

Árboles de búsqueda mediante monte carlo para juegos

Juan Sebastián Mora Rueda - 2181960

Jose Fabian Jimenez Ovalle - 2181685

Jorge Luis Sandoval Anaya - 2182028

Javier Fernando Carvajal Sanabria - 2173035

Facultad de Ingeniería Fisicomecánicas, Universidad Industrial de Santander

Ingeniería de Sistemas

Profesor Sergio Augusto Gélvez Cortés

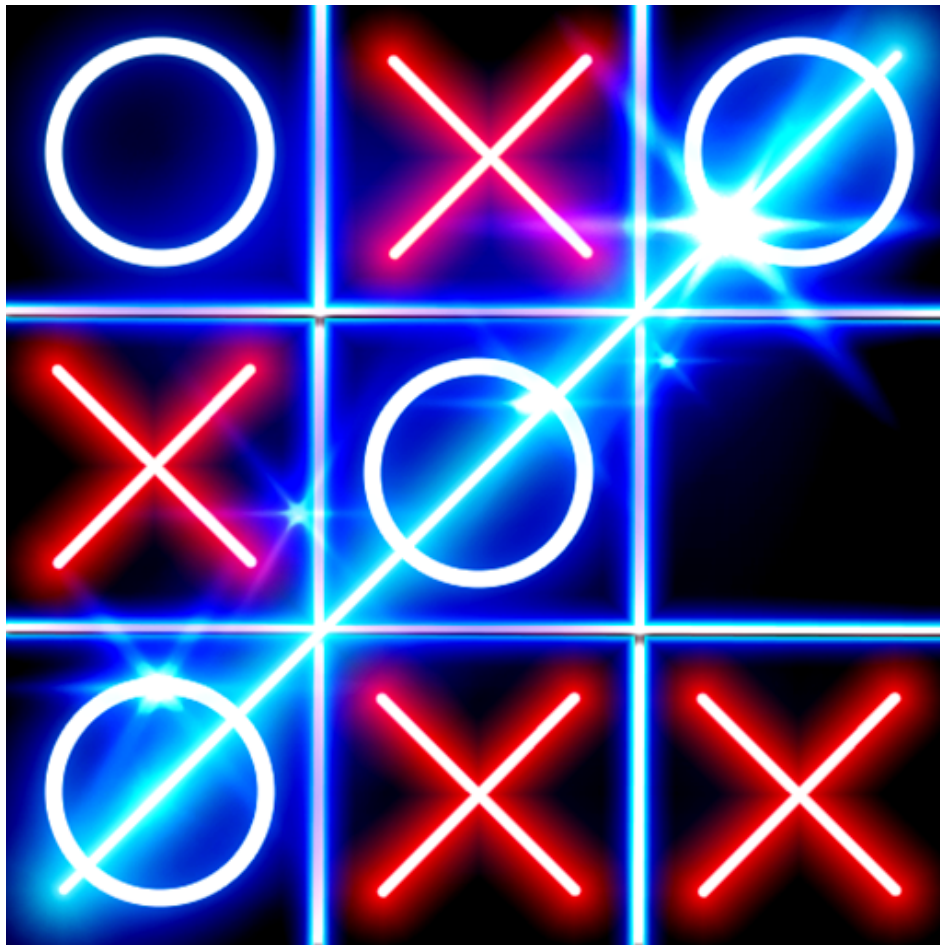
de septiembre de 2022

Tabla de contenido

1. Resumen.....	3
2. Introducción.....	4
3. Marco teórico.....	5
4. Desarrollo.....	6
5. Resultados.....	7 8 9
6. Conclusiones.....	10
7. Referencias.....	10

1. Resumen

Dada la existencia de diversos problemas deterministas los cuales se desarrollan a modo de procesos de decisión, esto lo podemos ver de manera más común en los juegos de mesa, donde de acuerdo a la situación se exploran las opciones y se determina la mejor opción. En ese contexto, MCTS se utiliza para resolver el árbol de juego. MCTS ha sido implementado en diferentes software de juegos de mesa como poker, ajedrez y su variante japonesa llamada shogi, y en este caso analizar su funcionamiento en el juego tic tac toe.



2. Introducción

En un mundo donde el avance de las tecnologías se hace una prioridad la automatización de los procesos donde el valor dado al tiempo es fundamental y el descubrimiento de nuevos problemas por resolver. Algunos de estos problemas donde existen procesos de decisión, en un principio se tiene que solo el ser humano puede a través del análisis y experiencia poder elegir la que a su parecer es la mejor opción, pero debido a que estos factores que son excepcionales, en el momento que incursiona la informática, se empiezan a crear modelos o técnicas para resolver estos problemas en un inicio se plantean los árboles de búsqueda que consiste una estructura de datos de tipo árbol utilizado para localizar llaves concretas dentro de un conjunto. En tiempos actuales el uso de éstas estructuras se pueden observar en el desarrollo de IA de un juego, es decir, videojuegos que simulan juegos de mesa y juegos de estrategia por turnos. En ese sentido el caso de estudio es cómo a través de un tipo de árbol de decisión que combina el método de Monte Carlo podemos simular el desarrollo del juego tic tac toe, donde la búsqueda de árboles de Monte Carlo (MCTS), que es un método para encontrar decisiones óptimas en un dominio dado tomando muestras aleatorias en el espacio de decisión y construyendo un árbol de búsqueda según los resultados.

3. Marco teórico

Método de Monte Carlo: Son una amplia clase de algoritmos computacionales que se basan en un muestreo aleatorio repetido para obtener resultados numéricos.

Exploración: Búsqueda de diferentes opciones para el desarrollo de un problema.

Explotación: Desarrolla las mejores acciones y estrategias encontradas al momento.

Selección: Desde la raíz recorrer los hijos y analizar, desde el más prometedor, hasta el menos prometedor en esa etapa que podría ser la mejor.

Expansión: Creación de nuevos hijos si este punto es diferente al final.

Simulación: Ejecución de un juego al azar del hijo seleccionado, hasta llegar a una configuración final

Retropropagación: Utiliza el resultado del juego de forma aleatoria y actualiza la información en la rama desde el nodo hijo hasta la raíz.

Heurística: Método para aumentar el conocimiento.

4. Desarrollo

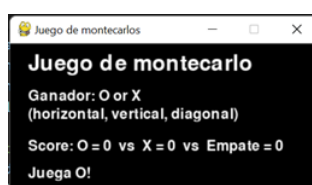
Para la ejecución del método MCTS en el juego tic tac toe, utilizamos python y diferentes librerías como numpy, matplotlib y pygame para la parte visual del software, luego para implementar el método MCTS, seguimos cuatro pasos fundamentales selección, expansión, simulación y actualización, en la selección tomamos de manera aleatoria un nodo hasta obtener la estadísticas necesarias para escoger el camino más comprometedor, posteriormente pasamos a la expansión este paso se lleva a cabo siempre y cuando la partida no se encuentre finalizada, se crea un nodo hijo aleatoriamente que no ha sido visitado y se genera un nuevo registro que se agrega al árbol, ahora comienza la simulación la cual sigue los mismos parámetros de una simulación de Monte Carlo, es decir se van añadiendo eventos de manera aleatoria hasta obtener una gran cantidad y poder llegar a una probabilidad certera de cada movimiento del juego hasta terminar la partida, finalmente llegamos a la actualización la cual ocurre cuando el juego ha terminado, se actualizan las estadísticas de todas las posiciones visitadas previamente durante la partida.

$$UCT(v_i, v) = \frac{Q(v_i)}{N(v_i)} + c \sqrt{\frac{\log(N(v))}{N(v_i)}}$$

La función UCT se encarga de minimizar riesgos, es decir si se tiene todas las estadísticas de bondad se puede encontrar el nodo hijo con el potencial más comprometedor para resolver la simulación de manera eficaz.

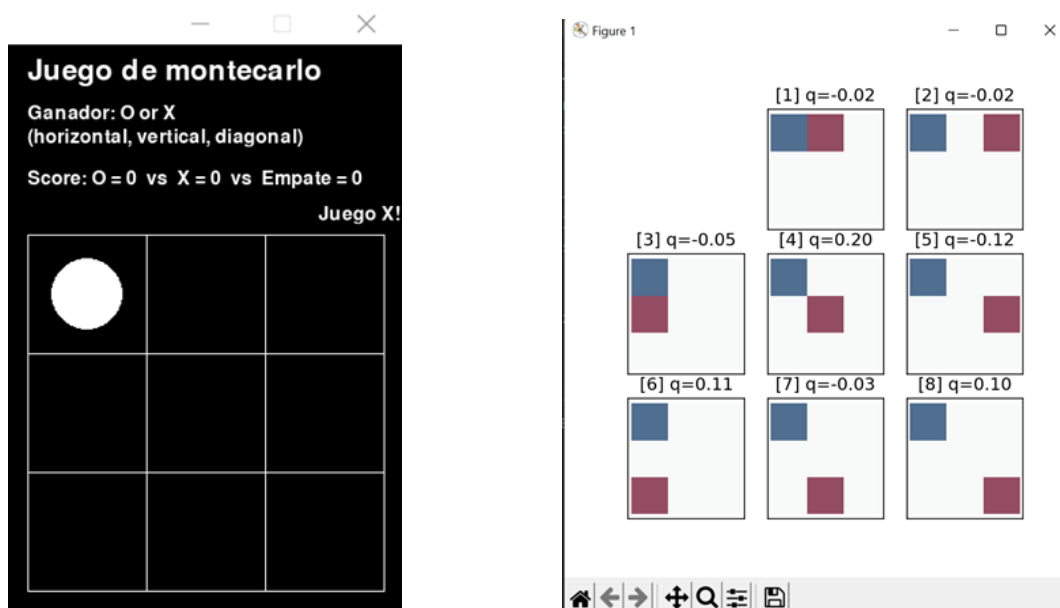
5. Resultados

A continuación, se mostrará el escenario en el cual se va a llevar a cabo la simulación.

