PARCIAL 2, INFORMÁTICA 2.

Juego Othello

Fabian Camilo Falla Ramírez FABIAN.FALLA@UDEA.EDU.CO
Estudiante Ingeniería de Telecomunicaciones
Universidad de Antioquia Medellín,
Antioquia

Sofia Marín Cacante SOFIA.MARINC@UDEA.EDU.CO
Estudiante Ingeniería de Telecomunicaciones
Universidad de Antioquia
Medellín, Antioquia

a) Contextualización breve del problema

Para el desarrollo de la solución se utilizó la estrategia divide y vencerás, analizando el hecho de que es un juego que existe desde hace un tiempo y ya tiene sus reglas de juego, su lógica y sus objetivos. El juego tiene como base un tablero que se puede interpretar como una matriz 2d, es un juego de estrategia que se debe ir actualizando en cada jugada, lo cual implica tener un buen uso de memoria y además tiene varias condiciones para los movimientos lo cual hace que podamos explorar más la programación orientada a objetos.

b) Análisis

Empezamos planteándonos las clases que íbamos crear, la primer idea de clase fue para el tablero, ya que es la base para el juego por lo tanto sería de mayor relevancia, en esta clase se implementan las funciones necesarias validar los ingresos del usuario, la creación del tablero, utilizando punteros y memoria dinámica para buscar la eficiencia en el uso de memoria, se implementan las condiciones iniciales, además de configurar los jugadores y las fichas que les corresponden, también se implementan las funciones necesarias para el manejo del puntaje y para guardar la información y así buscar un ganador. Algo también importante en el análisis de la solución es el desarrollo de las funciones para manejar los movimientos ya que es justo ahí donde se debían implementar las reglas de juego. Por otro lado, en el main utilizamos la clase tablero junto a la implementación de otras funciones necesarias como iniciar el juego y mostrar el historial para ver los resultados de partidas anteriores, en caso de que el usuario elija la opción de jugar se utilizaran todos los métodos de la clase tablero, si queremos revisar el historial solo abriremos el archivo texto de puntajes y nos mostrara en pantalla los datos.

c) Diseño

La lógica de los algoritmos implementados se hizo en base a un tablero que sería nuestra matriz, en este caso de 8x8, cada espacio de esta matriz va a representar las casillas del tablero, para hacerlo más organizado utilizamos el carácter "|", además en este tablero podremos visualizar en la parte superior las columnas que serán representadas con letras de la A a la H y las filas serán representadas con los numero del 1 al 8, esto con la intención de ser más amigables con el usuario. Como en el juego de mesa original hay unas condiciones iniciales y es que en el centro hay dos fichas negras y dos fichas blancas representadas en nuestro parcial por un ¨–¨ y un ¨*¨.

Después se crea una función para configurar los jugadores, haciendo que el jugador 1 siempre ingrese con la ficha –, además se implementa una función que restringe al usuario a solo ingresar letras en su nombre.

En la función mostrar puntaje creamos un puntero que nos permite acceder y ver cuántos puntos tiene cada jugador, la lógica de esta función se da iterando la matriz anteriormente creada y contando cuantas fichas hay de cada color, en este caso de cada carácter (* o -).

Con la función realizar movimiento lo que se hace es buscar en todas las direcciones, iterando la matriz y usando variables nuevas en x y y, dichas variables nos permiten revisar las condiciones y en que casos se puede encerrar una ficha enemiga, cuando validemos todos los movimientos, se rompe la función y éste método lo complementamos con el método tomar movimiento, el cual imprime un mensaje diciendo como ingresar la columna y la fila y en el caso que dicha casilla esté desocupada, llamara el método realizar movimiento para buscar la forma de encerrar la ficha enemiga.

