



Yoshi's World

Integrantes:

Santiago Duque Chacón
2180099

Carlos Mauricio Tovar Parra
1741699

Docente:

Oscar Fernando Bedoya Leyva, PhD.

Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Inteligencia Artificial

Capítulo 1

Introducción

En este proyecto se lleva a cabo la implementación del algoritmo **minimax** en el contexto de un juego con decisiones imperfectas.

En el juego se enfrenta a el jugador (usuario humano) con un enemigo (controlado por la IA), donde el objetivo de ambos es "pintar" la mayor cantidad de casillas posibles con su color. Para lograr el objetivo, el jugador y el enemigo se mueven al igual que un caballo de ajedrez. El juego empieza con ambos participantes en posiciones aleatorias y con un movimiento del enemigo. Las casillas que pinta cada uno (rojo y verde, respectivamente), son todas las casillas por las que pasan, incluidas las posiciones iniciales.

El juego terminará cuando AMBOS participantes agoten sus posibles movimientos (es decir, se queden "cerrados"). De esta manera, cuando un participante se quede sin movimientos el otro seguirá jugando hasta agotar los suyos propios. El puntaje corresponde a la cantidad de casillas pintadas por cada uno.

El comportamiento del enemigo sigue un algoritmo minimax donde la función heurística se basa en la manera de obtener el puntaje final, con un detalle adicional: se dará gran importancia a "cerrar" los movimientos del jugador/enemigo. Esta consideración se debe a que al evitar más posibilidades de movimiento para el contrario, resulta natural una mayor probabilidad de quedarse encerrado y por consiguiente, dejar de puntuar en menos turnos.

A continuación se explicará entonces la función heurística que se diseñó para implementar el algoritmo minimax.

Capítulo 2

La heurística

2.1. Definición y explicación de la función

Como se mencionó, la función heurística implementada gira en torno a 3 valores, los cuales son las casillas pintadas por cada participante, los posibles movimientos en cada estado y la distancia entre jugador-enemigo, a los que se les dan ciertos pesos ajustados tras múltiples pruebas. Adicionalmente, se distinguen 2 casos particulares relacionados con los movimientos posibles, que corresponden al agotamiento de movimientos por parte del jugador y el enemigo.

Teniendo esto en cuenta, la función tendrá en cuenta: el valor *PuntajeJugador – Enemigo*, con un peso de 0.3 (se multiplica el valor obtenido por ese factor); el valor *PosiblesMov.Jugador – Enemigo*, que tiene un peso de 0.6, donde en caso de que algún participante se quede sin posibles movimientos, se asigna directamente un valor de -5 o 5, según corresponda; por último, al valor de la distancia entre participantes, se le asigna un peso de 0.1.

De esta forma, la función final que se estableció es la siguiente:

$$\begin{aligned}P_c &= \text{Player Controlled,} \\E_c &= \text{Enemy Controlled,} \\P_m &= \text{Player Movements,} \\E_m &= \text{Enemy Movements.} \\D_{pe} &= \text{Distance Player-Enemy}\end{aligned}$$

$$H = 0,4 \cdot (P_c - E_c) + \begin{cases} -5,0 & \text{si } P_m = 0 \\ 5,0 & \text{si } E_m = 0 \\ 0,6 \cdot (P_m - E_m) & \text{en caso contrario} \end{cases} - 0,1 \cdot D_{pe}$$

2.2. Consideraciones finales y conclusiones

La principal característica y ventaja de esta versión final para definir la función heurística, radica en considerar no solo los puntajes que van a tener los participantes del juego a lo largo de este, si no que en caso de alargarse, tener las consideraciones sobre distancia y posibles movimientos arroja mejores predicciones para profundidades más allá de la máxima en los árboles generados por el algoritmo minimax.

Respecto a versiones anteriores y más simplificadas para la definición de la función, esta versión demuestra dar un comportamiento más competitivo al enemigo, especialmente en las dificultades mayores. Esto logra crear un juego entretenido y con un nivel de dificultad bastante aceptable para el usuario, a la vez que dota de un poder de decisión mayor al enemigo, donde las mayores ventajas sobre versiones primitivas de la función heurística se ven más claras en instancias del juego que finalizan con mayores turnos totales.