

# Implementación de Minmax para Hex

Francisco Prestamo Bernardez

## 1. Descripción General

Un agente de IA simple para jugar al juego de tablero Hex en el cual se combinan diferentes estrategias y algoritmos para realizar movimientos óptimos dentro de restricciones de tiempo.

## 2. Componentes Principales

### 2.1. Integración Python-C

- Utiliza `ctypes` para interactuar entre Python y C
- Compila código C como biblioteca compartida para rendimiento
- Gestiona memoria y transferencia de datos entre lenguajes

### 2.2. Sistema de Toma de Decisiones

- Implementa algoritmo Minimax con poda alfa-beta
- Profundidad de búsqueda configurable (predeterminado = 4)
- Búsqueda limitada por tiempo mediante un hilo de tiempo de espera separado
- Los primeros movimientos se manejan con estrategias especiales

### 2.3. Evaluación del Tablero

La evaluación del tablero se basa en una combinación ponderada de cuatro heurísticas principales:

#### 1. Rango de Conectividad

- Corresponde al mayor rango alcanzado en el eje relevante (horizontal para el jugador 1, vertical para el jugador 2) dentro del conjunto disjunto del que forma parte la celda.
- Representa la expansión máxima del grupo conectado del jugador.

## 2. Proximidad a Conexión Óptima

- Utiliza una métrica basada en el algoritmo de Dijkstra para estimar la distancia mínima desde la celda a ambos lados del tablero.
- Una menor suma de distancias se considera más favorable.

## 3. Valor de Bloqueo

- Mide la cantidad de posiciones enemigas potencialmente bloqueadas desde la celda evaluada.
- Favorece los movimientos que interrumpen la conectividad del oponente.

## 4. Conexiones Estratégicas

- Evalúa la capacidad de la celda para conectarse con otras piezas propias de valor estratégico.
- Favorece la formación de caminos robustos o conexiones críticas.

## 2.4. Optimizaciones de Rendimiento

- Ordenamiento de movimientos usando QuickSort
- Uso de Disjoint Set para detección de victoria
- Terminación anticipada con poda alfa-beta
- Exploración de movimientos basada en prioridades

## 3. Uso

La IA puede utilizarse creando una instancia de la clase `AstroBot` con un ID de jugador (1 o 2) y llamando a su método `play()` con:

- Estado actual del tablero
- Límite de tiempo para el cálculo del movimiento

Nota: Es necesario estar en la carpeta del player y el archivo `c` al mismo tiempo para la ejecución del código