Practica 3

Memoria de practicas

Yunhao Lin

**INTRODUCCION**

El objetivo de esta práctica es diseñar e implementar un agente deliberativo para el juegodesconecta-4 boom en la cual nuestro objetivo es hacer que nuestro adversario haga un conecta 4.

Para ello emplearemos la teoría vista en clase y los conocimientos de c++ previos que tengamos.

**IMPLEMENTACION**

Para empezar, el guion de practicas nos recomendaban implementar primero el minimax. Siguiendo el seudocódigo que se puede encontrarse fácilmente en la Wikipedia, he hecho una adaptación de ese seudocódigo para crear la implementación final.

Una vez logrado el minimax, el siguiente paso era lograr una poda alfa beta, para ello seguí el mismo proceso anteriormente descrito.

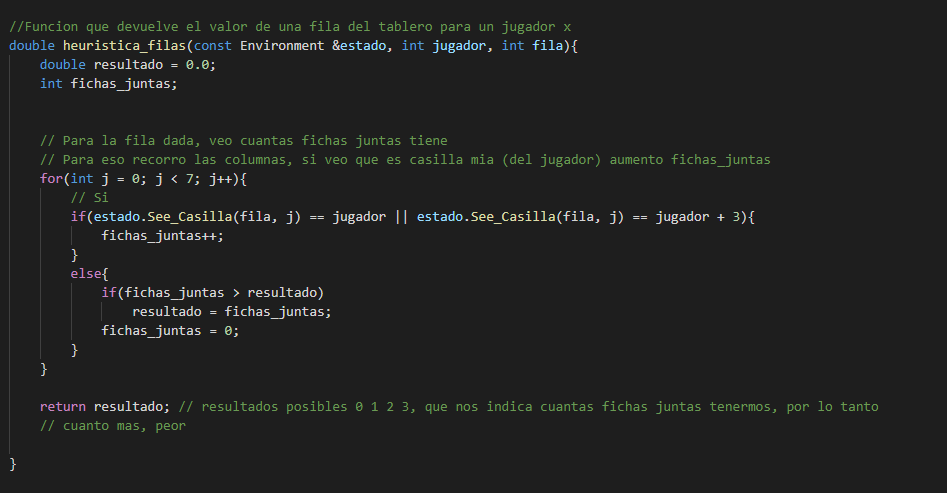
No me voy a extender en el funcionamiento de ambos métodos ya que se ha visto extensivamente en clase.

Los problemas con los que me he encontrado en esta primera fase es navegar hasta los siguientes hijos/movimientos. Esta parte se resolvió con el método GenerateAllMoves en el cual nos devolvía el numero de hijos/movimientos y cuales eran (almacenados en un vector Enviroment)

**IMPLEMENTACION DE LA HEURISTICA**

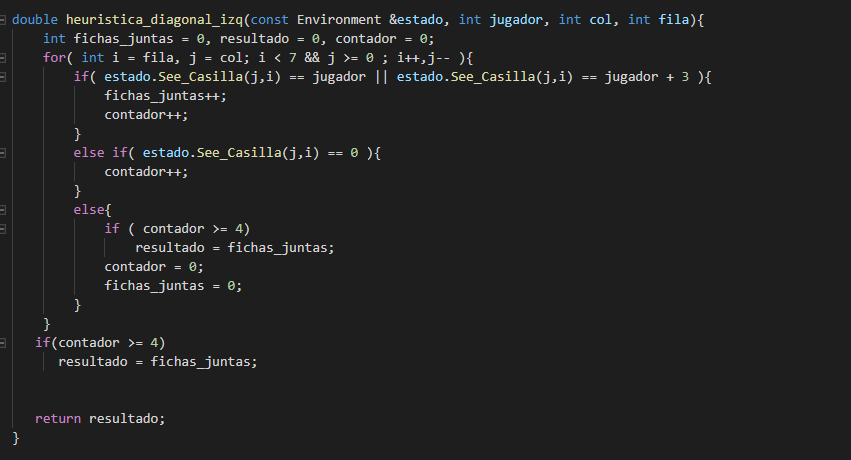
En esta parte, el objetivo era darles una puntuación a los diferentes movimientos, para ello valoramos las fichas seguidas que tienen en las casillas adyacentes (izquierda, derecha, arriba, abajo y diagonales).

