

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5



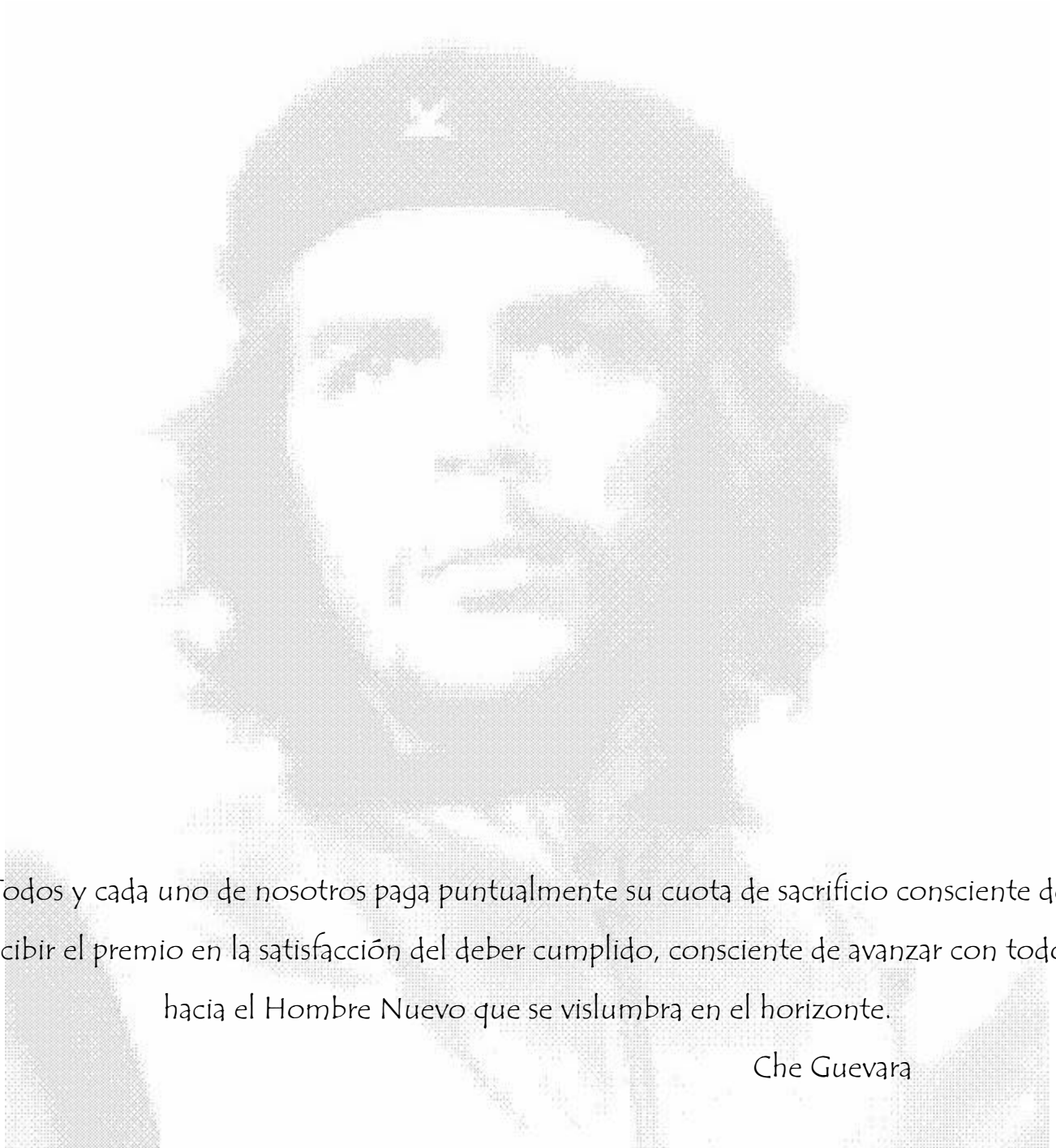
TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

TÍTULO: Aplicación del algoritmo Min -Max en el desarrollo de juegos de contrincantes en un entorno virtual.

AUTOR: Alexander Sardina Pompa.

TUTOR: Lic. José Miguel de la Rosa Trevín.

2 de Julio de 2008



Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, consciente de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Che Guevara

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Alexander Sardina Pompa

José Miguel de la Rosa Trevín

DATOS DE CONTACTO

Nombre y Apellidos: José Miguel de la Rosa Trevin.

Edad: 25.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Licenciado en Ciencias de la Computación.

Categoría Docente: Profesor Instructor.

E-mail: josemdelarosa@uci.cu

AGRADECIMIENTOS

A nuestro Comandante en Jefe Fidel Castro
por tener la maravillosa idea
de crear la Universidad de las Ciencias Informáticas.

A mis padres y familiares
por confiar en mí.

A mi novia Kenia.

A mi tutor
por ayudarme en todo.

A mis amigos
que compartieron todos estos años conmigo,
en especial a Yoan,

Jorge L, Javier, Luís Daniel
y Leonardo R.

A todos los que de una forma u otra
contribuyeron a la realización de este trabajo.

DEDICATORIA

A mis padres
por haber confiado
en mí y apoyarme en todo momento.
A mis abuelos, tíos, primos y hermanos
por estar siempre a mi lado.
A mi novia Kenia por aguantarme durante
tanto tiempo.
A mis hermanos de la UCI .
Alexander

RESUMEN

Unas de las técnicas más utilizadas en la actualidad para desarrollar la inteligencia en los juegos de contrincantes son, el algoritmo Min-Max y algoritmo Min-Max con recorte Alfa-Beta. Estas técnicas son ideales para juegos de contrincantes, donde el árbol de juego sea relativamente pequeño y se puedan explorar hasta lo más profundo, objetivo que de lograrse haría a cualquier ordenador invencible, al ser recursivos estos algoritmos generan una gran cantidad de combinaciones de jugadas, que en algunos casos como el ajedrez es imposible generar todo el árbol de juego, es aquí donde se usa una técnica de inteligencia artificial conocida como heurística. En el presente trabajo se proponen juegos a los que se les puede aplicar estos algoritmos.

Con el desarrollo de la Realidad Virtual (RV) los juegos han alcanzado un alto nivel de realismo por lo que los usuarios prefieren juegos con esta técnica, es por ello que para el desarrollo de los juegos propuestos, estos deberán ser resueltos en un entorno virtual.

PALABRAS CLAVE

Algoritmo, Min-Max, Búsqueda, Árbol de Juego, Juego, Contrincante, Entorno Virtual.

INDICE	
AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Introducción.	4
1.2 Inteligencia Artificial.	4
1.2.1. Definición de Inteligencia Artificial (IA).	4
1.3. Surgimiento y evolución del Algoritmo Min-Max.	5
1.4. Min-Max	7
1.4.1. Definición del algoritmo Min -Max.	7
1.4.2. Estrategia Min-Max.	8
1.4.3. Poda Alfa-Beta.	14
1.5. Diferentes tipos de juegos.	17
1.5.1. Juegos simétricos y asimétricos.	17
1.5.2. Juegos suma cero.....	17
1.5.3. Juegos cooperativos.	18
1.5.4. Juegos simultáneos y secuenciales.	18
1.5.5. Juegos con información perfecta.	18
1.6. Aplicaciones del Algoritmo Min-Max.	19
1.6.1. El juego del Gato o Tres en Raya.	19
1.6.2. Ajedrez.....	20
1.6.3. Damas.....	22
1.7. Realidad Virtual.	24
1.7.1. Definición de Realidad Virtual.	24
1.8. Relación entre IA y RV.	25

CAPÍTULO 2: PROPUESTA DE JUEGOS.	27
2.1. Introducción.	27
2.2. Juegos.	27
2.2.1. Conejo y Perros.	27
2.2.2. Tres en Raya.	30
2.2.3. Damas.	33
2.3. Algoritmo Min-Max para el Juego de Ajedrez.	36
CAPÍTULO 3: CASO DE ESTUDIO	39
3.1. Introducción	39
3.2. Implementación del caso de estudio.	39
3.2.1. Demo:.....	39
3.2.2. Análisis del tiempo que demora el algoritmo.	40
3.2.3. Heurística para el juego Cuatro en Rayas y el N en Raya	43
CONCLUSIONES.	44
RECOMENDACIONES	45
BIBLIOGRAFÍA	46
BIBLIOGRAFÍA CONSULTADA.....	47
ANEXOS.....	48
GLOSARIO	53

Índice de Figuras

Figura # 1. Árbol para representar juegos.....	10
Figura # 2. Árbol generado según el procedimiento Min -Max.....	11
Figura # 3. Camino seguido por el juego para las mejores tiradas de Max y Min. Suponiendo un horizonte de juego que llegue hasta el final.....	14
Figura # 4. Max genera el árbol con horizonte de un nivel y toma la decisión adecuada según la función de utilidad a los nodos que es capaz de ver.....	15
Figura # 5. MAX vuelve a generar el árbol desde la posición del juego que le ha situado MIN con su decisión. Aplica de nuevo la función de utilidad para los nodos que es capaz de ver en el horizonte y decide la mejor tirada. El juego lo finaliza MIN con su tirada, tal y como se puede seguir en el árbol de la figura 1.....	15
Figura # 6. Árbol con la poda Alfa -Beta. Los nodos hojas 4 y 8, numerando desde la izquierda, se puede podar.....	16
Figura # 7. Tablero para el juego de los Perros y el Conejo.....	30
Figura # 8. Árbol de juego para el Tres en Raya.....	33
Figura # 9. Tres cruces en forma diagonal gana Max.....	33
Figura # 10. Tres ceros en una misma columna gana Min.....	34
Figura # 11. Empate.....	34
Figura # 12. Tablero para el juego de las Damas.....	36
Figura # 13. Diagrama de clases para el juego del Gato o Tres en Raya	41
Figura # 14. Tablero vacío para el inicio del juego	43

INTRODUCCIÓN

A lo largo de la historia de la Inteligencia Artificial (IA), los juegos han sido los problemas de mayor atención entre los investigadores de esta rama. Su estudio ha inspirado a científicos de todos los tiempos para el desarrollo de teorías y modelos matemáticos. Una rama de las matemáticas como la estadística surgió precisamente de los cálculos realizados para diseñar estrategias que llevaran a los jugadores de los juegos de azar a lograr la victoria, pero la Teoría de Juegos de contrincantes está muy lejos de la estadística. Su objetivo no es precisamente el análisis del azar o de los elementos aleatorios sino de l comportamiento de una estrategia de los jugadores, creando en el contrincante un nivel de incertidumbre.

En el mundo con el que interactuamos diariamente, en las relaciones económicas como en las políticas o sociales, es muy frecuente encontrarnos con situaciones en las que el resultado depende de la conjunción de decisiones de diferentes agentes al igual que en los juegos.

La teoría matemática de los juegos de estrategia tiene por objeto proporcionar métodos para la toma de decisiones en las situaciones de conflicto o competencia. Su estudio comenzó en el Siglo XVI, por Cardan y Kepler, continuando su estudio varios investigadores en los siglos siguientes hasta la fecha, pero no fue hasta el Siglo XX, en el año 1921, que el matemático Emile Borel presentó el primer estudio orgánico. Sin embargo, el primer avance importante no ocurre hasta la década de los cuarenta, con la publicación del libro sobre Teoría de Juegos de John Von Neumann y Oskar Morgenstern (1944) en el cual se divulgó una formalización general de juegos en su forma extendida, introduciendo el concepto de estrategia, desarrollando esta teoría orientada hacia el análisis de las situaciones de conflicto en cuestiones económica; no obstante, sus conclusiones son también aplicables a situaciones que se presentan en competencias deportivas, tomando como ejemplo el juego de ajedrez.

Los juegos representan de una forma abstracta, situaciones de la vida real que requieren inteligencia. Es por ello que desde los mismos comienzos del estudio de la IA se inician trabajos sobre programas que pretenden jugar, independientemente del tipo de juego que sea. Una de las motivaciones principales de investigación en juegos es que son difíciles de resolver.

La IA es un área que en los últimos tiempos ha despertado gran interés y se ha desarrollado, esto se debe en gran medida a la capacidad de poder resolver situaciones imitando el razonamiento lógico de las personas. Al comienzo del estudio de los juegos estos fueron estudiados por una rama de la ciencia denominada Investigación Operativa, la cual proporcionaba técnicas que solo podrían ser aplicables si existía un procedimiento finito. Debido a este rápido desarrollo de la IA se crean nuevos algoritmos de búsquedas que permiten desarrollar soluciones dentro de procedimientos no finitos.

El presente trabajo estará dirigido al algoritmo Min-Max, este algoritmo se aplica a juegos de contrincantes con información perfecta (es decir, en los que no interviene el azar y en los que cada jugador puede ver toda la información del otro, pero desconoce su estrategia) y de suma cero (sólo se puede GANAR, PERDER o EMPATAR), algunos de estos juegos son las Damas, Ajedrez, Tres en Raya, etc. Estos juegos por lo general pueden ser representados a través de un árbol de estados, en los que cada estado representa un tablero diferente y en donde los lazos de unión entre estado del árbol indican posibles movimientos para pasar de un estado del tablero a otro estado, buscando una estrategia que lo lleve a la victoria.

Si se tratara de una búsqueda normal, un adversario solamente tendría que tomar el camino más adecuado por el árbol hasta encontrar la mejor solución, pero en realidad en un juego de contrincante no es tan fácil, ya que el adversario se encargará de desviar al jugador en su camino hacia la victoria; además, no suele ser posible computar todos los estados de un juego. Para poder operar sobre los juegos como árboles de búsqueda se utilizarán los horizontes limitados y las estimaciones heurísticas de utilidades, de aquí la utilidad del algoritmo Min -Max en estos juegos.

En la actualidad la mayoría de los proyectos para el desarrollo de juegos de contrincantes en la UCI carecen de técnicas avanzadas de Inteligencia Artificial (IA), algoritmos y estrategias ya conocidas y probadas en el mundo, limitando en estos el nivel de realismo e inteligencia.

Dada la **situación problemática** antes planteada y la necesidad de desarrollar juegos de contrincantes a los que se les pueda aplicar el algoritmo Min-Max, los esfuerzos de este trabajo estarán encaminados a

resolver el siguiente **problema**: ¿Cómo utilizar el algoritmo Mini-Max para el desarrollo de juegos de contrincantes?

