

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

E.P. INGENIERÍA DE SISTEMAS

ASIGNATURA DE INTELIGENCIA ARTIFICIAL



DOCENTE: Maguiña Pérez, Rolando Alberto

Sección: 1

Trabajo Computacional N°1

“ATRAPADOS”

ESTUDIANTE:

Valdivia Marcelo, Sergio Manuel

LIMA- PERÚ

2023

Contenido

1. Descripción del juego.....	3
2. Representación como Espacio de Estados	4
3. Implementación.....	8
3.1. Inicio del juego	8
3.1.1. Modo de juego.....	8
3.1.2. Dificultad	9
3.2. Desarrollo del juego.....	9
3.3. Fin del juego	9
4. Estrategia para las jugadas	10
4.1. Estrategia Aleatorio	10
4.2. Estrategia Primero el Mejor.....	10
5. Experimentos.....	11
5.1. Experimento 1	11
5.2. Experimento 2.....	12
5.3. Experimento 3.....	13
5.4. Experimento 4.....	14
5.5. Experimento 5.....	15
6. Conclusiones	16
7. Dificultades	16
8. Recomendaciones.....	16

1. Descripción del juego

Atrapados es un juego de estrategia para dos jugadores que se juega en una tabla de 4 filas y 8 columnas. Cada jugador comienza con cuatro fichas, todas del mismo color. El objetivo del juego es atrapar todas las fichas del oponente, es decir, impedir que se muevan.

En cada turno, un jugador mueve una de sus fichas hacia adelante o hacia atrás a lo largo de su fila, tantas casillas como desee, siempre y cuando no salte por encima de otra ficha. El jugador que logra atrapar todas las fichas del oponente gana la partida.

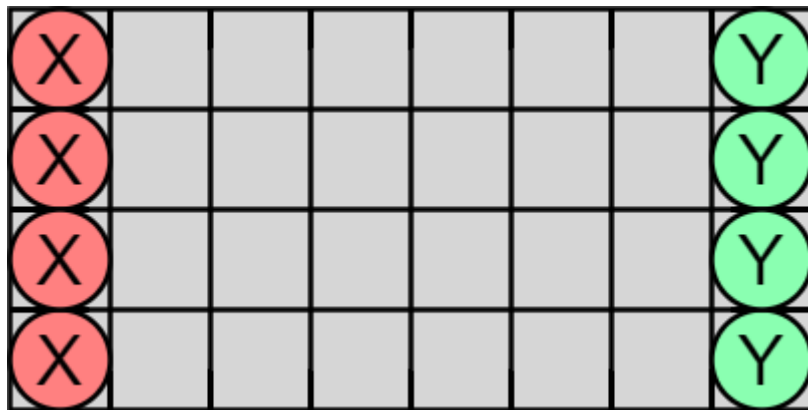
El juego se caracteriza por la complejidad de las posibilidades de movimiento, ya que los jugadores pueden desplazar sus fichas en cualquier dirección, siempre y cuando no salten por encima de las fichas del oponente. Por lo tanto, se requiere una estrategia cuidadosa y una anticipación de los movimientos del oponente.

2. Representación como Espacio de Estados

El espacio de estados para el juego "Atrapados" se define por una tabla de 4 filas y 8 columnas, lo que da un total de 32 espacios jugables. Cada espacio puede estar vacío o contener una de las 8 fichas disponibles para cada jugador. El jugador 1 tiene las fichas "X" y el jugador 2 tiene las fichas "Y", cada jugador tiene 4 fichas en total. El objetivo del juego es capturar las fichas del oponente, inmovilizándolas entre dos fichas propias, lo que las saca del juego.

Figura 1

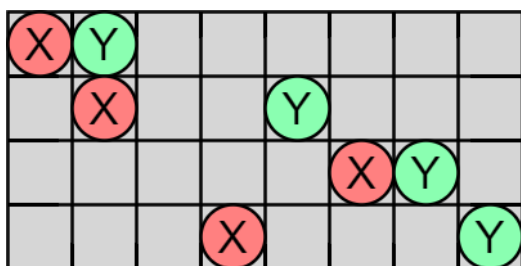
Estado inicial del juego



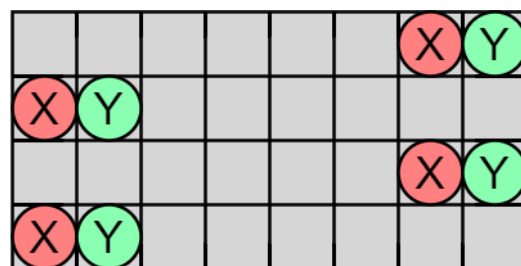
El estado inicial del juego consiste en una distribución igualitaria de las fichas de cada jugador en la tabla. Los jugadores se turnan para mover una de sus fichas, siguiendo las reglas establecidas, con el objetivo de capturar las fichas del oponente y bloquear sus movimientos. El juego termina cuando uno de los jugadores ha capturado todas las fichas del oponente o ha bloqueado todas las fichas del oponente de tal manera que no puede mover ninguna más.

