Sebastián Giraldo Álvarez

Informática II

Informe Parcial

Parcial II

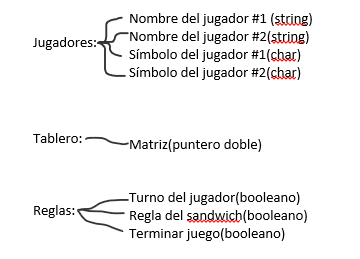
Universidad de Antioquia

En este documento se hará un informe donde se evidenciará el proceso del desarrollo de la solución del problema que se plantea para este parcial. Se sabe que se tiene que programar un juego de mesa llamado Othello, para una mejor familiarización con este se brinda en el documento un link donde se puede testear como tiene que funcionar el mismo cuando se ejecute.

En el documento anterior de contextualización ya se hizo una atracción respecto a lo que se desea hacer; se establecieron desde una perspectiva externa que funciones, clases y variables serán necesarias para que todo el problema se desenvuelva de una manera efectiva, productiva y lo más simplificada que se pueda.

26 de octubre.

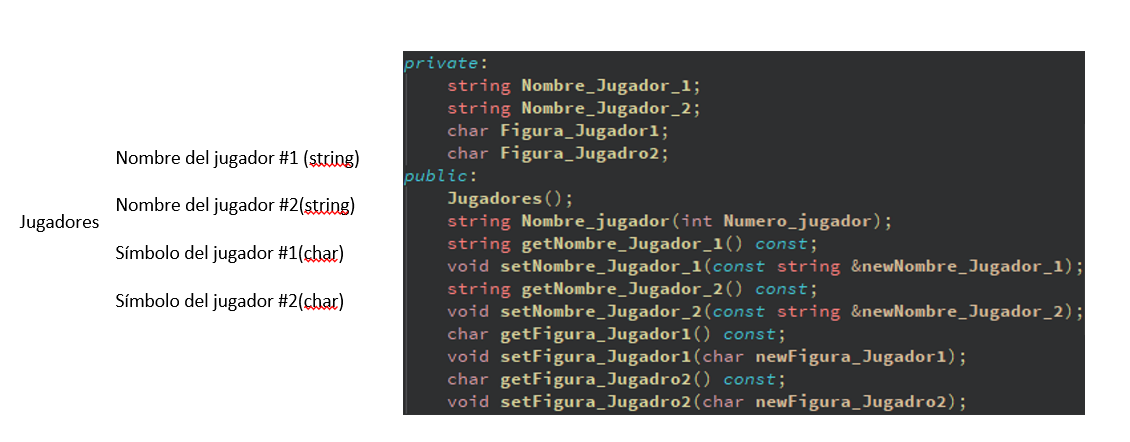
Este día se realizó una abstracción profunda del problema, y a la conclusión que se llego fue que habría por el momento 3 clases, la primera será la de los jugadores, esta contará con atributos privados que permitirán almacenar los nombres de los jugadores, esto con el objetivo de tener toda la información necesaria para otra característica que se necesita al final, otro atributo para tener la ficha/signo que le toco a cada jugador, y los cuales eran lo que representaban blanco (\*) y negro (-). La otra clase será aquella con la que se podrá almacenar todo el desarrollo del juego mientras se ejecuta. La tercera clase es aquella que va a verificar respecto al tablero si se puede o no poner en un lugar en específico uno de los signos.



Este es un pequeño diagrama que tampoco especifica mucho el cómo quiero que funcione el programa, pero me sirve como una base para comenzar.

27 de octubre

Este día comencé con el desarrollo del proyecto, comenzando a estructurar la función jugador.



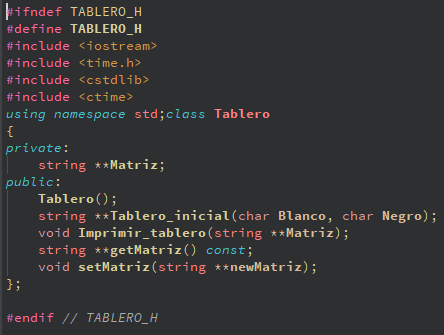
Cada atributo privado cuenta con su respectivo método gett y sett, esto para poder manipular esta información entre clases. Los atributos privados Nombre jugador 1 y Nombre jugador 2 son string porque es un tipo de variable que me permite utilizar de manera más cómoda. Y los signos que representan la ficha de cada jugador se declara como un char, esto para poder ingresarlo normalmente en el tablero.