El **objeto de estudio** es el algoritmo Min-Max y estrategias existentes para el desarrollo de juegos de contrincantes.

El **campo de acción** es el Algoritmo Mini-Max como estrategia para el desarrollo de juegos de contrincantes en entornos virtuales.

La investigación de este trabajo de diploma tiene como **objetivo general**: Proponer juegos de contrincantes a los cuales se les pueda aplicar el algoritmo Min –Max, que permitan fomentar el desarrollo de la línea de juegos en la facultad 5.

Para dar cumplimiento a este objetivo se plantean las siguientes **tareas investigativas**:

1. Analizar aplicaciones existentes del algoritmo Min-Max en las estrategias de juegos de contrincantes.
2. Analizar la situación actual en la facultad sobre el desarrollo de juegos de contrincante en un entorno virtual.
3. Valorar las variantes del algoritmo Min -Max que se pueden aplicar en las estrategias para el desarrollo de la inteligencia artificial en juegos 3D.

Los **posibles resultados** de la investigación se enumeran a continuación:

1. Una base bibliográfica sobre el algoritmo Min-Max en las estrategias de juegos de contrincantes.
2. Propuesta de los posibles juegos a los que se le podría aplicar el algoritmo dentro del área de la RV.

Para el desarrollo de las tareas científicas se combinarán diferentes métodos y técnicas en la búsqueda y procesamiento de la información, entre los que se encuentran:

A nivel Teórico

Histórico-Lógico: Para estudiar la evolución y desarrollo del objeto de estudio de la investigación.

Analítico-Sintético: Durante la definición de los conceptos básicos necesarios para la investigación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

En este capítulo se abordará una reseña histórica del surgimiento y evolución del Algoritmo Min-Max, así como algunas de las aplicaciones generales de los mismos en cualquier rama de las ciencias y fundamentalmente en la construcción de aplicaciones de RV en juegos. También se abordará el concepto de IA, y RV, la relación entre ambas, así como los diferentes tipos de juegos, profundizando en los juegos de adversario y el algoritmo Min-Max.

1.2 Inteligencia Artificial.

1.2.1. Definición de Inteligencia Artificial (IA).

La IA dentro de las ciencias de la computación es la encargada de la creación de máquinas que implementan tareas relacionadas con el comportamiento humano, logrando gran similitud al pensamiento del hombre.

La IA es una de las disciplinas más nuevas dentro de las ciencias de la computación, se puede decir que su estudio formal comienza en el año 1956 cuando se acuñó por primera vez este término, pero el estudio del comportamiento de la inteligencia contemplada con el razonamiento humano, se viene estudiando por filósofos desde hace más de 2 mil años.[10]

Se puede decir también que la IA es la ciencia que investiga la posibilidad de que un ordenador sea capaz de modificar la programación para el cual fue creado en función de la experiencia adquirida y que pueda aprender de esa experiencia.

Los científicos que se centran en esta área de investigación sueñan con la creación de un cerebro inteligente, aunque este sueño este muy bien lejos de hacerse realidad la aplicación de la IA ha servido para elaborar sistemas y dispositivos con cierto grado de inteligencia, por solo citar algunos ejemplos, en la actualidad la mayoría de los aviones de combate en los países desarrollados no llevan pilotos, disminuyendo en gran medida las pérdidas de vidas humanas, brazos robotizados con varios grados de libertad, demostración de teoremas y sistemas expertos que diagnostican enfermedades y prescriben remedios, así como la simulación de la inteligencia natural a través de dos vertientes: la

computación conexionista (redes neuronales) y la computación simbólica (aplicación de las reglas lógicas).

1.3. Surgimiento y evolución del Algoritmo Min-Max.

Considerando una clase de juego simple de dos personas de suma cero, Félix Édouard Justin Emile Borel estableció que es necesario investigar si es posible determinar un método de juego que es mejor que todos los otros. Este método de juego fue entendido como un código que determina para cada posible circunstancia finita, exactamente lo que la persona debe hacer. Con estas afirmaciones Borel se sintió libre para ignorar la estructura temporal extensiva de los juegos, de manera que pudiesen expresarse como una matriz de números que especifican el valor esperado para cada jugador con sus métodos de juego.

Años mas tarde Von Neumann continuó con la idea de Borel y explícitamente formuló un modelo general de juegos extensivos, en los cuales los jugadores se mueven secuencialmente en el tiempo con información imperfecta sobre los movimientos de cada uno de los otros jugadores. Debido a que los jugadores pueden obtener alguna información sobre los movimientos previos de otros jugadores, no podemos asumir que los movimientos de los jugadores son independientes en esta clase de juegos extensivos. Siguiendo a Borel, Von Neumann definió una estrategia para cada jugador como un completo plan, el cual especifica un movimiento para el jugador, a cada estadio o escenario cuando le corresponde está en modo activo, en función de su información en el estadio.

Un jugador racional puede elegir su estrategia antes de que el juego inicie, sin pérdida de generalidad, debido a que la estrategia le permite especificar un movimiento diferente para cada situación en la cual el jugador puede encontrarse durante el juego (Árbol de decisiones). Como esto sucede antes de iniciar el juego, sin consecuencias todavía, Von Neumann concluye que cada jugador debe elegir su estrategia sin estar informado de las elecciones estratégicas de los otros jugadores. Así Von Neumann argumenta que cualquier juego competitivo puede ser modelado por un juego matemático con la siguiente estructura simple:

- ✍ Un conjunto de jugadores.
- ✍ Cada jugador con un conjunto de estrategias.
- ✍ Cada jugador tiene una función de pagos que proviene del producto cartesiano de estos conjuntos de estrategias en números reales.

☞ Cada jugador debe elegir su estrategia independientemente de los otros jugadores.

Von Neumann publicó una serie de artículos en 1928 donde generaliza el teorema Min-Max, adjudicándose ser el creador de la teoría. Estos resultados fueron ampliados más tarde en su libro de 1944, *The Theory of Games and Economic Behavior*, escrito junto con Oskar Morgenstern. Este esquema fue denominado por Von Neumann y Morgenstern como la forma normal para la representación general de juegos extensivos.

Von Neumann no aplicó de manera consistente el principio de independencia estratégica esbozado. En su análisis de juegos con más de 2 jugadores, asumió que los jugadores podrían no solamente elegir sus estrategias independientemente, sino coordinar sus estrategias en coaliciones. Adicionalmente agregó dos restricciones adicionales a la forma normal: los pagos son transferibles, y todos los juegos son de suma cero, en parte debido a su aporte sobre el teorema Min-Max.

En 1928, Von Neumann demostró la existencia general de soluciones Min-Max en estrategias aleatorizadas para juegos de suma cero para dos jugadores. Para estos juegos, el teorema Min-Max es lógicamente equivalente a la existencia del equilibrio de Nash, empleando formulaciones matemáticas como el uso del teorema de punto fijo para probar la existencia de soluciones para juegos con aleatorización de estrategias. Von Neumann formuló el teorema Min-Max como una igualdad entre los valores que cada jugador puede garantizar por sí mismo, sin considerar lo que el oponente pueda hacer, no como una optimalidad mutua sobre un grupo particular de estrategias. Al formularse así el teorema no puede ser extendido más allá del caso de dos jugadores para juegos de suma cero.

En 1947, Von Neumann y Morgenstern publicaron su tercera contribución principal a la teoría de juegos; la derivación axiomática de la maximización de utilidad esperada desde un argumento de sustitución. Durante este período, el trabajo sobre teoría de juegos se centró sobre todo, en teoría de juegos cooperativos. Este tipo de teoría de juegos analiza las estrategias óptimas para grupos de individuos, suponiendo que pueden establecer acuerdos entre sí acerca de las estrategias más apropiadas.

El algoritmo Min-Max en su búsqueda para la solución de problemas, muchas veces tiene que evaluar ramas innecesarias que no le traerán beneficio alguno para obtener el mejor resultado. Por lo tanto, es

aquí donde se plantea la solución mediante la poda de esas ramas que no beneficiaran en nada a la mejor jugada: La técnica de Poda Alpha-Beta.

Este algoritmo es el más utilizado en las aplicaciones referidas a juegos, dada su excepcional utilidad en el aumento de la velocidad de la búsqueda sin producir pérdida de la información. Es una extensión en particular del algoritmo de Búsqueda Min-Max en juegos de dos contrincantes. [1]

1.4. Min-Max

1.4.1. Definición del algoritmo Min-Max.

Min-Max es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario y con información perfecta.

El funcionamiento de Min-Max puede resumirse de la siguiente manera: cómo elegir el mejor movimiento (Max) suponiendo que tu contrincante escogerá el peor (Min). En este método lo que se hace es observar todas las recompensas que se puedan obtener con cada estrategia y por cada una de ellas encontrar la mínima recompensa. La idea consiste en comenzar en la raíz o en la posición donde nos encontramos y usar el generador de movimientos posibles para generar las posibles posiciones sucesivas hasta un cierto límite de niveles.

A continuación se aplica la función de evaluación estática a las posiciones obtenidas y se elige la mejor posición para el jugador correspondiente, llevando los valores a un nivel hacia atrás para continuar la evaluación en todos los niveles anteriores. Una función de evaluación estática devuelve valores elevados para indicar buenas situaciones y valores negativos para indicar buenas situaciones para el oponente. Visto de esta manera, la meta es maximizar el valor de la función estática de la siguiente posición en el tablero. El nombre del algoritmo deriva de considerar que, dada una función estática que devuelve valores en relación al jugador maximizante, éste procura maximizar su valor mientras que su oponente procura minimizarlo. En un árbol de juego donde los valores de la función estática están en relación al jugador maximizante, se maximiza y minimiza alternadamente de un nivel a otro.

El algoritmo Min-Max es un procedimiento recursivo y el corte de la recursión está dado por alguna de las siguientes condiciones:

- ☞ Gana algún jugador.

- ✍ Se han explorado N capas, siendo N el límite establecido
- ✍ Se ha agotado el tiempo de exploración
- ✍ Se ha llegado a una situación estática donde no hay grandes cambios de un nivel a otro.

Una vez encontradas las mínimas recompensas para cada estrategia, se toma la mayor de ellas, siendo esta la que utilizaremos, ya que nos garantiza que al utilizar esta estrategia nuestra recompensa no sería menor al valor indicado. Para saber lo que a nuestro oponente le conviene decidir se hace el análisis inverso, es decir, se obtienen las máximas recompensas de cada estrategia y al final se toma la mayor recompensa obtenida.

La intersección de las estrategias es la solución al juego.

Complejidad del Algoritmo.

La complejidad en tiempo del algoritmo Min-Max es $O(b^m)$ donde b es el factor de ramificación y m la profundidad donde se encuentra el nodo terminal mas próximo, si bien la búsqueda se hace primero en profundidad, se deben explorar todos los nodos terminales, salvo que se ponga un límite de profundidad, con lo cual la decisión es imperfecta y no se puede garantizar el resultado.[2]

1.4.2. Estrategia Min-Max.

La estrategia Min-Max se aplica a un juego de información perfecta, y de suma cero, donde los jugadores son llamados Max y Min, en el que se van alternando los turnos hasta llegar al final del juego. Se considera que el jugador catalogado como Max realizará el primer movimiento, momento a partir del cual los dos jugadores van realizando jugadas alternas. Los nodos en los que Max ha de realizar movimiento, los de profundidad par (la raíz tiene profundidad 0), serán catalogados como nodos Max, mientras que los impares en los que tiene que realizar movimiento Min, se llamarán nodos Min. Los movimientos Min se corresponden con un movimiento de nuestro contrincante mientras que los movimientos Max se corresponden con uno de nuestros movimientos. [3]

El juego se representa mediante un árbol de búsqueda en el que cada nodo representa un estado posible del juego, siendo el nodo raíz la situación inicial. Los nodos que se encuentran en un mismo nivel representan el turno de un mismo jugador, indicando las posibles jugadas del jugador.

En la siguiente ilustración se puede ver un árbol de un juego para la estrategia Min-Max.

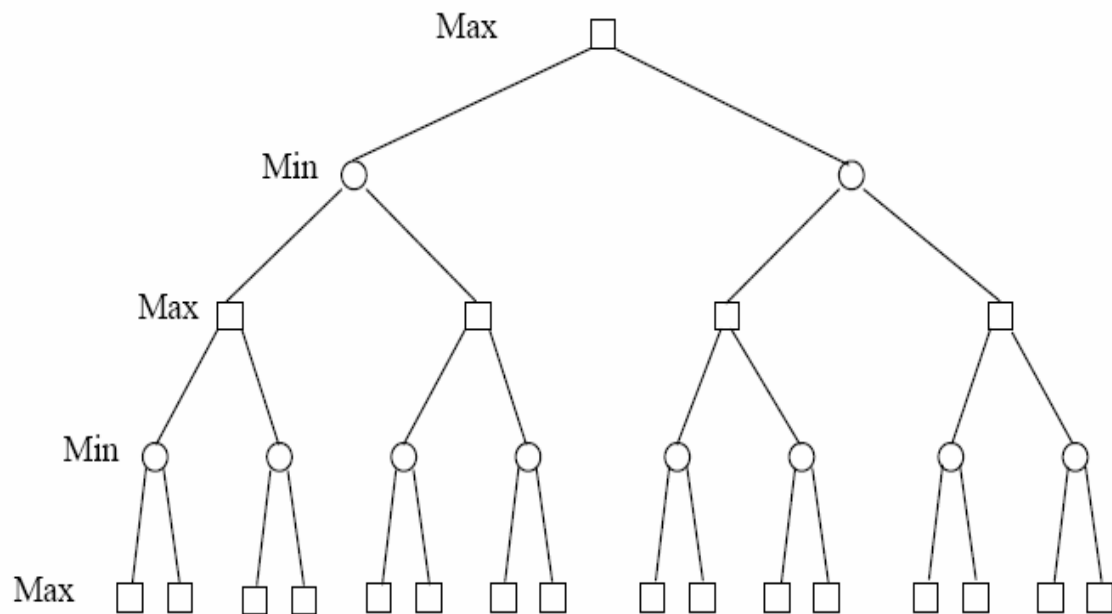


Figura # 1. Árbol que representa un juego aplicando el algoritmo Min-Max.

En la figura anterior los nodos terminales se corresponden con un movimiento que lleva al final del juego y por tanto a una puntuación.