Figura 2

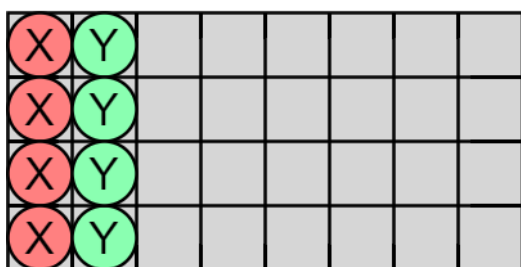
Posibles estados del juego



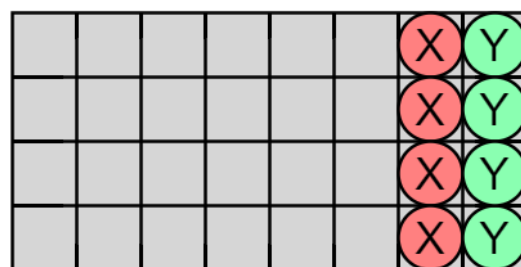
Caso 1



Caso 2



Caso 3



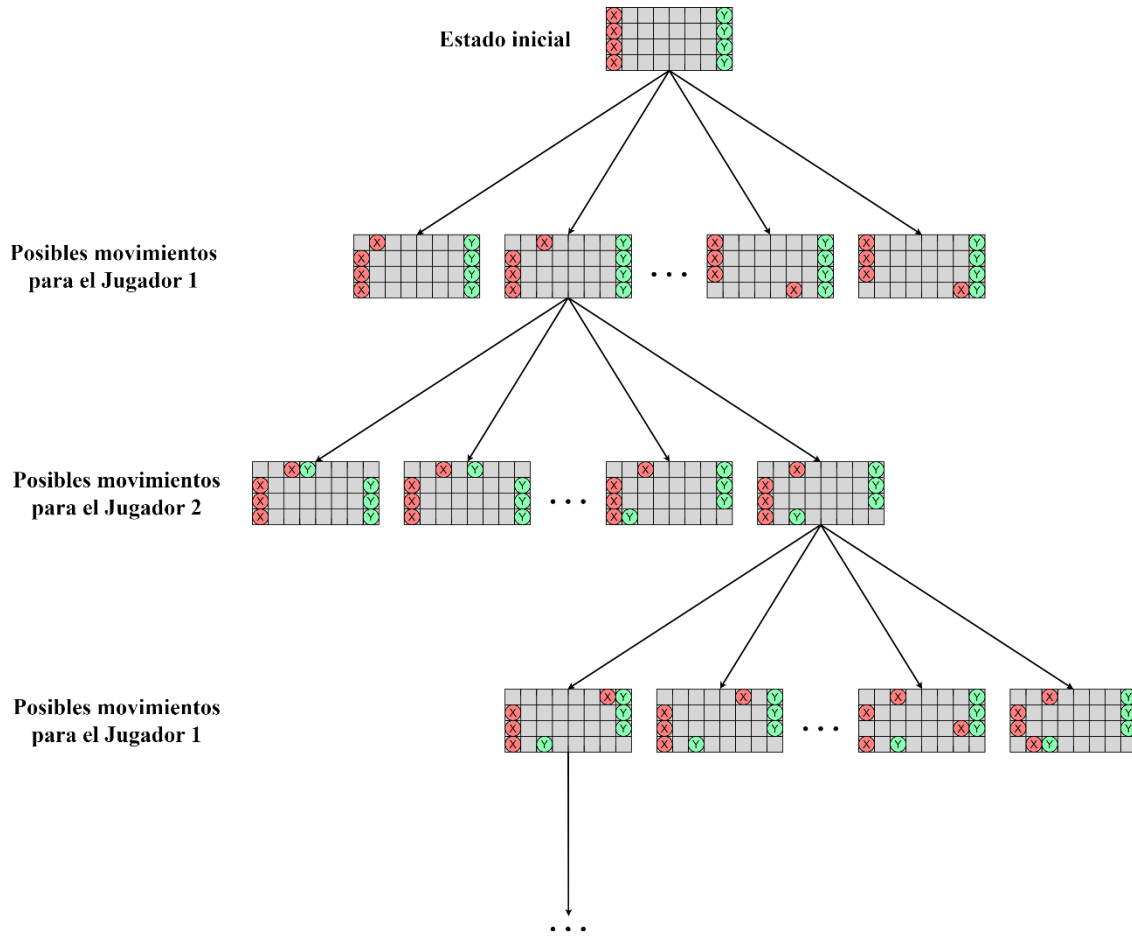
Caso 4

Nota: Los Casos 1 y 2 corresponden a estados aleatorios, posibles y validos del juego, mientras que los Casos 3 y 4 corresponden a las 2 situaciones en donde un jugador gana: Caso 3 gana el jugador 2 (Fichas “Y”), y el Caso 4 gana el jugador 1 (Fichas “X”).

El espacio de estados para este juego es extremadamente grande, lo que significa que la búsqueda de una solución