



Inteligencia Artificial:
Primer Proyecto : Cuatro en Línea

Ronald Esquivel López 2018093269
Ricardo Murillo Jiménez 2018173697

Profesores:
Maria Auxiliadora Mora Cross

2021

II SEMESTRE

INGENIERÍA EN COMPUTACIÓN

[Descripción del sistema](#)

[Descripción del agente](#)

[Descripción detallada de la función Eval](#)

[Experimentos](#)

[Experimento 1](#)

[Experimento 2](#)

[Conclusiones](#)

[Bibliografía](#)

Descripción del sistema

El presente proyecto se desarrolla con el fin de poner en práctica conceptos e ideas de la Inteligencia Artificial. Usando para ello el algoritmo minimax junto a su agregado: la poda Alfa-Beta ("Poda alfa-beta", 2020). Alfa-Beta es un método o técnica usada en este tipo de algoritmo con la cual se busca reducir el número de nodos (pertenecientes al árbol creado por el algoritmo minimax) que se deben de evaluar. Se hace así, debido a que la evaluación, conocida como evaluación estática que se aplica en estos algoritmos, generalmente es un proceso que requiere de gran poder computacional.

Esta evaluación estática se crea con el fin de darle un peso a los nodos del árbol, de esta manera se puede aplicar el algoritmo alfa-beta. Es importante considerar que esta evaluación se realiza mediante heurísticas, ya que realmente no existe algo que nos diga exactamente cuánto es el valor de un nodo X del árbol. Por lo que hay que poner especial atención en esta evaluación, ya que es la que dictará cuán bueno es un nodo X en comparación a otros, y una mala heurística ("Significado de Heurística", 2017), nos podría llevar a procesos o elecciones de decisiones deficientes. Por lo que para poner en práctica estos conceptos se desarrollará un agente inteligente capaz de jugar al juego conocido como "4 en Línea" de forma satisfactoria contra un humano.

Para esto hay que saber cómo es el juego: Básicamente el mismo consiste en que hay 2 jugadores y una "rejilla o matriz" en la cual los jugadores pueden "insertar" fichas (Se identifica al jugador dueño de la ficha mediante un color u otro método de identificación; para este proyecto se decidió que un número 2 sea la IA y el 1 un jugador) y que usualmente es de 6 filas y 7 columnas. Los jugadores deberán de elegir cuál de los 2 empieza a colocar piezas de primero.

Así, tendrán que ir insertando fichas por la parte superior de la rejilla, la cual caerá hasta encontrar el siguiente lugar vacío en la columna seleccionada. Claramente no se trata de colocar solamente fichas al azar, los jugadores tratarán de ir formando líneas de 4 fichas seguidas de su mismo color(número), ya que esto supondría un gane; a la vez que intentan reducir al máximo las oportunidades de su oponente de ganar.

Descripción del agente

Un aspecto importante a la hora de diseñar estos sistemas es algo a lo que se le llama agente inteligente. El agente básicamente percibe, razona y actúa, y lo que hace es mapear lo

que percibe y convertirlo en acciones.

Los agentes básicamente poseen un entorno, forma de percibir su entorno, acciones basadas en el entorno, funciones y diferentes medidas de desempeño según la acción realizada (Mora, 2021) .

En este caso nuestro agente sería la Inteligencia Artificial que se encargará de jugar al 4 en línea.