El algoritmo Min-Max busca hacia delante en el árbol de juegos, esta búsqueda se puede realizar hasta una profundidad limitada por el tiempo de juego, pero la solución no sería la mejor, lo ideal sería que la búsqueda se realice hasta el final del árbol de juego, o hasta que se encuentre una solución factible. El nivel inferior alcanzado es el horizonte de búsqueda y contiene los nodos terminales en el caso de que el tiempo fuera lo suficientemente grande o se haya realizado una búsqueda completa, en este caso se aplica una función heurística, llamada función de utilidad o también de resultado que califica a cada estado final con un valor indicando lo bueno o malo que es un estado del juego para el jugador respecto a su adversario.

La segunda fase del algoritmo consiste en propagar recursivamente en dirección a la raíz los valores obtenidos de la función de utilidad para los nodos o estados del horizonte. En los niveles que tenga que jugar Min se tomará el mínimo de los valores de los nodos descendientes, mientras que en los

nodos que tenga que jugar Max se tomará el mayor de los valores de los hijos. De este modo se consigue un árbol con puntuaciones en cada uno de los nodos. En el siguiente árbol se muestra el ejemplo de árbol con los valores obtenidos según el algoritmo Min-Max:

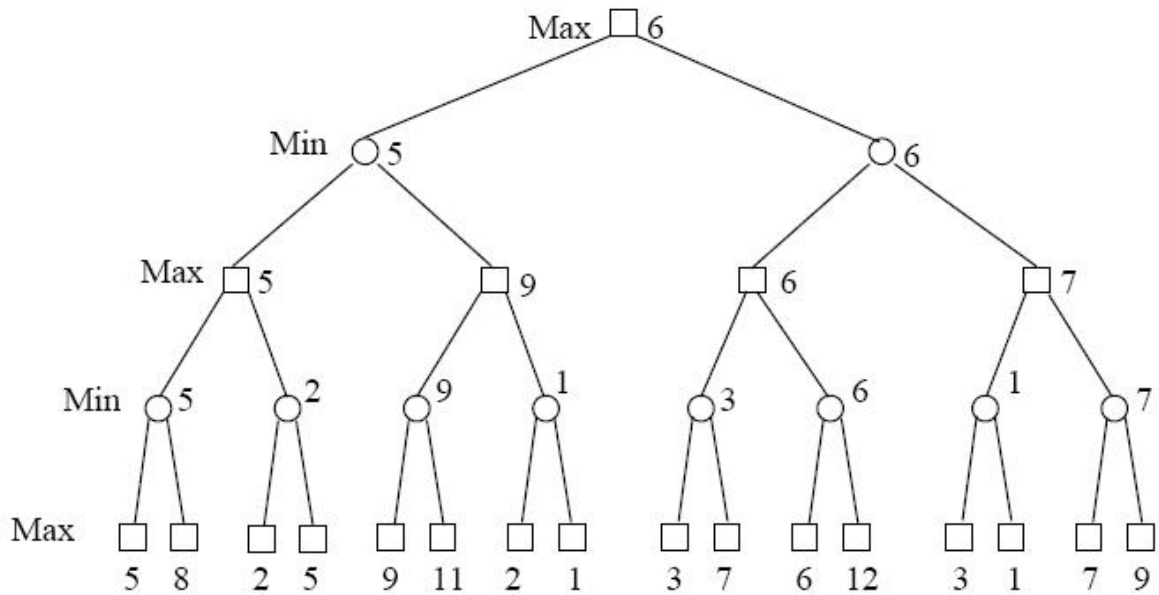


Figura # 2. Árbol generado según el procedimiento Min-Max.

Para un mejor entendimiento del árbol generado por el algoritmo Mn-Max, en la figura anterior, diremos que el primer estado o inicio del juego donde se encuentra Max será el nivel 1, seguidamente Min tiene nivel 2 y así sucesivamente hasta el último nivel.

Pasos:

- 1 El árbol se recorre de izquierda a derecha, para comenzar la búsqueda, se empieza por el nivel más bajo, en este caso es el nivel 4, turno para el jugador Min, en este caso el algoritmo escoge el menor valor para Min, escogiendo el 5.
- 2 Subiendo en el árbol de juego el siguiente nivel es el 3, turno para Max, este escogerá la mejor jugada, en este caso sería el mayor número, hasta el momento el valor que se tiene 5, pero este valor no es definitivo para este nivel, ya que el árbol tiene más hijos y no han sido analizados, ahora se visita el hijo derecho y se recuerda que se encuentra en el árbol del nivel 3.

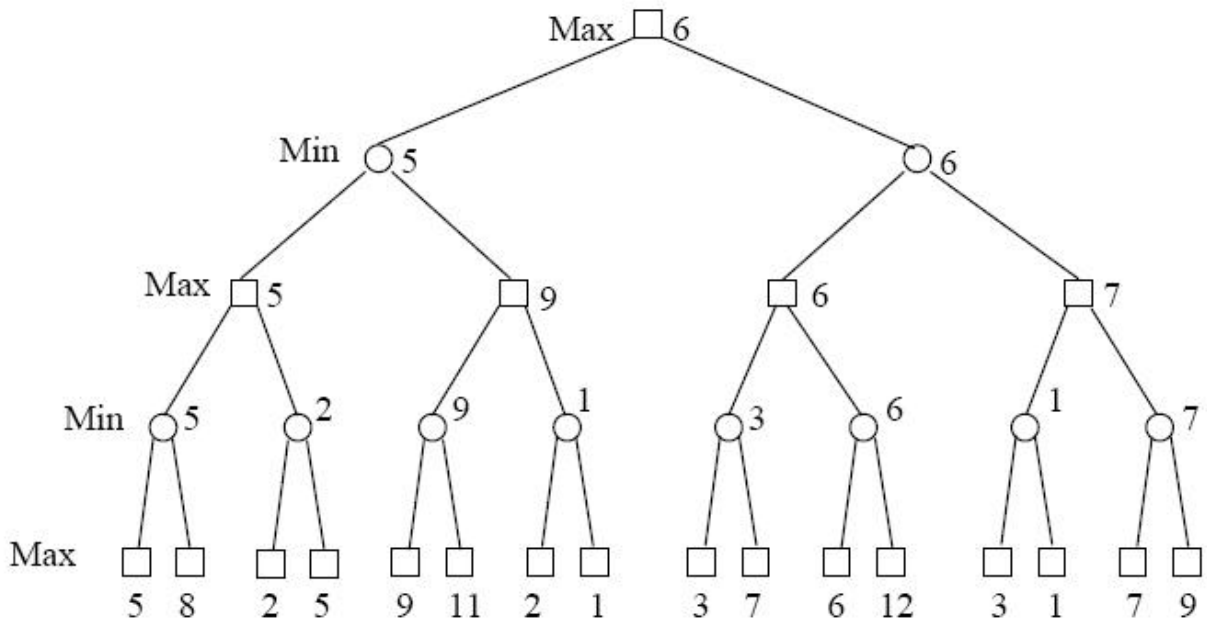
- 3 Para visitar el árbol hijo de 3, se baja el valor que tiene el nivel 3, que todavía tiene valor 5, ahora se visita el hijo que está en el nivel 4 y el turno es para Min, por lo que se tiene que escoger el menor valor, como los hijos de el nodo del nivel 4 tienen valores (2 y 5) se coge el menor que es 2 y es el valor que toma el nodo del nivel 4 para esta jugada.
- 4 Para volver al nivel 3 se tiene en cuenta el valor del nodo hijo, en el que se encontraba anteriormente, que tenía valor 2; ahora se está en condiciones de darle el verdadero valor al nodo del nivel 3, por el primer camino que se escogió en el paso(1) se tenía un valor de 5 y por el camino del paso(3) se tiene un valor de 2, como en este nivel es turno para Max, se escogerá el mayor número, entonces se tomará el valor 5, demostrando de esta manera que la mejor jugada es la que genera el árbol, por el hijo izquierdo y para el cual se obtuvo el valor de 5.
- 5 Escalando por el árbol, ahora se llega al nivel 2, se recuerda que en el nivel anterior, se tenía como mejor jugada a el 5, por lo tanto ahora en este nivel se garantiza un valor al menos de 5, como se explicó anteriormente este valor no es definitorio para este nodo, hay que visitar cada uno de sus hijos, para que se esté seguro que no hay una jugada mejor que 5, ahora se debe de bajar por todo el hijo derecho del nodo del nivel 2 hasta encontrar las hojas, al llegar al final se debe de encontrar con dos hojas con valores diferentes, una con un costo de 9 y otro con un costo de 11, como en el nivel 4 es turno para el jugador Min, se escoge el nodo de menor costo, en este caso se tomaría el 9, subiendo hacia el otro nivel se puede garantizar al menos un valor de 9, pero al tener este nodo un hijo derecho hay que visitarlo, se hace el procedimiento anterior hasta llegar a las hojas, que tienen como valor 2 y 1, como es turno para Min en el siguiente nivel se toma el de menor costo y se debe subir, en el próximo nivel turno para Max se tiene como costo 9, en este caso por ser precisamente turno para Max, se toma el mayor valor que es 9, ahora ya se debe de estar en condiciones de darle el valor al nodo que se encuentra en el nivel 2, este nodo anteriormente ya tenía un valor de 5, al ser turno para Min, se debe de tomar el menor de los dos entre el 9, que era la mejor jugada si se tomaba la opción por el hijo derecho y el 5 que se tenía de la jugada obtenida por el hijo izquierdo, al ser 5 menor que 9 el nodo del nivel 2 toma el valor de 5, indicando que la mejor jugada hasta el momento es la del hijo izquierdo, y como último paso, se debe subir el 5 para la raíz.
- 6 El recorrido del árbol no termina aquí, se debe de recordar que falta recorrer todo el hijo derecho de la raíz, los pasos antes mencionados se van haciendo a cada uno de los nodos restantes y a sus respectivos hijos hasta recorrer el árbol completo, si se observa bien la figura anterior se puede ver como se hace el recorrido al árbol completo, se puede percibir que al

final del recorrido del árbol el valor que toma Max en el nivel 1 es 6, esto quiere decir que se tiene como mejor jugada las que lleva al nodo de valor 6 y termina la búsqueda.

Una vez que se tiene el árbol con los valores de utilidad propagados desde los nodos del horizonte hasta la raíz. Se juega de la siguiente forma:

- Turno de Max: tomará la rama que le conduzca hacia el hijo que mayor valor tenga.
- Turno de Min: tomará la rama que le conduzca hacia el hijo de menor valor.

Aplicando las reglas de recorrido del árbol, el camino será el señalado por las líneas más gruesas en la siguiente ilustración, suponiendo que Min y Max toman decisiones perfectas al conocer el árbol completo. El resultado al que se ha llegado es el de mayor utilidad teniendo en cuenta que ambos juegan para ganar:



**Figura # 3. Camino seguido por el juego para las mejores tiradas de Max y Min.
Suponiendo un horizonte de juego que llegue hasta el final.**

En el ejemplo de la figura # 3 se ha supuesto que el juego es tan sencillo que es posible representar todas y cada una de las jugadas por medio del árbol, de modo que los nodos hojas se corresponde con estados finales del juego en los que se aplica la función de evaluación.

En un caso más complejo se dibuja el árbol hasta el horizonte de juego, que no corresponde con los estados finales del juego, de modo que se aplica la función de evaluación, no de utilidad, a estados intermedios del juego. En este caso, Max juega con los mejores datos que tiene hasta el horizonte de búsqueda. En la siguiente jugada Min toma la decisión respecto de los datos que tenga y de su algoritmo de movimiento (se considera a Min como un ente externo para este ejemplo). En la siguiente jugada Max tiene que volver a realizar el árbol del juego pero esta vez con una profundidad mayor, lo que permitirá ampliar el horizonte de juego y volver a aplicar la función de evaluación a los nodos finales, que serán más realistas que los del árbol obtenido en primer lugar puesto que están más cerca del final del juego que éstos. En la siguiente figura se puede ver cómo serían los pasos de generación del árbol para el árbol con un horizonte de 1 nivel, en el ejemplo que se está utilizando.



Figura # 4. Max genera el árbol con horizonte de un nivel y toma la decisión adecuada según la función de utilidad a los nodos que es capaz de ver.

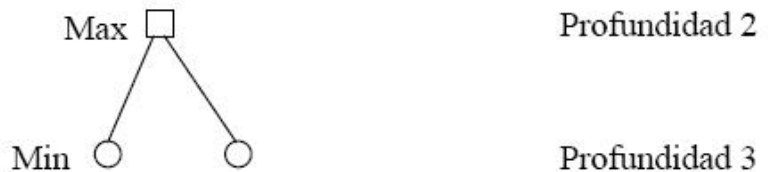


Figura # 5. MAX vuelve a generar el árbol desde la posición del juego que le ha situado MIN con su decisión. Aplica de nuevo la función de utilidad para los nodos que es capaz de ver en el horizonte y decide la mejor tirada. El juego lo finaliza MIN con su tirada, tal y como se puede seguir en el árbol de la figura 1.

En síntesis, los algoritmos de tipo Min-Max se componen de los siguientes pasos[3]:

- Generación de todo el árbol del juego desde el inicio hasta alcanzar los estados terminales o de horizonte de búsqueda limitado por el tiempo de juego que dispone el jugador en el turno.
- Aplicación de la función de evaluación a cada uno de los estados terminales para obtener un valor que califique a cada estado. En el caso de ser estados terminales el valor de la función de evaluación coincidirá con la función de utilidad (perfecta).
- Utilizar el valor de utilidad obtenido en el caso anterior para calcular la utilidad de los estados del nivel inmediatamente superior al de los terminales.
- Continuar calculando los valores de los nodos de las capas superiores.
- Llegar con los valores hasta la raíz, donde iniciará el jugador denotado como Max, que elegirá la opción que le lleve al valor más alto. En el turno de Min, éste tomará el valor más bajo.

