



Informe proyecto II IA

Jerson Alexis Ortiz Velasco
202410014
Jerson.ortiz@correounivalle.edu.co

Inteligencia Artificial
Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Programa Académico de Ingeniería de Sistemas
Cali, Noviembre 28 de 2025

Introducción

Este informe describe la función de utilidad heurística diseñada e implementada para el agente inteligente (Caballo Blanco) en el juego "Smart Horses". El objetivo de esta función es evaluar numéricamente qué tan favorable es un estado del tablero para la IA, permitiendo al algoritmo Minimax tomar decisiones informadas incluso cuando no puede calcular hasta el final del juego debido a la profundidad limitada.

Definición de la Función de Utilidad

La función de evaluación `evaluate_board(game_state)` combina dos factores estratégicos fundamentales: la **Diferencia de Puntos** y la **Movilidad**.

La fórmula matemática utilizada es:

$$Utilidad = (Puntaje_{IA} - Puntaje_{Jugador}) + (Movilidad_{IA} - Movilidad_{Jugador}) * 0.5$$

1. Diferencia de Puntos (Factor Primario)

El componente más crítico de la evaluación es la diferencia neta entre el puntaje acumulado por la IA y el del oponente.

- **Implementación:** `game_state.white_horse.score - game_state.black_horse.score`

- **Justificación:** El objetivo final del juego es terminar con una puntuación mayor que el adversario. Maximizar esta diferencia es la estrategia directa para la victoria. La IA priorizará capturar casillas con valores positivos altos (+10, +5) y forzar al oponente a casillas negativas o de bajo valor.

2. Movilidad (Factor Secundario)

Se otorga un valor adicional basado en la cantidad de movimientos legales disponibles para cada jugador en el estado actual.

Implementación: `(white_moves - black_moves) * 0.5`

Peso: Se multiplica por `0.5` para que sea un factor de desempate y estratégico, pero que no supere la importancia de capturar puntos reales.

Justificación:

- **Evitar el Bloqueo:** En **Smart Horses**, si un jugador se queda sin movimientos mientras su oponente aún puede jugar, recibe una **penalización de -4 puntos**. Al maximizar su propia movilidad y minimizar la del oponente, la IA reduce el riesgo de sufrir esta penalización y aumenta la probabilidad de infligir al jugador.
- **Control del Tablero:** Un caballo con más opciones de movimiento tiene mayor control sobre el tablero y flexibilidad para futuros turnos.

Consideraciones sobre la Penalización (-4 Puntos)

Además de la heurística estática, el algoritmo Minimax simula explícitamente la regla de penalización en el árbol de búsqueda.

- Si durante la exploración del árbol (simulación de futuros turnos), la IA detecta un estado donde un jugador no tiene movimientos pero el otro sí, **aplica inmediatamente la penalización de -4 puntos** al puntaje de ese jugador en el estado simulado.
- Esto permite que la función de utilidad `evaluate_board` capture con precisión el impacto negativo de quedarse encerrado, guiando a la IA para evitar caminos que lleven a esa situación.

Conclusión

Esta combinación de **maximización de puntaje** y **preservación de movilidad** permite a la IA comportarse de manera inteligente en los tres niveles de dificultad:

- **Principiante (Profundidad 2):** Toma decisiones tácticas inmediatas.
- **Amateur (Profundidad 4):** Planea secuencias cortas y evita trampas simples.
- **Experto (Profundidad 6):** Ejecuta estrategias a largo plazo, sacrificando ventajas inmediatas por posiciones superiores y buscando activamente bloquear al jugador humano.