**Informe Proyecto Final- Programación Avanzada en C++**

**Integrantes:** Daniel Martínez y Wilmer Cárdenas

**Fecha:** 19/12/2023

**NRC:** 3307

**Introducción:**

Este informe trata sobre la creación y desarrollo del método 'Mover' y los métodos 'MovimientoValido' en un juego de ajedrez virtual hecho en C++.

**Metodología de Implementación:**

**1. Cómo Funciona el Movimiento ('Mover'):**

- El método 'Mover' se encarga de cambiar las coordenadas que da el jugador en algo que el juego entiende y del movimiento de las piezas en el tablero.

- Primero cambia lo que el jugador dice (como "a2 a a4") en números del tablero (del 0 al 7).

- Luego hace varias comprobaciones:

- Revisa si hay una pieza en el lugar de donde se quiere mover.

- Asegura que a donde se mueve la pieza esté permitido para ese tipo de pieza, usando el método 'MovimientoValido'.

- Chequea que en el lugar a donde se mueve no haya una pieza del mismo equipo.

- Si todo está bien, mueve la pieza, cambia el tablero y pone el movimiento en el registro.

**2. Cómo se Comprueba un Movimiento ('MovimientoValido'):**

- Cada tipo de pieza en el juego tiene su propia manera de usar 'MovimientoValido', siguiendo las reglas de cómo se mueve cada pieza en el ajedrez.

- Por ejemplo, un peón puede moverse una casilla hacia delante o dos si es la primera vez, pero una torre puede ir de lado o hacia arriba y abajo en el tablero.

**Desafíos:**

- Fue difícil asegurarse de que el registro no tuviera movimientos que no eran válidos.

- Otro problema fue no solo plantear bien las reglas de movimiento para cada pieza, algo muy importante en el ajedrez, sino también poder conectar las clases derivadas con la clase tablero y aplicarlas en el main.

**Soluciones:**

- Para solucionar lo del registro, se cambió el método 'Mover' para que solo añadiera movimientos al registro si eran correctos y válidos.

- Se tuvo mucho cuidado al hacer las reglas de movimiento de cada pieza, pensando bien en las reglas del ajedrez.

**Lecciones Aprendidas:**

- Aprendimos lo importante que es comprobar bien los datos que entran, especialmente en juegos con reglas complicadas como el ajedrez.

- También vimos lo útil que es usar herencia y polimorfismo en C++ para hacer una programación limpia y que se pueda extender a diferentes tipos de piezas.

**Conclusión:**

La forma en que se realizó el método 'Mover' y los métodos 'MovimientoValido' funcionó bien para manejar el juego de ajedrez. Los desafíos y las soluciones nos ayudaron a entender mejor las reglas del ajedrez y cómo programar en C++. Este proyecto mostró lo importante que es tener una buena comprobación y un diseño cuidadoso cuando se trabaja con sistemas de reglas complejas.