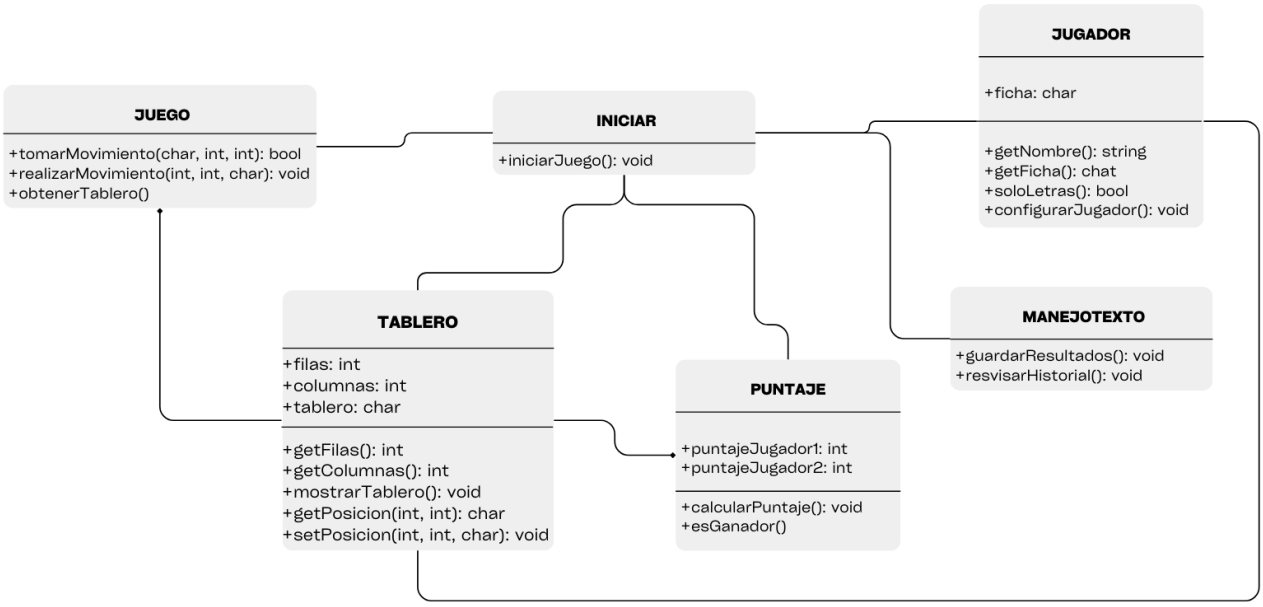
El ganador solo se dará cuando el tablero este lleno completamente o sea que nuestra matriz 8x8 este llena de fichas, cuando esto suceda se comparan la cantidad de fichas del jugador 1 con las fichas del jugador 2 y el que tenga mayor cantidad ganará. Después de esto se imprime un mensaje en pantalla, dicho mensaje lo hacemos con el método guardar resultados que abre un archivo de texto previamente creado con la siguiente estructura:

Jugador 1: nombre (Ficha -) > Puntaje: 41, Jugador 2: nombre (Ficha *) > Puntaje: 22, Ganador: nombre, Fecha y hora: 2023-11-02 22:57:17.

En nuestro main se implementó un switch case que permitirá al usuario elegir si desea jugar, revisar el historial o salir del juego, depende de lo que elija el usuario, se invocaran las funciones necesarias.

Mapa de Clases: Parcial II

Parcial II - Informática 2
Juego de Othello
Fabian Falla - Sofía Marín



d) Algoritmos implementados Enlace al repositorio: **https://github.com/sofiamarinc/Parcial_II.git**

e) Experiencia de aprendizaje

Durante el desarrollo de nuestro parcial, tuvimos algunos inconvenientes, nos costó bastante la implementación para encerrar las casillas y el método del sándwich, la manera de iterar la matriz con punteros también fue algo que nos generó algunos problemas ya que necesitábamos saber que cuál era la ficha de la casilla que representaba el puntero. Otra dificultad que tuvimos, más enfocada en el tiempo del desarrollo, fue que en el momento de implementar el parcial estábamos con prácticas de bastante complejidad, con avances del proyecto final y además utilizando elementos como los contenedores y como la interfaz gráfica que eran nuevos y que no utilizaríamos en este parcial por lo que dividir el tiempo fue bastante complejo, sin contar con las demás materias que no son menos importantes que esta.

Enlace al video: https://youtu.be/LGa2K1WE_4U