óptima es muy difícil. Por lo tanto, es común utilizar técnicas de búsqueda heurística para encontrar una solución satisfactoria en un tiempo razonable. Una función de evaluación efectiva debe tener en cuenta el número de fichas capturadas y la posición de las fichas restantes en la tabla para determinar la puntuación de un estado.

Figura 3

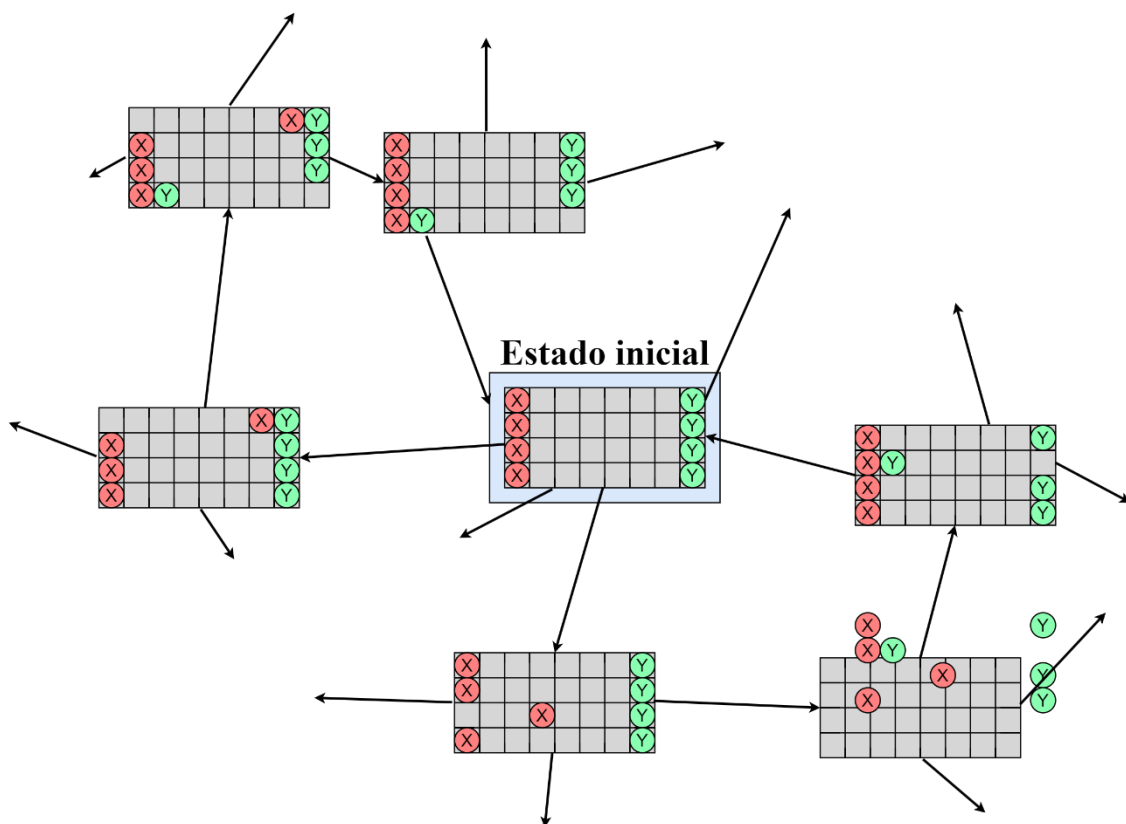
Espacio de Estados representado como Flujo de Juego



Nota: Debido a la naturaleza del juego y a la gran cantidad de estados posibles y válidos, no es viable representar todos los estados.

