Ajedrez

Raúl Nuño Valdés

Noviembre 2019

1. ¿Cuál fue el proceso de programación del ajedrez?
   Primero se planteo el problema (nos dijo cuál era el proyecto que tenía en mente). Luego, empezó a escribir las clases que le serían necesarias, como `Game` para probar el programa, `Board` para hacer el tablero, o `Rook` que fue la primera pieza. Una vez que ya se había checadoo el funcionamiento, revisó las clases y simplifico su contenido, creando nuevas clases que hicieran trabajos generales, como `Piece`. Tambien dejó de usar. También dejó de usar a `Game` y creó una clase para la interfaz gráfica.

2. ¿Cuál es la complejidad del programa completo?
   n a la 2.

3. ¿Cúal es el algoritmo o la función que es más compleja de ejecutar?
   El método drawBoard() es el más complejo.

4. ¿Qué conceptos vistos en clase aplicaste y en dónde?
   Herencia, en las pieza, por ejemplo. Clases abstractas en `Piece.java`. Encapsulación, con el paquete `core`.

5. ¿Es un proyecto difícil?
   Con la ayuda que nos dio, no fue tan difícil; sin embargo, aún hay cosas que no podría hacer, como el cambio del peón cuando llega al final del tablero.

6. Después de haberlo hecho entre todos ¿Crees que podrías ahora implementarlo completo tú solo?

Tal vez, pero aún hay movimientos de piezas y acciones con ellas que me serían complicadas, como el cambio del peón, o registrar cuantos turnos han pasdo (que podría ser útil para el enroque y comer a paso).

7. Describe con tus palabras cómo implementarías la regla Peón al paso
   Dberíamos incluir una variable `boolean` que registre si el peón se ha movido dos casillas en el turno anterior y cambiar si en el turno siguiente del adversario no se come a la pieza. En caso de que si se vayan a comer la pieza, habría que comprobar, en caso de que esté el peón en la casilla y=3 para blanco y y=4 para negro, deberá comprobar siempre las casillas a los lados, si hay un peón cuya variable boolena sea true, entonces se eliminará el peón enemigo y se moverá en diagonal el peón que come.

8. Describe con tus palabras cómo implementarías detectar que hay un jaque
   Cada pieza, debería tener un método jaque o parecido que reciba List¡Postition¿ y compruebe si el rey de color contrario a la pieza está en una de esa posibles posiciones. En caso de que sí esté el rey, entonces debería mandar un mensaje escrito, como el método `writeTurn`, pero que indique que color está en jaque.

9. Describe con tus palabras cómo implementarías enroque
   Podría ser nesesaria una variable entera que contara cuántos movimientos han efectuado las piezas, o una boolena que indique si se ha movido al menos una vez. Si tantoel rey y una torre no se han movido según la variable y, además, comprobamos que los 3 o 2 espacios (por lado de la reina o por lado del rey) están vacios, entonces podemos mover al rey dos casillas hacia el lado y habría que indicar que la torre también se mueve. Luevo la variable entera o booleana cambiaría para que no pueda volver a efectuarse la acción.