



INFORMATICA II
ANALISIS CLASES Y DIAGRAMAS

JUEGO OTHELLO

LUIS DAVID MUÑOZ
RAUL DAZA LIÑAN

DEPARTAMENTO DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
FACULTAD DE INGENIERIA
UNIVERSIDAD DE ANTIOQUIA
MEDELLIN 27/10/2023

Othello es un juego de estrategia que se lleva a cabo entre 2 jugadores en un tablero de 64 posiciones con 64 fichas disponibles para cada jugador, estas se distinguen por el color diferente para cada jugador, al final gana el jugador que tenga la mayor cantidad de fichas sobre el tablero, ya sea que el tablero este completamente lleno o parcialmente lleno, pero sin movimientos disponibles para los jugadores.

Se define una serie de clases con sus atributos y métodos que son necesarias para modelar el juego en consola.

Se desea iniciar el juego mostrando el historial de partidas jugadas que están guardadas en un archivo.txt. Se inicia la partida creando los objetos de jugadores, fichas, y tablero, seguido se llena el tablero con las posiciones iniciales de las fichas.

Inicia el primer turno para el jugador1 realizar jugada inicial, se muestra en el tablero los movimientos disponibles para realizar marcados con una O en las casillas, el jugador por consola debe ingresar el numero de fila y de columna en el cual se encuentra la casilla a la cual va a realizar jugada.

Se valida si la casilla ingresada por consola corresponde a un movimiento valido, si el movimiento no es válido, se le pide al jugador que ingrese nuevamente las coordenadas hasta que el movimiento sea válido, si se verifica que es un momento valido, se procede a implementar la lógica para cambiar las fichas al identificador del jugador en turno que acaba de realizar el movimiento.

Se verifica si hay movimientos disponibles o si el tablero esta lleno, en caso de que el tablero aun no este lleno y si haya movimientos disponibles, se cambia de turno de jugador y se repite el proceso. Si se encuentra que no hay movimientos disponibles o que el tablero esta lleno, en este caso la partida finaliza.

Una vez la partida finalice se cuentan las fichas de cada jugador y se muestran los resultados en pantalla, luego estos resultados se guardan en el archivo.txt de historial de resultados para tener registro de las partidas.

Luego le pregunta al usuario si desea jugar una nueva partida, si es afirmativo, se vuelve a iniciar otra partida. En caso negativo se finaliza el juego.

Clase ficha: Esta clase se define para modelar el color de las fichas de los jugadores, en este caso modelara un carácter que simularan a su vez los colores. Color blanco (*), color negro (-).

Clase ficha

Atributos:

char idd

Métodos:

ficha ()

getter y setter ficha ()

Clase tablero: Con esta clase se pretende realizar todas las acciones que involucren al tablero de juego, se crea el tablero de juego que será una matriz de 8X8 para juego general y estándar incluyendo las fichas por que hacen parte del tablero, se inicializara el tablero con las 4 fichas iniciales ubicadas en su respectiva posición inicial, se realizan jugadas, se verifica si los movimientos son válidos, obtiene posiciones de las celdas donde se desean poner las fichas de una jugada a realizar, invierte el carácter de las fichas involucradas en un movimiento, se imprime el tablero de juego cada que se realicen jugadas, verifica si el tablero esta lleno o si no quedan movimientos disponibles para finalizar el juego.

Clase tablero

Atributos:

Fichas* posición[filas][columnas]

Métodos:

tablero ()

iniTablero ()

jugada ()

jugadaValida ()

obtenerPosicion ()

cambiarFichas ()

impriTablero ()

verificaTablero ()

Clase jugador: Esta clase permite modelar los participantes de la partida con el nombre y el identificador de las fichas. Color blanco (*), color negro (-).

Clase jugador

Atributos:

char idd
string nombre

Métodos:

jugador ()
getters setters ()

Clase resultados: Se encarga de los resultados de una partida cuando finaliza, nombres de los jugadores y la cantidad de fichas que obtuvieron cuando finalizo el juego, y proporciona el nombre del jugador ganador.

Clase resultados

Atributos:

string win
string name1
string name2
int fichas1
int fichas2

Metodos:

resultados ()
getters y setters para name1, name2
getWin ()
getFichas1 ()
getFichas2 ()

Clase partida: Esta clase se va a encargar de gestionar la partida de los jugadores, las interacciones, cambio de turno para jugar y mostrar los resultados al finalizar la partida.

Clase partida

Atributos:

Jugadores
bool turno1

Métodos:

partida ()
void jugar ()
void resultadosPartida ()

Clase historial: Esta clase se encarga de mostrar el histórico de partidas que se han jugado, antes de empezar una partida muestra el historial de partidas registradas con los resultados, y al finalizar una partida guarda los resultados en un archivo de texto.

Clase historial

Atributos:

Métodos:

- historial ()
- void mostrarHistorial ()
- void agregarResultados ()

Diagrama de flujo del juego:



