



Proyecto: Ajedrez Alicia con Inteligencia Artificial

Alejandro Sierra.
Juan Pablo Castaño.
Juan Manuel Ramirez.
Kevin Ramirez.

December 4, 2024

1. Objetivos del Proyecto

Objetivo General

Desarrollar una versión funcional del ajedrez variante *Alicia* con jugabilidad entre una persona y una inteligencia artificial utilizando algoritmos de *minimax* con poda alfa-beta y una heurística personalizada.

Objetivos Específicos

- Implementar las reglas del ajedrez de Alicia (movimientos entre dos tableros paralelos).
- Diseñar una representación de los tableros y piezas en código.
- Implementar un algoritmo de IA con *minimax* y poda alfa-beta.

- Desarrollar una función heurística que permita evaluar los estados del tablero.
- Crear una interfaz gráfica o de consola para la interacción entre el jugador y la IA.
- Realizar pruebas para asegurar la funcionalidad y mejorar la heurística.

2. Metodología

Para garantizar un desarrollo ordenado y funcional, se seguirá un enfoque **iterativo y modular**, dividiendo el proyecto en etapas manejables.

Etapa 1: Análisis y Planificación

1. Estudio de las reglas del ajedrez de Alicia:

- Definir cómo funciona el cambio entre tableros.
- Analizar las condiciones de captura y movimiento.

2. Definición de requisitos técnicos:

- Herramientas a utilizar (e.g., lenguaje Python, librerías como `python-chess` o `PyQt5`).
- Decidir entre una interfaz gráfica o una versión de consola.

3. Especificar los componentes del sistema:

- Representación de tableros y piezas.
- Implementación de reglas y validación de movimientos.
- Diseño del algoritmo *minimax*.
- Interfaz para la jugabilidad.

Etapa 2: Diseño del Sistema

1. Arquitectura del Juego: Diseñar la estructura del proyecto, incluyendo los módulos principales:

- **Módulo de Tablero:** Manejo de los dos tableros paralelos, piezas y reglas.
- **Módulo de Reglas:** Verificación de movimientos válidos y condiciones de captura.
- **Módulo de IA:** Implementación de *minimax* con poda alfa-beta y heurística.
- **Módulo de Interfaz:** Interacción jugador-IA.
- **Módulo Principal:** Orquestación de la jugabilidad.

2. Flujo de Datos y Jugabilidad: Crear un diagrama conceptual que describa cómo se comunican los módulos y cómo fluye el turno de juego:

- El jugador realiza su movimiento → Cambia de tablero → Verificación de victoria.
- La IA calcula su mejor movimiento → Ejecuta el cambio de tablero → Verificación de victoria.

Etapa 3: Implementación por Fases

1. Fase 1: Representación de Tableros y Piezas

- Crear la representación de los dos tableros (matrices de 8x8).
- Diseñar clases para las piezas con métodos para identificar su tipo (peón, torre, etc.) y reglas de movimiento.

2. Fase 2: Implementación de Reglas

- Desarrollar funciones para verificar:
 - Movimientos válidos para cada tipo de pieza.
 - Capturas y cambios de tablero.
 - Condiciones de victoria (jaque mate, tablas).

3. Fase 3: Algoritmo *Minimax* con Poda Alfa-Beta

- Implementar el algoritmo *minimax*:
 - Explorar los posibles movimientos hasta una profundidad definida.
 - Incorporar poda alfa-beta para optimizar la búsqueda.
 - Usar una función heurística para evaluar el valor de los estados del tablero.

4. Fase 4: Desarrollo de la Interfaz

- Crear una interfaz en consola o gráfica:
 - Mostrar los dos tableros.
 - Permitir que el jugador seleccione y mueva piezas.
 - Mostrar los movimientos de la IA.

Etapa 4: Pruebas y Ajustes

1. Pruebas Unitarias:

- Asegurar que cada módulo funcione correctamente de manera aislada.
- Probar reglas de movimiento, capturas y cambio de tablero.

2. Pruebas de la IA:

- Evaluar si la IA elige movimientos coherentes con base en la heurística.
- Ajustar la profundidad del *minimax* para equilibrar dificultad y rendimiento.

3. Pruebas de Jugabilidad:

- Verificar que el juego es funcional de principio a fin.
- Revisar la experiencia del jugador y la interacción con la IA.

Etapa 5: Optimización y Documentación

- **Optimización:** Mejorar la eficiencia del algoritmo *minimax* y refinar la función heurística.
- **Documentación:** Explicar cómo funciona cada módulo e incluir ejemplos de uso y un manual para futuros ajustes.

3. Cronograma Tentativo

- **Semana 1:** Análisis y diseño del sistema.
- **Semana 2:** Implementación de tableros, piezas y reglas básicas.
- **Semana 3:** Desarrollo del algoritmo *minimax* con poda alfa-beta.
- **Semana 4:** Implementación de la interfaz y pruebas iniciales.
- **Semana 5:** Optimización, pruebas finales y documentación.