Figura 4

Espacio de Estados representado como Grafo



Nota: Debido a la naturaleza del juego y a la gran cantidad de estados posibles y válidos, no es viable representar todos los estados.

3. Implementación

Para la implementación del juego en base algoritmos, se utilizará el lenguaje Python. Exactamente se hará uso de Jupyter Notebook, mediante su distribución en Anaconda. El juego se ejecutará en consola. Se estará haciendo referencia a las funciones, variables y resultados del cuaderno computacional adjuntado con el informe, en donde se podrá encontrar todo el código en Python del juego, y la explicación de su funcionamiento. También se puede obtener el cuaderno computacional mediante el siguiente enlace de repositorio en GitHub: <https://github.com/SeRGioVaLD/AtrapadosPython.git>

Para la representación del tablero, se utilizará una lista, la cual tendrá una longitud de 32, correspondiente a las casillas del tablero. Donde el Jugador 1 se representara con el numero 1; y el Jugador 2 con el numero dos; las casillas basias tendran el valor 0.

Figura 5

Posiciones en el tablero

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

3.1.Inicio del juego

El juego inicia con la función **inicio()**, en donde se va a crear el tablero inicial con la función **crearTablero()**. Muestra en pantalla el tablero inicial y el tablero con sus posiciones. Luego de esto, te presenta un menú de modo de juego a elegir con cuatro opciones.

3.1.1. Modo de juego

El menú modo de juego se presenta al jugador, en donde se pueden elegir las siguientes opciones:

3.1.1.1.HUMANO vs PC

El jugador humano juega contra el jugador PC, ingresando por consola sus movimientos. También se le mostrará un pequeño menú para elegir si comenzar primero o dejar que la PC empiece. En cualquier caso, al jugador que empieza se le asigna siempre las fichas “X”.

3.1.1.2.HUMANO vs HUMANO

Se enfrentan dos jugadores humanos. Al primer jugador se le asigna siempre las fichas “X” y al segundo jugador las fichas “Y”. Todos los movimientos se hacen por entrada en consola.

3.1.1.3.PC vs PC

Se enfrentan dos jugadores PC, en donde se le asigna a cada jugador las fichas “X” y “Y” a cada uno. Se representan las jugadas y el resultado final.

La cuarta opción en el menú sirve para salir del programa completamente.

3.1.2. Dificultad

El programa implementa dos dificultades, las cuales corresponden a los dos tipos de estrategia que se pueden elegir para realizar los movimientos del jugador PC.

Aparecerá un pequeño menú con tres opciones, este menú solo se mostrará si anteriormente se eligió el modo de juego 1 (humano vs pc) o 3 (pc vs pc).

3.1.2.1.Dificultad fácil

En esta dificultad, las jugadas del jugador PC se elegirán mediante la estrategia “Aleatoria”

3.1.2.2.Dificultad media

En esta dificultad, las jugadas del jugador PC se elegirán mediante la estrategia “Primero el Mejor”

La tercera opción en el menú sirve para salir del programa completamente.

3.2.Desarrollo del juego

Luego de la elección del modo de juego y la dificultad, el juego procede a realizarse. Según el modo elegido, comenzará el jugador PC o el jugador Humano.

Si empieza el jugador Humano, se ejecutará la función **movimientoHumano()**, que recibirá las jugadas del jugador humano y las validará para poder realizarse. De igual si es que empieza el jugador PC, cuyas jugadas se realizarán según la dificultad (estrategia) elegida y se ejecutan con la función **movimientoAutonomo()**.

El juego se desarrollará así hasta que alguien gane, ya que no se tiene un estado de empate

3.3.Fin del juego

El juego se dará por terminado cuando uno de los dos jugadores gane. Esto será establecido con la función **ganadorPartida()**, que retorna verdadero o falso en la evaluación del tablero que es pasado como parámetro, el cual representa la última jugada que ha realizado cualquiera de los jugadores en ese turno.

Un jugador realiza una jugada, lo que genera un nuevo estado. Este estado se evalúa con la función, si retorna verdad el jugador se declara ganador, sino el juego continúa.

4. Estrategia para las jugadas

Debido al planteamiento del juego, existen dos tipos diferentes de estrategias que se pueden elegir para que el jugador PC elija sus jugadas: Estrategia Aleatorio y Estrategia Primero el Mejor

4.1.Estrategia Aleatorio

En esta estrategia, se generan los sucesores del estado actual, y se elige uno aleatoriamente. Este estado no necesariamente es la mejor opción, simplemente es uno de los estados posibles y validos que se generan a partir del estado actual.

