I’A capaz de jugar una partida de Tres en línea

Inteligencia Artificial capaz de jugar una partida de Tres en línea empleando búsqueda en profundidad de movimientos predefinidos en un árbol de juego.

Sebastián Franco Gómez

Andrés Mejía Vélez

Jhoao Martínez

Santiago Sánchez Pulgarín

*Ing. Sistemas, Universidad Tecnológica de Pereira, Pereira – Colombia*

andres.mejia@utp.edu.co

s.sanchez123@utp.edu.co

jhoao.martinez@utp.edu.co

Sebastian.franco@utp.edu.co

***Resumen* - En el problema se plantea una inteligencia artificial capaz de jugar tres en línea reconociendo las posiciones del tablero de juego 3x3, donde hay 2 jugadores, uno que juega con las Equis (*X)* y otro que juega con los Círculos (*O*); Gana el primeroque logre poner 3 símbolos seguidos hacia cualquier dirección (Horizontal, Vertical o Diagonales), si ningún jugador logra el objetivo, el juego se declara empate. En el programa se tiene predefinidas los movimientos en un árbol de juego, para saber cuál es la más adecuada se filtra mediante condicionales y posteriormente realizar la búsqueda en amplitud de la jugada apropiada a ese movimiento.**

***Palabras clave* – Bot, inteligencia artificial, búsqueda por amplitud, python.**

***Abstract-* The problem involves an artificial intelligence capable of playing Tic Tac Toe recognizing the positions of the 3x3 game board, where there are 2 players, one who plays with the (X) and another who plays with the Circles (O); The first player to achieve 3 consecutive symbols in any direction (Horizontal, Vertical or Diagonal) wins. If no player achieves the objective, the game is declared a draw. In the program, the movements are predefined in a game tree, to find out which is the most appropriate, it is filtered by means of conditionals, and then to carry out an in-depth search for the move appropriate to that movement.**

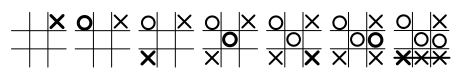
***Keywords - Bot, Artificial intelligence, search by amplitude, python.***

I. INTRODUCCIÓN

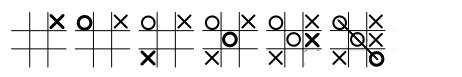
Se planea generar búsquedas en amplitud con el objetivo de encontrar una jugada que le permita ganar el juego a la IA.

Existe tres posibles finales en la partida:

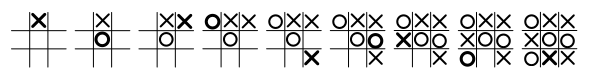
-Que gane el primer jugador (**X**):



-Que gane el segundo jugador(**O**):

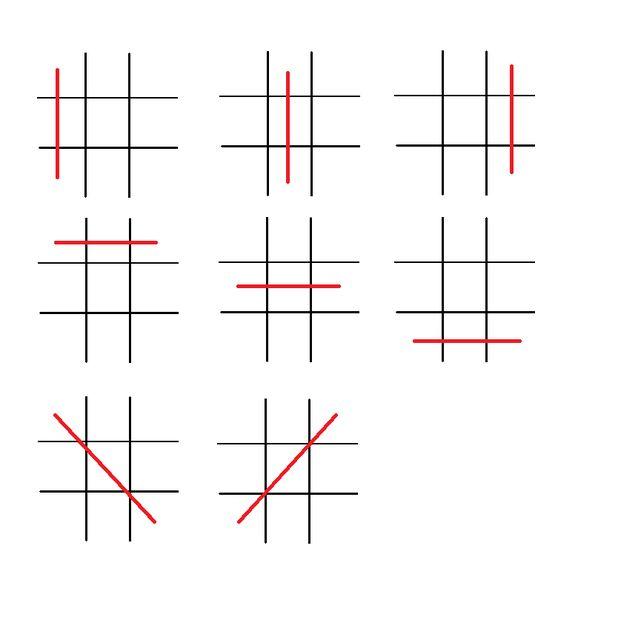


-Que termine en **empate**:



Objetivo:

Juntar 3 Símbolos Seguidos en cualquier dirección, Existen 8 posibles formas de ganar por cada jugador.



II. CONTENIDO

Para entender el funcionamiento de la inteligencia artificial desarrollada en Python la dividiremos en módulos explicados a continuación:

Módulo 1: **Definición de la importancia de las casillas en el tablero**.

