



PROYECTO 2

Inteligencia Artificial

Saavedra Escobar Jennifer 1410249, Serna Isabella 1722103, Rojas Tello Alexis 1610115
Facultad de Ingeniería – Ingeniería en Sistemas, Universidad del Valle
Cali, Colombia

Introducción

El proyecto consiste en diseñar un juego llamado **Horses Go (Gotta destroy them all!)**. Sobre un tablero de ajedrez cada jugador tiene un caballo (Un jugador y la maquina). Las casillas por las que pasa cada caballo se destruyen y por lo tanto nunca se pueden volver a ocupar. El juego lo gana quien haga el último movimiento. En el juego se manejan tres niveles de dificultad: principiante, intermedio, y avanzado y el tamaño del tablero se define según el nivel de dificultad en 4X4, 6X6 y 8X8 respectivamente. Para lograr el objetivo se deben aplicar el algoritmo minimax con decisiones imperfectas para ayudar a la maquina a hacer sus movimientos, manejando arboles de profundidad 4, 6 y 8 según el nivel en que se juegue. La implementación se trabajó con el lenguaje de programación orientado a objetos, JAVA.

EL PROBLEMA

En el desarrollo del proyecto se plantea como objetivo básico implementar el algoritmo minimax para el caso de juego de dos participantes donde un humano se enfrenta a una maquina buscando obtener la victoria. El juego consiste en: En un tablero de ajedrez, un par de caballos (Blanco y Negro) se mueven con el movimiento característico de un caballo de ajedrez (en L) por cada movimiento que realiza cada jugador se inhabilita la casilla en la que este se encontraba. Se deben realizar la cantidad de movimientos que sean posibles, el caballo que realice el último movimiento gana el juego. El ambiente es representado con una matriz de enteros de tamaño 4X4, 6X6 y 8X8 según el nivel en que sea elegido por el humano (principiante, intermedio y avanzado respectivamente).

La máquina utilizara el caballo blanco y la posición inicial de los caballos se presentan de forma aleatoria (Los jugadores no pueden iniciar en la misma casilla).

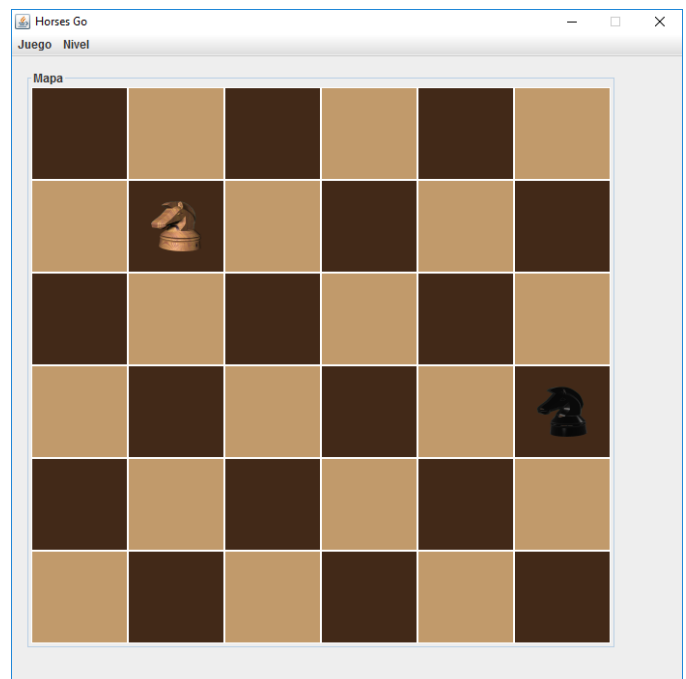


Figura 2
Interfaz Grafica

En la interfaz se observa la ubicación del caballo que representa la maquina en la casilla (1,1) donde se encuentra la imagen del caballo blanco, la ubicación del jugador en la casilla (,) donde se encuentra la imagen del caballo negro. Las casillas deshabilitadas son representadas por la X. Ver Figura 3.



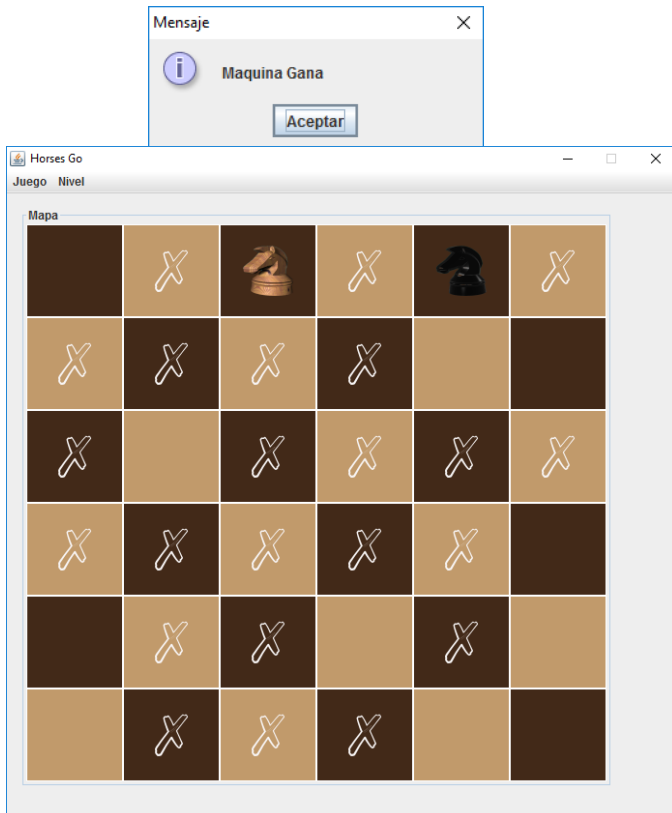
Figura 3

De Izquierda a Derecha (Caballo Blanco, Caballo Negro, X)