4.2.Estrategia Primero el Mejor

Esta estrategia utiliza el método de búsqueda informada Primero el Mejor, en donde, mediante una función de evaluación se otorga un valor a cada estado sucesor generado a partir del estado actual. Luego de esto, se elige el sucesor con mejor valor de evaluación, ya que este se considera como la mejor jugada posible y valida de realizar.

La función de evaluación es la que nos va a otorgar el peso de cada estado. Para este caso, la mejor función de evaluación debe tener en cuenta la cantidad de posibles movimientos del oponente y los propios. No se considera la cantidad de fichas capturadas debido a que estas ya se cuentan en lo mencionado anterior, y se estaría tomando doblemente su valor. Por lo cual, la función de evaluación seria la siguiente:

Figura 6

Función de Evaluación

$$f(n) = (N^{\circ} \text{ Movimientos validos jugador } X) - (N^{\circ} \text{ Movimientos validos jugador } Y)$$

Donde: $X, Y \rightarrow 1, 2$

Nota: Si “X” toma el jugador uno, “Y” obligatoriamente es el jugador dos y viceversa.

Para la implementación del juego, el jugador elegirá que estrategia quiere que siga el jugador PC. Se considera la Estrategia Aleatorio como un modo de juego fácil, ya que el jugador PC no necesariamente va a elegir la mejor jugada. De hecho, es posible que este elija la peor jugada posible. Mientras que la Estrategia Primero el Mejor se considera de dificultad media, ya que elige el mejor sucesor entre los generados en base al estado actual en base a la función de evaluación. El problema esta en esto mismo, ya que no se puede saber si al elegir este sucesor, los siguientes sucesores van a ser mejores o peores que el primero.

5. Experimentos

Se realizarán 5 experimentos, los primeros 2 utilizando la Estrategia Aleatorio, en otros dos experimentos se utilizará la Estrategia Primero el Mejor, y el ultimo experimento será jugador humano vs jugador humano. Los experimentos hacen referencia a los dos modos de juego; se hará que el jugador PC inicie siempre los juegos.

5.1.Experimento 1

- Modo: Humano vs PC
- Comienza: PC
- Dificultad: Fácil
- Resultado:

Figura 7

Experimento N°1

JUGADA NUMERO: 16								
Ahora toca jugar a humano JUGADOR 2								
Ingrese la numero de ficha que quiere mover: 2								
Ingrese su movimiento: 10								
Movimiento: 10								
1	2	3	4	5	6	7	8	
9	10	11	12	13	14	15	16	
17	18	19	20	21	22	23	24	
25	26	27	28	29	30	31	32	
----- TABLERO -----								
0	X							
0	X							
0	X							
0	X							
----- GANA EL JUGADOR 2 -----								

Se logra ver que, luego de 16 jugadas, el jugador humano resulta el ganador

5.2.Experimento 2

- Modo: PC vs PC
- Dificultad: Fácil
- Resultado:

Figura 8

Experimento N°2

JUGADA NUMERO: 100								
Ahora toca jugar a computadora JUGADOR 2								
Movimiento: 30								
Estrategia: Movimiento por Aleatorio								
	1		2		3		4	
	5		6		7		8	
	9		10		11		12	
	13		14		15		16	
	17		18		19		20	
	21		22		23		24	
	25		26		27		28	
	29		30		31		32	
----- TABLERO -----								
					0			
	0				X			
			0					
			0				X	
---CANTIDAD DE JUGADAS AGOTADAS---								

En la implementación del program, se pone un limite de 100 jugadas para evitar dos cosas: primero, que el juego simulado entre jugadores PC no sea tan largo, en especial en con la Estrategia Aleatorio, donde el juego puede durar indefinidamente; segundo, al evitar que el juego dure infinitamente, se reduce la posibilidad de que el programa falle o se produzca un error por el alto consumo de recursos. El resultado final fue que no se logró terminar el juego antes de consumir todas las jugadas (esto no se considera empate ya que, como se explicó en un inicio, no es posible un empate).

