}".format(p1_won))
print("Games P2 won: {}".format(p2_won))
print("Games Drawn: {}".format(draw))

```

```

Optimal vs Optimal Output with 1st input in middle cell:
Games P1 won: 50
Games P2 won: 0
Games Drawn: 0

```

- Optimal Vs Optimal [ when the middle cell is not chosen ] :

If both players play optimally, ideally the game will be always a draw if the first player does NOT choose the middle cell as his/her first move.

The result obtained:

```

In [8]: print('\033[1m' + 'Optimal vs Optimal Output with 1st input not in middle cell:' + '\033[0m')
print("Games P1 won: {}".format(p1_won))
print("Games P2 won: {}".format(p2_won))
print("Games Drawn: {}".format(draw))

```

```

Optimal vs Optimal Output with 1st input not in middle cell:
Games P1 won: 0
Games P2 won: 0
Games Drawn: 50

```

- Random Vs Optimal

If one player is optimal and the other is random, ideally the optimal player will win most of the games and the random player will never win.

The game is designed in a way that Player 1 plays randomly, and Player 2 plays optimally.

The results obtained are:

```
In [14]: print('\033[1m' + 'Player P1 plays randomly, and Player P2 plays optimally:' + '\033[0m')
print("Games P1 won: {}".format(p1_won))
print("Games P2 won: {}".format(p2_won))
print("Games Drawn: {}".format(draw))
```

```
Player P1 plays randomly, and Player P2 plays optimally:
Games P1 won: 0
Games P2 won: 48
Games Drawn: 2
```

- You Vs Optimal

The result obtained:

```
Please enter a valid cell (3, 4, 5, 6, 7): 6
Please enter a valid character (x, o): o
```

P1 :

```
x | o | x |
-----
3 | 4 | 5 |
-----
o | 7 | o |
-----
```

P2 :

```
x | o | x |
-----
3 | 4 | 5 |
-----
o | o | o |
-----
```

P2 Wins!

```
Out[0]: 'P2'
```

### Task Division and Project Reflection:

1. Simple tic tac toe – Combined effort
2. Wild tic tac toe – Combined effort
3. Documentation – Combined effort

### Challenges faced:

1. The wild tic tac toe took a lot of time to implement and run
2. We were unable to understand how to give the choice of 'x' and 'o'