**INFORME DESARROLLO DEL PROYECTO**

**Contextualización del problema.**

El objetivo de este parcial es crear un juego cuyo nombre es ‘Othello’ a través del lenguaje C++, haciendo uso de la programación orientada a objetos principalmente, para encontrar una solución satisfactoria al problema es sumamente importante tener claro los conceptos vistos sobre POO en las sesiones de clase teóricas.

**Análisis:**

**-**Definir muy bien los atributos y funciones que tendrán las clases a implementar. Las cuales son principalmente la clase jugador y la clase tablero.

-Se utilizará un archivo .cpp y .h que tendrá las funciones necesarias para hacer el uso de manejo de archivos.

- En un turno de un jugador, informar cuáles movimientos validos presenta, si tiene, pedir su movimiento y actualizar los datos de la matriz, que contiene la información del tablero.

-Tener siempre en cuenta el estado inicial del tablero en cada partida.

**Diseño**

**Clase jugador**: Se crea esta clase con el objetivo de crear dos objetos tipo jugador correspondientes a las fichas negras y blancas al inicio de cada partida, de esta manera la clase jugador tendrá la información necesaria para modelar cada objeto, ya sean las fichas negras o blancas. En los atributos de esta clase tendremos: cantidad de fichas del objeto, símbolo en consola correspondiente al objeto (‘\*’ o ‘-‘) y una variable puntero doble , en la cual se almacena utilizando memoria dinámica el estado del tablero.

En las funciones (public) de esta clase tendremos: una Función encargada de retornar un tipo de dato booleano para saber si el objeto de clase puede hacer algún movimiento, se creará otra función encargada de recibir los datos de la casilla en la cual se desea agregar una nueva ficha si el objeto presenta movimientos validos en su turno.

**Clase tablero:** Esta clase se creará principalmente para crear el arreglo en memoria dinámica de la matriz 8x8, en la cual cada elemento de esta es un dato tipo char que puede ser un asterisco, guion o espacio en blanco (indicando que la casilla esta vacía). Esta matriz luego de crearla mediante un puntero doble es la que se pasara como argumento para crear los objetos de la clase jugador a través del constructor de dicha clase. Atributos: cantidad de fichas totales que hay en el tablero, puntero doble en donde se almacena el estado del tablero.

En las funciones (public) de la clase tendremos: una función que se encargara de actualizar las fichas del tablero, luego que un objeto de la clase jugador realizo un “sandwich”, una función que se encargara de imprimir el estado del tablero, una función para comprobar si el juego termino, es decir, que hay 64 fichas totales en el tablero.

Se creará un archivo .cpp y .h el cual contendrá algunas funciones necesarias para guardar en un archivo .de texto, nombre de los jugadores participantes, la fecha, la hora, quién ganó y con cuantas fichas.

**Algoritmos implementados**

**Link repositorio**: <https://github.com/Kelogi/Parcial2-Info2.git>

**Experiencia de aprendizaje**

-Uno de los principales inconvenientes fue determinar cuantas y cuales clases se utilizarían.

- Otro inconveniente que se presento fue pensar muy bien la lógica que se utilizaría para saber que movimientos validos presenta un jugador de la clase jugador, además de la lógica necesaria en la clase tablero para actualizar el estado del tablero (matriz), luego que se hizo “sandwich”.

- Se fomenta la capacidad de análisis en la solución de problemas que requieran el uso de programación orientada a objetos a través de C++.