

Implementación del juego de mesa Othello

Parcial #2, Informática II 2023-2



I.- Descripción de la asignación

A raíz del auge tecnológico, los juegos de mesa se han visto en la necesidad de reinventarse, migrando hacia versiones digitales que les permitan mantener su popularidad entre las nuevas generaciones. Una etapa fundamental de este proceso consiste en modelar la lógica detrás de reglas del juego de manera que posteriormente se pueda crear la capa de interfaz necesaria para que este pueda jugarse desde la web, móvil, etc.

Othello es un juego de estrategia de tablero para dos jugadores, que se les presentará más abajo en este documento. Su tarea consistirá en implementar el juego Othello en C++ utilizando un enfoque de programación orientada a objetos. Debe proporcionar una base de código bien documentada, un video y un informe detallando el contexto, el análisis, el diseño y la implementación del código.

II.- Objetivos

El objetivo principal poner en práctica sus destrezas en el análisis de problemas y manejo de la POO en el lenguaje C++. Si usted ha llevado un proceso disciplinado de aprendizaje a lo largo del semestre, podrá plantear una solución viable y su resultado será satisfactorio. En caso contrario, podrá identificar sus debilidades y deberá tomar medidas para fortalecer su aprendizaje.

No olvide la importancia de dedicar tiempo a cada etapa del desarrollo, en especial las etapas de análisis y diseño, fundamentales para el enfoque de POO.

Este parcial representa el 20% de la nota final del curso y su contenido ha sido revisado por los profesores Aníbal Guerra y Augusto Salazar.

III.- Instrucciones generales

A continuación, se describen los requisitos que se deben cumplir. El incumplimiento de cualquiera de ellos será considerado al momento de calificarle.

1. Genere un informe en donde se detalle el desarrollo del proyecto. Aborde:
 - a. Contextualización breve del problema.
 - b. Análisis: consideraciones para el abordaje del problema y desarrollo de propuesta de la estrategia de solución.
 - c. Diseño: un diagrama donde refleje y describa las clases que conforman su solución. Incluya los miembros de clase y el especificador de acceso en cada caso. Además, aborde la lógica detrás de los principales algoritmos implementados en los métodos y funciones desarrolladas. Justifique sus decisiones.
 - d. Algoritmos implementados: anexe un enlace al repositorio con los archivos fuente.
 - e. Experiencia de aprendizaje: Problemas de desarrollo que afrontó, evolución de la solución, consideraciones y resultado del aprendizaje alcanzado durante el desarrollo.
 2. La solución debe ser implementada en lenguaje C++ usando POO. Debe considerar el uso eficiente de la memoria, para lo cual debe utilizar punteros, arreglos y memoria dinámica. También considere la eficiencia en los procesos de sus algoritmos.
 3. Se debe crear un repositorio público para almacenar su informe, código fuente, enlace al video de *youtube* y otros anexos. Una vez cumplida la fecha de entrega, no se podrá hacer cambios a este repositorio. Los retrasos en la entrega generarán una penalidad por cada día.
 4. Se deben hacer *commits* de forma regular (al menos, cada dos días), de tal forma que se evidencie la evolución de la propuesta de solución y su implementación.
 5. El video de *youtube* que debe incluir lo siguiente:
 - Presentación de la solución planteada. Análisis realizado y explicación de la arquitectura del software (3 minutos máximo).
 - Demostración de funcionamiento del sistema. Ejecutarlo y mostrar cómo funciona el juego, sus reglas y validaciones (3 minutos máximo).
 - Explicación del código fuente. Justifique las decisiones de sus estructuras de datos y estructuras de control (5 minutos máximo).
- La duración total del video no debe exceder 11 minutos ni ser inferior a 5 minutos.

- Asegúrese que el video tenga buen sonido y que se puede visualizar con resolución suficiente para apreciar bien los componentes presentados.

6. El plazo de entrega se divide en dos momentos:

- a. El día 27 de octubre para adjuntar la evidencia de la contextualización, análisis y diseño de la solución.
- b. El día 3 de noviembre, para adjuntar la evidencia de la implementación.

IV. Reglas del juego y requisitos

El juego se desarrolla en un tablero de 8x8, con fichas negras y blancas. El objetivo es tener la mayor cantidad fichas de tu color en el tablero cuando termine el juego. Puedes probarlo en: <https://www.eothello.com/>.

Existen otras variantes del juego con tablero de otras dimensiones. Aunque no se le pide implementarlas, su solución debe tener un grado de generalidad que facilite la incorporación de esta posibilidad a futuro.