-Para el módulo uno se deben definir una jerarquía para las casillas que pertenecen al tablero de juego de acuerdo a la facilidad o capacidad que tenga cada casilla para ganar una partida.

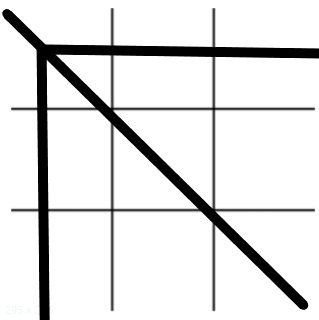
en el tablero, existen 3 tipos de casillas (esquinas, aristas y centro).

**Esquina:**

En la esquina **s**e puede ganar de tres formas

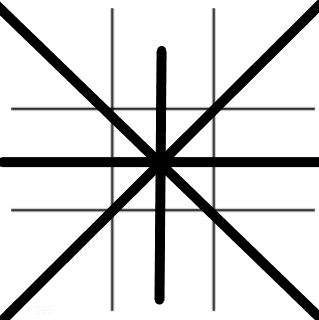
Diferentes, por lo tanto asignaremos un

3 a su nivelde importancia

****

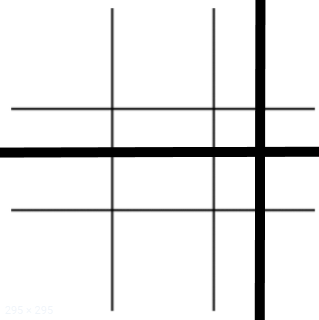
**Centro**:

El centro le asignaremos 4 a su nivel de importancia, ya que hay 4 formas de ganar usándolo.



**Arista:**

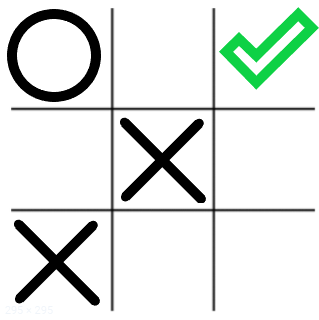
Por último la arista tiene un nivel de importancia de 2.



Módulo 2: **Cuando se gana.**

-En el módulo dos se debe establecer cuando se gana, se una partida.

Si el Bot juega con las X o O debe entender todos los posibles movimientos que lo llevaran a la victoria, Ejemplo:

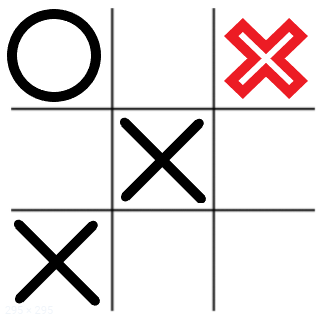


El jugador **X** debe entender que si esa casilla no es llenada por el jugador **O** su próxima jugada será la ganadora.

Módulo 3: **Cuando se pierde.**

-En el módulo tres se debe establecer cuando se pierde, se una partida.

Si el Bot juega con las O debe entender todos los posibles movimientos que lo llevaran a la derrota, Ejemplo:

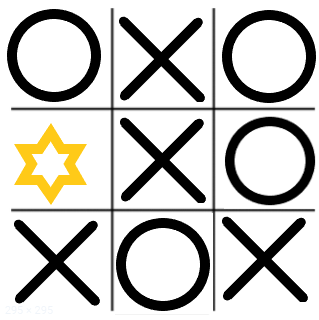


El jugador **O** debe entender que si esa casilla no es llenada perderá el juego, entonces le dará prioridad a esta casilla en particular.

Módulo 4: **Cuando se empata.**

-En el módulo cuatro se debe establecer cuando se empata, se una partida.

El Bot debe entender que si sucede este tipo de movimiento la partida no se gana, pero tampoco se pierde, Ejemplo:



1. CONCLUSIONES

Para que el Bot funcione correctamente debe cumplir con las condiciones mínimas de reconocimiento y del tablero entre ellas están:

-**Percepción de condiciones de victoria**:

Lo primero que percibe el Bot es si en la jugada actual puede ganar, si no es así continua a la siguiente capa (condiciones de derrota).

-**Percepción de condiciones de derrota:**

Después el Bot evalúa si en el siguiente movimiento puede perder, de no ser así continua a la siguiente capa (Importancia de la casilla),

-**Percepción de la importancia de la casilla:**

Por último el Bot buscara una casilla con el nivel más alto de importancia para jugar.

REFERENCIAS

[1] The Solution of a Simple Game Daniel I. A. Cohen

[2] A PERFECT TIC-TAC-TOE A.I. vcubingx