

# **PROYECTO FINAL INTELIGENCIA ARTIFICIAL**

## **AGENTE QUE JUEGA GENIAL**

### **INTEGRANTES:**

**MARIA ALEJANDRA AGUIAR VASQUEZ**

**LUIS FELIPE CASTAÑO LEDESMA**

**PROFESOR: ING. JOSHUA TRIANA**



**UNIVERSIDAD DEL VALLE SEDE TULUA  
FACULTAD DE INGENIERIA  
PROGRAMA ACADÉMICO DE INGENIERIA DE SISTEMAS  
VALLE DEL CAUCA - TULUA  
2017**

## 1 INTRODUCCIÓN AL JUEGO

---

**Genial** es un juego abstracto en el que los jugadores colocarán sobre un tablero unas fichas compuestas por 2 hexágonos con una figura geométrica y un color asociado en cada uno de ellos. Los jugadores puntuarán en cada uno de estos colores/formas al colocar cada loseta sumando un punto por cada forma coincidente con la de la loseta según una línea trazada en cada una de las cinco direcciones posibles respecto a la pieza para cada hexágono. Habrá que progresar equilibradamente, ya que la puntuación final dependerá de los colores menos avanzados.

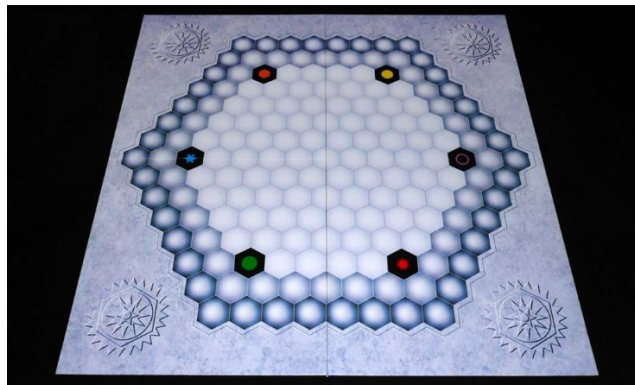
### **Componentes:**

Empezando con las **fichas**, los jugadores gestionarán una serie de fichas formadas por 2 hexágonos unidos por uno de sus lados. Cada uno de estos hexágonos mostrará una figura geométrica de seis posibles: Estrella Azul, Circulo Verde, Hexágono Naranja, Asterisco Rojo, Aro Morado y Rueda Dentada Amarilla. Entre las 120 fichas que componen el juego se encuentran todas las combinaciones posibles.



*Ilustración 1 - Fichas*

El **Tablero principal** mostrará una serie de casillas hexagonales en las que los jugadores irán colocando las fichas anteriores. En los vértices del se aprecian 6 casillas con una de las formas anteriores, que servirán de punto de partida.



*Ilustración 2 - Tablero*

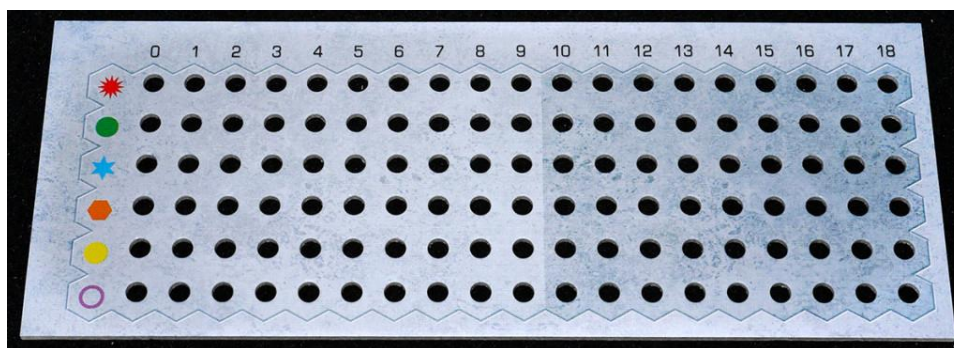
Los jugadores tendrán una mano de 6 fichas, las cuales se sacarán de la bolsa de tela al inicio de la partida.



