



Fecha de la defensa: 9 de mayo de 2023

Fecha de entrega: 24 de mayo de 2023

Índice de contenidos

1. Introducción	3
2. Planificación	3
3. Diagrama de clases	4
4. Diagrama de secuencia	4
5. Casos de prueba	5
6. Excepciones	5
7. Conclusiones	6

1. Introducción

Nuestro proyecto trata sobre un juego por turnos. En este el jugador controla a tres héroes que deberá mantener al menos uno con vida para no perder. Para ello cuenta con opciones como atacar, cubrirse, utilizar objeto o cambiar personaje. Una vez acaba el turno, el enemigo ataca a uno de los héroes al azar bajando la vida de este. Así hasta que bien el villano o bien los tres héroes pierdan toda la vida y la partida se dé por finalizada.

Los objetivos del proyecto son los siguientes:

- Conseguir un juego entretenido y cómodo de jugar.
- Aprender a diseñar/implementar cosas no dadas en clases magistrales.
- Ser capaces de crear un diseño propio y editarlo a nuestras necesidades.
- Conseguir encontrar casos extremos a la vez que se desarrolla el diseño.

2. Planificación

En un principio la planificación estaba hecha de la siguiente manera:

- Gorka: diseño diagrama secuencia, implementación y entregas.
- Iker: diseño diagrama clases, casos extremos y entregas.
- Unai: diseño diagrama clases, implementación y entregas.

Al final se ha hecho de la siguiente manera:

- Gorka: diseño diagrama secuencia y clases, implementación y entregas.
- Iker: diseño clases (no tanto como se esperaba) y casos extremos.
- Unai: diseño clases, implementación (no tanto como se esperaba).

Se puede ver que hay cierta incoherencia en la planificación, esto es debido a que las cosas no salían y unos participantes del grupo tuvieron que tirar del resto, trabajando estos mucho más que los otros.

Si hablamos de horas de trabajo estaría bien dividirlo por entregas:

Para las primeras entregas nuestro contenido fue bastante escaso, por lo que las horas trabajadas también lo reflejan:

- Para la primera entrega lo hicimos un día al salir de clase y estuvimos 2-3 horas entre los tres hasta terminarlo.
- Para la segunda hubo compañeros empezando a descolgarse. Tampoco es que se hiciera una entrega muy completa; pero este fue el tiempo invertido de cada uno: Gorka 8 horas, Unai 1 hora e Iker 1 hora.
- Para el video de presentación no solo teníamos que preparar eso, sino que teníamos que trabajar lo no trabajado anteriormente, es por eso que el tiempo es bastante mayor: Gorka 17 horas, Unai 5 horas, Iker 5 horas.

3. Diseño: diagrama de clases

El diagrama de clases final se encuentra en la carpeta del proyecto.

Respecto a los cambios realizados desde los primeros diseños hasta este han sido básicamente los siguientes: añadir nuevas clases que nos permitan expandir la capacidad del juego, añadir varios getters y setters que deben utilizarse para poder cambiar ciertos atributos y la corrección de clases tipo MAE.

4. Diseño: diagrama de secuencia

El diagrama de secuencia también se encuentra en la carpeta del proyecto.

Para leerlo de manera correcta se deben seguir los siguientes pasos:

- 1) Una vez se abre la carpeta se encuentran la Imagen1, Imagen2 e Imagen3; las cuales representan el bucle inicial, de principio a fin.
- 2) El método representado con el número 12 (realizarAccion) se extiende; y este está en la carpeta Espadachin, en el cual se siguen las imágenes desde la uno a la tres.
(Para el mago el funcionamiento es el mismo, solo que en vez de utilizar el método 42 utiliza el 68, y el resto lo hace igual)
(La curandera también, hace lo mismo quitando esa parte final donde en vez de llamar a 42 llama al 69 el cual llama a la ListaHeroes con curación(), el cual llama a cada héroe con subirVida(100):void)
(En resumen, los héroes solamente cambian en la parte de la habilidad especial, el resto de su diagrama es igual)
- 3) De los diagramas de los héroes el método 29 está extendido en la carpeta atacar.
- 4) Del diagrama de atacar está extendido el método 45 en la carpeta ataque.
- 5) Ahora volvemos al diagrama de los héroes y el método 40 está extendido en la carpeta cambiarPersonaje.
- 6) Ahora volvemos a las imágenes del bucle inicial, y tenemos el método 25, el cual se extiende en la carpeta RealizarAccionDF//RealizarAcción.
- 7) Dentro de esta carpeta se encuentra el método 57, el cual está extendido en la carpeta RealizarAccionDF//Acción, la imagen accionDF.
- 8) Por último, esta imagen tiene varios métodos extendidos en esa carpeta: el 64 el cual es la imagen Hab2, el 65 el cual es la imagen Hab1 y el 66 el cual es la

