

Grado en Ingeniería Informática
Inteligencia Artificial
Curso 2019/2020

Guion 4

Java

Juegos



UNIVERSIDAD DE JAÉN

Profesor: Angel Miguel Garcia Vico

Grupo: 2

Rafael Osuna Ventura

31024321N

rov00005@red.ujaen.es

José Antonio Morales Velasco

76593264K

jamv0007@red.ujaen.es

Índice

Informe del Juego.....	2
Estrategia Utilizada.....	3
Función Recorrer/Inorden.....	3
Función CopiaTablero.....	3
Función CaculaValor.....	3
Función Comprueba.....	4
Función TurnoJugada.....	4
Función MostrarRama / MostrarRamaInorden.....	5
Función Mostrar/MostrarInorden.....	5
Conclusión.....	5

Informe del Juego

Ejecución	Niveles	Movimientos	Ganador
1	4	16	ALFABETA
2		16	ALFABETA
3		14	ALFABETA
1	5	16	ALFABETA
2		12	ALFABETA
3		28	ALFABETA
1	6		
2		No Funciona	
3			
1	7		
2		No Funciona	
3			
1	8		
2		No Funciona	
3			

Como podemos observar en la tabla de resultados para los niveles 4 y 5 el algoritmo casi siempre es ganado con una media de unos 15.3 para el nivel 4 y 18.6 para el nivel 5 aunque estos valores dependen mucho de los movimientos del usuario ya que si se dedica a jugar defensivamente el juego se puede alargar y con él el número de movimientos.

Por desgracia el algoritmo minimax implementado para el tamaño de tablero 6x7 solo nos permite la ejecución con los niveles 5 o inferior ya que con el nivel 6 se puede jugar aunque tras colocar un par de fichas dará errores de memoria. Con los niveles 7 y superior no funciona por el mismo problema desde la generación del primer árbol.

El error consta en que aunque no ha agotado la memoria, el 98% está consumida y el recolector de basura consume todo el tiempo en intentar extraer memoria de la basura y al tardar tanto devuelve una excepción provocando que el juego no pueda continuar.

Estrategia Utilizada

Función Recorrer/Inorden

Para la estrategia utilizada, en primer lugar para el nodo con el movimiento del jugador, se calculan los posibles movimientos de la máquina colocando si es posible una ficha en cada columna del tablero actual con lo que se obtendría un hijo. Realiza una copia del tablero padre con la función `CopiaTablero` y coloca en ella la ficha en la columna correspondiente. A su vez para este se repite el proceso para el primer hijo hasta un nodo con solución, en el que uno de los jugadores gane o que llegue al nivel de restricción asignado. Entonces calculamos el valor heurístico del tablero.