*Ilustración 3 - Fichas en mano*

Lo más interesante de este juego es la **puntuación**. Al colocar una ficha, desde cada uno de los dos hexágonos se trazará una línea imaginaria en cinco direcciones (cada lado del hexágono a excepción del que sirve de unión con el otro). Por cada forma coincidente y consecutiva en cada una de esas líneas con la del hexágono, el jugador anotará un punto en dicha forma/color.

Para llevar la cuenta de los puntos los jugadores dispondrán de unos **tableros de puntuación** con un marcador para cada uno de los seis colores/formas. El máximo de puntos que se puede acumular para un determinado color será 18.



*Ilustración 4 – Tablero de puntuación*

## 2 REGLAS DEL JUEGO

---

Cada turno consistirá en los siguientes pasos:

**1. Selección del tablero:**

Al ser una partida de dos jugadores, solo se podrá colocar fichas en las casillas centrales (las más claras) delimitadas por las casillas iniciales. Teniéndose entonces 11 filas distribuidas de la siguiente manera:

- Desde la primera fila con seis casillas incrementando en una hasta la sexta fila con once casillas.
- Desde la séptima fila con diez casillas decreciendo en una hasta la onceava fila con seis casillas.

**2. Colocar una ficha en el tablero:**

El jugador debe seleccionar una de las 6 fichas de las que dispone y colocarla dos espacios adyacentes del tablero que se encuentren vacíos, de otra forma no se permite poner la ficha.

- El jugador solo consigue puntos si coloca la ficha de forma que los símbolos coincidan con los símbolos de casillas adyacentes.
- Durante la primera ronda de juego se debe colocar la ficha adyacente a uno de los símbolos iniciales del tablero.
- No se puede colocar una ficha o parte de la ficha fuera de los límites del tablero.
- En cada turno hay que colocar una ficha, aunque con ella no se ganen puntos.

**3. Puntuar:**

El jugador puntúa por separado cada hexágono de la ficha (Sin tener en cuenta la pareja), traza cinco líneas imaginarias desde cada uno de los dos hexágonos de la ficha colocada.

- Sólo se puntúan las fichas adyacentes y con formas iguales en línea recta.
- En cada dirección, cuenta el número de símbolos adyacentes que coinciden con el de la pieza jugada

**4. Fin de la partida:**

La partida finaliza cuando no sea posible colocar una nueva ficha en el tablero (Cuando ya no hay dos casillas vacías adyacentes). Se examina el marcador y el ganador será el jugador cuyo color con menos puntos, tenga un valor más alto.

### 3 RESTRICCIONES DE LA IMPLEMENTACIÓN.

---

Por medio del análisis del juego para la construcción del árbol Minimax, se pudo observar a grandes rasgos que la complejidad de dicho árbol es muy alta. Por ende, se seleccionaron dos de las reglas anteriormente mencionadas con el objetivo de reducir el tamaño del árbol MiniMax.

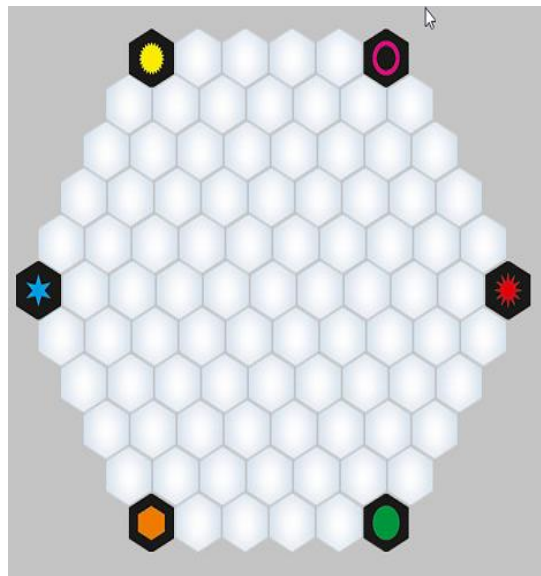
1. Los jugadores no conocen las fichas de su adversario.
2. En el caso de ser dos jugadores, el tablero se constituirá de las casillas que poseen el color de fondo más claro, delimitando el campo de acción de los jugadores.