1.4.3. Poda Alfa-Beta.

Existe un problema en el Algoritmo Min-Max, el cual tiene que ver con la complejidad del mismo, ya que el número de estados a examinar o evaluar es exponencial en el número de movimientos y el árbol de juego va creciendo de forma considerada. Sin embargo aunque este exponente no se pueda eliminar, existe una posibilidad de reducirlo en gran medida, esto se logra mediante la realización de una poda al árbol Min-Max.[3]

El método de Poda Alfa-Beta permite aplicar el algoritmo Min-Max de forma correcta sin tener que soportar el costo de recorrer todo el árbol, evitando recorrer ciertas ramas que serán inútiles para el resultado final. Aplicando la poda Alfa-Beta, del árbol Min-Max se obtienen las mismas jugadas que sin podar, pero recorriendo solo los nodos necesarios. Para ilustrar la técnica de poda Alfa-Beta se mostrará un ejemplo de un fragmente de árbol (la raíz está un nivel más arriba):

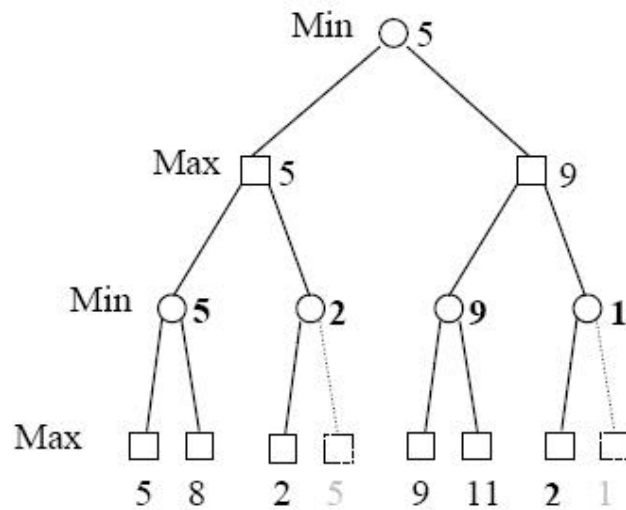


Figura # 6. Árbol con la poda Alfa-Beta. Los nodos hojas 4 y 8, numerando desde la izquierda, se puede podar.

Para un mejor entendimiento del algoritmo Min-Max con la poda Alfa-Beta, explicaremos la figura anterior, partiremos diciendo que los niveles del árbol los numeraremos de arriba hacia abajo comenzando por el 1, teniendo de esta forma el árbol 4 niveles.

1. En la figura, se calcula la utilidad de los nodos 1 y 2, siempre numerar de izquierda a derecha, refiriéndose a los nodos terminales. Los valores que deben de tomar estos nodos son 5 y 8, al pasar al siguiente nivel se debe de tomar el menor valor que es 5 para el nodo Min, para una mejor certeza de que no se va a encontrar un valor menor que 5, se busca en el otro hijo el valor, debido a que el otro valor es mayor, se concluye con que el nodo 1 con valor 5 es el mejor movimiento.
2. Hay que tener en cuenta que para el Max se tomará el máximo entre 5 y el nodo que tiene a su derecha en el mismo nivel, hasta el momento se sabe que en este nivel se tiene asegurado al menos un valor máximo de 5. Para calcular el valor de este nodo, se hace el mínimo de sus hijos. En la primera asignación se ve que el tercer nodo terminal tiene valor 2, ahora bien como en el nivel anterior de modo que en el peor de los casos (en el que el 4º

terminal tenga un valor inferior) el nodo Min sobre el que se hace la comparación con el Min que vale 5, valdrá 2, que es inferior a 5. Por lo tanto es inútil calcular el valor del terminal 4, puesto que para que sea tenido en cuenta habrá de valer menos que 2 y por lo tanto será menor que 5. De esta misma forma se le va aplicando el algoritmo a todos los nodos del árbol hasta recorrer todo el árbol y encontrar el mejor camino que lleve a la victoria.

La poda Alfa-Beta cumple el siguiente principio:

Si se tienen dos nodos de un árbol, n y m , ambos posibles destinos a los que puede ir un jugador, en el momento que m sea mejor opción que n nunca se alcanzará n , por lo que no es necesario considerarla. Para realizar esta comprobación se necesitará conocer en cierto momento con certeza que el nodo n es menos atractiva frente al nodo m para un jugador, aún cuando no se hayan calculado completamente todos sus descendientes, momento en el cual se realiza la poda y se deshecha el cálculo.

La aplicación de la poda se realiza durante la generación del árbol. En esta estrategia la evaluación de los nodos hojas, la propagación de los valores hacia la raíz y la generación del árbol se realizan de forma simultánea, de modo que cuando sea posible se realiza la poda.

En el algoritmo se tienen dos valores[3]:

- Alfa: cota inferior, que en caso de los nodos Max nunca puede disminuir. El valor alfa de un nodo Max es el mayor de los valores de sus sucesores.
- Beta: cota superior, que en los nodos Min nunca puede aumentar, puesto que tratan de minimizar el valor de los nodos. El valor beta de un nodo Min es el menor de sus sucesores.

Las siguientes reglas establecen cuándo se puede terminar el proceso de búsqueda en un nodo:

- Se puede suspender la exploración por debajo de un nodo Min cuando su valor Beta (cota superior) sea igual o inferior al valor Alfa (cota inferior) de cualquiera de sus nodos antecesores Max, de modo que el valor de propagación de este nodo será Beta.

- La exploración se puede suspender debajo de cualquier nodo Max con un valor Alfa (cota inferior) mayor o igual al valor beta (cota superior) de cualquiera de sus antecesores.

1.5. Diferentes tipos de juegos.

Los juegos se pueden clasificar de diferentes maneras una de ellas es de acuerdo a los métodos que son aplicados para resolverlos, por lo que se tienen las siguientes categorizaciones.

1.5.1. Juegos simétricos y asimétricos.

Un juego simétrico es un juego donde las recompensas por jugar una estrategia particular dependen solamente de las otras estrategias empleadas, no de quién las juegue. Si se pueden intercambiar las identidades de los jugadores sin cambiar las recompensas, el juego es simétrico. La simetría puede aparecer en diferentes formas. Los juegos ordinalmente simétricos son juegos simétricos respecto a la estructura ordinal de las recompensas. Un juego es cuantitativamente simétrico si y sólo si es simétrico respecto a las recompensas exactas. Los juegos asimétricos más estudiados son los juegos donde no hay conjuntos de estrategias idénticas para ambos jugadores. No obstante, puede haber juegos asimétricos con estrategias idénticas para cada jugador.[4]

1.5.2. Juegos suma cero.

Se dice que un juego es un juego de suma cero si la suma de las recompensas es cero, es decir, si uno de los bandos pierde exactamente tanto como lo que el otro gana. En los juegos de suma cero las metas que persiguen los jugadores son totalmente opuestas.

Si los intereses de los jugadores son completamente opuestos (juegos estrictamente competitivos) son clasificados como juegos de suma cero donde el beneficio total para todos los jugadores del juego, en cada combinación de estrategias, siempre suma cero (es decir, un jugador se beneficia solamente a expensas de otros, uno gana en lo que el otro pierde). El ajedrez, el monopolio y el Póker son ejemplos de juegos de suma cero, porque se gana exactamente la cantidad que pierde el oponente.

Si por el contrario los intereses de los jugadores coinciden al menos parcialmente es decir no son estrictamente competitivos se tratara de juegos de suma no cero. La mayoría de ejemplos reales en negocios y política, son juegos de suma no cero, es decir, la ganancia de un jugador no necesariamente se corresponde con la pérdida de otro. Por ejemplo, un contrato de negocios involucra idealmente un desenlace de suma positiva, donde cada oponente termina en una posición mejor que la

que tendría si no se hubiera dado la negociación. Ni siquiera en la guerra se pretende la aniquilación completa del enemigo, mas bien las negociaciones de paz sirven para la repartición de beneficios.[4]

1.5.3. Juegos cooperativos.

Un juego cooperativo es un juego de suma no cero en el cual dos o más jugadores no compiten, sino más bien se esfuerzan por conseguir el mismo objetivo y por lo tanto ganan o pierden como un grupo. Un ejemplo es la operación de un negocio exitoso, debido a que todos los participantes se benefician si el negocio tiene éxito y perjudican si falla.

Dentro de los juegos cooperativos se diferencian aquellos donde los jugadores pueden comunicarse entre si para acordar sus movimientos y firmar acuerdos vinculantes y reparto de beneficios, por ejemplo, dos conductores tratando de evitar un choque frontal.

En un juego de suma cero o estrictamente competitivo la comunicación y los pactos no son relevantes porque no habría ninguna razón para confiar en los oponentes y es de esperar que mientan si le es beneficioso.[4]

1.5.4. Juegos simultáneos y secuenciales.

Los juegos simultáneos son juegos en los que los jugadores se mueven simultáneamente o en el que los jugadores no conocen los movimientos de los anteriores. Los juegos secuenciales (o dinámicos) son juegos en los que los jugadores posteriores tienen algún conocimiento de las acciones anteriores. Este conocimiento no hace falta que sea perfecto; puede ser algo de información. Por ejemplo, un jugador puede conocer que un jugador no realizó una acción determinada, pero no sabe cuál de las otras acciones disponibles eligió.

La representación de la forma normal se usa para representar juegos simultáneos, y la extensa para representar juegos secuenciales.[4]

1.5.5. Juegos con información perfecta.

Un subconjunto importante de los juegos secuenciales es el conjunto de los juegos de información perfecta. Un juego es de información perfecta si todos los jugadores conocen los movimientos que han efectuado previamente todos los otros jugadores; así que sólo los juegos secuenciales pueden ser

juegos de información perfecta, pues en los juegos simultáneos no todos los jugadores (a menudo ninguno) conocen las acciones del resto.

La mayoría de juegos estudiados en la Teoría de Juegos son de información imperfecta. (Papel, tijera y piedra, el monopolio- así se sepa los activos y pasivos de los jugadores y en que casillas han caído, no se puede prever cual de las cartas de casualidad y arca comunal le tocan, esta es la razón por la cual no se consideran de información perfecta).

En los juegos de información perfecta cada jugador conoce su estrategia y las recompensas de los otros jugadores (información relevante al juego) así como las acciones tomadas por aquellos, pero nunca la estrategia de su contrincante.

El Ajedrez y las Damas ejemplifican juegos de información completa, donde se conoce todos los esquemas de acciones e incentivos (reglas y desenlaces). [4]

1.6. Aplicaciones del Algoritmo Min-Max.

El Min-Max es un algoritmo creado para el desarrollo de la IA en juegos de contrincantes, con información perfecta y de suma cero, debido al gran desarrollo del mercado de los juegos, en la actualidad hay un sinnúmero de juegos de tablero, para los cuales se han creado jugadores de computadora para competir, ya sea contra otros jugadores de computadora o contra humanos, que tienen implementado este algoritmo, de ellos se pueden citar algunos como:

1.6.1. El juego del Gato o Tres en Raya.

Los algoritmos Min-Max y Min-Max con recorte poda Alfa-Beta son los algoritmos usados para al juego del Gato, o Tres en Raya. El juego usa un tablero de 9 casillas, dividido en tres renglones con 3 casillas cada uno. Inicialmente el tablero se encuentra vacío, Max será quien inicie el juego y seguidamente los jugadores alternan su turno. En este caso lo que tiene que hacer Max, es determinar la secuencia de jugadas que conduzca a un estado Terminal ganador o favorecedor, por su parte lo Min buscará una estrategia que desvíe a Max, en su camino hacia la victoria. Cada uno de ellos tiene un símbolo (O ó X). Para ganar es necesario llenar una línea (vertical, horizontal o diagonal) con tres símbolos iguales. Si todas las casillas están ocupadas sin que haya una raya de 3 fichas del mismo tipo, hay empate.

En el camino hacia la victoria los dos jugadores muchas veces tienen que evaluar ramas innecesarias que no le traerán beneficio alguno para obtener el mejor resultado. Por lo tanto, es aquí donde se plantea la solución mediante la poda de esas ramas que no beneficiaran en nada a la mejor jugada: La técnica de Poda Alpha-Beta.

Este algoritmo es el más utilizado en las aplicaciones referidas a juegos de contrincantes, por su excepcional utilidad en el aumento de la velocidad de la búsqueda, al no producir pérdida de la información y no tener que buscar en ramas innecesarias. Este algoritmo es una extensión en particular del algoritmo de Búsqueda Min-Max en juegos de dos contrincantes.[5, 6]

Los valores que adopta Alfa son un límite inferior para los valores de Max y los valores de Beta son un límite superior para la elección de Min.

- i) Los valores Alfa de los nodos Max son crecientes.
- ii) Los valores Beta de los nodos Min son decrecientes.

1.6.2. Ajedrez.

En el juego de ajedrez, es un juego adecuado para tratarlo mediante técnicas de IA, los dos jugadores saben la posición exacta donde están las piezas, por lo que alternan movimientos, y están libres de hacer cualquier movimiento legal. El objetivo del juego es dar jaque mate al otro jugador, por su parte el otro jugador su objetivo es evitar el jaque mate, o alcanzar una tabla si se esta en una posición de desventaja dada las circunstancias del juego. Un programa del ajedrez selecciona los movimientos mediante el uso de una función de búsqueda. Una función de búsqueda es una función que se pasa información sobre el juego, e intenta encontrar el mejor movimiento para el lado que el programa está jugando. Una clase obvia de función de búsqueda a utilizar es una función con un árbol de búsqueda. En un juego de ajedrez se puede considerar como árbol n-ario grande generalmente (los medios n-ario de cada nodo en el árbol, son un cierto número arbitrario de las ramas que conducen a otros nodos). El árbol del ajedrez es muy espeso (generalmente cerca de 35 ramas de cada posición), y muy profundo. Sería posible representar cada juego concebible del ajedrez en un árbol n-ary gigante, entonces realizar la búsqueda de los mejores movimientos para cada jugador. El algoritmo Min-Max para el desarrollo del juego sería obvio a utilizar. Es posible utilizar la búsqueda Min-Max para solucionar juegos simples como el tic-tac-toe. El árbol de juego en tic-tac-toe no es muy espeso y no es muy profundo, así que el árbol entero puede ser atravesado, el juego puede ser entendido totalmente, y un movimiento se puede hacer en cualquier posición que esté garantizada para ser el

mejor movimiento. Es posible hacer esto con ajedrez en un sentido matemático, pero no es posible hacerlo con la tecnología actual, y aun en un futuro lejano. Sin embargo, sigue siendo posible utilizar la búsqueda Min-Max como la base de un programa de ajedrez.