Las reglas del juego son las siguientes:

1. A cada jugador se le asigna un color. El juego comienza con dos fichas negras y dos fichas blancas colocadas en el centro del tablero de forma diagonal (ver Fig. 1).

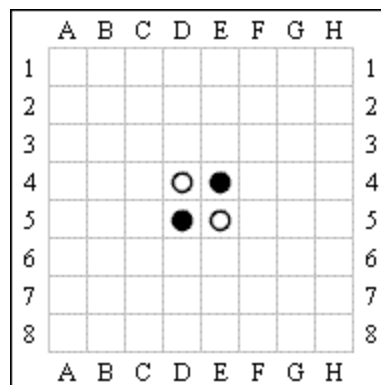


Figura 1. Configuración inicial de Othello. Fuente: Wikipedia

2. Los jugadores se turnan para colocar una ficha de su color en el tablero. El jugador sólo puede colocar su ficha si esta produce un encierro tipo "sandwich".
3. En un encierro tipo "sándwich" quedan atrapadas una o más fichas del oponente, entre la ficha que se ha colocado y otra ficha del mismo color. Los encierros pueden ocurrir en sentido vertical (hacia arriba y hacia abajo), horizontal (hacia la izquierda y hacia la derecha) y en diagonal (ambas diagonales). La consecuencia de un encierro

es que todas las fichas del oponente que quedaron encerradas se cambiarán al color del jugador actual. En la figura 2 se observa un ejemplo.

	A	B	C	D	E	F	G	H	
1			○	○	○	○	○		1
2				○	●	●	○		2
3	○	○	○	○	○	○	●	○	3
4	○	●	○	●	●	●	↓		4
5	○	●	●	○	○	●	○		5
6	○	○	○	○	●	●	○		6
7			●	●	●	●			7
8			●	●	●	●			8
	A	B	C	D	E	F	G	H	

	A	B	C	D	E	F	G	H	
1			○	○	○	○	○		1
2				○	●	○	○		2
3	○	○	○	○	○	○	○		3
4	○	●	○	○	○	○	○		4
5	○	●	●	○	○	●	○		5
6	○	○	○	○	●	●	○		6
7			●	●	●	●			7
8			●	●	●	●			8
	A	B	C	D	E	F	G	H	

Figura 2. Encierro tipo “sandwich”. En el tablero de la izquierda se ha marcado con una flecha la posición (G4) donde el jugador de blanco pondrá su ficha. Al hacerlo (ver tablero de la derecha), las fichas negras de las posiciones D4, E4, F4 (en sentido horizontal) y E2,F3 (en sentido diagonal) han quedado encerradas entre dos fichas blancas. Ahora, todas las fichas encerradas pasan a ser del color blanco. Fuente: Wikipedia.

4. Si un jugador no puede colocar su ficha encerrando fichas de su oponente, debe pasar su turno. En este caso, debe validarse que efectivamente el jugador no tiene ningún movimiento posible.
5. El juego termina cuando ningún jugador puede colocar fichas (según la regla 2) o cuando el tablero está completamente lleno.
6. Gana el jugador con la mayor cantidad de fichas de su color cuando termine el juego.

Requisitos:

1. Implemente las clases necesarias para modelar el tablero, jugadores, administrar el juego (según las reglas previas) y representar cualquier entidad que considere relevante. La modularidad debe ser una característica fundamental del código a desarrollar.
2. Implemente la visualización del tablero de la forma más amigablemente posible. Actualice dicho tablero luego de cada jugada, representando las fichas de color blanco con un asterisco y las de color negro con un guion. Todas las entrada y salidas del juego deben darse a través de la consola.
3. Al finalizar cada partida, los resultados deben guardarse. Para ello, debe guardar el nombre de los jugadores participantes, la fecha, la hora, quién ganó y con cuantas fichas. Cuando se cierre el juego, debe almacenar la información de las partidas jugadas en un archivo de texto, usando el formato de archivo que usted considere pertinente.

Incorpore en su interfaz una opción que permita imprimir por pantalla el histórico de todas las partidas.

4. Considere un uso adecuado de excepciones para manejar los posibles errores.

5. Documente su código internamente.

V. Restricciones

a.- En su solución no debe usar contenedores de la STL.

b.- Para esta solución, todo el código debe ser de desarrollo propio, no debe reusar soluciones disponibles en la web ni generadas por IA.

c.- Para este parcial no debe desarrollar interfaces gráficas de usuario, la visualización debe manejarse únicamente por la consola. Puede utilizar sobrecarga y plantillas en caso de que lo considere necesario.