**INFORME TAREA ARBOLES DE JUEGO**

**INTEGRANTES:**

* Darian Proaño.
* Erick Rosero.
* Gustavo Guevara.
* Juan Carrillo.

**Objetivo:**

1. Implementar un menú para poder correr el programa proporcionado por el profesor de manera correcta.
2. Entender la forma en la que funciona el código y la funcionalidad e importancia del algoritmo MiniMax en el mismo.
3. Realizar las pruebas necesarias con el código ya modificado.

**Marco Teórico:**

***Descripción del juego tres en raya.***

Un juego entre dos jugadores: O y X, que se turnan para marcar espacios en un tablero de 3×3. El jugador gana si logra hacer una línea de tres símbolos: la línea puede ser horizontal, vertical o diagonal.

***Algoritmo MiniMax***

En teoría de juegos, "minimax" es un método de decisión que minimiza la pérdida máxima esperada en un juego con un oponente e información completa. Minimax es un algoritmo recursivo. La forma en que funciona minimax se puede resumir asumiendo que tu oponente elegirá el peor movimiento para ti y el mejor movimiento para ti.

**Pasos del algoritmo minimax:**

1. Generación de árboles de juego. Todos los nodos se generarán hasta que se alcance el estado final.
2. Cálculo de los valores de la función de utilidad para cada nodo terminal.
3. Calcular el valor de los nodos superiores a partir del valor de los inferiores. Según nivel si es MAX o MIN se elegirán los valores mínimos y máximos representando los movimientos del jugador y del oponente, de ahí el nombre de minimax.
4. Elegir la jugada valorando los valores que han llegado al nivel superior.

El algoritmo explorará los nodos del árbol, asignándoles un valor numérico usando una función de evaluación, comenzando desde los nodos finales y trabajando hacia la raíz. La función de asistencia determinará qué tan buena es la posición del jugador cuando llegue allí.

Ante el dilema del prisionero, minimax siempre elegirá la opción que maximice su resultado, dado que el oponente está tratando de minimizarlo y nos hace perder.

***Funcionamiento del algoritmo Min Max en un tres en raya.***

Juega al error del oponente sin buscar beneficio en el turno siguiente.

Almacena jugadas realizadas de acuerdo al movimiento.

Genera un todo el nodo hasta llegar a un estado terminal.

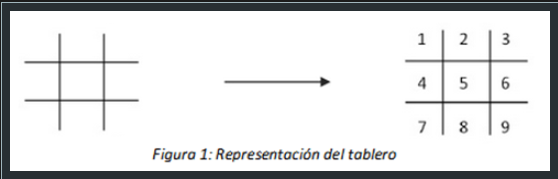
Calcula los valores de la función de utilidad para cada nodo terminal.

Calcula el valor de los nodos superiores a partir del valor de los inferiores. Alternativamente se elegirán los valores mínimos y máximos representado los movimientos del jugador y del oponente.

Explora los nodos del árbol dando, un valor numérico mediante una función, empezando por los nodos terminales y subiendo hacia la raíz.

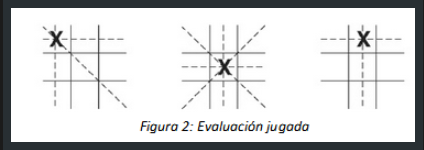
Colocar 0 o 1 en los nodos terminales dependiendo si gana Min o Max.

Inicialmente se parte de un tablero de ajedrez de 3x3 representado por un vector de nueve componentes, donde las componentes del vector corresponden a las posiciones del tablero, así:



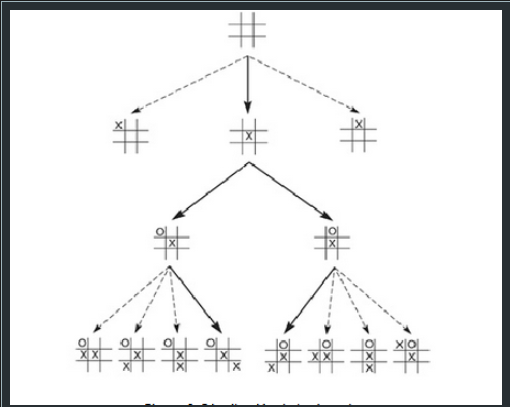
También consideramos una variable que define la posición en el vector para calcular si el campo está vacío, para calcular si el juego está completo o aún en progreso. Para ello se define una variable constante que define el estado del juego.

Para determinar cómo se evalúa cada nodo o estado de característica, cada característica se evalúa de la siguiente manera:



Para determinar cuál de todas las posiciones posibles es mejor, siga los pasos a continuación para cada posición.

* Ver si se produce la victoria. Si ocurre catalogarla como la mejor dándole el mejor puesto en la clasificación.
* En caso contrario, considerar todos los posibles movimientos que el oponente puede realizar en la siguiente jugada. Asumir que el oponente realizará este movimiento. Cualquier puesto que tenga la jugada, asignarla al nodo que se está considerando.
* El mejor nodo es el que resulte con un puesto más alto.

  
**Organización del desarrollo del trabajo.**

Todos los participantes actuaron de manera activa en la realización del presente proyecto.

**Responsabilidad de cada integrante:**

Erick Rosero: Codificación e investigación.

Darian Proaño: Informe e investigación.

Gustavo Guevara: Codificación e investigación.

Juan Carrillo: Informe e investigación.

**Presentación del problema.**

El creador del juego de Minimax, John Von Neumann, afirmó que un juego es una situación conflictiva en la que uno debe tomar una decisión sabiendo que los demás también toman decisiones, y que el resultado del conflicto se determina, de algún modo, a partir de todas las decisiones realizadas, y también, que siempre existe una forma racional de actuar en juegos de dos participantes, si los intereses que los gobiernan son completamente opuestos.

Para poner a prueba el juego de Minimax de la forma visual más simple posible, se decidió que el TicTacToe, o tres en raya, que es uno de los juegos con reglas simples entre dos personas, sería el ejemplo más adecuado para representar visual e interactivamente el Minimax.

**Conclusiones:**

1. Pudimos observar que el algorimo minimax es una herramienta de inteligencia artificial muy poderosa y sumamente eficaz, ya que siempre encuentra la forma de nunca dejar ganar a un usuario común.
2. El algoritmo de minimax es eficiente a la hora de simular ser un oponente en un juego simple de dos personas.
3. Un juego que consideramos simple, como el tres en raya, son una serie de decisiones concatenadas qué, aunque son casi instintivas para nosotros, en realidad tienen muchas opciones y condiciones detrás de cada movimiento. Y qué, si no fuera por los algoritmos recursivos como el Minimax, serían una pesadilla de hacer y producir en computadora.