El juego se desarrolla sobre un tablero de 8 filas y 8 columnas, en el cual se colocan fichas de 2 colores que simbolizan a los 2 oponentes, estos colores son generalmente: Blanco y Negro. Este juego además posee un sin numero de fichas, de diferente tipo y movimiento[7]

A continuación hacemos refería a las fichas presentes en el juego de Ajedrez:

- ♔ Rey.
- ♚ Reina.
- ♖ Torre.
- ♘ Alfil.
- ♞ Caballo.
- ♟ Peón.

Al iniciar el juego cada jugador tiene una cantidad equivalente a 16 fichas. Sin embargo, la configuración de cada movimiento que se pueda generar y las posibles acciones que estas puedan sugerir, hacen de este juego algo interesante, motivo por el cual se ha tomado énfasis en estudiar este juego empezando el uso de búsquedas sistemáticas para diseñar programas para ordenadores.

En el desarrollo de este juego aparecen una gran cantidad de complicaciones a la hora diseñar una búsqueda adecuada, se podría decir que una de ellas o la principal, es la complejidad espacial en un ordenador, motivo por el cual muchas veces se utiliza una determinada profundidad hasta donde se puede llegar a evaluar.

La mayoría de los desarrolladores de este juego para calcular la heurística le asignan un peso a cada una de las piezas, en dependencia del color, a continuación ilustramos un ejemplo de cómo podría ser.

• **Piezas Blancas:**

- Rey: 50
- Reina: 10
- Torre: 5

- Alfil: 3
- Caballo: 3
- Peón: 1

• **Piezas Negras:**

- Rey: -50
- Reina: -10
- Torre:- 5
- Alfil: -3
- Caballo: -3
- Peón: -1

1.6.3. Damas.

Desde los principios de la computación el ser humano ha pretendido imitar su propio pensamiento, imaginando que algunas de sus funciones podrían ser imitadas por las maquinas.

En muchos casos se ha llegado más lejos, surgiendo incluso teorías psicológicas que toman como modelo el computador para encontrar analogías con la mente humana, comparando procesos e intentando sistematizar éstos.

El primer juego que se intento computarizar y para el que se desarrollo un programa fue precisamente el juego de Damas[8]. Posteriormente otros juegos fueron tratados en este aspecto, llegando al momento actual en que es el ajedrez el más estudiado y al que mayores esfuerzos se dedican en pos de la consecución del jugador perfecto.

Los historiadores afirmaron hasta 1986 que el juego de las damas era un invento francés. Otros decían incluso que era un invento de los egipcios. Y así hasta la fecha nadie sabía su origen. A pesar de que la mayoría de los importantes libros fueron escritos en España desde el año 1547 hasta 1901, España no fue tomada en cuenta.

Después de una investigación de diez años por parte de Govert Westerveld (Monnickendam - Holanda, 1947), la cosa ha cambiado en beneficio de los españoles, y es que ahora reconocen que efectivamente España fue la madre creadora del juego de las Damas, gracias a los árabes. Existen

básicamente dos formas de jugar. Una es el juego español con 12 fichas, y otro es el juego con 20 fichas.

Marion Tinsley, fue el histórico campeón de damas que inició su reinado en 1954 y que sólo perdió nueve partidas en los siguientes 40 años. Sin embargo, en 1994 perdió su cetro ante un programa de ordenador, y ese mismo programa se ha convertido ahora en imbatible.

El programa Chinook comenzó a ser implementado en la Universidad de Alberta en Canadá en 1989, y estaba centrado en el juego de las damas americanas que se juegan sobre un tablero de 8x8 casillas. Chinook, realiza el mejor movimiento posible en cualquier combinación de las fichas, de modo que si su oponente juega a la perfección, lo más que puede conseguir es acabar en tablas. Chinook no puede, nunca jamás, perder una partida; dicho de otra manera: las damas han sido resueltas como juego[9].

Los informáticos llevan trabajando en el programa desde 1989, aunque entonces tenía un objetivo más modesto, conseguir ganar el campeonato del mundo de damas. En primer lugar hubo que conseguir permiso para que un programa de ordenador pudiera participar, lo cual se logra en 1990. La primera participación de Chinook en el campeonato fue en 1992, pero no logra ganarlo. Sin embargo, en 1994 gana el campeonato (la primera vez que un programa de ordenador gana un campeonato humano en cualquier juego de inteligencia).

Sin embargo, Jonathan Schaeffer y su equipo no estaban contentos con eso, querían que Chinook alcanzase la perfección absoluta en el juego. Con la ayuda de jugadores expertos, los informáticos programaron una serie de reglas heurísticas con las que el programa decidía que movimiento realizar en cada posición. Durante años han utilizado múltiples ordenadores (50 de media, 200 en los picos de trabajo) para calcular todas las posibilidades, estimar los movimientos más eficaces, perfeccionar las reglas y aumentar su número.

Sus desarrolladores han logrado que este programa no tenga que analizar cada una de las $5+10^{20}$ posibles combinaciones en cada movimiento, y de hecho para llegar a la combinación ganadora sólo se acaban analizando una fracción de 1/5.000.000 del total, esto se logra utilizando la poda Alfa-Beta, una mejora al método Min-Max, algoritmo que fue utilizado para desarrollar el programa Chinook.

El resultado final es que matemáticamente un juego perfecto de damas acaba siempre en tablas. Un ordenador que empieza a jugar con las negras, con todos los conocimientos del análisis, es capaz de no perder nunca, y podría incluso ganar si el adversario comete un error. Pero si ambos contendientes juegan siempre «la partida perfecta» en base al análisis completo y perfecto, las tablas están garantizadas.[9]

1.7. Realidad Virtual.

1.7.1. Definición de Realidad Virtual.

El desarrollo de la informática en los últimos tiempos a traído consigo, el surgimiento y desarrollo de un nuevo termino “RV”, iniciada en los programas de entrenamiento militares, simuladores de vuelo, centros de investigación y académicos, ha pasado a los programas más variados en el ámbito profesional y doméstico. Hoy se empieza a aplicar en ingeniería, medicina, arquitectura, educación, juegos, entre otros campos.

Hoy en día dar una definición concreta de RV es difícil, existen muchas definiciones quizás tantas como investigadores relacionados con esta área, esto se debe en gran medida a su reciente y rápida evolución. No resultaría extraño que el concepto de RV suela ser relativo para diferentes investigadores del tema en diferentes situaciones. Lejos de dar una definición veremos un conjunto de definiciones, planteadas anteriormente donde en cada una se valora una idea de lo que se debería entender por RV:[11]

RV: es una simulación tridimensional interactiva por computador en la que el usuario se siente introducido en un ambiente artificial, y que lo percibe como real basado en estímulos a los órganos sensoriales.

RV: es un paso más allá de lo que sería la simulación por computadores, tratándose más bien de una simulación interactiva, dinámica y en tiempo real de un sistema.

RV: es un entorno de tres dimensiones sintetizado por computadora, en el que participantes acoplados de forma adecuada pueden manipular elementos físicos simulados en el entorno y, de alguna manera, relacionarse con las representaciones de otras personas pasadas, presentes o ficticias, o con criaturas inventadas.

RV es simulación por computadora, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y variables complejas, durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a mundos que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial. Una nueva y sorprendente forma de navegar información.

La RV puede ser de dos tipos: inmersiva y no inmersiva. Los métodos inmersivos de RV con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano. La RV no inmersiva utiliza medios como el que actualmente nos ofrece Internet en el cual podemos interactuar en tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora.

La RV no inmersiva ofrece un nuevo mundo a través de una ventana de escritorio. Este enfoque no inmersivo tiene varias ventajas sobre el enfoque inmersivo como: bajo costo y fácil y rápida aceptación de los usuarios. Los dispositivos inmersivos son de alto costo y generalmente el usuario prefiere manipular el ambiente virtual por medio de dispositivos familiares como son el teclado y el ratón que por medio de cascos pesados o guantes.

En la actualidad la RV se usa ampliamente en la Industria, permite minimizar los costes de la realización de prototipos, simulación, entrenamiento, marketing y otros usos. También la tecnología actual permite obtener un realismo sin precedentes.[11]

1.8. Relación entre IA y RV.

RV es un término que se refiere a la simulación mediante un ordenador de entornos del mundo real o imaginarios que utilizan gráficos 3D y dispositivos externos como guantes o cascos para permitir a los usuarios interactuar con la simulación, donde el usuario lo percibe como real basado en estímulos a los órganos sensoriales.

Los usuarios se pueden mover a través de un entorno de RV e interactuar con los objetos del entorno o con otros usuarios. Un Agente Virtual Inteligente, por otro lado, es un programa que adopta una

representación virtual, el cual por lo general se asocia con la forma humana dentro del Entorno Virtual y, mediante técnicas de IA, es capaz de responder de manera inteligente a los sucesos del entorno, perseguir metas, o interactuar con los usuarios o con otros agentes.

CAPÍTULO 2: PROPUESTA DE JUEGOS.

2.1. Introducción.

En este capítulo se tratará de demostrar que es posible aplicar la técnica del Algoritmo Min -Max en el desarrollo de aplicaciones de realidad virtual, para lograrlo se partirá del ejemplo de juegos de contrincantes entre dos jugadores. Estos juegos por tener la característica de suma cero y de información perfecta son un clasico ejemplo para poner en practica el alto nivel de inteligencia que este algoritmo presenta. Se explicará el algoritmo Min-Max y Min-Max con recorte Alfa-Beta durante todo el desarrollo del capítulo.

2.2. Juegos.

A pesar de que son muy diferentes unos de otros los juegos de Ajedrez, Conejo y Perros y Tic -Tac-Toe, por solo citar algunos, también tienen una gran similitud, esto se debe a que todos ellos son juegos para dos jugadores, dicho con otras palabras, son juegos de contrincantes. Estos juegos se clasifican según su forma de resolverlos en juegos de suma cero y de información perfecta. Al presentar semejantes propiedades es posible crear un árbol de juego para facilitar la selección del siguiente movimiento una vez comenzado el juego.

2.2.1. Conejo y Perros.

Especificaciones del Juego:

Desarrollar el juego de Perros y Conejo usando el algoritmo Min-Max.

Instrucciones:

El juego de **El Conejo y los Perros** consta de un tablero con 11 posiciones distintas, interconectadas con determinado número de arcos, en las cuales se ubican 3 perros y 1 conejo, como se muestra en la figura # 7. En cada turno, un solo perro se mueve a una posición vacía directamente conectada, y a continuación el conejo hace lo mismo. Los perros pueden moverse verticalmente o en línea recta, nunca hacia atrás. El conejo puede moverse en cualquier dirección. Los perros ganan atrapando al conejo de tal manera que le sea imposible moverse a cualquier posición.

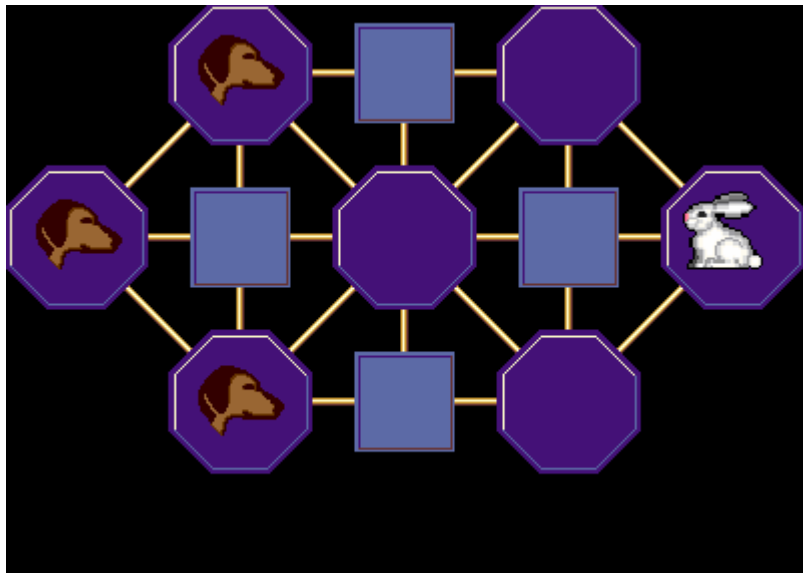


Figura # 7. Tablero para el juego de los Perros y el Conejo.

El conejo puede ganar de dos maneras, cuando deja atrás a los perros, es decir dejándolos a las derechas, o cuando los perros realicen 17 jugadas sin atrapar el conejo, por ejemplo los perros solo se muevan arriba y abajo entonces los perros están atrapados y el conejo gana. [12]

Para jugar:

- ✎ Si decide jugar con los perros seleccione una opción, que le diga la posibilidad de que quiere jugar con los perros y deberá programar el conejo.
- ✎ Si decide jugar con el conejo (en contra de los perros) debe coger una opción que le permita jugar con el conejo y competir contra un jugador preprogramado usando el algoritmo Min-Max y con una Poda Alfa Beta. Recordar que en este caso la máquina jugara primero.
- ✎ No cambia el tipo de jugador mientras dura el juego, ya que la computadora seguirá usándola opción inicial.

El aprendizaje por medio del algoritmo de Min-Max con poda Alfa-Beta se implementará una búsqueda exhaustiva, recorriendo el árbol de juego completo. Para el desarrollo de este juego se deben tomar diversas condiciones que se enuncian a continuación. [12]

- ✂ Minimizar los movimientos posibles del conejo.
- ✂ Revisar que el conejo no rebasará al último perro.
- ✂ Revisar que el conejo no ocupará el centro del tablero.
- ✂ El conejo NO debe llegar a los niveles 1 y 2 (Los niveles representan la posición en el tablero, es decir, existen 5 niveles y se numeran de izquierda a derecha).
- ✂ Los perros deben permanecer de preferencia cada uno en su profundidad correspondiente (Las profundidades representan el renglón en el tablero, existen 3 profundidades y se numeran de arriba a abajo).
- ✂ El conejo debe dirigirse de preferencia sólo hacia un nivel igual o mayor al nivel en que se encuentre.
- ✂ Los perros no deben hacer una jugada perdedora.
- ✂ Los perros deben realizar una jugada ganadora.

Para el desarrollo de este juego, al igual que el Juego Tres en Raya por la condición de ser un juego sencillo, que solo tiene 11 posiciones no se cree necesario realizar una función heurística para evaluar el tablero, sino recorrer el árbol completo de juego para los dos jugadores (Perros y Conejo). En este caso se usará una función de utilidad que tomará para el caso de Max valor de 1, señalando que cuando se llegue a ese valor no se añadirán más nodos, siendo esta una jugada ganadora, por su parte Min tomará valor -1, indicando, que sería la mejor jugada para el contrincante. La mayor dificultad que presenta esta técnica, sin el uso de una función heurística, que minimice el número de movimientos es generar el árbol de juego.