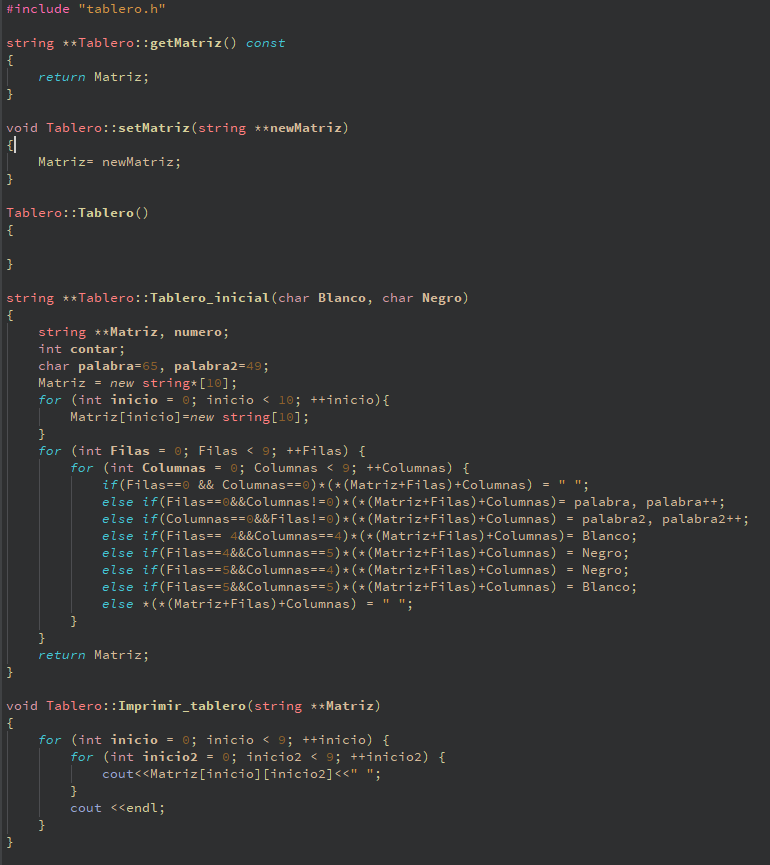
28 de octubre

Este día se me presento un problema, cuando le asignaba un valor a el espacio de memoria correspondiente a la posición que el jugador quiere poner su ficha me generaba un error. Tuve que buscar en internet como asignarle el espacio de memoria de una manera correcta a el puntero doble. Aparte de esto estructuré la clase tablero, al momento de comenzar a ingresar los datos fue cuando me di cuenta de que estaba este error, este día no pude solucionarlo, pero cree funciones en un apartado que lo llame Eficiencia y esto me permitió hacer un código más limpio y eficiente.

29 de octubre

Este día solucioné el problema que anteriormente no me permitió avanzar en el proyecto, la manera que encontré de asignar a el espacio de memoria del puntero doble fue la siguiente “\*(\*(Matriz+Filas)+Columnas)”, ya con esto me permitía ejecutar con normalidad el código. Después de solucionar lo anterior, termine de estructurar la clase tablero, esta cuenta con dos métodos públicos, uno para mostrar la matriz donde se están guardando las posiciones del juego, y la otra es para almacenar la información en esta.



