

MANSIÓN ZOMBI

VERSIÓN 2

Docente: Hugo María Vegas Carrasco

Alumno: Jonathan Marín Chicharro

Asignatura: Programación

Fecha límite de entrega: 14/05/2024

ÍNDICE

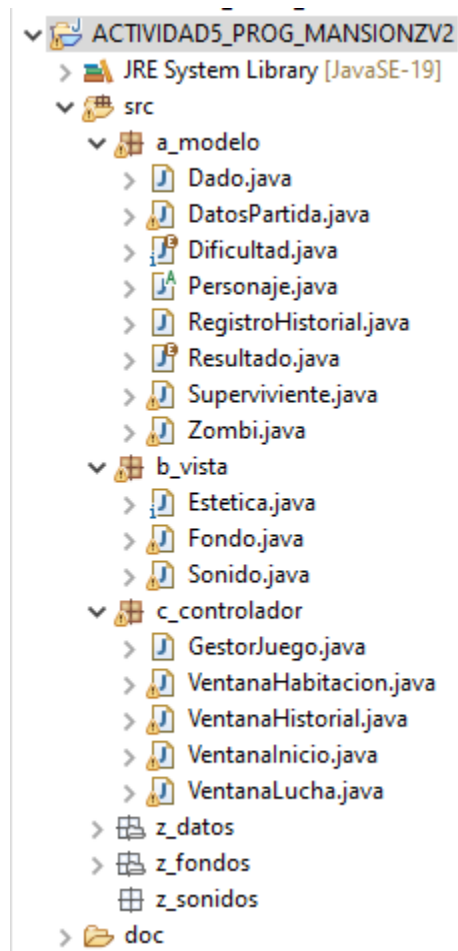
Introducción	4
Manual técnico	5
Conclusión	8
¿Qué me ha parecido y me ha aportado la actividad?	8
Dificultades y retos encontrados	8
Alguna propuesta de mejora de la misma	9

Introducción

En esta práctica se ha trabajado en una aplicación gráfica por medio de componentes Java Swing (también AWT) para el juego “Mansión Zombi”, ya desarrollado anteriormente.

Manual técnico

He intentado clasificar las clases y los contenidos según el patrón MVC. Al principio me resultó muy útil, cuando la cantidad de cosas por pensar podía resultar abrumadora. Era algo con lo que empezar a trabajar.



En el plan original (que hice antes de empezar si quiera con el código) ya tenía algunas clases pensadas pero desde luego que no todas.

Contenido

MODELO

- Contenido de la mansión Zombie

clases:
zombie, usuario,
dato,

Media

VISTA

- Estático todo

- Recibe lo del controlador por referencia y lo modifica

clases:
gestorFondos,
gestorSonidos

Visual

CONTROLADOR

- El usuario interactua y juega en la Mansión

- Se dan órdenes a vista para que cambie cosas

clases:
gestorJuego,
ventanaInicio,
ventanaHistorial,
ventanaLucha

Tanto el modelo como la vista contienen muchos atributos y métodos estáticos a los que acceden las clases que controlan lo que está sucediendo en el juego. Luego en todo el código de manera global se ha podido acceder a estos recursos con facilidad.

Los datos también se almacenan en objetos que pueden guardarse en ficheros binarios. Se crea un objeto nuevo tanto al guardar, como al cargar, como al asignar a los atributos estáticos.

Pienso que esta no es la mejor manera de escribir código dado que los atributos estáticos pueden comprometer mucho la integridad de los datos, pero he recurrido a ella para acelerar la realización de la tarea.

Si no estoy equivocado, he podido controlar las posibilidades de “corrupción” de los datos. Me he asegurado de que al cargar los datos siempre se lean de un fichero, y de que al asignarlos a los atributos

estáticos, esto se haga con otro objeto distinto. No creo que sea lo más eficiente, pero creo que el programa funciona adecuadamente.

El uso de la clase 'DatosPartida' para almacenar datos estáticos y además crear objetos puede ser algo confuso. He intentado señalarlo lo mejor posible en el código.

Conclusión

¿Qué me ha parecido y me ha aportado la actividad?

Ha supuesto un reto planificar y organizar las clases de forma que tuvieran sentido, pero también ha sido muy divertido trabajar en este programa.

Dificultades y retos encontrados

Planifiqué la ventana del historial de una manera diferente, rellenando las filas con atributos de un objeto en lugar de simples String, por lo que el almacenar en ficheros de texto no era lo más adecuado (esto me pasa por no leer lo suficiente las especificaciones al principio). Al final hice una solución híbrida: se añaden filas a través de objetos y a través de Strings. Esto puede ser algo confuso.

La mayor parte del funcionamiento del juego lo programé en muy poco tiempo. Estaba todo lo suficientemente organizado como para que pudiera trabajar rápidamente, pero creo que es bastante mejorable. Hay código que sé que no será accesible, y funciones que se repiten innecesariamente, aunque todo ello le da algo de seguridad. Es algo que quiero revisar.

También tengo claro que he abusado de métodos y atributos estáticos para trabajar más rápido. Existe una falta de encapsulamiento que puede resultar peligrosa, como ya he señalado.

Alguna propuesta de mejora de la misma

La ventana del historial de resultados podría incluir un botón para borrar los datos almacenados.