

Sistemas Inteligentes

Práctica 2: Algoritmo Minimax

Descripción de la práctica

Se trata de hacer un programa que juegue a conecta 4 contra el programa desarrollado por el profesor. El objetivo del juego consiste en conseguir 4 en raya en un tablero de 7 columnas y 6 filas, en el que las fichas caen desde arriba.

5							
4							
3					O	X	
2				O	X	O	
1			X	X	O	O	
0			X	O	X	X	
	0	1	2	3	4	5	6

El alumno debe implementar en la clase **conecta4.MiEstado** las siguientes funciones:

- Constructor por defecto (sin parámetros) para crear un estado sin fichas en el tablero.
- Constructor para crear un objeto con las mismas fichas que otro. Este constructor es necesario para que el alumno use su propia clase **conecta4.MiEstado** en lugar de la del profesor.
- **getSucesor**: calculará un sucesor de la clase **conecta4.MiEstado**.
- **getFicha**: indicará qué ficha hay en la posición indicada: **false**, **true** o **null** si se trata de una ficha de MIN, de MAX, o no hay ficha, respectivamente.
- **getFichas**: devolverá el número de fichas que hay en la columna indicada.
- **getUtilidad**: devolverá el número de cuatros en raya en filas, columnas y diagonales que podría hacer MAX menos los de que podría hacer MIN. Si en el estado gana MAX, el valor debe ser mayor o igual que **GANADOR**, y si gana MIN, el valor debe ser menor o igual que **-GANADOR**.
- **esFinal**: indicará si se trata de un estado ganador o empate.

El algoritmo Minimax se debe implementar en la clase **conecta4.Main**. La función **main** debe aceptar dos parámetros para limitar la profundidad de búsqueda de MIN y de MAX. De esta forma se podrá comprobar que generalmente el jugador ganador es el que más explora.

Para jugar contra el profesor, hay que llamar a una de las siguientes funciones según sea MIN o MAX de forma alternada con la implementación del alumno, tal como se ve en la implementación inicial de la clase **conecta4.Main**:

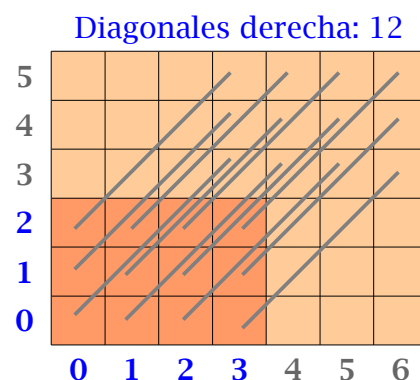
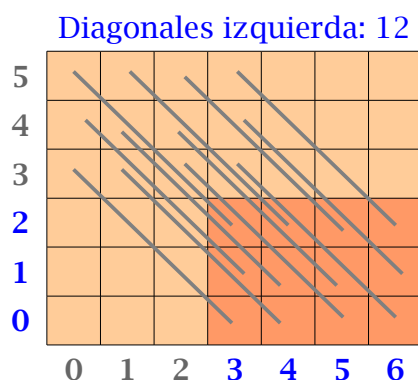
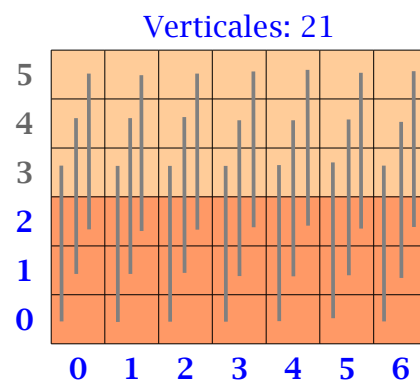
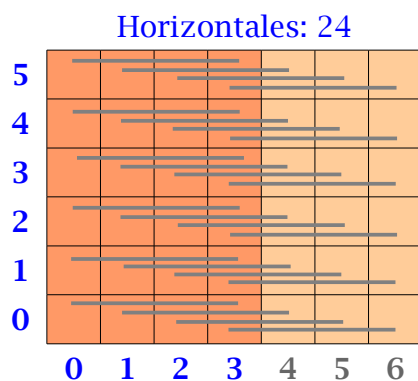
- Estado Profesor.maxValor(Estado e, int limite);
- Estado Profesor.minValor(Estado e, int limite);

Función de utilidad

El cálculo de esta función consiste en buscar cuántos 4 en raya puede realizar MAX menos los que puede realizar MIN.

Para que, por ejemplo, MAX pueda realizar un 4 en raya en ciertas casillas, dichas casillas deberán estar vacías o sólo ocupadas por MAX. Si hubiera una casilla ocupada por MIN, no se tendría en cuenta este cuatro en raya. Si las 4 casillas estuvieran ocupadas por MAX, entonces se trataría de un estado ganador, y por tanto, el valor de utilidad sería GANADOR.

En las siguientes figuras se muestran todos los 4 en raya que se podrían hacer en un tablero vacío. Las casillas más oscuras podrían tomarse como puntos de partida para buscar los 4 en raya.



En un tablero vacío cada jugador podría hacer $24 + 21 + 12 + 12 = 69$ cuatros en raya, pero el valor de utilidad debería ser 69 de MAX - 69 de MIN = 0 .

La búsqueda de los 4 en raya se podría hacer usando dos bucles anidados que recorrieran las celdas oscuras con x (columna) e y (fila), y para cada una de estas celdas se podría usar otro bucle que iterara una variable i de 0 a 3 . De esta forma, las coordenadas de los 4 en raya serían:

- Horizontal: $x+i, y$
- Vertical: $x, y+i$
- Diagonal izquierda: $x-i, y+i$
- Diagonal derecha: $x+i, y+i$

Tareas a realizar

Las tareas concretas a realizar en la práctica son las siguientes:

1. Implementar las funciones necesarias en la clase **conecta4.MiEstado**.
2. Implementar el algoritmo Minimax en la clase **conecta4.Main**.
3. Realizar varias ejecuciones con distintos límites de profundidad para MIN y para MAX y anotar quién resulta ganador y cuántos nodos se han explorado.

Entorno de trabajo

Para el desarrollo de la práctica es necesario añadir el fichero **profesor.jar** como una librería del proyecto. El proyecto debería crearse usando la codificación UTF-8.

La ejecución por consola se puede realizar con esta sentencia:

```
java -cp ruta_clases:profesor.jar conecta4.Main
```

La implementación del profesor se puede ejecutar con la siguiente sentencia:

```
java -jar profesor.jar
```

Normas de entrega

La práctica se realizará de forma individual o en parejas.

Hay que entregar el código Java y una pequeña memoria en PDF donde se explique las pruebas que se han realizado y los resultados que se han obtenido:

- Límites de profundidad para MIN y MAX.
- Jugador ganador.
- Nodos explorados.
- Tiempo de ejecución.

La fecha límite para entregar la práctica es el 22 de Diciembre de 2019.