- El entorno en este caso del 4 en línea sería el tablero o rejilla a utilizar.
 - **Entorno total:** El agente siempre puede observar todo el tablero de juego.
 - **Determinista:** El estado siguiente, va a estar totalmente determinado por el estado actual del tablero y la acción ejecutada por la IA.
 - **Episódico:** el Estado actual si depende de lo sucedido anteriormente.
 - **Semi-Estático:** Ya que mucha parte del juego el entorno no cambiará, es hasta que se coloque una ficha que el entorno cambia un poco.
 - Es un **entorno discreto**, se pueden contar la cantidad de combinaciones posibles en el tablero.
 - **MultiAgente:** Para poder jugar debemos de tener dos entidades diferentes.
- La percepción que posee el bot, es saber donde están ubicadas las fichas en el tablero.
- **Acción principal:** Es la colocación de una nueva pieza en la rejilla (en este caso una matriz).
- **Función y Actuadores:** Buscar la jugada que maximice mi valor como tablero, y la acción sería colocar la pieza en el lugar elegido.
- **Medida de desempeño:** En este caso, esta característica se mide mediante la función de evaluación, y el movimiento realizado por el agente, si el valor del tablero va yendo hacia arriba, quiere decir que el agente está cerca de tener la solución.

Descripción detallada de la función Eval

A continuación se explicará cómo se realizó la evaluación estática mediante una heurística: Primeramente, cuando un tablero cualquiera entra en función:

Se le asigna una puntuación inicial contando la cantidad de fichas “X” que haya en la columna central. Esto porque caso todos los ganos que se den, a excepción de la mayoría de los ganos verticales, tendrá que pasar al menos una ficha por la columna central.

Se decidió sumar un valor de 3 por cada ficha encontrada. Después de esto, se asignará una puntuación basado en recorrer todo el tablero, de forma horizontal, vertical, Diagonal hacia al “frente del tablero” (↘) y hacia atrás del tablero (↙) en unidades de 4 fichas .

Recorriendo así todas las posibles direcciones en las que se podría formar un 4 en línea. Estas unidades de 4 fichas, serán evaluadas por otra función y se les asignará una nota a cada una, que sumará al score universal del tablero.

Se decidió que sí: Hay 4 fichas X iguales, se sumará al score, una nota de 100, indicando que ese es un muy buen tablero al cual podría llegar.

En el caso de que haya 3 Fichas X y un espacio en blanco, solamente se sumará 5 puntos a la nota del tablero.

En el caso de que se encuentren 2 fichas X y 2 vacías solamente se sumarán 2 puntos.

Si llegara a encontrar 3 fichas del oponente y una vacía, entonces vamos a restarle 4 puntos al tablero, ya que el enemigo, está cerca de una victoria, lo que no nos interesa, entonces nos alejamos de ese tablero.

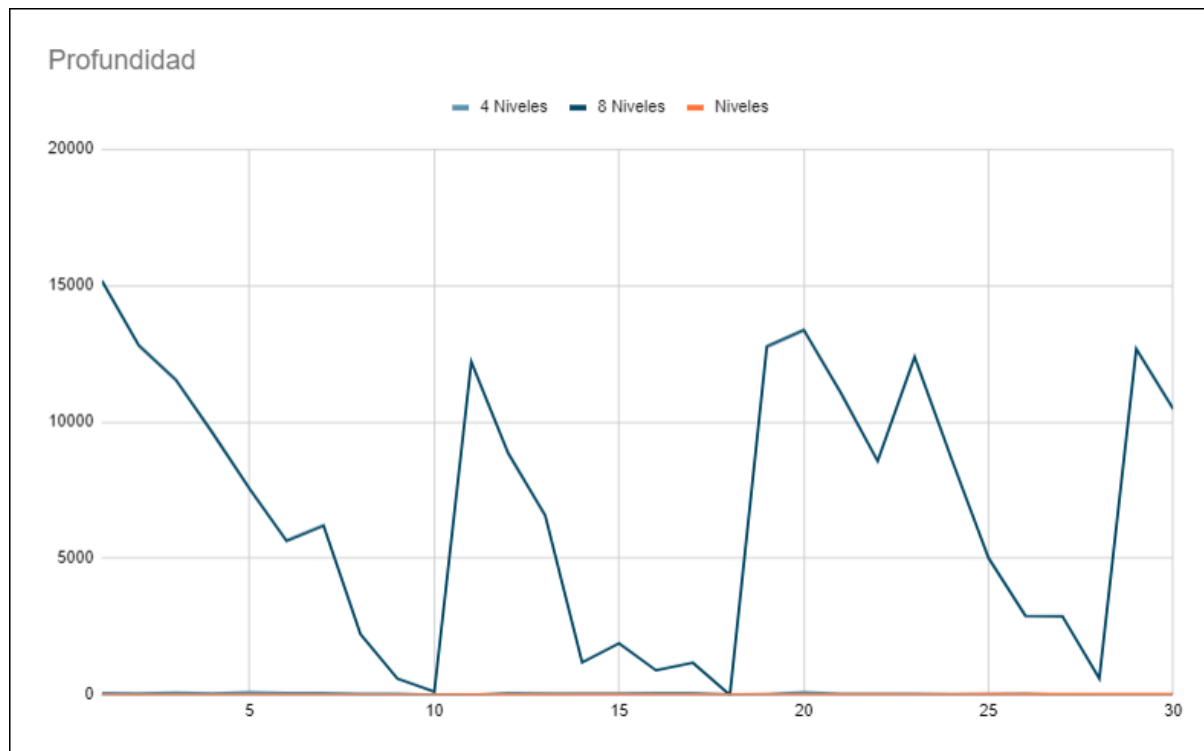
Y si no se cumple ninguna de esas opciones, el tablero simplemente se queda con el mismo score.

