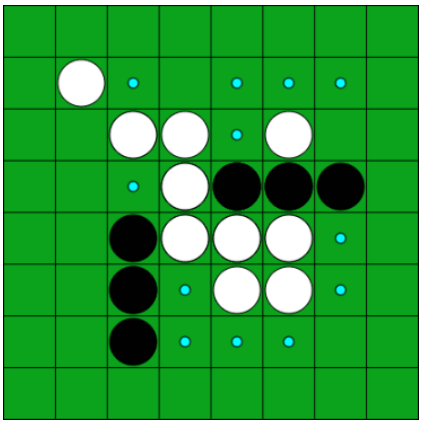
INFORMÁTICA II

Miguel Angel Botero Gonzalez 1040182106
Mariana Vasquez Castiblanco 1003433478

Parcial 2

24 Octubre 2023
28 Octubre 2023

Visión general



"**Othello**" es un juego de estrategia de tablero (típicamente de 8x8), tradicionalmente clasificado como un juego de mesa, concebido para dos participantes. La premisa del juego radica en la adquisición de fichas o discos pertenecientes al oponente mediante una maniobra de encierro de tipo "**sándwich**", que se efectúa mediante la colocación de una de las fichas propias en una posición que se encuentre entre dos fichas enemigas, ya sea en una disposición vertical, horizontal o diagonal. Esta estratégica maniobra provoca una conversión de las fichas apresadas, transformándose en fichas del color del jugador que ejecutó la maniobra.

El objetivo primordial del juego consiste en alcanzar la ocupación de la mayor cantidad de posiciones en el tablero con fichas del color asignado a cada jugador, que son blanco y negro. La victoria se declara cuando uno de los jugadores logra controlar la mayoría de las casillas del tablero con sus fichas de color correspondiente o cuando se agotan las posibilidades de realizar movimientos en el tablero.

Diseño de clases

1. **Clase Tablero:** Esta clase **'Tablero'** representa el tablero de othello. contiene un matriz 2D para representar el estado del juego donde cada celda contiene valores, Contiene métodos para dibujar el tablero luego de cada jugada.
2. **Clase Jugador:** Crea una clase **Jugador** representa al jugador 1 y jugador 2 tiene atributos como el color de la ficha (negra o blanca) y métodos para realizar movimientos válidos.
3. **Clase Juego:** Crea una clase **Juego** que administre el flujo del juego. Contiene métodos para iniciar el juego, realizar movimientos, verificar reglas y determinar al ganador.

Análisis de las clases

Clases y métodos

Tablero(int filas, int columnas): El constructor de la clase Tablero inicializa el tablero con las dimensiones especificadas (filas y columnas). Debes usar una matriz dinámica (por ejemplo, un puntero doble) para representar el estado del tablero. Inicializa el tablero con las fichas iniciales en el centro.

~Tablero(): El destructor se encarga de liberar la memoria utilizada para el tablero.

void ImprimirTablero() const: Este método imprime el estado actual del tablero en la consola, mostrando las fichas de cada jugador y el estado de las casillas vacías.

bool EsMovimientoValido(int fila, int columna, char jugador) const: Verifica si un movimiento es válido. Debe comprobar si la casilla indicada está vacía y si cumple con las reglas de movimiento esto determinara si la nueva jugada resulta en fichas del oponente siendo volteadas.

> Esta jugada valida debe tener encuenta un lineamiento muy importante; solo se podra poner una nueva ficha si esta esta en alguna

relacion con otra ficha de su mismo color, ya sea de manera horizontal, vertical o diagonal; Adicionalmente a la condicion anterior esta nueva ficha solo se puede poner si se realiza un “encierro” con la ficha original.

El encierro o sandwich es una condicion que se debe cumplir en la funcion de movimiento; consta de encerrar 1 o mas fichas del jugador contrario entre una ficha de una nueva jugada y una ficha de una jugada antigua.

void RealizarMovimiento(int fila, int columna, char jugador): Este método coloca la ficha del jugador en la casilla especificada y voltea las fichas del oponente si corresponde.

bool HayMovimientosDisponibles(char jugador) const: Verifica si el jugador actual tiene movimientos válidos disponibles en el tablero.

int ContarFichas(char jugador) const: Cuenta y devuelve la cantidad de fichas de cada jugador en el tablero.

bool JuegoFinalizado() const: Verifica si el juego ha terminado (ya sea porque no hay más movimientos disponibles o el tablero está lleno).

char ObtenerGanador() const: Devuelve el ganador del juego o un carácter especial si es un empate.

Clase Jugador:

Jugador(char ficha): El constructor de la clase Jugador tomará un parámetro para especificar el color de ficha que usará el jugador (negro o blanco).

int ElegirMovimiento(const Tablero& tablero) const: En el caso de un juego entre dos jugadores humanos, este método puede ser modificado para solicitar al jugador las coordenadas del movimiento desde la entrada estándar (teclado). Por ejemplo, podria std::cin para obtener las coordenadas de la fila y la columna del movimiento.

Clase Juego:

Juego(const Jugador& jugador1, const Jugador& jugador2, int filas, int columnas): El constructor inicializa el juego con los jugadores y las dimensiones del tablero, pero no es necesario hacer ningún cambio importante en esta clase.

void IniciarJuego(): Inicia el juego y comienza a alternar entre los turnos de los jugadores. El método RealizarTurno puede permanecer igual, ya que solicitará al jugador actual que elija un movimiento y lo realizará en el tablero.

void MostrarResultado() const: Después de que el juego termine, muestra el resultado final, el ganador o un empate, en la consola.