Considerando el tamaño original del tablero para un enfrentamiento uno contra uno (ver ilustración 5) la cantidad prevista de posibles jugadas para una ficha en el primer movimiento para el árbol Minimax es de 444, ahora haciendo algunos cálculos podemos afirmar lo siguiente:

*Posibles jugadas para una ficha Max = 444*

*Fichas en mano para Max = 6*

*Cantidad total de posibilidades Max =  $444 \times 6 = 2.664$*



*Ilustración 5 - Tablero original*

Es decir, habrá 2664 nodos de profundidad uno en el árbol. si se continúan expandiendo los nodos de profundidad uno, se podrá calcular la cantidad total de nodos del árbol Minimax:

*Posibles jugadas para una ficha de Min = 430*

*Fichas en mano para Min = desconocida = probabilidad de 25 tipos de fichas*

Debido a que el agente no conoce las fichas que posee el jugador y al ser éstas entregadas aleatoriamente podemos afirmar que el árbol de juego se ve afectado por el azar. Por lo tanto es necesario incluir nodos aleatorios, teniendo en cuenta la probabilidad de que el jugador tenga una ficha con determinada distribución de colores (ver tabla 1).

Ingenious Tile Distribution	
Red/Red	5
Blue/Blue	5
Green/Green	5
Yellow/Yellow	5
Orange/Orange	5
Purple/Purple	5
Red/Blue	6
Red/Green	6
Red/Yellow	6
Red/ Orange	6
Red/Purple	6
Blue/Green	6
Blue/Yellow	6
Blue/Orange	6
Blue/Purple	6
Green/Yellow	6
Green/Orange	6
Green/Purple	6
Yellow/Orange	6
Yellow/Purple	6
Orange/Purple	6
<b>TOTAL</b>	<b>120</b>

*Tabla 1 - Distribución de fichas*

La probabilidad para cada distribución (en total 25 distribuciones) se calcula tomando el número de fichas de esa distribución y dividiéndola por el número total de fichas en el juego, lo cual genera un crecimiento considerable en la cantidad de nodos de profundidad dos:

*Cantidad total de posibilidades Min =  $430 \times 25 = 10.750$*

*Cantidad total de nodos en el árbol = 28'638.000*

Al observar que la cantidad de nodos en el árbol era muy elevada se consideraron dos opciones: la primera opción estudiada fue dar a conocer al agente las fichas que posee el jugador y la segunda fue reducir el tablero, esto con el fin de disminuir el tamaño del árbol. Los resultados fueron los siguientes:

#### **Opción 1 - Dar a conocer las fichas del jugador al agente**

*Posibles jugadas para una ficha Max = 444*

*Fichas en mano para Max = 6*

*Cantidad total de posibilidades Max =  $444 \times 6 = 2.664$*

*Posibles jugadas para una ficha de Min = 430*

*Fichas en mano para Min = 6*

*Cantidad total de posibilidades Min =  $430 \times 6 = 2.580$*

*Cantidad total de nodos en el árbol = 6'873.120*

#### **Opción 2 - Reducir el tamaño del tablero (ver ilustración 6)**

*Posibles jugadas para una ficha Max = 276*

*Fichas en mano para Max = 6*

*Cantidad total de posibilidades Max =  $276 \times 6 = 1.656$*

*Posibles jugadas para una ficha de Min = 262*

*Fichas en mano para Min = 6*

*Cantidad total de posibilidades Min =  $262 \times 6 = 1.572$*

*Cantidad total de nodos en el árbol = 2'603.232*



*Ilustración 6 - Tablero reducido*

A continuación, se presentan los resultados obtenidos gracias a las modificaciones en relación al tiempo de ejecución:

Descripción	Cantidad de nodos en el árbol	Tiempo estimado de ejecución
Árbol Minimax sin aplicar cambios al problema	28'638.000	1 minuto 57 segundos
Árbol Minimax donde el agente conoce la mano del contrincante	6'873.120	28 segundos
Árbol Minimax con reducción del tablero	2'603.232	11 segundos

*Tabla 2 - Resultados obtenidos*



## 4 IMPLEMENTACIÓN DEL AGENTE

---

### 4.1 IMPLEMENTACIÓN DEL ÁRBOL MINIMAX:

Para la representación del árbol Minimax en java se implementaron dos clases, una clase llamada `ArbolMinMax.java` que representa la estructura general del árbol y sus funciones principales y otra clase llamada `Nodo.java` que representa un nodo con punteros a otros nodos con los que tiene relación (listas doblemente enlazadas).

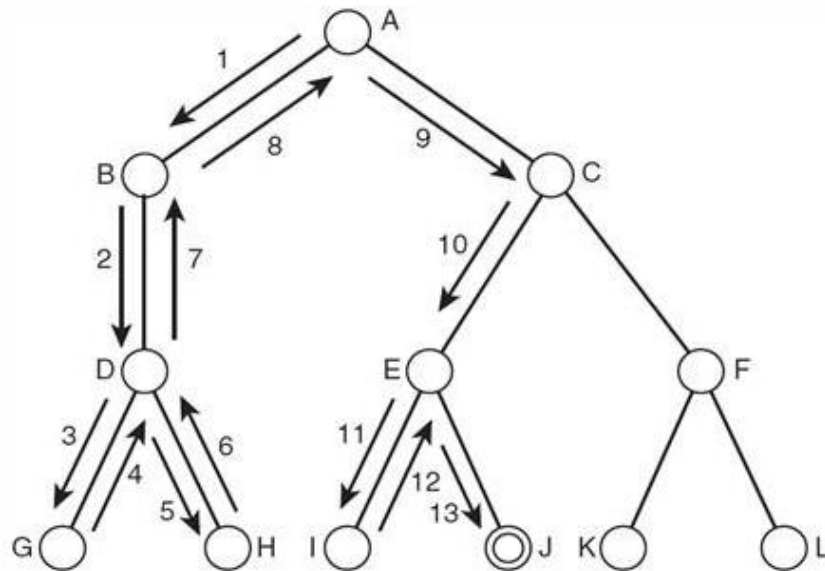
***AbolMinMax.java:*** Para implementar el comportamiento de un árbol Minimax crearon seis funciones, cuatro principales y dos auxiliares. Al final de la ejecución el árbol Minimax arroja como resultado la ficha de la mano que se debería jugar y su posición correspondiente en el tablero. A continuación, se detalla cada función:

**funcionPrincipal:** Esta función es la encargada de ingresar el nodo raíz en la cola de prioridad para posteriormente expandirse. Luego de eso, expande todos los nodos que se generen hasta terminar el árbol y finalmente los hijos del nivel más bajo envían su utilidad a su padre y así sucesivamente hasta llegar al nodo raíz. Cabe destacar que la expansión de un nodo no es ejecutada en esta función, sino que **FuncionPrincipal** hace uso de otra función para esta tarea.

**clonarPuntos:** Es necesario clonar los arreglos de puntuación tanto del jugador como del agente al momento de crear un nuevo nodo y de esta forma evitar la utilización de punteros de java, esto se debe a que cada nodo produce una puntuación diferente la cual es independiente de los demás nodos en esa profundidad.

**expandir:** En esta función se expanden los nodos del árbol, donde las profundidades son 3: profundidad cero para el nodo raíz, uno para las posibles jugadas de Max y dos para las posibles jugadas de Min. Al ejecutar la función **Expandir** lo primero en revisarse es la profundidad del árbol y de acuerdo a esto se toman las medidas necesarias para la expansión de dicho nodo (esto incluye búsqueda de posibles jugadas y creación de nodos hijos de ser necesario).

**agregarNodo:** Esta función se encarga de guardar los nodos nuevos resultantes de cada expansión para posteriormente ser expandidos, asegurando que se guarden en la cola con un orden específico de prioridad. Debido a que se quiere utilizar el recorrido preferente por profundidad (ver ilustración 7) es necesario que la prioridad en la cola sean los nodos con mayor profundidad.