Experimentos

A continuación se presentan los experimentos realizados en el juego.

Experimento 1

Este primer experimento consiste en ejecutar el juego con distintas profundidades de expansión del árbol minmax con poda. En este caso se evaluará el tiempo de decisión de cada jugada con profundidades de 4,6 y 8, observando el comportamiento de la función EVAL del juego.



En este caso las pruebas con 30 casos para cada profundidad nos demuestran que entre más alta la profundidad, más alto el tiempo de ejecución (ms). Además gracias a este experimento se puede notar como decrementa el tiempo de ejecución conforme se avanza en los juegos. Cuando se comienza la partida el algoritmo es más lento, pero conforme hay más entradas en el tablero, el algoritmo va siendo más rápido.

Experimento 2

Realmente, no se logró observar una gran diferencia de comportamiento, una vez que los jugadores iniciales eran intercambiados, el algoritmo incluso pudo ganar cuando se lo ponía como segundo jugador. Por lo que realmente no se logra obtener una buena conclusión de este experimento.

Experimento 3

El algoritmo corrió aproximadamente unas 12 veces, de las cuales logró ganar unas 6, dando así un acierto de aprox el 50%. En estas “partidas”, muchas veces el algoritmo logró poner al jugador en situaciones especiales, del tipo, si pone esta pieza, pierde. El algoritmo lo consideramos como inteligente, debido a que de vez en cuando hacía buenas jugadas.

Conclusiones

Se logra observar basado en los experimentos, que las hipótesis planteadas si se cumplen. Y además de que minmax y su función EVAL sí logran un comportamiento inteligente a la hora de tomar decisiones.

Este proyecto fué bastante interesante y productivo para nosotros, nos ayudó a entender mucho acerca de los agentes, de cómo estos se desarrollan y funcionan y también aprender acerca de la programación funcional.

Minmax con poda es un algoritmo excelente que tiene muchas aplicaciones en el mundo de los videojuegos. Aunque no se mostró una interfaz gráfica agradable para el disfrute del juego, este puede ser ejecutado y ser funcional.

Bibliografía

Mora, M. (2021). Diseño de bases de datos [Ebook]. Retrieved 6 September 2021, from

<https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC6200/S-2-2021.AL.IC6200.20/file-storage/view/clases%2F03-modelo-de-agentes%2F01-Agentesv2.pdf>.

Poda alfa-beta. Es.wikipedia.org. (2020). Retrieved 6 September 2021, from https://es.wikipedia.org/wiki/Poda_alfa-beta.

Significado de Heurística. Significados. (2017). Retrieved 6 September 2021, from <https://www.significados.com/heuristica/>.