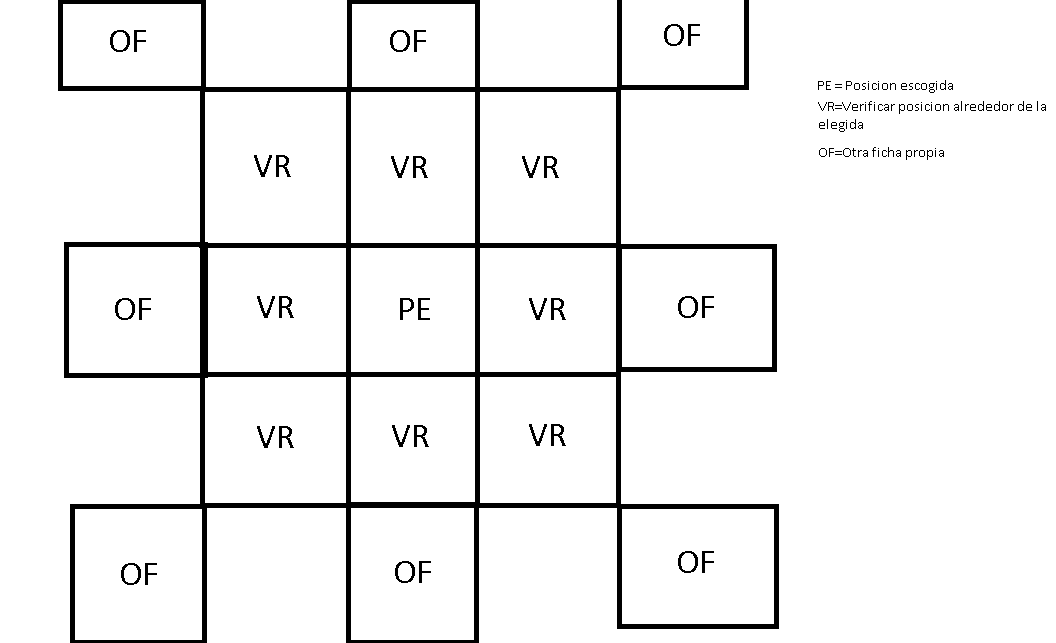


30 de octubre

Comencé a hacer la última clase, que posiblemente sea la parte más complicada del proyecto, pues representa aquel fragmento del parcial donde hay que verificar diferentes aspectos dentro de la matriz, aspectos tales como, el verificar si la jugada que se desea hacer es válida, y si se aplica la regla del sándwich. Esta clase tendrá métodos muy específicos, la idea que tengo por el momento es una para la regla del sándwich, otra para terminar el juego según el número de fichas que se tengan en el tablero, así que tal vez podría hacer un método que lleve un conteo de ambas fichas por separado y después la sume para determinar en el final del juego quien gana y quien no.

31 de octubre

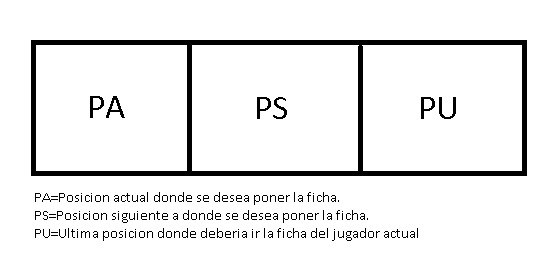
Se me presento una dificultad este día, pues para el método de la regla del sándwich no se ejecuta de la manera en la que se debería, pues yo plantee mi solución de esta manera, el jugador va a escoger una posición, la regla dice que para que una jugada sea válida tiene que formarse un encierro tipo sándwich, ósea, al menos una ficha del jugador contrario tiene que estar en algún lado alrededor, y además después de esta ficha contraria debe haber una ficha de la propiedad del jugador que tiene el turno en ese momento.



La dificultad recae en que el programa cuando lo pruebo solo está cambiando las fichas específicamente cuando esta una ficha del otro jugador e inmediatamente otra del jugador que tiene el turno, básicamente si hay una cadena de fichas del jugador contrario entre dos del jugador del turno actual no funciona.