Desarrollo del problema

Para llegar al objetivo final que es ganar el juego el algoritmo MiniMax debe ayudar a la maquina a tomar la mejor decisión hasta el punto en que el jugador humano (caballo negro) no tenga opciones de movimiento. Se desarrolló el algoritmo de juegos para dos participantes MINIMAX visto en clase, desde la interfaz gráfica se tiene la opción de elegir el nivel que desea jugar el usuario y se debe elegir la opción de cargar el mapapara iniciar el juego.

Una vez aplicado un algoritmo MINIMAX el caballo blanco que corresponde a la maquina se mueve e inhabilita la casilla inicial. Tal como se muestra en la Figura 4.



Explicación de la Heurística

Para el desarrollo del algoritmo MINIMAX para se utiliza la estrategia de heurísticas que son criterios para decidir cuál entre varias acciones promete ser la mejor para alcanzar la meta.

En la implementación del algoritmo MINIMAX la heurística es una función que asigna a cada nodo un valor que corresponde al costo estimado de llegar a la meta estando en dicho nodo.

Teniendo en cuenta que los métodos de búsqueda heurística disponen de alguna información sobre la proximidad de cada estado a un estado objetivo, se diseña una heurística que nos ayude a obtener un estimado de lo que podría costarle al caballo llegar a la meta. La heurística elegida debe tomar su valor más alto en el punto de partida o posición inicial del caballo y debe ser 0 en la posición final o meta.

Debido a que el caballo solo maneja movimientos en “L” la heurística la definimos como la cantidad de movimientos posibles del caballo.

```
public int heuristica() {
    int MovimientosJugador = getMovimientosPosibles(this.posicionJugador, this.mapaEstado).size();
    int MovimientosPc = getMovimientosPosibles(this.posicionPc, this.mapaEstado).size();
    if (MovimientosJugador == 0 && MovimientosPc == 0) {
        return tipoNodo == TIPO_NODO_MAX ? -99 : 99;
    }
    if (MovimientosJugador == 0) {
        return 99;
    }
    if (MovimientosPc == 0) {
        return -99;
    }
    return MovimientosPc - MovimientosJugador;
}
```

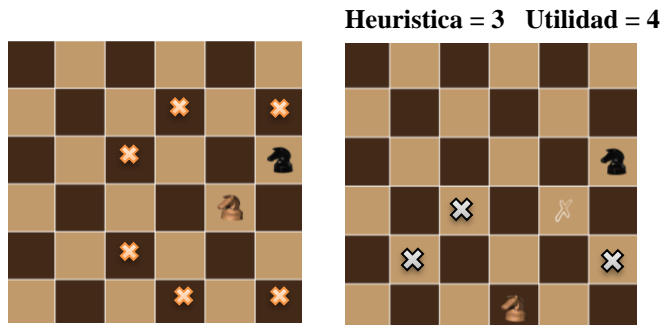
Admisibilidad de la heurística planteada.

Una heurística se considera admisible si nunca sobreestima el costo de alcanzar el objetivo, o sea, que en el punto actual utilidad del nodo nunca es mayor que el menor costo posible.

Se considera admisible la heurística planteada ya que el valor que ofrece la heurística cantidad de movimientos, en el peor de los casos es igual al mejor avance posible, es decir a la utilidad del nodo.

ESTADO INICIAL

ESTADO META

**Figura 4**

Estado meta y estado inicial de un ambiente

En la figura 4 podemos observar que según la heurística el valor de llegar a la meta desde el nodo inicial es 3.

Mientras que la mejor ruta posible teniendo en cuenta el costo tendría un valor de 4.

Al comparar el valor estimado $H(X_1, X_2) = 3$ con el valor real $U(X_1, X_2) = 4$ podemos observar que el valor de la heurística es menor al valor real. Al no sobrepasar el costo real de llegar al objetivo podemos concluir que la heurística es admisible.

CONCLUSIONES.

La implementación de este proyecto logró demostrar que el uso del algoritmo MINIMAX hace al agente tan inteligente como para ganarle a un humano en un juego de dos jugadores. La escogencia del nivel afecta el tiempo de ejecución del juego ya que la escogencia del nivel define la profundidad del árbol MINIMAX, y a su vez la exigencia referente a los recursos de memoria que finalmente desemboca en el tipo de maquina requerida. La importancia de realizar un buen diseño en la heurística para la decisión minimax es imprescindible, de allí su dependencia para hallar la mejor solución sin sobre estimar los costos reales.

BIBLIOGRAFÍA.

- Tema6. Algoritmo MiniMax”, Escuela de Ingeniería de Sistemas y Computación, Universidad Del Valle.
- Russell, S.J.; Norvig, P. (2002). Artificial Intelligence: A Modern Approach.