Memoria del 4 en raya Crush

Práctica 3

Emilio Chica Jiménez

11/04/2015

Diseño de un agente deliberativo para hallar la solución al juego 4 en raya crush

1. Objetivo

El objetivo de esta práctica ha sido el de crear un agente deliberativo el cual sea capaz de ganar al juego 4 en raya crush. Para ello necesitamos el algoritmo de poda alpha-beta para hallar recorrer el árbol de estados y encontrar la mejor solución podando los nodos que no nos ofrezcan una solución mejor que la que tenemos. Por último se requiere una heurística que haga que nuestro agente juegue de la mejor forma.

2. Análisis del problema

Algoritmo Poda Alpha-Beta

Comencemos analizando el algoritmo poda alpha-beta. Este algoritmo requiere de una previa implementación del algoritmo minimax, por lo que nuestro primer objetivo es construir dicho algoritmo. Sabemos que se basa en la búsqueda del mayor beneficio con respecto al jugador max y con respecto al jugador min la jugada menos beneficiosa para el max, con respecto a la jugada próxima que vayan a realizar.

Partiendo de esa base y sabiendo que tenemos que ir generando el árbol de estados conforme nuestro contrincante haga su jugada, ya sabemos cómo tenemos que implementar nuestro algoritmo minimax.

La poda alpha-beta sería ayudar a nuestro algoritmo minimax a podar ramas del árbol de estados que ya no nos son útiles. Para ello utilizamos la cota alpha para podar nodos beta y la cota beta para podar nodos alpha. Por lo tanto tenemos que propagar los valores de alpha y beta que tenemos actualmente hacia los nodos que vayamos generando para así poder hacer la poda.

Heurística

Para obtener un punto de enfoque del juego y realizar el análisis correcto del mismo he comprendido que para realizar una buena heurística lo mejor que puedo hacer es analizar las mejores jugadas a las cuales les puedo dar la mejor puntuación.

Para empezar con algo sencillo el juego se puede dividir en tres tipos de jugadas:

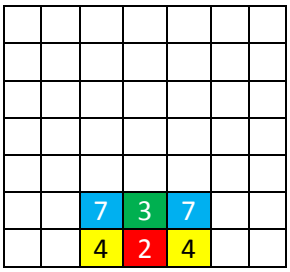
- Jugadas iniciales:

  Las primeras fichas son importantes, ya que estudiando por internet las mejores jugadas descubres que si eres el jugador inicial la mejor posición para comenzar es la central, cosa que tiene lógica, tienes más posibilidades de crear una línea hacia más direcciones distintas. Si eres el jugador secundario tendremos varias opciones a elegir.

- Jugadas de 2 en línea:

Este tipo de jugadas son las que podríamos considerar de menor importancia con respecto a las de 3 en línea, sin embargo son muy útiles para que nuestra heurística sepa seleccionar las mejores jugadas de 2.

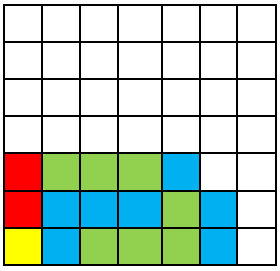
Ejemplo:

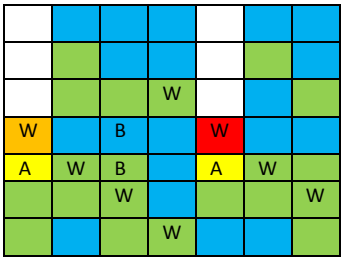


Con este esquema contemplamos todas las posibles jugadas y la importancia que le otorgamos a cada una de ellas con el número que aparece en cada casilla.

- Jugadas de 3 en línea:

En este caso nos decidimos muchas cosas ya que el hacer una jugada de 3 alerta al contrario a que debe poner una ficha para impedir tu 4 en raya. Por ello jugando hay que darse cuenta que este tipo de jugadas hay que realizarlas de modo que el contrario no pueda poner ficha para cortarte la jugada y posteriormente terminar la misma obligando al contrario a poner la ficha que te acabe el movimiento. Para ello un ejemplo de este tipo de jugadas es muy útil:





Como podemos observar en ambas situaciones se dan jugadas potenciales de victoria. En la primera tendríamos que obligar al jugador verde a poner en la casilla amarilla para que pudiéramos hacer nuestro 4 en raya, para ello deberíamos dejar ese tipo de casillas para las últimas. No es una condición de victoria segura pero ayuda para el final.

Para la segunda jugada se requiere un poco más de astucia ya que si nos fijamos tal y como se han ido poniendo las fichas el jugador verde es el que tiene las de ganar ya que es el turno del jugador azul y da igual donde ponga el jugador azul que ha perdido, por lo tanto de este tipo de jugadas sacamos en claro que la importancia de las diagonales es más que evidente.