1 de noviembre

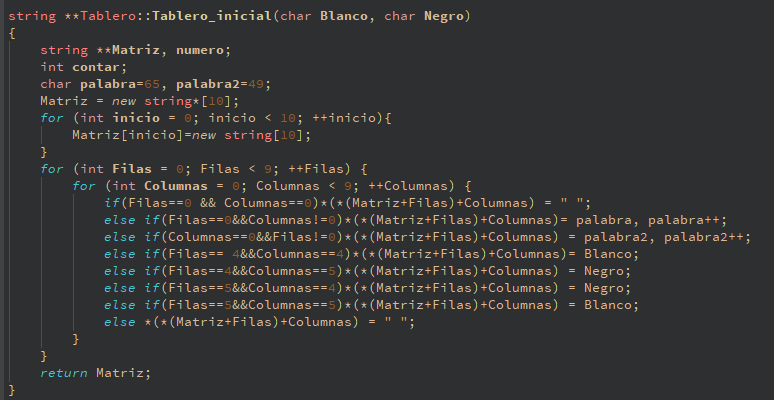
Este día pude solucionar este problema, simplemente poniendo primero un ciclo que recorra la matriz, específicamente todas las líneas alrededor de la posición que escogió el jugador del turno actual, y verificar que primero que todo haya directamente a su lado una ficha de del jugador contrario, además, que verifique si en toda esta línea hay alguna ficha del jugador del turno actual, si la hay, que termine el ciclo y además, cambie todas las fichas blancas que hay entre ambas.



2 de noviembre

El último día realizando el proyecto, en general solo me falta desarrollar aqueda función que me permite guardar cada juego y decir quien gano, además del extra de poner la fecha y hora de aquellos juegos que se han hecho, el documento fue creado con normalidad a través de la clase string, y fstring. Además de terminar esto, también comente el código y limpie un poco el código, es decir, elimine algunos cout que había puesto para usarlo como un debugger, finalmente use la página que se brindó en el documento para comparar esta con el comportamiento del juego que ya había programado, esto con el objetivo de verificar el correcto funcionamiento de todas las clases, métodos, funciones y variables en general.

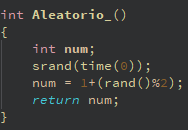
Explicación de algoritmos importantes en el proyecto.



Se define una matriz tipo string con memoria dinámica para poder poner las secciones para orientarse cuando se imprima, es decir, las letras y los números para poder ingresar la posición en la que se desea poner una ficha, en un principio se reservan los espacios de las filas, y gracias al primer ciclo se reserva para las columnas. En el segundo ciclo se puede observar cómo se le asigna a la matriz unos valores necesarios para tener una matriz base, tal y como indica el nombre del método.

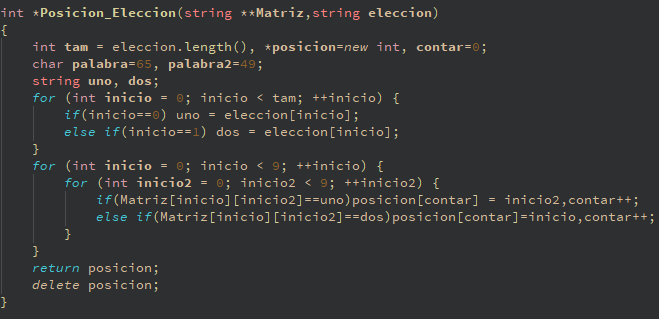


Como se le asigna a cada jugador una ficha:



Esta función permite retornar un número del 1 al 2, de esto depende si el jugador #1 le tocan las fichas blancas o negras, este número se retorna y en un condicional ubicado en el main se efectúa su debida asignación.

Función para reorganizar la posición que da el jugador y reorganizarla para que sea más fácil encontrarla en la matriz.



Simplemente por medio de un contador, y además un char que tiene como valor la representación de las letras en su número según la tabla ASCII, busco su posición en números enteros, por ejemplo, si el jugador escoge la posición A1, la función va a retornar la fila 1 y la columna 1, y el valor que se va a retornar de la función es un puntero con ambos valores.

Finalmente, en el parcial pude desarrollar mi manera de pensar en los problemas, el cómo analizar los proyectos, como dividirlos para que quede algo tanto compacto como reutilizable y productivo. Me gustó mucho el tipo de parcial que es, además de cómo se presta para que nosotros como estudiantes busquemos una manera idónea para un correcto desarrollo.