



# OTHELLO

Inteligencia Artificial

Alejandro Gómez Rueda y Francisco Jesús de la Rubia García

M16C10

## Explicación Algoritmo MiniMax

Para cada iteración en caso de que sea el fin del juego o haya una profundidad de cero, se devolverá el valor determinado por la heurística  $h_1$  (que posteriormente explicaremos).

En caso de que el jugador actual no pueda jugar, se pasará turno, posteriormente almacenaremos en un ArrayList los posibles movimientos que puede hacer el jugador, a continuación, realizamos una copia del tablero y, para cada casilla situada en el array, se asigna la celda con su correspondiente color en dicho tablero.

Después calculamos el valor del tablero una vez colocada la ficha y, dependiendo de si es un nodo máximo, el valor actual del tablero es mayor que alfa, se actualiza el valor de alfa con el valor actual y para el caso del nodo mínimo el valor actual del tablero sería menor que beta y se actualizaría con el valor actual, realizando esto para todos los nodos del árbol.

En caso de una partida en la que el jugador1 sea el humano y el jugador2 sea la máquina, el valor de alfa representara el humano y el de beta la máquina.

## Explicación Algoritmo PodaAlfaBeta.

Para cada iteración en caso de que sea el fin del juego o haya una profundidad de cero, se devolverá el valor determinado por la heurística  $h_1$  (que posteriormente explicaremos).

En caso de que el jugador actual no pueda jugar, se pasará turno, posteriormente almacenaremos en un ArrayList los posibles movimientos que puede hacer el jugador, a continuación, realizamos una copia del tablero y, para cada casilla situada en el array, se asigna la celda con su correspondiente color en dicho tablero.

Después calculamos el valor del tablero una vez colocada la ficha y, dependiendo de si es un nodo máximo, el valor actual del tablero es mayor que alfa, se actualiza el valor de alfa con el valor actual y, si alfa es mayor o igual que beta se devuelve el valor de alfa. Para el caso del nodo mínimo el valor actual del tablero sería menor que beta y se actualizaría con el valor actual y, si el valor de beta es menor o igual que el de alfa se devuelve el valor de beta realizando esto para todos los nodos del árbol.

## Explicación de la heurística empleada

Para la utilización de esta heurística le pasaremos como parámetros el tablero sobre el que se está jugando y el color del jugador, lo primero que haremos será calcular la puntuación actual

a través del método Puntos, como la que hay implementada en la heurística h2, cuya función es la de comprobar si se ha finalizado el juego y en el caso de que la puntuación sea mayor que 0 se devuelve 100 y en caso contrario -100.

Para cada color de las fichas, se va aumentando el valor de la heurística con las esquinas, es decir, si somos las fichas blancas y, en una esquina hay una ficha blanca la puntuación es de 10, si tenemos dos sería 20 y así sucesivamente. El procedimiento con las fichas negras es el mismo que las blancas.

El método Puntos, al cual hemos hecho referencia anteriormente se recorre todo el tablero y, dependiendo del color del jugador va incrementando una variable contador.

### Análisis del comportamiento de los jugadores

Ejecución	Jugador1	Algoritmo1	Heurística	Jugador2	Algoritmo2	Heurística2	Negras	Fichas1	Fichas2
1	Humano	-----	-----	Máquina	MiniMax	H1	Jugador1	26	38
2	Máquina	Poda	H1	Máquina	Poda	H1	Jugador1	57	7
3	Máquina	MiniMax	H2	Máquina	MiniMax	H2	Jugador1	34	30
4	Humano	-----	-----	Máquina	Poda	H2	Jugador1	18	46
5	Humano	-----	-----	Máquina	MiniMax	H2	Jugador1	34	30

Si nosotros pusiésemos a ejecutar de nuevo un algoritmo PodaAlfaBeta entre maquina contra maquina con la heurística H1, por ejemplo, y después ejecutamos el algoritmo MiniMax con la misma heurística, el resultado que nos proporcionara ambos algoritmos va a ser el mismo, ya que el PodaAlfaBeta es una mejora del algoritmo MiniMax como ya hemos visto anteriormente.

En la tabla no hemos ilustrado el caso de máquina vs máquina cada una utilizando distintos algoritmos dado que la propia lógica del programa no permite realizar dicha operación, tan sólo se podría enfrentar ambas máquinas pero utilizando el mismo algoritmo.