TP FINAL PROGRAMACIÓN 2

SANTIAGO PRATO

1. Lectura y manejo del archivo en C

El primer paso fue decidir cómo interpretar el archivo que contiene los nombres de los jugadores, sus colores, quién empieza y todas las jugadas. Para eso armé funciones separadas:

-jugadoresAsignacion para leer los nombres y colores.
-empezo para saber qué color juega primero.
-procesarJugadasDesdeArchivo para recorrer y aplicar cada jugada.

La idea fue mantener toda la parte de lectura de archivo completamente separada de la lógica del juego. Esto ayudó mucho a evitar inconsistencias en el tablero y también hizo que probar cada parte fuera más simple.

2. Representación del tablero

En C representé el tablero con una matriz de estructuras Celda, y en Python usé una matriz de caracteres. En ambos casos usé la misma convención:

'X' para casillas vacías
'B' para las fichas blancas
'N' para las negras

3. Validación y aplicación de jugadas

Esta fue una de las partes más importantes del proyecto. Para validar una jugada, tanto en C como en Python, recorrí las ocho direcciones posibles buscando la secuencia “fichas del rival seguidas por una ficha propia”. Ese enfoque resultó muy útil porque:

-Evita duplicar código.
-Permite que la lógica del juego sea casi idéntica en ambos lenguajes.
-Hace que detectar errores sea más fácil, porque cada caso está bien aislado.

Cuando una jugada era válida, la función aplicarJugada se encargaba de dar vuelta las fichas correspondientes, usando exactamente las mismas direcciones que la validación.

4. Estado del juego en C

La función estadoJuego resume cómo queda todo una vez aplicadas todas las jugadas del archivo. Para eso sigue un orden bastante claro:

-Mostrar el tablero final.
-Revisar si el jugador que sigue tiene jugadas disponibles o si corresponde saltear turno.
-Determinar si la partida ya terminó.
-Contar las fichas de cada color para definir un ganador.

Este orden ayuda a que el flujo sea fácil de seguir y evita comportamientos ambiguos.

5. Interacción y niveles de juego en Python

En Python implementé la parte interactiva:

-El jugador humano ingresa sus jugadas manualmente.
-La computadora ofrece dos niveles:
  Nivel 0: elige una jugada válida al azar.
  Nivel 1: analiza todas las jugadas posibles y elige la que más fichas voltea.

Para esto resultó clave reutilizar todas las funciones ya probadas (jugadasPosibles, jugadaValida, contarFichasVueltas, etc.), lo que mantuvo la coherencia con la versión en C.

Decisiones importantes

Las decisiones que más influenciaron la estructura final fueron:

-Usar vectores de direcciones para validar y aplicar jugadas.
Simplificó muchísimo el código.

-Separar completamente la lectura del archivo, la lógica del juego y la presentación.
Ayudó a probar cada parte por separado.

-Crear los tests desde el principio.
Esto permitió detectar problemas rápidamente y evitó que los errores se acumulen más adelante.