

# AlephNull: Un Jugador Autónomo de Hex

Luis Ernesto Amat Cárdenas C-312

11 de abril de 2025

## Resumen

Este documento detalla el proceso de desarrollo de **AlephNull**, un agente autónomo para el juego Hex que combina técnicas de búsqueda heurística con optimizaciones algorítmicas. El jugador resultante supera consistentemente a oponentes básicos (aleatorios y greedy) y ofrece resistencia contra humanos.

## 1. Objetivo

Desarrollar un jugador de Hex que:

- Derrote consistentemente a oponentes automatizados
- Minimice el número de movimientos para victoria (gane lo más rápido posible)
- Mantenga competitividad contra jugadores humanos (sea entretenido)

## 2. Metodología

### 2.1. Evolución del Algoritmo

#### Fase 1: A\* Básico

- Implementación inicial con heurística de camino más corto
- Problemas: Complejidad temporal  $O(b^d)$  inviable ( $b \approx 121$  para tablero  $11 \times 11$ )
- Logro: Base para la heurística de distancia optimizada

#### Fase 2: Minimax Temprano

- Búsqueda de camino de costo mínimo a lo ancho
- Heurística inicial:
$$h(n) = d_{victoria}$$
- Limitación: Enfoque naive sin estrategia ofensiva

## Evolución de los Algoritmos de Hex

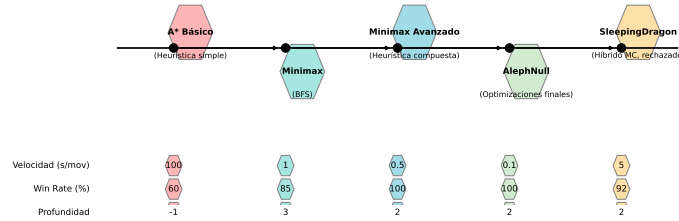


Figura 1: Progresión de las versiones del jugador

### Fase 3: Minimax Avanzado

Heurística final con componentes:

$$h(n) = \underbrace{2(s-d)}_{Victoria} + \underbrace{0,5c_1 + 0,2c_2}_{Estructura} + \underbrace{0,01o}_{Defensa}$$

Donde:

- $s$ : Tamaño del tablero
- $d$ : Distancia del camino más corto (en particular, piezas faltantes en el camino de costo mínimo)
- $c_n$ : Piezas propias a distancia  $n$  del camino
- $o$ : Piezas oponentes adyacentes al camino más corto (combinado con las otras heurísticas, causa que se bloquee al oponente en los "puentes")

En general, el jugador se dirige de un lado al otro del tablero lo más rapido posible mientras genera estructuras defensivas y se protege de ataques menores. Las simulaciones indican que esta estrategia asegura la victoria en  $s/2$  turnos **de no ser interrumpida**.

### 2.2. Abordajes Rechazados

Técnica	Razón de rechazo
Heurística Manhattan	Incompatible con topología hexagonal (véase manhattan.pdf)
Híbrido Minimax-Monte-Carlo	Menos eficiente e inferior a Minimax puro
A* optimizado	Minimax superó en rendimiento
Reescritura de la heurística en C	Rechazado en favor de pobar la efectividad de M-C

### 3. Optimizaciones Clave

#### 3.1. Representación del Grafo

- **Nodos fantasma:** Conexiones virtuales a bordes opuestos para minimizar los cambios requeridos en el algoritmo
- **Reinterpretación del grafo:** Eliminación de nodos oponentes en BFS y 'compresión' de los nodos propios (véase la estructura Árbol de Aristas Puentes donde se usa una técnica parecida).
- **Coste computacional** reducido de  $O(n^2)$  de Dijkstra a  $O(V + E)$  de BFS

#### 3.2. Minimax con Alpha-Beta

- Poda de gran parte de nodos en promedio

### 4. Resultados

#### 4.1. Métricas de Rendimiento

Oponente	Win Rate	Mov. Promedio
Monke (Aleatorio)	100 %	$s/2$
Usagi (Greedy)	100 %	$s/2$
M-C (players probados)	100 %	$s/2$

#### 4.2. Conclusiones

- Una heurística bien balanceada supera a búsquedas no informadas al inicio
- La estructura de puentes temprana es crucial
- Es mejor algoritmo es el que gana más (y mejor aún que si gana rápido)
- La paralelización beneficia más a Monte Carlo que a Minimax
- El enfoque híbrido no siempre proporciona ventajas

### Referencias

- [1] Knuth, D. *Analysis of Alpha-Beta Pruning*. 1975.