

# Proyecto de IA: Hex

Noel Pérez Calvo

## Descripción General

Este proyecto implementa una Inteligencia Artificial para el juego **Hex**, utilizando el algoritmo **Monte Carlo Tree Search (MCTS)**. El objetivo de la IA es seleccionar el mejor movimiento posible, simulando miles de partidas en un tiempo limitado para evaluar la calidad de cada posible jugada.

## ¿Qué es MCTS?

**Monte Carlo Tree Search (MCTS)** es un algoritmo basado en simulaciones que se usa para tomar decisiones en entornos de gran complejidad como los juegos. El algoritmo construye un árbol de búsqueda de forma dinámica y progresiva mediante simulaciones aleatorias, tratando de balancear dos aspectos clave:

- **Exploración:** Buscar nuevas jugadas que podrían ser buenas.
- **Explotación:** Profundizar en jugadas que ya han demostrado ser prometedoras.

## Fórmula UCT

Durante la selección del mejor nodo, se utiliza la fórmula **UCT (Upper Confidence Bound for Trees)** para tomar decisiones equilibradas:

$$\text{UCT}(i) = \frac{w_i}{n_i} + C \cdot \sqrt{\frac{\ln N}{n_i}}$$

Donde:

- $w_i$ : número de victorias en el nodo  $i$ ,
- $n_i$ : número de visitas al nodo  $i$ ,
- $N$ : número total de visitas al nodo padre,
- $C$ : constante de exploración (usualmente  $\sqrt{2} \approx 1.4$ ).

Esta fórmula asegura que el algoritmo favorezca los movimientos con mejores resultados, pero sin dejar de explorar otros posibles caminos.

## ¿Cómo funciona el algoritmo?

El algoritmo MCTS se divide en varias fases que se repiten hasta que se cumple un límite de tiempo o se alcanza un número máximo de simulaciones.

### 1. Inicialización

- Se crea un nodo raíz a partir del estado actual del tablero.
- Se inicia un bucle que se ejecuta hasta que se cumple un límite de tiempo (por ejemplo, 10 segundos).

### 2. Fase de Selección

- Se recorre el árbol desde la raíz hasta una hoja usando la fórmula UCT.
- En cada paso se elige el hijo con mayor valor UCT.
- Se simula cómo evoluciona el tablero aplicando los movimientos seleccionados.

### 3. Fase de Expansión

- Si el nodo actual tiene jugadas no exploradas, se elige una de ellas aleatoriamente.
- Se expande el árbol creando un nuevo nodo hijo para esa jugada.

### 4. Fase de Simulación

- A partir del estado del nuevo nodo, se simula una partida aleatoria completa hasta que haya un ganador o no queden movimientos posibles.

### 5. Fase de Retropropagación

- Se actualiza el árbol hacia atrás, desde el nodo donde terminó la simulación hasta la raíz.
- Se incrementan los contadores de visitas y victorias según el resultado de la simulación.

### 6. Reversión de Cambios

- Al terminar cada simulación, se deshacen todos los movimientos aplicados para dejar el tablero como al principio.

## **7. Elección del Movimiento Final**

- Una vez agotado el tiempo de búsqueda, se selecciona el movimiento con mayor cantidad de visitas desde el nodo raíz como jugada final de la IA.