Para esta parte, he intentado dos maneras, la primera, dada una posición i, j valoraba cuantas de las mías tenia cerca, y dependiendo de las que tenia cerca, le daba una puntuación u otra. Si es el enemigo le doy mas puntos, y si es mía le doy menos puntos, para que no valore ese movimiento. Sin embargo, mis intentos para hacer funcionar este método fueron infructuosos. Por lo cual me decante por un nuevo acercamiento. Esta vez, divido cada parte en funciones. Por lo tanto, tengo una parte que me devuelve las fichas que tengo juntas en una fila.



Para el caso de las columnas seria idéntico, solo que se recorre las filas.

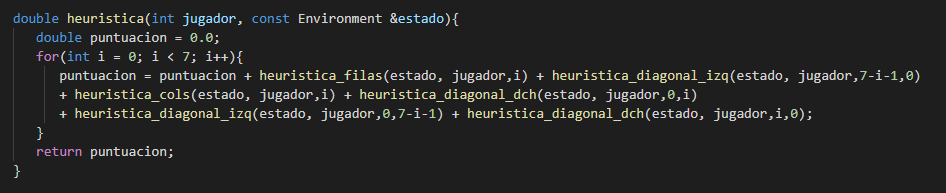
Una vez implementado estas dos partes, los casos que nos faltarían por valorar seria las diagonales:



Para valorar las diagonales lo que hago es comprobar si las adyacentes son mías, pero además cuento los espacios en blanco que hay. Si el contador supera 4 eso quiere decir que puede haber movimientos en los que puedo hacer un 4 en raya, y por tanto me perjudique, pero si no hay “espacio” para que haga cuatro en raya, no me perjudicaría, y devuelvo 0.

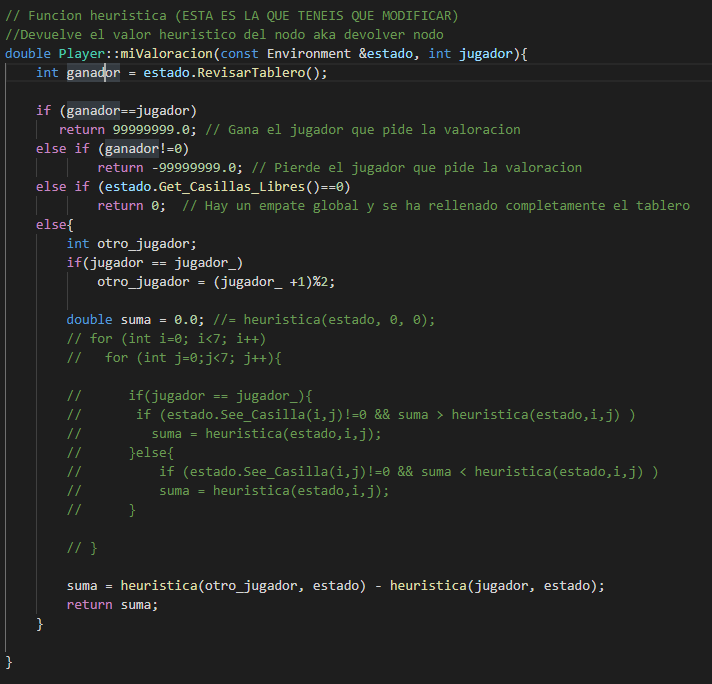
Para el caso de las diagonales a la derecha seguiríamos una lógica muy parecida.

Finalmente reúno todas las puntuaciones previas de la siguiente manera:



Por lo tanto, según esta heurística, una puntuación mas alta, significa un tablero más malo.

**VALORACION DE LAS PUNTUACIONES**

Si no se hay ganador y además quede espacios libres en el tablero, devuelvo la suma de las heurísticas.

Si la suma sale negativa significa que jugador tiene peor tablero, y si positivo, al contrario.

**CONCLUSION**

En esta practica he aprendido a implementar la poda alfa beta (además del minimax), con la cual, a partir de una serie de valores me da el mejor de ellos. Por lo tanto, nuestro agente, en cada paso intenta hacer la mejor acción posible entre las disponibles. Y por lo tanto creo que he cumplido con los objetivos por la cual estaba diseñada esta práctica.