3. Descripción de la solución planteada

La solución a los dos problemas que se plantean son:

Algoritmo poda alpha-beta

Para este problema he usado la generación de un nivel completo con respecto al nodo actual que tengo, con el método *GenerateAllMoves.* A partir de este nivel completo de jugadas posibles voy a generar una a una las siguientes jugadas posibles para cada nodo de este nivel con el método *GenerateNextMove,* y voy a estudiar su valor heurístico y su poda correspondiente de sus sucesores a 8 niveles.

Para ello utilizo mi método miniMax que me devuelve el valor de la mejor jugada del nivel completo que he generado habiendo descendido los 8 niveles necesarios para encontrar la mejor. Este método miniMax también realiza la poda de los nodos innecesarios.

Mi método PodaAlfaBeta es el que va a comprobar si el valor que me devuelve minMax es mayor que el valor de otra jugada anterior, si es así cambio la acción por la del tablero que me ha dado mejor resultado, si no, sigo buscando.

Si por alguna razón el valor de alpha acaba en menos infinito significa que no ha habido ninguna jugada que haya superado el valor de menos infinito y por lo tanto sólo me queda una jugada por hacer, por lo tanto devuelvo el siguiente movimiento. Aunque sepa que vaya a perder.

Heurística aplicada

La heurística aplicada al juego se basa en las mejores jugadas que puede hacer un jugador de 4 en raya, dando prioridad a la casilla central en el inicio del jugador max y la casilla número 5 en el caso del jugador min, a las casillas de la izquierda a mitad del juego y sobre todo y más importante a las diagonales.

Para ello he construido un sistema de valores el cual se puede ver en esta tabla por prioridad de movimientos:

Movimientos	**Prioridad**	**Casillas MAX**		**Casillas Min**	
		X	**Y**	**X**	**Y**
Inicial	2	3	0	5	0
Vertical de 2	2	-	y-1	-	y-1
Vertical de 2 central	5	3	y-1	3	y-1
Vertical de 2 izquierda	3	2	y-1	2	y-1
Vertical de 2 derecha	3	4	y-1	4	y-1
Vertical de 3	5	-	y-1,y-2	-	y-1,y-2
Vertical de 3 central	9	3	y-1,y-2	3	y-1,y-2
Vertical de 3 con espacio arriba	8	-	y-1,y-2, y-3=0	-	y-1,y-2, y-3=0
Horizontal de 2	2	x+1, x<6	-	x+1, x<6	-
Horizontal de 2 hacia la izquierda cerca del centro	4	x+1,x>1, x<4	-	x+1,x>1, x<4	-
Horizontal de 3	9	x+1,x+2	-	x+1,x+2	-
Horizontal de 3 hacia la izquierda cerca del centro	14	x+1,x+2, x>0,x<4	-	x+1,x+2, x>0,x<4	-
Horizontal de 3 con espacios laterales	24	x+3=0, x-1=0	-	x+3=0, x-1=0	-
Diagonal derecha de 2	2	x+1	y-1	x+1	y-1
Diagonal derecha de 2 más a la izquierda para poder terminarla	5	x<4, x+1	y>0, y-1	x<4, x+1	y>0, y-1
Diagonal izquierda de 2	2	x-1	y-1	x-1	y-1
Diagonal izquierda de 2 más a la derecha para poder terminarla	3	x>2, x-1	y>2, y-1	x>2, x-1	y>2, y-1
Diagonal derecha de 3	8	x+1,x+2	y-1,y-2	x+1,x+2	y-1,y-2
Diagonal derecha de 3 más a la izquierda para poder terminarla	9	x+1,x+2, x<4	y-1,y-2, y>2	x+1,x+2, x<4	y-1,y-2, y>2
Diagonal derecha de 3 con espacios laterales	19	x+1,x+2, x-1=0, x+3=0	y-1,x-2, y-3=0, y+1=0	x+1,x+2, x-1=0, x+3=0	y-1,x-2, y-3=0, y+1=0
Diagonal izquierda de 3	8	x-1,x-2	y-1,y-2	x-1,x-2	y-1,y-2
Diagonal izquierda de 3 más a la derecha para poder terminarla	8	x+1,x+2, x>2	y-1,y-2, y>2	x+1,x+2, x>2	y-1,y-2, y>2
Diagonal izquierda de 3 con espacios laterales	18	x-1,x-2, x-1=0, x+3=0	y-1,y-2, x-1=0, x+3=0	x-1,x-2, x-1=0, x+3=0	y-1,y-2, x-1=0, x+3=0
COMBINACIONES CON ESPACIO EN MEDIO	10	x+1=0, x+2, x+3	-	x+1=0, x+2, x+3	-
COMBINACIONES CON ESPACIO EN MEDIO	10	x+1, x+2=0, x+3	-	x+1, x+2=0, x+3	-

Los valores no son exactos ya que dependiendo de donde este la X puede sumar más al valor si es la casilla central o la casilla 5. Y también hay intersecciones que no se contemplan en estos valores con respecto a horizontales mas a la izquierda, mas a la derecha, verticales, diagonales, etc..