Proyecto Parcial 1 Informática 2

ACTIVIDAD

Implementación del juego de mesa Othello (PreInforme)



Facultad de Ingeniería

Departamento de Ingeniería Electrónica y

Telecomunicaciones

**Estudiante:**

Emiliano De Jesús Lince Diaz

Universidad de Antioquia

Medellín

2021-2

Tabla de contenido

[Objetivos 2](#_Toc149328681)

[Descripción de la asignación 2](#_Toc149328682)

[Introducción (PRE-INFORME) 2](#_Toc149328683)

[**Reglas del juego y requisitos** 3](#_Toc149328684)

[**¿Como puedo empezar a crear el juego?** 4](#_Toc149328685)

[**Enlaces** 5](#_Toc149328686)

[***Enlaces web*** 5](#_Toc149328687)

# 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.

# 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.

# **Introducción (PRE-INFORME)**

Primero debemos dar una exploración acerca del juego que vamos a diseñar en el lenguaje C++, para ello vamos a investigar en diferentes paginas web y videos, para luego poder desarrollarlo en código.

Para ello en este documento subiré enlaces a videos y paginas web no solo sobre el juego si no tambien ayudas que nos servían para un buen desarrollo de esta practica y estará todo anexo al final de este documento.

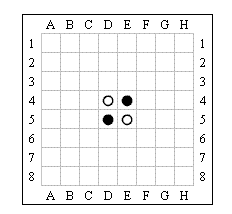
Primero que todo vamos con las reglas y los requisitos que tenemos tener en cuenta para el desarrollo del juego.

## **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.

Imagen que contiene Forma

Descripción generada automáticamente

En la figura 2 se observa un ejemplo. 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.

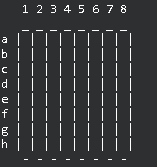
6. Gana el jugador con la mayor cantidad de fichas de su color cuando termine el juego.

Ahora viene la parte importante, ¿cómo se va a desarrollar el juego?, Para ello tenemos que empezar desde el comienzo desde lo más básico y eso es el entorno del juego, ¿cómo así?, para poder jugarlo primero debemos desarrollar como una “proyección” del tablero para saber la ubicación de las fichas ya colocadas, saber en qué posición están, para ahí si poder saber si en cualquier posición del tablero que yo escoja esta libre u ocupada, o saber si ya se termino el juego y alguien gano o fue un empate, para ello vamos a usar valores, datos, nombres para poder identificar y activar cualquier acción que tengamos que efectuar en el juego.

## **¿Como puedo empezar a crear el juego?**

Si no tenemos ni idea de cómo empezar o no tenemos claridad podemos siempre recorrer al internet, aunque se sugiere que todo lo desarrollado en esta práctica tratase de hacerse intuyendo en uno solución para resolver los problemas y no recorriendo al internet tanto, para así aprender mas y que queden los conocimientos en nosotros, igual al final mandare videos de ejemplos de otros tipos de juego de mesa hechos en entornos de desarrollo, igualmente este documento, junto con el próximo documento será tambien de ayuda para desarrollarlo.

Podríamos empezar armando un tablero en consola como este:



Donde en los espacios en blanco irán las fichas blancas y negras.

std::cout << std::endl;

std::cout << " 1 2 3 4 5 6 7 8 " << std::endl;

std::cout << " \_ \_ \_ \_ \_ \_ \_ \_ " << std::endl;

std::cout << "a | | | | | | | | | " << std::endl;

std::cout << "b | | | | | | | | | " << std::endl;

std::cout << "c | | | | | | | | | " << std::endl;

std::cout << "d | | | | | | | | | " << std::endl;

std::cout << "e | | | | | | | | | " << std::endl;

std::cout << "f | | | | | | | | | " << std::endl;

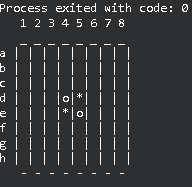
std::cout << "g | | | | | | | | | " << std::endl;

std::cout << "h | | | | | | | | | " << std::endl;

std::cout << " - - - - - - - - " << std::endl;

use este código para dibujar dicho tablero y con tabla ascii podemos hacer que nos dibuje las fichas ya que nosotros usando funciones, constructores y variables para poder colocar las fichas en las posiciones indicadas por el usuario. Y que se guarden para luego verificar si se gano la partida o no.

Tambien podríamos usar asteriscos o “o” para las fichas negras y las blancas respectivamente, asi como lo hare yo:



## 

# Informe

Para este desarrollo de la practica empezaremos primero con el diseño de nuestro código en C++, mirando primero que clases vamos a usar y su manejo como tal, para ello lo primero que me llego para la solución del problema, es el usar la ubicación como clave principal de todo el desarrollo, ya que con ello puedo hacer cualquier proceso con mucha efectividad, luego el crear una clase madre la cual será tablero y su atributo principal es el tener la ubicación de todas las fichas en general. Luego pensando en sus métodos pude deducir que necesitaría de unos claves como MostrarTablero donde podremos mostrar el tablero del jugo con todas las fichas colocadas hasta el momento por el jugador y el VerificarFicha donde buscaremos en la ubicación general de las fichas y miraremos que la posición deseada por el jugador no este ocupada o sea errónea y por último el método ColocarFicha que lo que hará será mirar si la ficha es lanzada por el jugador 1 o el jugador 2 y dependiendo de eso mandara el carácter correspondiente a cada jugador, junto con su posición exacta a MostrarTablero.

Aquí subo desarrollo del código llevado hasta ahora, aunque no esta completamente terminado ya que tiene errores que seguiré corrigiendo, aunque el transcurso de la práctica se acabara. Mis disculpas por los errores y el no tener completo a tiempo lo pedido y solicitado:



Primero subiré los pantallazos del archivo tablero.cpp donde se definen cada método y que hace

## **Texto Descripción generada automáticamente**

Texto

Descripción generada automáticamente

## **Texto Descripción generada automáticamente**

Texto

Descripción generada automáticamente

Ese ultimo es el header donde defino los atributos y métodos de la clase.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Y aquí es donde llamo a los métodos de la clase a ejecutar.

## **Enlaces**

### ***Enlaces web***

<https://service.mattel.com/instruction_sheets/B3165-SP.pdf>.

***Enlaces videos YouTube*.**

<https://www.youtube.com/watch?v=Q3rKjAKXKCI>.

<https://www.youtube.com/watch?v=VzBuqhiJUw8>.

Otros proyectos Ayuda

<https://www.youtube.com/watch?v=0d5Ud_HRxUw>.

<https://www.youtube.com/watch?v=vKnzxaVbw4k>.

<https://www.youtube.com/watch?v=ovorFAFXH0k>.