Para desarrollar el árbol de juego, se crea un nodo inicial que será la raíz y de ahí en adelante se crean nodos que indican los movimientos, la raíz estará en el primer nivel, ya para el segundo nivel estarán tantos nodos como posibles movimientos válidos se hayan generados del primer movimiento, y en el tercero estarán las posiciones válidas que se pueden generar a partir de las posiciones del segundo nivel

y así consecutivamente, de esta forma no se tendría un árbol con un nivel fijo, sino un árbol solo con los movimientos que se van generando.

Una vez generado el árbol de juego, se debe tener una función de evaluación, que evalúa cada una de las posiciones, para saber de esta forma cual es la más favorable, esto ocurre mediante una búsqueda por todo el árbol de juego mediante el algoritmo Min-Max con recorte Alfa-Beta, la función antes mencionada tendrá un valor de 1 para el caso del jugador Max y -1 para Min. En un nivel de juego escogemos el mínimo y luego el máximo de los mínimos, podando aquellas posiciones que no sean necesarias visitar. [12]

2.2.2. Tres en Raya.

Especificaciones del Juego:

Desarrollar el juego del Gato o Tres en Raya usando el algoritmo Min-Max.

Instrucciones.

En el presente juego se propone la realización de un agente inteligente, para el juego Tres en raya, este es un juego sencillo, las reglas del juego y la descripción del problema se explican a continuación. La idea consiste en un juego de contrincantes de información perfecta, en el que participan dos jugadores uno llamado Max y otro llamado Min. Para comenzar el juego, cada jugador coloca alternativamente una ficha en una matriz de 3x3 o un tablero de 3x3, para este caso el jugador que inicia la partida será conocido como Max, y jugará con la ficha(x) que representará sus movimientos, por su parte el otro jugador recibirá el nombre de Min y jugará con la ficha (0) .El primero que logre completar una fila, columna o diagonal con fichas de un mismo tipo ganará la partida.

Reglas:

Inicialmente el tablero se encontrará en cero, es decir no habrá ninguna ficha sobre él, Max será quien hará el primer movimiento y los jugadores se van alternando al poner sus fichas.

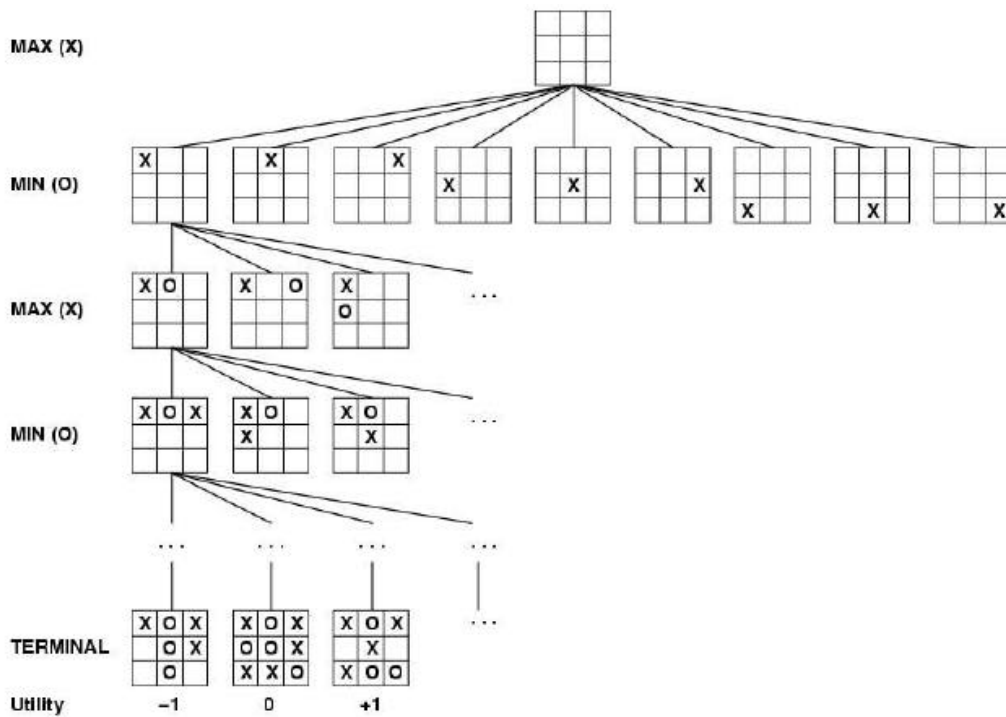


Figura # 8. Árbol de juego para el Tres en Raya.

En la figura anterior se muestra la forma en que se va expandiendo el árbol de juego. Max gana obteniendo una raya con tres (X).



Figura # 9. Tres cruces en forma diagonal gana Max.

Por su parte Min gana obteniendo una raya con tres (O).



Figura # 10. Tres ceros en una misma columna gana Min.

Si todas las casillas están ocupadas y no se han colocados tres fichas del mismo tipo en una fila, columna o diagonal el juego quedara en empate.

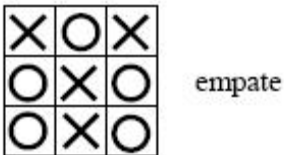


Figura # 11. Empate.

La heurística se calcula para saber que tan favorable es el tablero con respecto a la computadora. Entre más favorable sea la computadora tendrás más probabilidad de ganar. En este juego se toma la decisión de que en el juego Tres en Raya es demasiado sencillo como para usar una heurística en todos los casos, en la siguiente propuesta se plantea resolver el juego mediante una búsqueda exhaustiva en el árbol del juego, es decir, un árbol que contenga todos los movimientos posibles de ambos jugadores.

La mayor dificultad que traería esta búsqueda es desarrollar el árbol de juego, con esta variante se desarrollará un juego perfecto que conduce siempre a un empate o una victoria para la computadora, claro que podrían considerarse las probabilidades de los errores más comunes en las personas al jugar Tres en Raya y el algoritmo maximizaría la probabilidad de éxito eligiendo una jugada que conduzca a un posible error por parte de la persona. Para desarrollar el árbol de juego, consiste en tener como raíz una posición, el segundo nivel del árbol consiste en todas las posiciones válidas que se pueden generar de la raíz, y el tercero son las posiciones válidas que se pueden generar a partir de las posiciones del segundo nivel, y así consecutivamente, después hay que tener una forma de evaluar las posiciones para poder distinguir cual es mejor que otra, o cual es la menos favorable, esta se llama una función de evaluación.

Teniendo evaluadas las posiciones hay que deliberar cual es la mejor, esto lo haremos con una búsqueda en el árbol. Primero generamos las posiciones en el árbol de juego, después evaluamos a todos los nietos, a los cuales les pondremos una calificación numérica, y ahora cada hijo tomará el valor del hijo que tenga la mínima calificación. Y por último se escogerá como jugada al hijo con la mayor puntuación. En un nivel escogemos el mínimo y luego el máximo de los mínimos, por eso se llama Min-Max.

Niveles de Dificultad:

Para el desarrollo del juego, se plantearon tres niveles de juegos, para jugar fácil el árbol crece solamente hasta 3 niveles, para jugar en medio hasta 5 niveles y para jugar en difícil se debe expandir a 8 niveles, de esta forma es imposible vencer a la computadora. En nuestro contexto, un nivel representa una jugada (marcar con una O ó X). Estos niveles de expansión dependen de la situación del juego, por ejemplo si el juego está terminando es perfectamente posible que no se pueda expandir un estado a 8 niveles de profundidad. De esta forma quedaría conformado el árbol de búsqueda para el juego.

2.2.3. Damas.

Especificaciones del Juego:

Desarrollar el juego de Las Damas usando el algoritmo Min-Max.

Instrucciones.

Para el juego de Las Damas se utilizará el tablero clásico del Ajedrez, preferiblemente blanco y negro. En este juego se usaran términos como peones para indicar fichas con movimientos diagonales simples, dígame un solo salto, a no ser para comer más de una ficha y damas que pueden moverse de forma diagonal por todo el tablero.

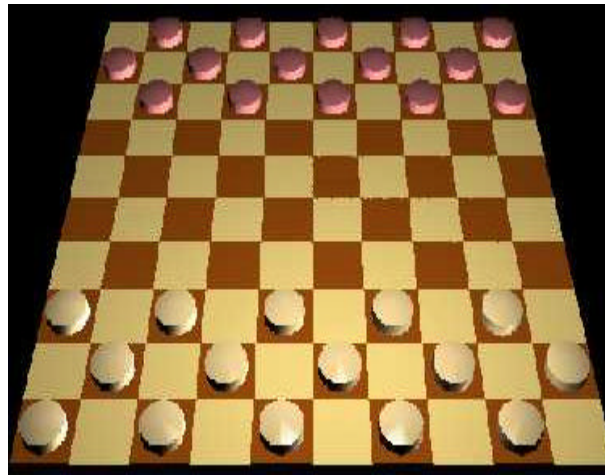


Figura # 12. Tablero para el juego de las Damas.

Algunos juegos como las Damas o el ajedrez, por tener información completa al inicio del juego, son idóneos para la búsqueda de caminos en un sistema complejo, pudiendo después aplicar lo descubierto en otros campos de la computación o de la ciencia en general. Posteriormente al intento de computarizar el juego de damas, otros juegos fueron tratados en este aspecto, llegando al momento actual en que es el ajedrez el más estudiado y al que mayores esfuerzos se dedican en pos de la consecución del jugador perfecto, como sucedió con los programas Deep Blue y actualmente, 3DFritz. [8]

Las técnicas y algoritmos que se usan para todos estos juegos son básicamente los mismos, cambiando solo alguna particularidad del juego tratado, como la generación de jugadas.

Para el desarrollo de este programa se debe utilizar un algoritmo Min -Max, fijando un nivel máximo de profundidad en el árbol de 6 niveles, que corresponde a tres movimientos de la máquina más los movimientos de respuesta del adversario.

En principios se pensó que se podrían resolver por búsqueda exhaustiva en el árbol de l juego, es decir, un árbol que contenga todos los movimientos posibles de ambos jugadores. Pero estos tendría inconvenientes, considerando por ejemplo el juego de las Damas, en el comienzo de la partida un jugador puede realizar 9 movimientos, por lo que en el árbol de juego habría 81 nodos en el segundo nivel. Por el contrario, en el juego de ajedrez, en el comienzo de la partida un jugador puede realizar 20 movimientos,

por lo que en el árbol de juego habría 400 nodos en el segundo nivel, así la complejidad aumentaría exponencialmente en cada nivel de profundidad del árbol, usando el algoritmo Min -Max. Resulta evidente que una simple búsqueda directa no es posible de realizar en la práctica, y por lo tanto es necesario algún tipo de procedimiento de búsqueda heurística, esto traería como resultado una mejor elección del camino a seguir mientras más capas se exploren antes de tomar la decisión. Partiendo de este principio se utiliza una función de evaluación estática (heurística) para elegir el movimiento más prometedor.

En cada nodo del árbol se almacena la información correspondiente al estado de la partida si se llevase a cabo dicho movimiento. Por ello, recorriendo el árbol se busca el movimiento que maximice la diferencia entre el número de piezas con las que juega la máquina y el número de piezas con las que juega el adversario (el usuario). Debemos tener en cuenta para determinar el número de piezas que pueden existir dos tipos de fichas para cada jugador: peones (piezas de salida) y damas. Debido a las mayores posibilidades de movimiento y acción de la Dama, se le ha dado un peso a cada ficha: 1 punto por peón y 3 por dama. Así, se maximiza, para todos los movimientos posibles para la máquina y solamente para el mejor o los mejores para el adversario, la función.[8]

$$3*\text{damasMáquina} + \text{peonesMáquina} - 3*\text{damasUsuario} - \text{peonesUsuario}$$

Es decir, se considerarán todos los movimientos que puede realizar la máquina, pero consideraremos que el usuario hará el mejor de sus movimientos, para ello, recorreremos el árbol de movimientos solamente para el mejor (o los mejores si el camino no es único) de los movimientos para cada caso del usuario.

Si el resultado de la búsqueda del mejor movimiento no es único, entonces se ha recurrido a la aleatoriedad para determinar el movimiento a realizar. A cada uno de estos movimientos posibles se le asigna un peso dependiendo del número de nodos del último nivel de profundidad (que dependan de él) que permitan llegar a un máximo de la función explicada anteriormente. Por tanto, cuantos más caminos llevan a dicho resultado, mayor será la probabilidad de seguir ese camino. Esta aleatoriedad permite que la máquina no sea tan determinista ante situaciones idénticas desde el punto de vista del análisis del árbol de movimientos.

El gestor de jugadas lleva una particularidad en este juego en relación con el ajedrez, ya que ciertos movimientos son obligatorios (en el juego de las Damas, es obligatorio el comer piezas si se puede), y dentro de varios movimientos obligatorios habrá que elegir uno de ellos, motivo por el que se aplicó el dar pesos a cada tipo de ficha, como se indicó anteriormente.

Con el fin de probar la inteligencia del juego de las Damas, se han realizado pruebas con varios niveles de profundidad del árbol de movimientos, demostrando que mientras mayor sea el número de niveles para el juego, mejor serán los resultados obtenidos. En esta propuesta se han fijado solo 6 niveles de profundidad, nos basamos en que con 8 niveles no siempre puede crearse el árbol de movimientos por falta de memoria en las máquinas. Por ello, se ha optado por utilizar 6 niveles en el programa. Incluso con 6 niveles de profundidad, dependiendo de si el número de damas de ambos jugadores es alto, no siempre ha sido posible crear el árbol. Para el caso de una máquina de muy poca memoria, que sea imposible crear el árbol de movimientos, se ha optado por intentar crear el árbol con menor profundidad (4 niveles, 2 por cada jugador). De este modo, aunque se barajen menos posibilidades, podrá seguir siendo un buen movimiento por observar 4 movimientos hacia el futuro. Cabe destacar que el tiempo de respuesta es prácticamente nulo siempre que el número de damas en el tablero no sea significativo.

