Taller II

Introducción

El objetivo principal del taller es implementar el algoritmo Minimax con poda alfa-beta para resolver el popular juego de estrategia "Conecta 4".

Descripción del Juego:

Conecta 4 es un juego de mesa en el que dos jugadores toman turnos dejando caer discos de colores en una cuadrícula vertical. El objetivo es conectar cuatro de sus propios discos del mismo color de manera consecutiva, ya sea horizontal, vertical o diagonalmente.

Objetivos del Taller:

- Comprender el funcionamiento del algoritmo Minimax y su aplicación en juegos.
- Implementar el algoritmo Minimax con poda alfa-beta en el contexto del juego Conecta 4.
- Analizar y comparar el rendimiento del algoritmo con y sin la poda alfa-beta.

ESCUELA DE INGENIERÍA
UCN COQUIMBO

Pasos a seguir

- Antes de comenzar con la implementación, los alumnos deben investigar los fundamentos del algoritmo Minimax y la poda alfa-beta. Familiarícese con cómo estos algoritmos se aplican en la resolución de juegos estratégicos.
- Cada alumno deberá implementar una versión básica del juego Conecta 4, que incluya la lógica para realizar movimientos y verificar la victoria.
- los alumnos deben agregar la implementación del algoritmo Minimax para la toma de decisiones del juego. Cada nodo del árbol de juego debe representar un estado del tablero.
- los alumnos deben extenderla para incluir la poda alfa-beta. Esta técnica ayuda a reducir el número de nodos evaluados, mejorando significativamente el rendimiento del algoritmo.
- Comparen los resultados y los tiempos de ejecución con y sin la poda alfa-beta. Anoten cualquier observación sobre cómo la poda afecta la eficiencia del algoritmo.

Condiciones de entrega

- Se debe generar un menú para elegir la dificultad del juego (fácil, medio, difícil).
- Se debe generar un csv que guarde todas las partidas hasta el momento y dentro del programa se debe poder ver la puntuación de jugador vs máquina.
- Por consola se debe poder visualizar el estado de la partida en el momento (tablero de 6x7).
- Se debe poder guardar el estado de la partida, en caso de que el jugador tenga que cerrar el programa por algún motivo (guardar estado en otro csv y después cargarlo al momento de iniciar el programa).

ESCUELA DE INGENIERÍA UCN COQUIMBO

Entregables

- Se debe subir un link de github con el proyecto en visualización publica a campus virtual con un plazo máximo hasta el 29/11/2023.
- El programa debe ser capaz de compilar con g++ (buscar información sobre mingw o utilizar codespace de github para realizar el proyecto). Se ocupará el comando "g++ -o main main.cpp" para compilar el proyecto.
- Dentro del README del repositorio se debe incluir información sobre la implementación, decisiones de diseño, resultados de pruebas (comparación de minimax sin poda y con poda).
- Se debe utilizar programación orientada a objetos para la programación de las estructuras utilizadas.

Pauta evaluación

	Pts totales	Pts obtenidos
Implementación del Juego (20 puntos)		
Implementación correcta de las reglas del juego Conecta 4.	5	
Correcta alternancia entre turnos de los jugadores.	5	
Implementación adecuada para detectar cuándo un jugador ha ganado.	5	
Interfaz clara y amigable que permita a los usuarios realizar movimientos y visualizar el estado del juego.	5	
Implementación de Minimax (30 puntos)		
Correcta representación de los estados del tablero como nodos en el árbol de juego.	10	
Implementación correcta de la evaluación de nodos terminales (ganancia, pérdida o empate).	10	
Correcta implementación de Minimax para la toma de decisiones del juego.	10	
Implementación de Poda Alfa-Beta (30 puntos)		
Integración de la poda alfa-beta: Correcta implementación de la poda alfa-beta en el algoritmo Minimax.	10	
Evidencia de una reducción significativa en el número de nodos evaluados con la implementación de la poda.	10	

El algoritmo sigue tomando decisiones óptimas después de la implementación de la poda.	10	
Evaluación del Rendimiento (10 puntos)		
Análisis claro y conciso del rendimiento del algoritmo con y sin la poda alfa-beta.	5	
Comentarios sobre la eficiencia del algoritmo y cómo la poda alfa-beta mejora el tiempo de ejecución.	5	

Implementación del Programa (35 puntos)		
Funcionalidad General (10 puntos)		
El programa se ejecuta sin errores graves.	10	
Funcionalidades Específicas (15 puntos)		
Las funcionalidades específicas, como grabar partidas, funcionan correctamente.	10	
Se han implementado al menos tres funcionalidades adicionales más allá de las básicas.	5	
Documentación (10 puntos)		
Se ha proporcionado documentación clara y legible en el código.	5	
Se incluyen comentarios que explican las secciones críticas del programa.	5	

Usabilidad y Presentación (20 puntos)		
Interfaz de Usuario (10 puntos)		
La interfaz de usuario es amigable y fácil de usar.	5	
El menú interactivo es intuitivo.	5	
Organización del Código (10 puntos)		
El código está organizado de manera ordenada y legible.	5	
Se siguen buenas prácticas de codificación.	5	

- Fecha de entrega 29/11 23.59 a través de campus virtual. Consultas -> jose.veas@ce.ucn.cl
- Se evaluará con nota 1 en caso de:
    - Copia
    - No entregar documentación
    - Programa no compila
    - No entregar en fechas indicadas

*El profesor se reserva el derecho de solicitar revisión con alumno en caso de encontrarse con situaciones sospechosas que podrían modificar la evaluación.