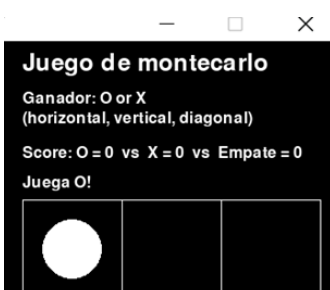


Como se puede ver en la imagen de nuestra simulación cuenta con un tablero 3*3 en el cual se va a llevar todo el proceso del juego tic tac toe, también lleva el registro de las partidas.

Se procede a iniciar la simulación, se marca el círculo en la posición (0,0), luego el software nos muestra las matrices con los diferentes movimientos y sus probabilidades.

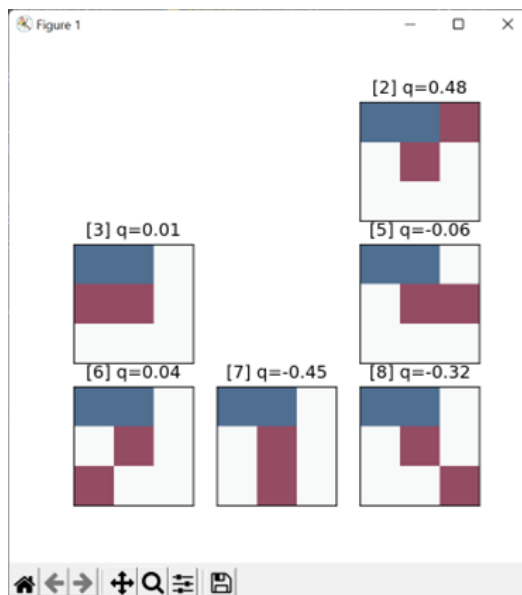
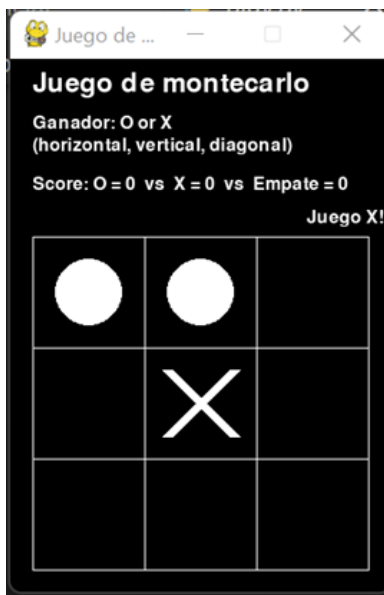


El software escoge el movimiento más asertivo, es decir el que tenía mayor probabilidad, en este caso la posición (1,1).

