



Juego de ajedrez

por Nicolás Kano Chavira

18-enero-2019

1. ¿Cuál fue el proceso de programación del ajedrez?

Primero creamos la clase Position para poder marcar las coordenadas de las piezas. Luego creamos el tablero en processing, usando las herramientas nativas de este programa. Después creamos un prototipo de pieza armando la pieza Rook para ver que cosas necesitaríamos para el ajedrez y ver como haríamos los movimientos, de ahí decidimos heredar sus movimientos a la clase Queen pero después decidimos que seria mejor hacer uso de la herencia por generalización y crear una clase abstracta llamada Piece de la cual heredarían las demás piezas, en Piece se comparten los atributos básicos (su posición y su color) y algunos métodos importantes que cada pieza debe tener, también hicimos dos tipos de enum uno para el tipo de cada pieza (Pawn, Rook, Knight, Bishop, Queen, King) y otro para los colores (Black, White, None) y con los enum y la clase pieza desarrollamos cada pieza individual creando métodos para conseguir los movimientos legales de cada pieza, al final acabamos de construir la parte de processing para que el programa tuviera todas las piezas y se marcaran los movimientos que puede hacer una pieza y también indicar a que jugador le toca jugar, y para acabar hacemos unas pruebas finales para revisar que todo el código funcione como debería y arreglar si algo sale mal o algo no funciona, por ultimo queda elaborar este archivo donde detallamos algunos detalles importantes de la practica.

2. ¿Cuál es la complejidad del programa completo?(Qué tanto poder de cómputo requiere)

la complejidad del programa es de $O(n^3)$

3. ¿Cuál es el algoritmo o la función que es más compleja de ejecutar?

Aunque hay muchas funciones con complejidad de $O(n^3)$ la mas compleja es draw ya que esta llama a drawboard (que tiene complejidad de n^3), drawPosibleMoves (que tiene complejidad de $O(n)$)y write turn (que tiene complejidad de $O(1)$)

4. ¿Qué conceptos vistos en clase aplicaste y en donde?

Vimos el uso de objetos en toda la practica ya que estamos haciéndola en un lenguaje orientado a objetos y estamos usando varios tipos de objetos de diferentes maneras para hacer que el programa funcione.

Estamos usando el concepto de manejo de flujo cuando comparamos el color de las piezas o cuando vemos cuando una pieza puede moverse mediante ifs y también lo ocupamos cuando hacemos bucles con fors por ejemplo cuando dibujamos el tablero o cuando llenamos la matriz con sus respectivas piezas.

Vimos el uso de herencia en las clase Piece y todas sus hijas(Pawn, Rook, Bishop, Knight, King, Queen) también la clase Piece aplica los conceptos de clase abstracta.

El concepto de agrupación de objetos es uno de los mas importantes que aplicamos para este proyecto ya que es la base de nuestro tablero con el cual trabajamos para hacer casi todo en el programa

5. ¿Es un proyecto difícil?

No creo que sea difícil, mas bien diría que es un proyecto laborioso ya que cada parte del programa individualmente no es muy difícil de hacer, tal vez la parte mas complicada seria la parte de pasar la matriz a processing pero fuera de ello no es complicado, pero si son muchos métodos y clases que tenemos que hacer así que si lleva un rato hacer el programa

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

Si, ya que la dificultad de este proyecto no es demasiada creo que si podría implementarlo desde cero sin ningún problema

7. Describe con tus palabras cómo implementarías la regla Peón al paso

en la clase Pawn la pondría un atributo digamos `int paso=0` que cuando haga comienzo largo cambie a 1, en el turno del oponente checaría con un método observador en el Pawn enemigo si ese atributo es 1 agrega el movimiento para poderse comer la pieza y también tendría que modificar el tablero para cambiar el peón que hizo comienzo largo a tipo NONE para que lo borre, si moviera con otra pieza y no me comiese al peón que hizo el comienzo largo cambiaría otra vez su atributo paso a 0

8. Describe con tus palabras cómo implementarías detectar que hay un jaque

haría un método que revisara las posiciones en la lista de movimientos validos y que piezas están contenidas en ellas (piezas de color contrario o piezas de tipo NONE) si hay una pieza de color contrario checaría que clase de pieza es y si es Rey marcaría con un recuadro rojo a la posición del rey para mostrar que hay un jaque

9. Describe con tus palabras cómo implementarías enroque

agregaría un atributo llamado enroque a la clase King y a la clase Rook que sea un bool y sea true de base, si haces un movimiento con la pieza se cambia el atributo a 0, luego se checa si entre King y Rook no hay piezas en medio, si es así y ambos tienen sus enroque en true se agrega el movimiento en la lista de movimiento valido y si se hace este se mueven las dos piezas (Rook y King)