

# Universidad politécnica salesiana

Estudiante: Rafael Angamarca

## El algoritmo Minimax y su aplicación

En este tutorial hablaremos sobre el algoritmo Minimax y veremos su aplicación mediante un ejemplo sencillo, el cual consistirá en el desarrollo de un tres en raya donde jugaremos contra la máquina.

Debido a la gran demanda de tecnología, la IA es una ciencia aplicada que consiste en incluir inteligencia a máquinas creadas por el hombre, estas tecnologías de IA también son aplicadas en juegos para darle una mejor interacción humano-computador.

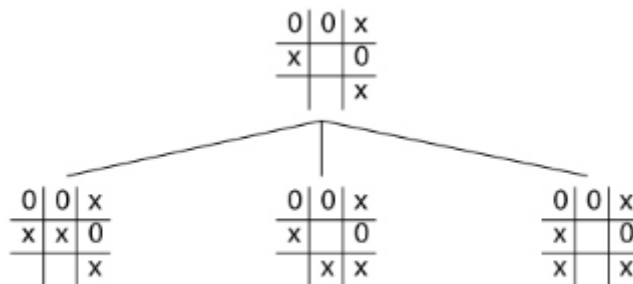
El algoritmo de minimax en simples palabras consiste en la elección del mejor movimiento para el computador, suponiendo que el contrincante escogerá uno que lo pueda perjudicar, para escoger la mejor opción este algoritmo realiza un árbol de búsqueda con todos los posibles movimientos, luego recorre todo el árbol de soluciones del juego a partir de un estado dado, es decir, según las casillas que ya han sido rellenadas. Por tanto, minimax se ejecutará cada vez que le toque mover a la IA.

En el algoritmo Minimax el espacio de búsqueda queda definido por:

Estado inicial: Es una configuración inicial del juego, es decir, un estado en el que se encuentre el juego. Para nuestro ejemplo sería:

0	0	x
x		0
		x

Operadores: Corresponden a las jugadas legales que se pueden hacer en el juego, en el caso del tres en raya no puedes marcar una casilla ya antes marcada.



Condición Terminal: Determina cuando el juego se acabó, en nuestro ejemplo el juego termina cuando un jugador marca tres casillas seguidas iguales, ya sea horizontalmente, verticalmente o en diagonal, o se marcan todas las casillas (empate) .

0	0	x
x	x	0
0	x	x

0	0	x
x	x	0
x	0	x

0	0	x
x	0	0
x	x	x

Función de Utilidad: Da un valor numérico a una configuración final de un juego. En un juego en donde se puede ganar, perder o empatar, los valores pueden ser 1, 0, o -1.

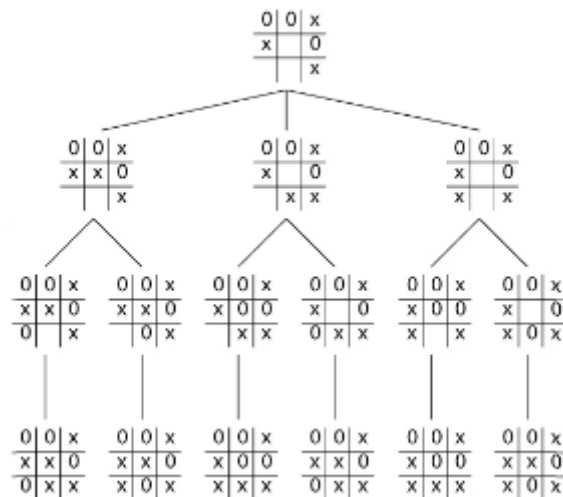
0	0	x
x	x	0
0	x	x

-1	0	0	x
x	0	0	
x	0	x	

1	0	0	x
x	x	0	
x	0	x	

Implementación Minimax: Los pasos que sigue minimax pueden variar, pero lo importante es tener una idea clara de cómo es su funcionamiento, los pasos a seguir son:

El algoritmo primero genera un árbol de soluciones completo a partir de un nodo dado. veamos el siguiente ejemplo:



Para cada nodo final, buscamos la función de utilidad de estos. En nuestro ejemplo usaremos un 0 para las partidas que terminen en empate, un 1 para las que gane la IA y un -1 para las que gane el jugador humano.

0	0	x
x	x	0
0	x	x

1	0	0	x
x	x	0	
x	0	x	

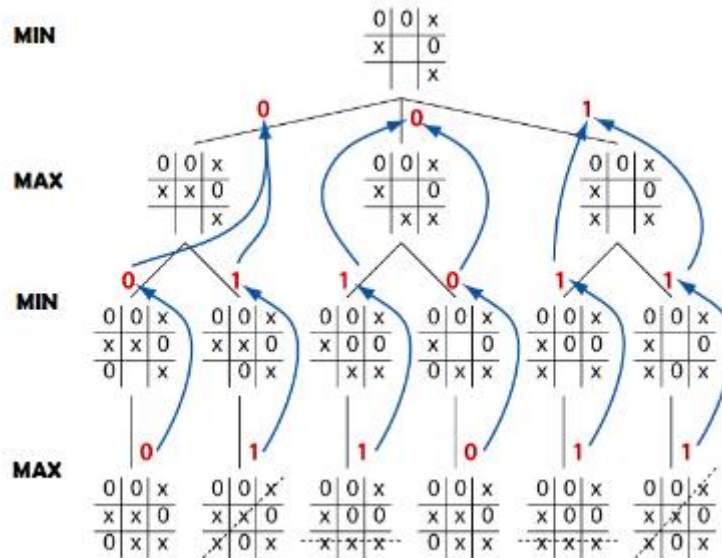
1	0	0	x
x	0	0	
x	x	x	

0	0	x
x	x	0
0	x	x

-1	0	0	x
x	0	0	
x	0	x	

1	0	0	x
x	x	0	
x	0	x	

Y lo que hará el algoritmo Minimax cuando vaya regresando hacia atrás, será comunicarle a la llamada recursiva superior cuál es el mejor nodo hoja alcanzado hasta el momento. Cada llamada recursiva tiene que saber a quién le toca jugar, para analizar si el movimiento realizado pertenece a la IA o al otro jugador, ya que cuando sea el turno de la IA nos interesa MAXIMIZAR el resultado, y cuando sea el turno del rival MINIMIZAR su resultado.



Al final el algoritmo nos devolverá la jugada que debe realizar la máquina para maximizar sus posibilidades y bloquear las posibilidades del rival.

Agregando una interfaz gráfica creada con ayuda de Qt Designer y PyQt (lo podemos ver en nuestro curso profesional de Python), obtendremos el siguiente resultado:



Espero que les haya gustado este tutorial y les pueda servir para que desarrollen sus propios juegos, no olviden que si tienen dudas y consultas, pueden dejarlas en los comentarios. Hasta la próxima.