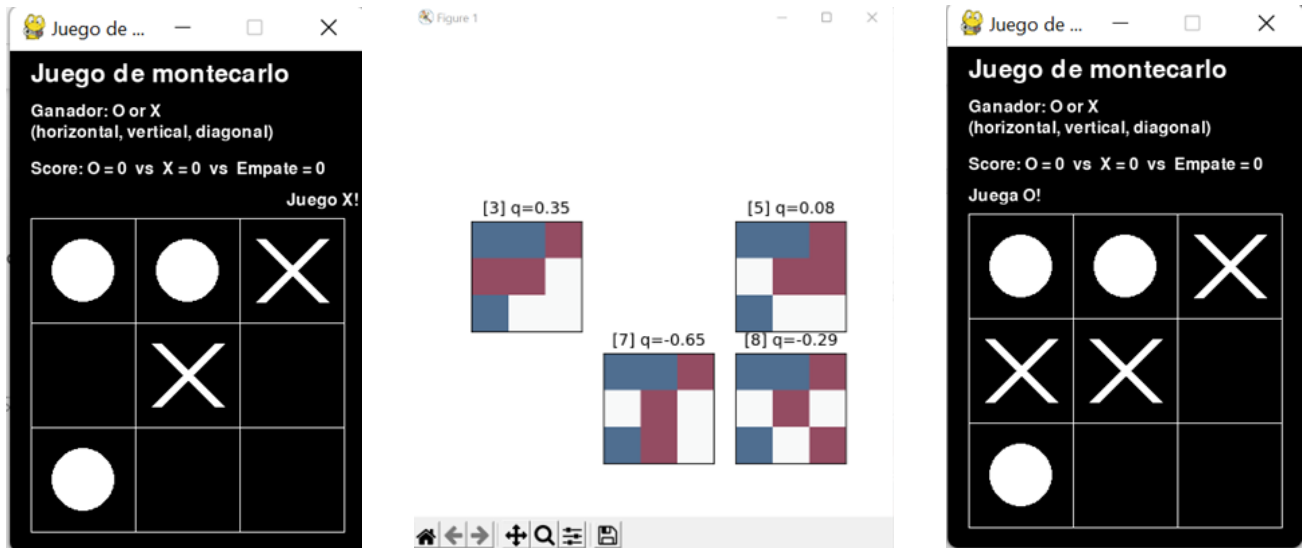


De esta manera se lleva a cabo el proceso de nuestra simulación a continuación, se mostrará el desenlace de esta partida de tic tac toe.

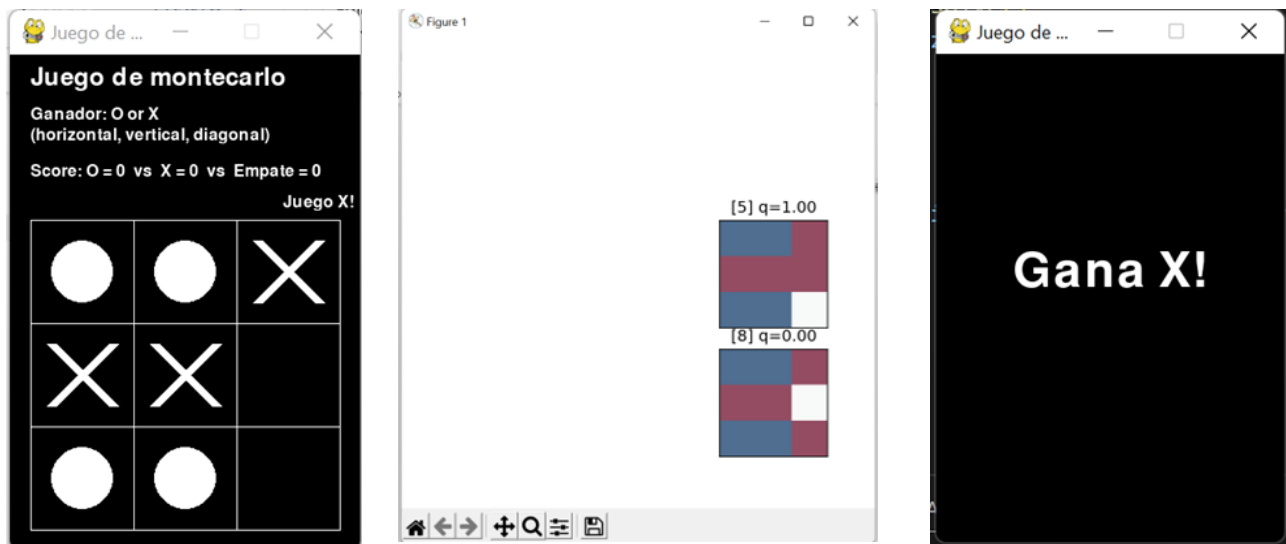
Se procede a marcar la posición (0,1), luego el software selecciona la casilla (0,2), es el movimiento con mayor probabilidad para este caso.



Se procede a marcar la posición (2,0), luego el software selecciona la casilla (1,0), es el movimiento con mayor probabilidad para este caso.



Se procede a marcar la posición (2,1), luego el software selecciona la casilla (1,2) y se finaliza el juego ya que se obtiene un ganador.



6. Conclusiones

- El método MonteCarlo tiene mucha importancia actualmente porque nos permite la resolución de ciertos problemas, pero esto debido al avance de los computadores. La eficiencia de este método se encuentra en que MonteCarlo no recorre todos los nodos del árbol sino que calcula lo que serían los mejores nodos y así se ahorra mucho recorrido.
- Es importante saber que cada vez la máquina es más parecida al ser humano, lo suficiente como para competir contra ella, y esto es claro en los juegos.

7. Bibliografía

- <https://programmerclick.com/article/95711486601/>
- <https://es.acervolima.com/ml-busqueda-de-arboles-de-montecarlo-mcts/>
- https://es.frwiki.wiki/wiki/Recherche_arborescente_Monte-Carlo
- http://opac.pucv.cl/pucv_txt/txt-7500/UCC7969_01.pdf