Parcial 02

Primera Entrega

CONTEXTUALIZACIÓN, ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

Sergio Giraldo
Julio Benavides

Departamento de Ingeniería Electrónica y Telecomunicaciones

Universidad de Antioquia Medellín

Octubre de 2023

Índice

1. Sinopsis del Juego

Othello, conocido como Reversi, es un juego de mesa estratégico para dos jugadores que se juega en un tablero de 8x8 casillas. Los jugadores se turnan para colocar fichas de su color con el objetivo de capturar las del oponente al rodearlas en cualquier dirección. El juego combina reglas sencillas con profundas estrategias, lo que lo hace atractivo para todos los niveles. La victoria se determina por la cantidad de fichas propias en el tablero al final de la partida. Othello es un desafío intelectual que ha perdurado como un juego de estrategia clásico.

2. Contextualización

El problema que aborda este proyecto es crear una aplicación de software que permita a dos jugadores competir en el juego de Othello en una computadora. El software debe proporcionar la mecánica del juego, incluida la colocación de fichas, la captura de fichas del oponente y la determinación del ganador.

Análisis: Consideraciones para el Abordaje del Problema y Desarrollo de Propuesta de Estrategia de Solución:

Diseño y Mecánica del Juego: El primer paso es comprender en profundidad las reglas y mecánicas del juego de Othello. Esto incluye cómo se coloca y captura fichas, las reglas para ganar y perder, y los aspectos estratégicos clave. La comprensión de las reglas es fundamental para el desarrollo preciso de la aplicación.

Representación del Tablero: Una de las consideraciones más críticas es cómo representar el tablero de 8x8 en la memoria de la computadora. La elección de una estructura de datos adecuada es esencial para facilitar la colocación y captura de fichas, así como para determinar el estado del juego.

Interfaz de Usuario: Para que los jugadores interactúen con la aplicación, es necesario crear una interfaz de usuario. Esto incluye la visualización del tablero y la capacidad de ingresar las jugadas de los jugadores.

Algoritmos de Juego: Para implementar las reglas de Othello, se deben desarrollar algoritmos que permitan verificar la validez de las jugadas, determinar las fichas capturadas y evaluar el estado del juego en busca de un ganador.

Turnos de Jugadores: El juego debe alternar los turnos de los jugadores, permitiéndoles realizar jugadas válidas. La aplicación debe asegurarse de que cada jugador juegue con su color asignado y que las jugadas sean legales.

Finalización del Juego: La aplicación debe detectar cuándo se ha alcanzado el final del juego, ya sea porque el tablero está lleno o porque ningún jugador puede realizar una jugada válida. En este punto, se deben contar las fichas y determinar el ganador.

Manejo de Errores: Se deben incorporar mecanismos para manejar situaciones de error, como jugadas inválidas o entradas incorrectas por parte de los jugadores.

Optimización y Eficiencia: A medida que el juego progresa, es importante optimizar la eficiencia de los algoritmos, especialmente en juegos avanzados donde el número de posibles jugadas puede ser considerable.

Pruebas y Depuración: La aplicación debe someterse a pruebas exhaustivas para garantizar que funcione según lo previsto y que cumpla con las reglas del juego en todo momento. Además, se deben abordar y corregir posibles errores o problemas.

Documentación y Comentarios: Es fundamental documentar el código de manera clara y proporcionar comentarios adecuados para que otros desarrolladores puedan comprender y mantener la aplicación en el futuro.

3. Análisis

- **Análisis para Abordar el Juego de Othello en C++ (Ejecución por Consola):**

- Estructura de Datos: Representar el tablero como una matriz 8x8 para mantener el estado del juego. Usar variables para rastrear el jugador actual y el número de fichas capturadas por cada jugador.

- Entrada de Usuario: Permitir que los jugadores ingresen sus movimientos a través de la consola, por ejemplo, "A4" para colocar una ficha en la fila A y columna 4.

- Reglas del Juego: Implementar las reglas del juego de Othello, que incluyen la validación de movimientos, la captura de fichas del oponente y la actualización del tablero.

- Finalización del Juego: Detectar cuándo finaliza el juego (por ejemplo, cuando el tablero está lleno o ambos jugadores no pueden realizar movimientos válidos).

- Visualización del Tablero: Diseñar una función para mostrar el estado actual del tablero en la consola.

- Gestión de Turnos: Alternar entre los turnos de los jugadores y verificar la validez de los movimientos en cada turno.

- Conteo de Fichas: Realizar el recuento de fichas al final del juego para determinar al ganador.

- Interfaz de Usuario Amigable: Proporcionar mensajes claros y solicitudes de entrada para que los jugadores comprendan y disfruten el juego.

- Loop Principal: Crear un bucle principal que permita a los jugadores realizar movimientos hasta que el juego termine.

- Diseño Modular: Considerar la modularidad para separar funciones relacionadas con la lógica del juego, entrada/salida y visualización del tablero.

- Con este análisis, puedes comenzar a desarrollar el juego de Othello en C++, centrándote en cada uno de estos aspectos para lograr una implementación funcional y entretenida para la consola. En la siguiente figura un posible diseño de la salida en consola del tablero de juego:

4. Diseño De La Solución

Las consideraciones clave que se llevaran a cabo en el desarrollo de este código:

- **Clase Othello:** El código utiliza una clase llamada Othello para modelar el juego. Esta clase encapsula la lógica del juego, incluyendo el tablero

y las reglas para validar y realizar jugadas.

- **Representación del tablero:** El tablero se representa como una matriz de 8x8 caracteres en la clase Othello. Se utiliza ' ' para casillas vacías, '*' para fichas del jugador 1 y '-' para fichas del jugador 2. Esto permite una representación clara del estado del juego.

- **Método jugar():** El método jugar() controla el flujo del juego. Muestra el tablero, solicita movimientos de los jugadores y aplica las reglas del juego para validar y realizar jugadas. El juego continúa hasta que no se pueden realizar más movimientos válidos.

- **Validación de jugadas:** El código implementa la lógica para validar si una jugada es válida. Esto implica verificar si la casilla seleccionada está dentro de los límites del tablero, si está vacía y si rodea fichas del oponente que pueden ser volteadas.

- **Realización de jugadas:** Cuando una jugada es válida, el código voltea las fichas del oponente atrapadas y coloca la ficha del jugador actual en la casilla seleccionada.

- **Visualización del tablero:** El método dibujarTablero() se encarga de mostrar el estado actual del tablero en la consola, lo que facilita el seguimiento del juego.

- **Control de turnos:** La variable jugadorActual se utiliza para llevar el control de los turnos de los jugadores. Se alterna entre '*' y '-' para alternar los turnos.

- En resumen, el código aborda el problema de implementar el juego de Othello en la consola, proporcionando una interfaz para que dos jugadores jueguen. Las consideraciones clave incluyen la representación del tablero, la validación y realización de jugadas, el control de turnos y la visualización del tablero. Es un ejemplo simple pero funcional de un juego de estrategia de tablero.

5. Referencias

[1] "Othello Programming." Stanford Encyclopedia of Philosophy. 2017. [Enlace:https://plato.stanford.edu/archives/fall2017/entries/games-and-gaming/othello-programming.html]

[2] Stanford University. "Programming Othello (Reversi)." [Enlace:https://web.stanford.edu/class/cs221/assignments/reversi/]