*Ilustración 7 - Búsqueda por profundidad*

**posiblesPosiciones:** Esta función recibe el estado actual del tablero y retorna una lista de arreglos de 4 posiciones (fila ficha, columna ficha, fila ficha pareja, columna ficha pareja) con todas las posibles jugadas para una ficha sin importar su color, apoyándose en la función auxiliar **ValidarEspacio**.

**Nodo.java:** El árbol Minimax se implementó mediante el uso de listas doblemente enlazadas, por ello fue necesario el uso de la clase **Nodo** para permitir el uso de punteros de un nodo a otro. A continuación, una breve descripción de las funciones:

**actualizarPuntaje:** Esta función es llamada por los nodos hijos y recibe el puntaje obtenido, la ficha y las posiciones necesarias para obtener dicho puntaje. Luego basándose en el tipo de nodo, es decir, si el nodo que recibe los datos es Min o Max actualiza el puntaje según las siguientes reglas: si el nodo actual es Max y la utilidad recibida es mayor a la utilidad actual se actualizan los valores, de lo contrario no; si el nodo actual es Min y la utilidad recibida es menor a la utilidad actual se actualizan los valores, de lo contrario no.

**actualizarPuntajeFicha:** Esta función recibe una ficha, un tablero determinado el cual tiene en cuenta las posiciones usadas por el nodo padre y los colores de la ficha. Es la encargada de calcular el puntaje total de la ficha ingresada en ese tablero.

**calcularUtilidad:** Esta función revisa los puntajes obtenidos por parte del agente y del jugador y de acuerdo a una serie de parámetros de los que se hablará más adelante, calcula la utilidad para un nodo en específico.

## 4.2 IMPLEMENTACIÓN DE LA FUNCIÓN DE UTILIDAD

Para calcular la utilidad se tuvieron en cuenta varios aspectos:

1. El principal aspecto estudiado fue la cantidad de fichas que se tienen en el puntaje más bajo por parte del agente, debido a que este factor es el que determinará el ganador.
2. También se tuvo en cuenta las jugadas del contrincante, por ello se consideró el número de fichas que el jugador (la persona) tiene en el puntaje más bajo.
3. Además de los aspectos anteriormente mencionados, se tiene en cuenta también la cantidad de puntos totales tanto del agente como del contrincante.

Aplicando todos estos aspectos obtenemos la siguiente ecuación para calcular la utilidad:

$$Utilidad = \frac{puntosTotalesIA - puntosTotalesJugador}{100}$$

$$Utilidad += 2 \times \frac{puntoMenorIA + 10}{cantidadFichasEnPuntoMenorIA}$$

$$Utilidad -= 2 \times \frac{puntoMenorJ + 10}{cantidadFichasEnPuntoMenorJ}$$

Los multiplicadores usados (2 y 100) para cada factor se escogieron para dar más prioridad a ciertas jugadas.

## 5 CONCLUSIONES

---

1. Escoger la función de utilidad en un árbol Minimax es muy importante, debido a que esta tiene la tarea de indicar al agente que jugada es mejor.
2. Un árbol Minimax puede analizar todas las posibles jugadas para un estado determinado del juego y en base a esto escoger la acción que debería realizar MAX para minimizar la pérdida en el juego
3. La implementación de un árbol Minimax está limitada por la cantidad de nodos, si esta es muy grande la ejecución se verá afectada.
4. En la realización del proyecto, se comprendió que para implementar un buen agente se debe hacer un análisis muy profundo del juego y a partir de este análisis formular una heurística que se acomode mejor a él, sin dejar de lado metodos que ayudarían a aumentar la eficiencia.

## 6 BIBLIOGRAFIA

---

[1] iMisut, <<Reseña: Genial>>, 2016. [En línea]. Disponible:

<http://misutmeeple.com/2016/03/resena-genial/>

[2] Simplemente genial - REGLAS, 2004 [En línea]. Disponible:

<https://boardgamegeek.com/filepage/7806/simplemente-genial-reglas-doc>