5.3.Experimento 3

- Modo: Humano vs PC
- Dificultad: Media
- Resultado:

Figura 9

Experimento N°3

JUGADA NUMERO: 28								
Ahora toca jugar a humano JUGADOR 2								
Ingrese la numero de ficha que quiere mover: 3								
Ingrese su movimiento: 18								
Movimiento: 18								
	1		2		3		4	
	5		6		7		8	
	9		10		11		12	
	13		14		15		16	
	17		18		19		20	
	21		22		23		24	
	25		26		27		28	
	29		30		31		32	
----- TABLERO -----								
	0		X					
	0		X					
	0		X					
	0		X					
----- GANA EL JUGADOR 2 -----								

En este caso, gana el jugador humano en 28 jugadas. Da igual si se comienza primero o segundo, ya que siempre es posible ganarle a la PC con buenas jugadas, y recordando las posibilidades de la estrategia que está utilizando el oponente.

5.4.Experimento 4

- Modo: PC vs PC
- Dificultad: Media
- Resultado:

Figura 10

Experimento N°4

JUGADA NUMERO: 28								
Ahora toca jugar a computadora JUGADOR 2								
Movimiento: 10								
Estrategia: Movimiento por Primero el Mejor								
	1		2		3		4	
	5		6		7		8	
	9		10		11		12	
	13		14		15		16	
	17		18		19		20	
	21		22		23		24	
	25		26		27		28	
	29		30		31		32	
----- TABLERO -----								
	0		X					
	0		X					
	0		X					
	0		X					
----- GANA EL JUGADOR 2 -----								

Este caso es parecido al experimento 3, ya que la PC jugador 2 utiliza el mismo método que utilizó el jugador humano, solo que lo realiza automáticamente.

5.5.Experimento 5

- Modo: Humano vs Humano
- Resultado:

Figura 11

Experimento N°5

JUGADA NUMERO: 8
Ahora toca jugar a humano JUGADOR 2

Ingrese la numero de ficha que quiere mover: 3

Ingrese su movimiento: 18
Movimiento: 18

	1		2		3		4		5		6		7		8	
	9		10		11		12		13		14		15		16	
	17		18		19		20		21		22		23		24	
	25		26		27		28		29		30		31		32	

----- TABLERO -----

	0		X													
	0		X													
	0		X													
	0		X													

----- GANA EL JUGADOR 2 -----

Este experimento únicamente se realiza como corroborar el buen funcionamiento del programa cuando se intenta utilizar como una simulación del juego real, donde dos jugadores humanos se enfrentan entre sí.

6. Conclusiones

- Los experimentos demuestran que la aplicación de las Estrategias Aleatorio y Primero el Mejor se realizaron correctamente.
- La función de evaluación es correcta y aplicable a otros casos, aunque puede que en otros tipos de búsqueda sea necesario su modificación.
- La complejidad del juego, su duración, y las posibilidades de juego dependen totalmente del tamaño del juego y sus reglas.
- Sobre los experimentos, pese a la posibilidad de generar jugadas y obtener los mejores resultados por parte de la PC, los jugadores humanos pueden derrotarla planteando una buena estrategia.
- El planteamiento del modo PC vs PC es muy útil para conocer el alcance de los metodos de búsqueda.
- El planteamiento del modo Humano vs Humano permite el utilizar un mismo programa por varios usuarios.
- Los resultados de los experimentos nos ayudan a comprender que tan complejo logra ser el juego.
- El planteamiento de este tipo de juegos utilizando metodos de búsqueda informada o aleatoria es realmente beneficioso para el estudio de la inteligencia artificial.

7. Dificultades

- El planteamiento del código y sus resultados resulta tedioso y necesitado de constante testeo.
- Es inviable representar todos los estados posibles validos del juego.
- El planteamiento de una función de evaluación optima resulta laborioso, además que es posible que se dejen atrás posibles estados o situaciones favorables a la implementación.
- La ejecución del programa en consolas es muy tediosa, ya que es necesario ingresar varios parametros para poder realizar una sola acción.
- Establecer un límite de jugadas resulta favorable para el optimo desarrollo del programa, pero disminuye sus posibilidades de uso y funcionamiento.

8. Recomendaciones

- Se recomienda el planteamiento de una tercera dificultad en el juego utilizando el método de búsqueda informada MINMAX.
- El código de la implementación del programa aún puede ser optimizado para que consuma menos recursos.
- Es muy recomendable incorpórale una interfaz gráfica al juego, ya que resultaría más cómodo de jugar de esa forma.
- También seria viable el agregar una opción en donde se modifique las dimensiones del tablero para plantear distintos modos de juego.
- Por último, se recomienda seguir realizando experimentos para conocer el grado de dificultad que puede llegar a tener el juego y su planteamiento como juego inteligente