

# INT404 ARTIFICIAL INTELLIGENCE

**PROJECT TITLE: TIC-TAC-TOE USING MINIMAX ALGORITHM**

## Name : K. Yaswanth

**Roll No : 54**

**Reg No : 11801592**

**Section : K18JC Submitted to : Sagar Pande sir**

**Table of Contents**

1. Introduction 3
   1. Objective if the project
   2. Motivation
   3. Context
2. Conceptual Framework 3-4
   1. Libraries used
   2. Minimax Algorithm
3. Scope of the project 5
   1. Milestones
4. Screenshots 05-12
   1. Code Screen Shots
   2. Working or output
5. Reference 12
6. Conclusion 12

**Report**

**Introduction:**

**Project Name**: Tic-Tac-Toe using minimax algorithm.

**Objective of the project** : Create an Ai which can play Tic-Tac-Toe game which can make its own choices and develop moves on its own

**Motivation**: I chose this project because I play games a lot so I wanted to do something which belongs to the field of games and chose Tic-Tac-Toe as I am a beginner and I wanted it to be simple.

**Context:**

**Tic-Tac-Toe** is a game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

Tic-tac-toe using Artificial intelligence we can create a game which can make its own choices and develop moves on its own.

The objective is to implement a Tic-tac-toe game with integrated ai which can make own moves using minmax algorithm.

### **Conceptual Framework**:

**Language Used**: Python

### **Libraries Used**:

1. Math library
2. Platform
3. Time

**Math** : This module provides access to the mathematical functions defined by the c standard

These functions cannot be used with complex numbers; use the functions of the same name from the cmath module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don’t is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an

exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

**Platform** : The platform module in Python is used to access the underlying platform’s data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform’s hardware, operating system, and interpreter version information where the program is running.

### **Minimax Algorithm**:

Minimax is an artificial intelligence applied in two player games, such as tic-tac- toe, checkers, chess and go. This games are known as zero-sum games, because in a mathematical representation: one player wins (+1) and other player loses (-

1. or both of anyone not to win (0).

The algorithm search, recursively, the best move that leads the Max player to win or not lose (draw). It consider the current state of the game and the available moves at that state, then for each valid move it plays (alternating min and max) until it finds a terminal state (win, draw or lose).

Code for minimax:

minimax(state, depth, player)

if (player = max) then

best = [null, -infinity]

else

best = [null, +infinity]

if (depth = 0 or gameover) then

score = evaluate this state for player return [null, score]

for each valid move m for player in state s do execute move m on s

[move, score] = minimax(s, depth - 1, -player) undo move m on s

if (player = max) then

if score > best.score then best = [move, score]

else

if score < best.score then best = [move, score]

return best

end

**Scope of The project:**

Milestones:

13/02/2020 – learning about minimax.

15/02/2020 - Application of minimax in two player games.

20/02/2020 - researching about libraries required to implement the Tic-Tac-Toe.

05/03/2020 - Initial Code

06/03/2020 - setting the minimax.

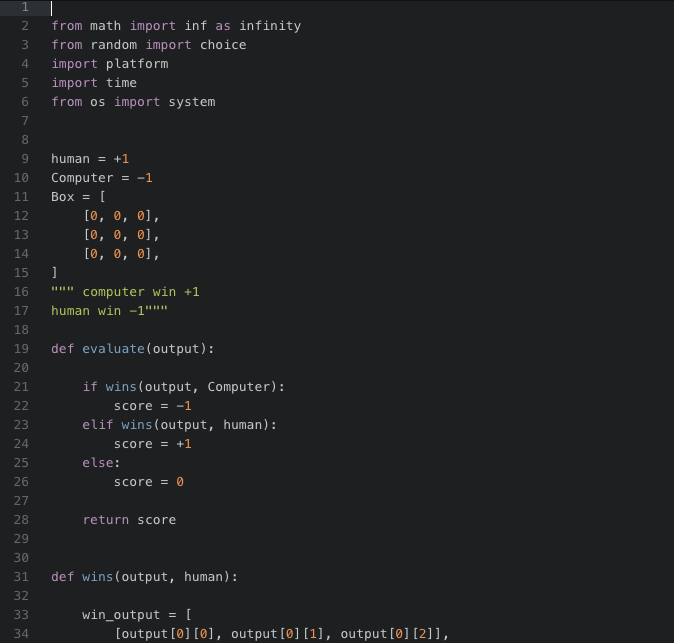
11/03/2020 – Testing the code.

16/03/2020 - Rechecking and adding extra details to the code like result display.

19/03/2020 - code for selecting player turn.

21/03/2020 - code is completed(code is fully functional )

**Code:**

****

### 

### 

### 

### **Code for turns of human and computer choice:**

### 

### **Code for Valid movie checking:**

### 

### **Main Function Cell:**

### 

### 

### 

### **Working :**

### 

### 

### 

### **References:**

* + I researched minimax algorithm and learnt it from GeeksforGeek’s.
  + How to use minimax algorithm for two-player games from YouTube
  + And researched the libraries required and the concept from YouTube.

### **Conclusion:**

Tic-tac-toe implemented with minimax algorithm looks for the best case in the tree and proceeds with the best case it chooses on its own based on possible ways. By doing this project I came to know uses of automation and how it can change the future of technology.