La construcción del árbol ha sido una de las tareas más complejas que se han desarrollado en esta práctica. Hoy en día muchos programas analizados que simulan un jugador de Damas (al igual que los programas de Ajedrez), utilizan bases de datos para mejorar los últimos movimientos de las partidas, cuando ya quedan pocas piezas sobre el tablero. De esta forma, pueden almacenarse finales de partidas ya jugadas y tomar un camino fijo hacia la victoria. Para este juego no se ha seguido esta técnica ya que se ha optado por un único algoritmo a utilizar durante toda la partida, que se ha intentado lo suficientemente genérico para cualquier situación que pueda darse en una partida. [8]

2.3. Algoritmo Min-Max para el Juego de Ajedrez.

El algoritmo Min-Max no siempre es factible usarlo para cualquier tipo de juego de contrincante. En el siguiente trabajo no se valora la posibilidad de proponer el Juego de Ajedrez, a continuación se dará una explicación de porque no se tiene en cuenta.

Al iniciar el juego de ajedrez las piezas blancas pueden elegir uno entre 20 movimientos posibles y seguidamente al movimiento de las blancas, las negras tienen 20 opciones de respuesta cualquiera haya sido la elección de las blancas. Como se puede apreciar tan solo de la primera jugada en el juego de Ajedrez pueden surgir 400 posiciones diferentes, después de dos jugadas el número de jugadas podría aumentar en 20 000, llegando a una cifra extremadamente grande luego de varias jugadas, por lo que el árbol que genera el juego también es extremadamente grande. Lo ideal para saber la mejor jugada de apertura sería examinar el árbol por completo, buscando todas las líneas de juego y como conclusiones determinar cuál sería el mejor movimiento inicial. Sin embargo, en la práctica el árbol es demasiado extenso para considerar este mecanismo, incluso el determinar el mejor movimiento buscando en el árbol de variantes a partir de una posición avanzada en el juego resultaría imposible. En el juego lo mejor que se podría realizar es buscar en un sector limitado del árbol de variantes, esperando obtener suficiente información con tal de decidir correctamente cual es el mejor movimiento.

En este caso se usaría el algoritmo Min-Max con poda Alfa-Beta reduciendo aproximadamente el número de nodos que debería buscar, en la raíz cuadrada, lo que hace más eficiente el algoritmo Min-Max.

El problema del algoritmo Min-Max con poda Alfa-Beta es muy similar al Min-Max, por su parte la poda Alfa-Beta tiene funciones adicionales que lo distingue del Min-Max. Esto resulta muy fácil conceptualmente, pero para el juego de ajedrez no resulta eficiente. Una capa adicional de profundidad se busca cada vez, trayendo como resultado que el tamaño del árbol buscado crece exponencial. Para el caso del ajedrez tiende cerca de 35 movimientos legales en cualquier posición dada. El Min-Max se utiliza para buscar a una capa de profundidad, se examinan cerca de 35 posiciones. Si la función se utiliza para buscar dos capas, debe examinar 35^2 posiciones y así sucesivamente. Esto al parecer no es malo hasta el momento, pero una vez que hemos avanzado en el juego el número crece muy rápidamente. Por ejemplo, en el caso de una búsqueda de 6 capas son casi dos mil millones de posiciones, y en el caso de 10 capas la búsqueda está sobre el quadrillion dos. Para lograr un resultado eficiente es importante buscar tan profundamente como sea posible, si la meta es crear a un jugador de ajedrez fuerte examinando las capas primeras del árbol del juego y aplicando una evaluación heurística en los nodos de

la hoja. Min-Max no permite una búsqueda muy profunda, porque el factor de ramificación eficaz es extremadamente alto.

CAPÍTULO 3: CASO DE ESTUDIO

3.1. Introducción

En el presente capítulo se demuestra la funcionalidad del algoritmo Min -Max, haciendo uso de un Demo para el Juego del Gato o Tres en Raya, demostrando de esta manera la eficiencia del algoritmo en cuestión.

3.2. Implementación del caso de estudio.

3.2.1. Demo:

Para probar el funcionamiento del algoritmo Min -Max en los juegos de contrincantes se tomó como ejemplo el Juego del Gato o Tres en Raya. Para desarrollar el agente inteligente, dígame la computadora se tuvieron en cuenta un conjunto de clases que se muestran en la figura # 13.

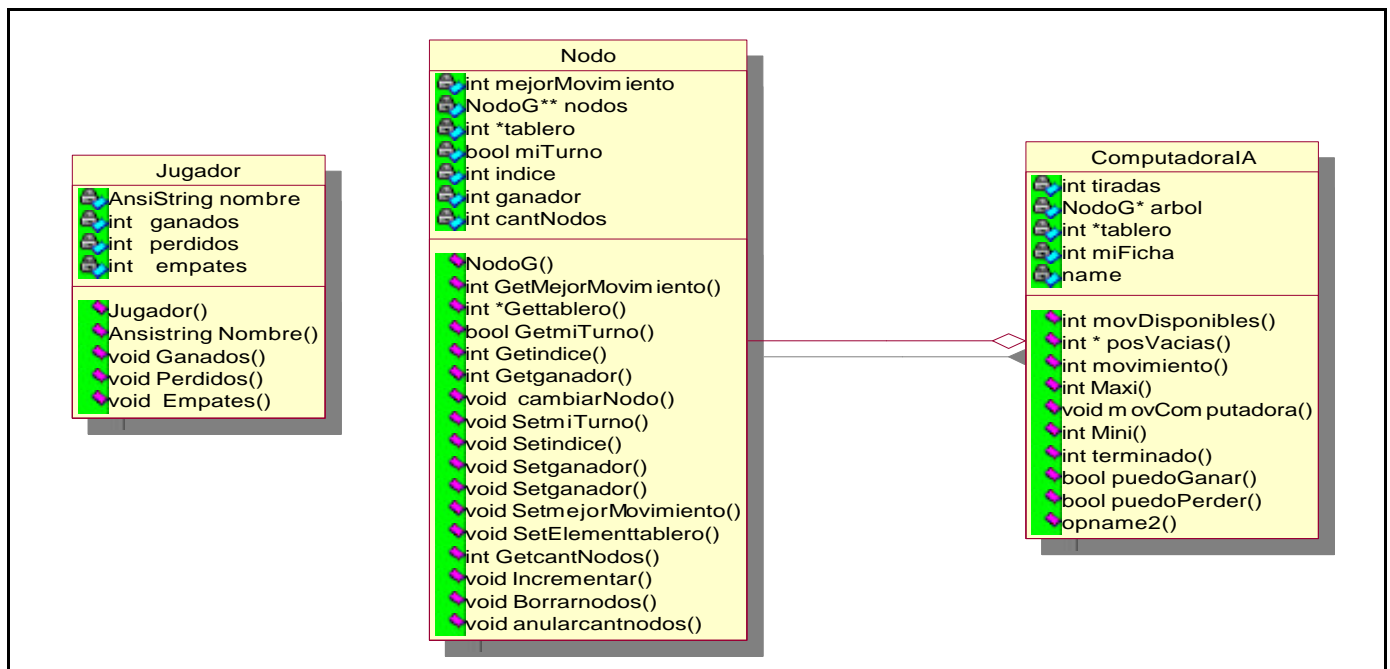


Figura # 13. Diagrama de clases para el juego del Gato o Tres en Raya.

Luego de ver el diagrama de clases mostrado, se seguirá explicando el funcionamiento del juego. Este consiste en tener como raíz el nodo que se encuentra en el primer nivel, el segundo nivel del árbol consiste en todas las posiciones válidas que se pueden generar de la raíz, y el tercero son las posiciones válidas que se pueden generar a partir de las posiciones del segundo nivel, y así consecutivamente. El demo cuenta con tres niveles de dificultad, fácil solo se explora el árbol hasta tres niveles, medio se explora hasta 5 niveles y donde se exploran todos los niveles de juegos, es decir se llegará hasta el nivel 8, se dice que se llegará hasta el nivel 8 porque este es el número máximo de nivel para que el juego sea perfecto e invencible.

Después hay que tener una forma de evaluar las posiciones para poder distinguir cual es mejor que otra, o cual es la menos factible. Esto se llama una función de evaluación. En el ejemplo la función que se tomó fue de 1 para el caso de Max y de esta forma se indica que es una jugada vencedora, por su parte para indicar una jugadora perdedora para Min esta función tendrá valor de -1, y en caso de empate tendrá valor de 0.

Teniendo evaluadas las posiciones hay que deliberar cual es la mejor, esto se logra realizando una búsqueda exhaustiva en el árbol, para esto caso se usa un algoritmo sumamente común en IA, el Min-Max. El cual es sumamente sencillo.

Primero se generan las posiciones en el árbol de juego, después se evalúa a todos los nietos, a los cuales se les pondrá una calificación numérica, y ahora cada hijo tomará el valor del hijo que tenga la mínima calificación o máxima en dependencia del nivel en que se encuentre, y por último en el nivel más alto en el árbol de juego se escogerá como jugada al hijo con la mayor puntuación.

3.2.2. Análisis del tiempo que demora el algoritmo.

El Tres en Raya al ser un juego para dos jugadores resultaría imposible implementar el algoritmo Min-Max para tres jugadores en un tablero 3D con 27 posiciones. Es evidente que a un jugador no le parecería bien esperar el movimiento del rival más que unos pocos segundos como máximo. Por lo que no sería viable

la implementación de árboles de búsqueda si no se limita estrictamente la profundidad umbral tras la que no se deberían de expandir nuevos nodos porque se dispararía el tiempo de cálculo y ni siquiera los ordenadores más potentes podrían resolver este problema.

El problema está en la incapacidad de barajar todas las posibles situaciones del tablero. Un Tres en Raya convencional tiene 9 casillas e implementar un algoritmo de búsqueda para este juego es perfectamente viable con una búsqueda exhaustiva en el árbol de juego. Se logra gracias a su pequeño orden de ramificación y su profundidad máxima del árbol.

Al tener 9 casillas, explorar todas las combinaciones tendría orden $O(n!)$ con $n=9$ fijo por lo que $9!=362880$ nodos o posibilidades. Esta cifra no supone desafío alguno para un ordenador.

Pero ese mismo orden de complejidad aplicándose en 27 casillas posibles supone, tratándose de una operación factorial, la asombrosa cifra de:

$$27! = 10888869450418352160768000000.$$

Esta cifra de 10 TRILLONES nos indica el número de nodos a explorar si consideramos todos los posibles movimientos hasta un final de partida en el que se han ocupado todas las casillas. Queda por tanto descartado que no se puede explorar la mayoría de los nodos para encontrar una secuencia de jugadas que lleve al ordenador a realizar todos los cálculos que lo llevaría a la victoria.

Si bien el juego Tres en Raya en 3D para tres jugadores y 27 posiciones es imposible resolver con el algoritmo Min-Max por el rendimiento o tiempo en segundos que se demora, esto no ocurre si se lleva a un entorno virtual para dos jugadores y 9 posiciones, donde el mayor tiempo que demoraría está dado por los cálculos del algoritmo Min-Max y no de el tiempo que demoraría cargar las imágenes del entorno virtual, para tener una idea de lo que demora realizando una búsqueda exhaustiva a continuación se muestra una tabla con los tiempos que demora el algoritmo para el Tres en Raya .

Nivel	Cant Nodos	Tiempo (s)
4	43680	0.00
5	15120	0.016
6	60480	0.047
7	181440	0.78
8	362880	0.109

Tabla # 1.

En la tabla se muestra perfectamente como realizando una búsqueda en todos los nodos que se puedan generar aplicando el algoritmo Min-Max solo demora 0.109 (s) y cargar el tablero demora aproximadamente 0.015 (s) en cualquiera de las variantes de este juego, viéndose aproximadamente a 57 Frame por segundo (FPS) en un ordenador Intel Pentium 4, con una velocidad del CPU de 3.0 GHZ y un 1GB de RAM, para otro ordenador Intel Pentium 4, con una velocidad del CPU de 3.0 GHZ, 1 GB de RAM y con tarjeta de video NVIDEA QUADRO FX 500/FX600 el tiempo que demora en cargar la escena es de 59 (FPS) por lo que estos tiempos no representan problema alguno para ningún ordenador.

Sin embargo con una variante del Tres en Raya, un juego que tiene un tablero 4x4 donde la idea es la misma del Tres en Raya pero esta vez se colocarían cuatro fichas del mismo tipo y no tres no ocurre lo mismo. Este juego para los 5 primeros niveles no representa ningún problema con el tiempo de respuesta del algoritmo pero ya para el nivel 6 no resulta eficiente el árbol de juego por el tiempo. A continuación se muestra una tabla con los tiempos.

Nivel	Cant Nodos	Tiempo(s)
3	3360	0.00
4	43680	0.031
5	524160	0.265
6	5 765 760	5.625

Tabla # 2.

Para los tres primeros niveles el tiempo de respuesta es casi nulo, ya para el nivel 4 cuando el algoritmo genera 43680 nodos demora 0.031 (s), para generar 524160 demora 0.265 (s) hasta este nivel no presenta ningún problema porque cualquier jugador puede esperar perfectamente este tiempo, donde lejos de molestar al jugador lo que le crearía es una sensación de que la computadora esta pensando, pero sin embargo para el nivel 6 el algoritmo genera más de 5 millones de nodos o posibles jugadas lo que resultaría imposible calcular por un ordenar común, es esta la razón por la cual para esta variante solo se exploraran los primeros 5 niveles y se usa una heurística para mejorar la inteligencia del juego o, porque a este nivel la inteligencia de la maquina no es tan bueno como en el Tres en Raya.

3.2.3. Heurística para el juego Cuatro en Rayas y el N en Raya

- ✍ Si se tienen dos fichas de diferentes tipos en una misma fila, columna o diagonal la heurística toma valor 0.
- ✍ Si se tienen fichas del ordenador en una misma fila, columna o diagonal la heurística toma valor positivo igual a la suma de la cantidad de fichas de ese tipo. Cada ficha del ordenador tiene valor 1.
- ✍ Si se tienen fichas del jugador en una misma fila, columna o diagonal la heurística toma valor negativo igual a la suma de la cantidad de fichas de ese tipo. Cada ficha del jugador tiene valor -1.
- ✍ Si se tiene tres fichas del ordenador en una misma fila, columna o diagonal la heurística toma valor infinito positivo, forzando de esta forma al ordenador a jugar en la casilla que falta para completar la cuarta ficha, indicando que no hay una jugada mejor para él que esta.
- ✍ Si se tienen tres fichas del jugador en una misma fila, columna o diagonal la heurística toma valor infinito negativo, forzando de esta forma al ordenador a jugar en la casilla que falta para completar la cuarta ficha, indicando que no hay una jugada peor para él que esta.

