Documentación referente a la práctica final.

Pedro Pablo Selas García.

75895266-G

selas@correo.ugr.es

pedro.selas@gmail.com

Ingeniería Informática Sistemas (B).

CONTENIDOS.

- **Diseño.**
- **Planificación y organización.**
- **Comentarios.**

Diseño.

El diseño que propuese para abordar la práctica, en principio em basé en las implementaciones de las clases en preprácticas y ejercicios voluntarios previos. La clase matriz e imagen por ejemplo.

A la hora del diseño, consulté con mi tutor (Javier Martínez Baena) proponiendo un diseño y el me citaba ideas de diseño, cosa que me facilitó después el trabajo a la hora de la implementación, debido que conforme se va avanzando en la abstracción de los problemas mejor aplicar un "divide y vencerás". Por ello opté por el diseño de clases que me facilitasen el posterior trabajo con otras clases.

Diseñé una clase vector de piezas para facilitar la interfaz con la bolsa de piezas y la cola de piezas, así evitava depurar errores típicos por culpa de la implentación de mucho código y de poca ordenación y abstracción.

Pensé en implementar una clase vector de imágenes, pero no lo vi necesario, a no ser que el diseño en vez de trabajar con matrices de enteros, trabajase con imágenes, por lo tanto la descarté porque sólo una función en todo el proyecto la necesitaba.

Diseñé la bolsa de pieza para no tener que trabajar con dos vectores en la misma clase del mismo tipo, y evitar fallos y tiempo en depuración.

Posteriormente diseñé la clase Tablero o acumulador para su fin, que trabaja con la clase matriz.

Opté por el diseño de una clase maracador para la gestión de la parte gráfica de los mismos, ya que eran 5.

La clase Interfaz, (para mi gusto la más interesante y entretenida) es con fin de gestionar toda la parte gráfica del juego en si y evitar ciertos fallos en el diseño y desarrollo, porque cuanto más sea la abstracción, mejor se trabaja y

mejor podrán trabajar en el programa los que vengas detrás, es decir hacerlo para todos y no hacer un programa sin protocolo para así evitar quebrantamientos de cabeza *(Ya tenemos lo que se llama "Obfuscated Code").*

Clase matriz.

Esta clase está destianda a manejar matrices de enetros de una manera más facil, utilizando memoria dinámica y con una implementación que lo que hace es trabajar con las matrices dinámicas como sino lo fueran.

Clase Imagen.

Esta clase es para ofrecer una interfaz más cómoda con las imágenes. Trabaja de manera conjunta con la biblioteca gráficos.

Clase Pieza.

Es la clase con la que se construyen las piezas y necesita de la clase matriz para su fin.

Clase Vpieza.

Esta clase gestiona un vector de piezas para ofrecer interfaz mucho más cómoda para la gestión de la bolsa de piezas y cola de piezas, utilizando métodos como Push(), Pop(), etc..

Clase Bolsa_piezas.

En esta clase se guardan las piezas del juego en si, ofreciendo una interfaz muy cómoda, utlizando métodos de la clase Vpieza, de aquí es donde se extraen las piezas para el vector de piezas.

Clase Cola_piezas.

Esta clase es la que va dando las piezas al juego, obteniendo las piezas de manera aleatoria desde la bolsa de piezas. Utiliza métodos de la clase Vpieza. La cola podía hacerse de muchas maneras, yo decidí por sacar la pieza desde la primera posición e insertar al final, regenerando la cola en cada Pop().

Clase Marcador.

Esta clase gestiona la parte gráfica de los marcadores.

Clase Interfaz.

En este clase se gestiona todo lo referente a la parte gráfica del juego, ofreceindo también una interfaz para la carga de la configuración desde un fichero externo.

Planificación y organización.

Empecé por el desarrollo de las clases en el siguiente orden.

- **Matriz.**
- **Imagen.**
- **Pieza.**
- **Tablero.**
- **Vpieza.**
- **Bolsa_piezas.**
- **Cola_piezas.**
- **Marcador.**
- **Interfaz.**
- **Implementación de las funciones extra para la gestión del juego en el main.**

  Configuracion. Carga la configuración del juego. Semilla, lineas, etc..

  Tablero. Carga la configuración del tablero.

  Cola. Configuración referente a la cola de piezas.

  Titulo. Configuración referente al marcador de título.

  Nivel. Configuración referente al marcador de nivel.

  Lineas. Configuración referente al marcador de lineas.

  Piezas. Configuración referente al marcador de piezas.

  Estado. Configuración referente al marcador de estado.

  CargarPiezas. Lee las piezas.

  CargarTablero. Lee la configuración del tablero.

  Pintar. Refresca la interaz del juego.

  Juego. Desarrollo del juego en si. "Evita líneas de código en el main"

  Saltar. Se encarga de la lectura de los ficheros con comentarios, etc.

Todas las fucniones que cargan la configuración del fichero, utilizan un flujo iostream para la lectura del fichero.

Comentarios.

- **He generado la documentación con doxygen pero lo he echo desde windows, al parece no me ha echo todo lo que se supone que tenía que hacer, porque los comentarios de las funciones lo los encuentro ni el de la biblioteca.**

- **No he podido abordar la práctica como me hubiera gustado, es decir hacerla muy completa por falta de tiempo con las otras asignaturas, sólo lo básico sin puntuaciones, etc...**

- **Me falta hacer el vector de imágenes de manera dinámica.**
- **Muchas gracias por sus propuestas de diseño, etc...**