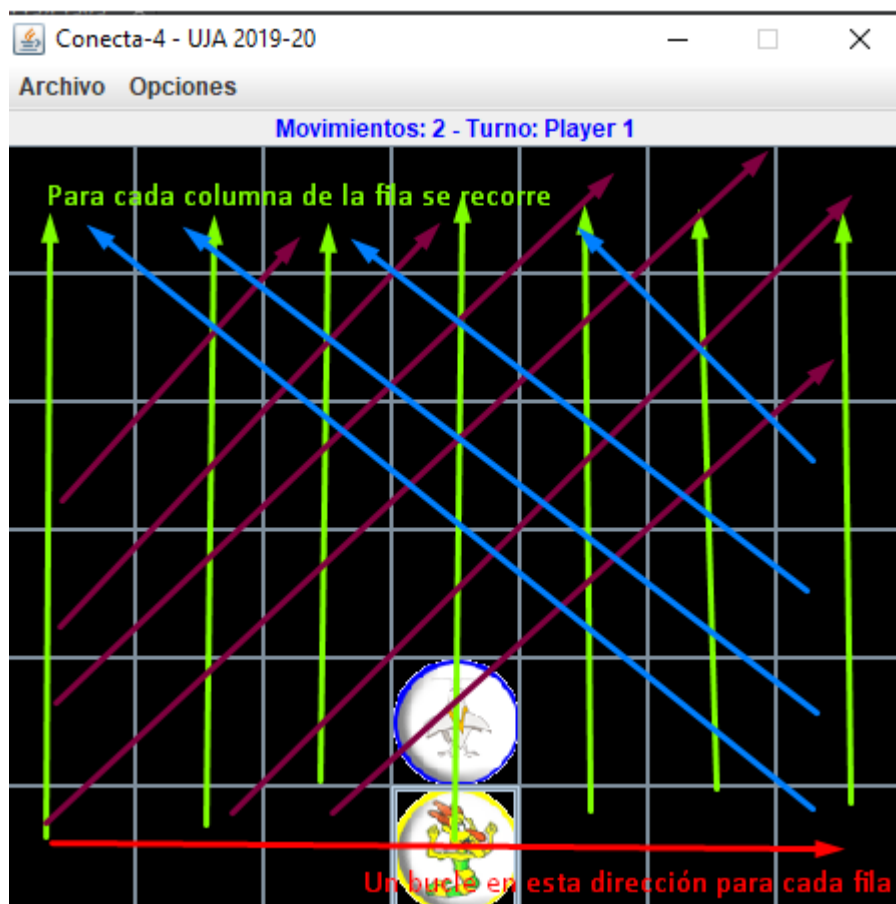
Para los padres de los nodos terminales, cogen el valor del hijo y modifican alfa o beta en función del tipo de nivel actual y calculan el siguiente hijo que llevará como valores alfa y beta los obtenidos del padre. Los valores se van propagando y si se cumple la fórmula de la poda realizará poda lo que hará que no genere los hijos siguientes.

Función CopiaTablero

Copia valor a valor el tablero del padre insertando las fichas en los lugares correspondientes.

Función CalculaValor

La función `CalculaValor`, que pasado el nodo calcula desde la esquina inferior izquierda, hasta la derecha con dos contadores, uno que almacena si la casilla actual es 0 o el elemento actual para el que se está calculando (máquina o jugador), si se cumple esto aumenta en 1 y además otra variable que cuenta las fichas en cada línea. A su vez mientras nos desplazamos por las filas, se calcula para las columnas si existe y a su vez también las diagonales. Si en el camino de cálculo de estas líneas se encuentra con una ficha del jugador contrario, comprueba si de alguna forma en el recorrido se puede hacer conecta (el valor introducido de fichas para ganar), si lo hace se calcula el valor y las variables se ponen a 0 sino pues las pone a 0 sin realizar cambio en el valor. Así lo repite para cada línea del tablero. Los valores de las diagonales solo se calculan una vez en el tablero, cuando están en la última fila del cuadrante.



Función Comprueba

La función Comprueba calcula si hay posibilidades de ganar, si hay en la línea al menos una ficha, y si de los valores que hay son iguales o mayores que el número de fichas para ganar, entonces dependiendo del jugador para el que este calculando, el valor de la heurística se suma o resta.

Función TurnoJugada

Se llama a las funciones anteriores para crear el árbol y heurística.

Después de generarlo calcula el hijo con el mismo valor y busca la columna en el distinta, donde se insertó la ficha y devuelve dicha columna.

Desde aquí también se llama a las funciones para mostrar el árbol o la primera rama solo, que es más fácil e intuitiva de ver.

Función MostrarRama / MostrarRamaInorden

Muestra unicamente de forma recursiva desde el nodo raiz, solamente el primer nodo hijo de cada uno hasta el nivel indicado.

Función Mostrar/MostrarInorden

Muestra la mitad de nodos hijos del actual, el actual y luego el resto. La salida es bastante confusa y larga lo que no es rentable para grandes árboles.

Conclusión

Tras muchas pruebas podemos concluir que el algoritmo minimax como mucho permite empatar con el si se escogen las jugadas con cuidado aunque la mayoría de las veces gana este algoritmo, por lo que se trata de un sistema que coge siempre el mejor caso aunque con un coste exesivo de memoria debido al gran arbol que debe generar por lo menos para este juego.