Mientras mayor sea el valor de la heurística, más favorable es la jugada para el ordenador, lo contrario ocurre si la heurística toma un valor pequeño.

CONCLUSIONES.

Una vez finalizada la investigación sobre el algoritmo Min-Max se puede afirmar que para lograr un mayor grado de inteligencia en los juegos de contrincantes se debe explorar el árbol lo más profundo posible, si se lograra generar todo el árbol de juego y realizar una búsqueda completa el algoritmo sería perfecto, para el juego Tres en Raya al lograrse perfectamente realizar una búsqueda exhaustiva, resulta imposible ganarle al ordenador. Se debe tener cuidado a la hora de implementar los juegos de contrincantes, porque no siempre es posible generar el árbol completo de juego debido al número extremadamente grande de jugadas que genera el árbol de juego. En este caso lo conveniente es el uso de una heurística, y limitar la búsqueda a un nivel fijo, de esta forma el algoritmo ya no sería perfecto como en el Tres en Raya pues dependería en gran medida de cuán eficiente es la heurística.

RECOMENDACIONES

Se recomienda.

- ✍ Implementar los juegos propuestos durante el trabajo .
- ✍ Seguir proponiendo juegos a los que se les pueda aplicar el algoritmo Min-Max.

BIBLIOGRAFÍA

1. R., I.A.M.R.L.A.M., *Concepciones básicas en economía de la información y la selección adversa en el objeto de estudio de la teoría organizacional*. 2005.
2. SÁNCHEZ, A.B.S., *Inteligencia en redes de comunicaciones*.
3. Ocaña, A.M., *Juegos como problemas de Búsqueda. Búsqueda con adversarios y grafos*
4. Fuentes, O.I., *Jugador Inteligente de Backgammon*. 2006.
5. Corchado, E., *BÚSQUEDA HEURÍSTICA EN SITUACIONES ANTAGÓNICAS*. 2004.
6. Nilsson, N., *INTRODUCCION A LA IA*.
7. MANCILLA, J.E., *DESARROLLO DE LA MAESTRÍA AJEDRECÍSTICA COMPUTACIONAL UTILIZANDO RECURSOS RESTRINGIDOS*.
8. Amo, C.A.F.I.G.d., *Juego de las Damas*.
9. Pastor, J., *Descubierto el algoritmo imbatible para el juego de damas*. 2007.
10. *Introducción de la IA*.
11. Malfanti, I.S., *Introducción*. 2001.
12. Vázquez, R.G.P.J.A.A.R.M.A.S., *Uso de IA mediante el esquema de una búsqueda Minimax con poda Alfa-Beta. El conejo y los perros*.

BIBLIOGRAFÍA CONSULTADA

Lezcano B. M, Lio D, Pérez R, Valdivia Z. *Introducción a la Inteligencia Artificial. ¿Qué es la IA?* Pandora, 2000.

Gómez M, Lezcano B. M, Pérez R, Reynoso A, Valdivia Z, *Aplicaciones de la IA*, Ediciones de la Noche, 2002.

Funge J.D. *Artificial Intelligence for Computer Games*, Sales and Customer Service Office, 2004.

Rabin E. *AI Game Programing Wisdom*, Charles River Media, 2002

Introducción a la teoría de juegos.

Disponible en: <http://www.taringa.net/posts/info/872598/La-Teoria-de-Juegos.html>

Diferentes tipos de juegos.

Disponible en: <http://www.ii.uam.es/~fdiez/docencia/04-05/IA/juegos.ppt> .

Algoritmo Min-Max.

Disponible en: <http://es.wikipedia.org/wiki/Minimax> .

Algoritmo Min-Max, poda Alfa-Beta y Heurística.

Disponible en: <http://ccia.ei.uvigo.es/docencia/IA/Tema2.pdf> .

Inteligencia Artificial .

Disponible en: http://es.wikipedia.org/wiki/Inteligencia_artificial .

Que es Inteligencia Artificial.

Disponible en: <http://www.cruzrojaquayas.org/inteligencia/Que%20es%20IA.htm>

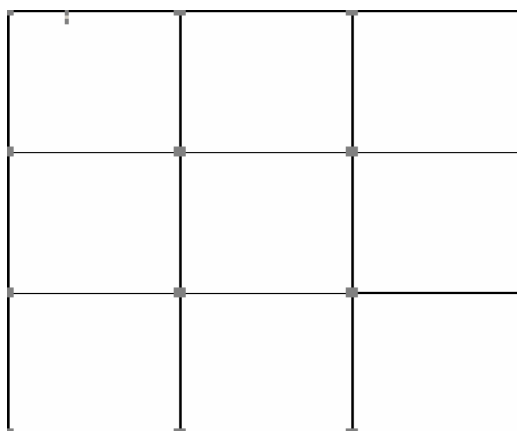
ANEXOS

Anexo I**Manual del caso de estudio.**

Para iniciar el juego se deba escoger primeramente la forma de juego, dígase humano contra la computadora o humano contra humano, después se escogerá el nivel de dificultad. Una vez seleccionada la forma de juego se procede a jugar, para indicar los turnos se toma la primera opción de juego; el jugador humano para indicar su jugada lo hará con las bolas, mientras que la computadora jugará con las cruces, si se toma la otra opción, el jugador uno, jugará con las bolas, y el jugador dos con las cruces.

Una vez que se selecciona la forma de juego, se procederá a realizar el primer movimiento según el tipo de juego seleccionado, si estamos en el primer caso el primero en jugar será el jugador # 1, por su parte si estamos en el otro caso jugará primero el humano. El juego terminará cuando se haya completado una fila, columna o diagonal de una misma ficha, y el ganador será el jugador que lo logre. Si en el desarrollo del juego no hay posibles movimientos y ningún jugador ah podido completar una fila, columna o diagonal con tres de sus fichas entonces el juego quedará en tablas.

Inicialmente el tablero se encuentra vacío como se muestra en la siguiente figura.



Tablero vacío para el inicio del juego.

Anexo II

Imagen que muestra el comienzo del juego (Persona vs Persona).



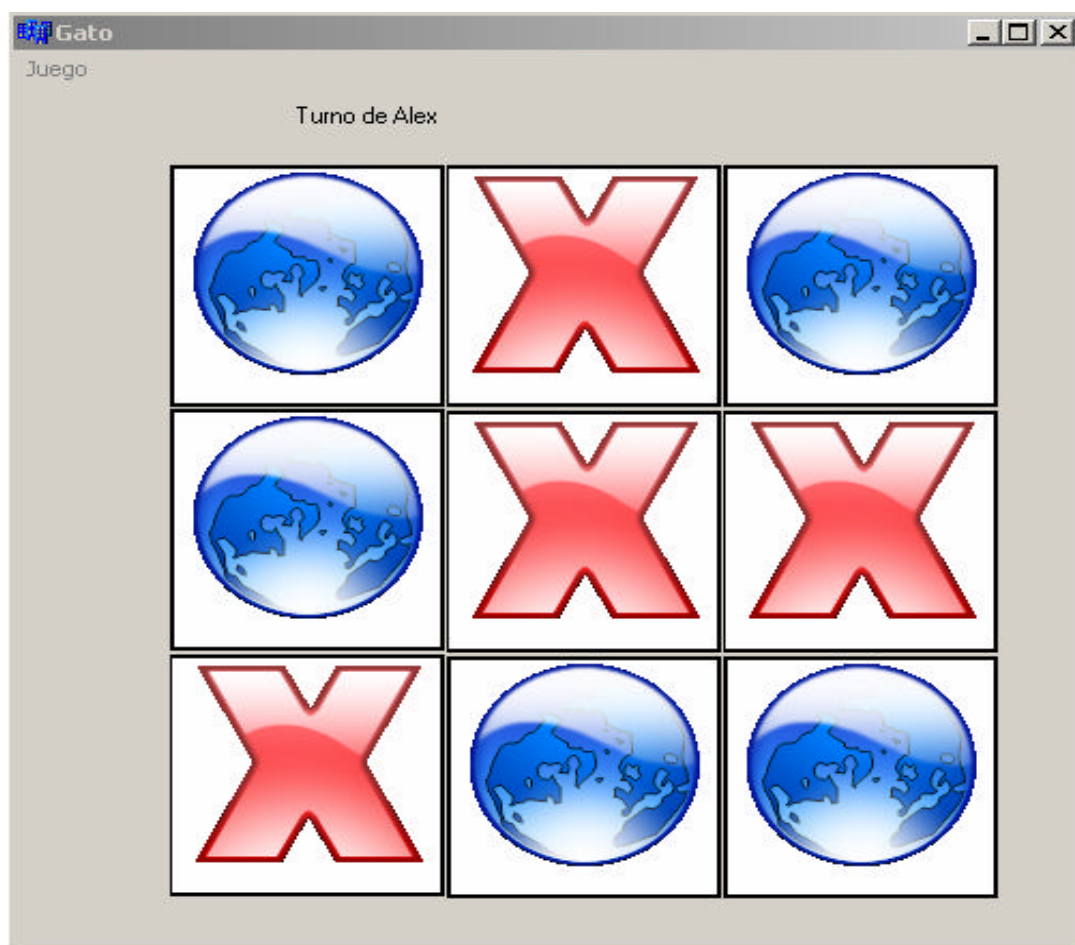
Anexo III

Imagen que muestra el comienzo del juego (Persona vs Computadora).



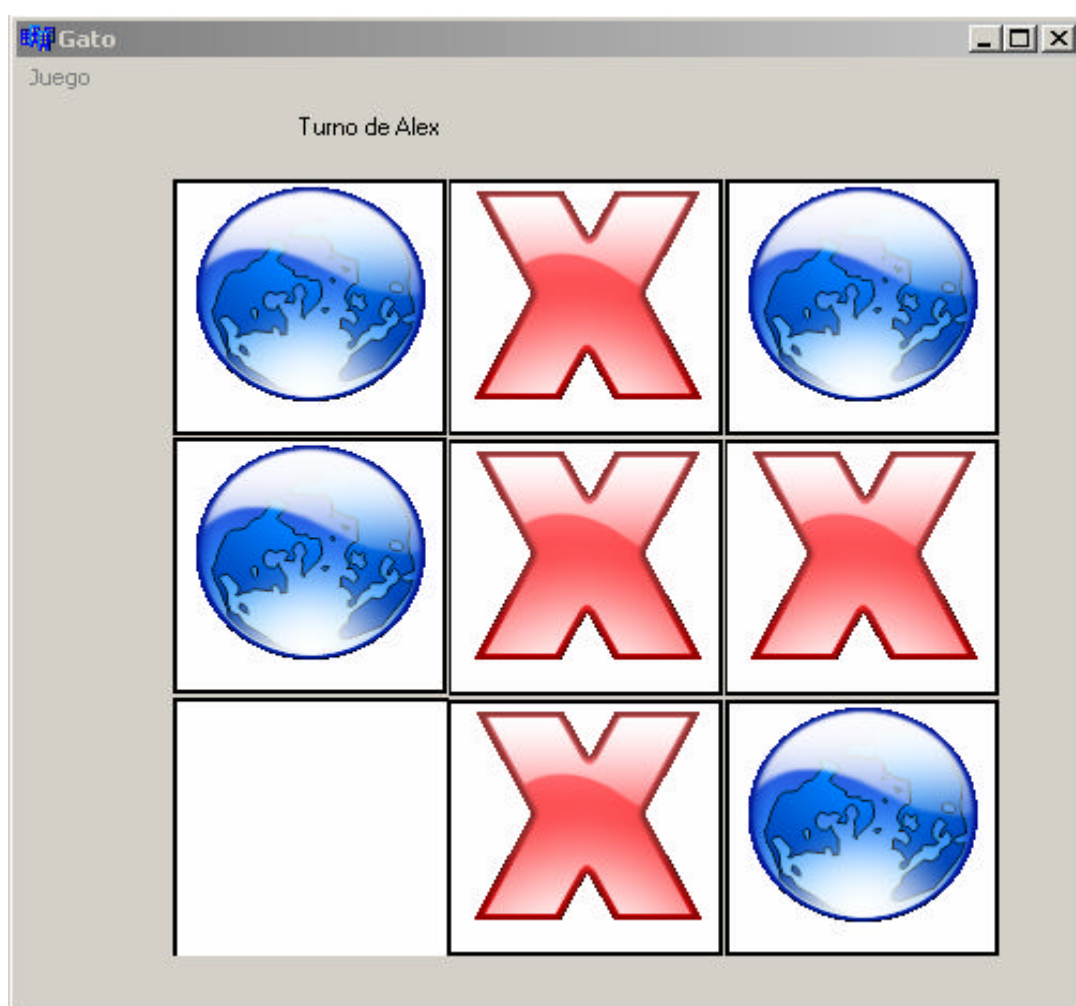
Anexo IV

Imagen que muestra un empate en el Juego.



Anexo V

Imagen que muestra cuando gana la Computadora.



GLOSARIO

A

Aleatorio: Al azar, que no sigue un patrón, secuencia u orden determinado.

Algoritmo: Un conjunto de reglas bien definidas para la solución de un problema en un número finito de pasos.

Alfa-Beta: Variante del algoritmo Min-Max que no explora nodos innecesarios.

B

Búsqueda Exhaustiva: Búsqueda completa en todo el árbol de juego.

H

Heurística: Regla que permite orientar un algoritmo hacia la solución de un problema. Técnica de programación que permite a un sistema la creación gradual de un valor óptimo para una variable específica por medio del registro de los valores obtenidos en operaciones anteriores. Técnica empleada en los sistemas de inteligencia artificial.

I

Inteligencia Artificial (IA): La IA dentro de las ciencias de la computación es la encargada de la creación de máquinas que implementan tareas relacionadas con el comportamiento humano, logrando gran similitud al pensamiento del hombre.

M

Min-Max: Min-Max es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario y con información perfecta.

R

Realidad Virtual (RV): Simulación de un medio ambiente real o imaginario que se puede experimentar visualmente en tres dimensiones. La realidad virtual puede además proporcionar una experiencia interactiva de percepción táctil, sonora y de movimiento.

S

STK: *SeceneToolkit*, herramienta para el manejo y representación de escenas tridimensionales .