5. Casos de prueba: diseño de las JUnit

Aquí va un listado de todos los casos extremos que se han identificado a la hora de hacer el diseño:

- El bucle principal se reinicia sin ningún problema.
- El bucle se para cuándo debe.
- No salta ningún error a la hora de elegir la acción del héroe.
- Si un héroe esta muerto, este no puede recibir ataques ni puede ser seleccionado para realizar una acción.
- Las restas de vida se hacen de manera correcta.
- No se excede el máximo de vida por personaje.
- Las pociones aumentan correctamente las estadísticas.
- No salta ningún error a la hora de elegir pociones.
- A la hora de cambiar el personaje no salta ningún error.
- Al seleccionar un personaje este se asigna para realizar la acción.
- Si el personaje esta muerto no se puede cambiar a este.
- Cuando se utiliza cubrirse, no debería atacar el enemigo pero si aumenta su contador de habilidad.
- Cuando se utilizan habilidades especiales excepto la de la curandera, no debería atacar el enemigo pero si aumenta su contador de habilidad.
- Los estados se aplican de manera correcta.
- Los estados se cambian de manera correcta.
- Cada estado limita lo que tiene que limitar.
- Se reinician correctamente los estados y estadísticas al final de cada turno.
- Cuando la vida de un personaje baje de cero, que no marque que tiene -x vida, sino que avise de que ha muerto.
- Las curaciones no deberían sumarle vida a los muertos.
- Saltan adecuadamente los golpes críticos y sus consecuencias.
- Las habilidades se activan en el turno correcto, y se lanzan bien.
- Al terminar el bucle marca si has ganado o perdido dependiendo del resultado.
- Los textos se imprimen de manera correcta y en el momento adecuado.

Estos casos se han ido probando lanzando el programa principal, ya que cada cambio provocado en los atributos y demás es impreso por pantalla. Para ello, se han trucado las estadísticas de los héroes para provocar estas situaciones y se ha corregido el error donde debía.

6. Excepciones

Solamente ha sido implementada una excepción. Esta ocurre en la clase Instrucciones a la hora de abrir un fichero de texto. Esto lo encapsulamos en un try catch ya que puede ocurrir que no encuentre el fichero, y por eso mismo salte la excepción.

7. Conclusiones

Principalmente podemos decir que nuestros objetivos han sido cumplidos. Queríamos crear un juego por turnos, en el que el jugador tuviera diferentes elecciones a la hora de realizar su turno. Y realmente, ha quedado mejor de lo que nos esperábamos.

Ahí es donde vemos nuestros puntos fuertes; por muy complejo que se estuviera volviendo siempre hemos ido más allá y gracias a eso hemos aprendido a programar/diseñar cosas que no se han impartido en la asignatura. Además, hemos conseguido trabajar juntos, a todos se nos han ido ocurriendo ideas y ninguno ha quedado fuera en ningún momento.

Respecto al trabajo individual aquí va la valoración personal de cada uno:

- Gorka: he trabajado sin parar para que el proyecto fuera hacia delante, me atribuyo el logro de conseguir unir al grupo y dirigirlo por un camino a la hora de diseñar. Además, he conseguido aprender mucho más por mi cuenta haciendo el proyecto que con los laboratorios programados para clase. Si tuviera que destacar algo malo, sería el hecho de que me he preocupado poco de preparar los primeros diseños y hemos tenido que hacerlo todo del tirón al final.
- Unai: gracias al proyecto he conseguido entender a la perfección como hacer un diseño correcto de un programa, lo que me ha ayudado también para el desarrollo de la asignatura. Ciertamente es que he andado un poco perdido a la hora de trabajar, pero mis compañeros me han ayudado y he podido estar un poco más centrado llevando el proyecto adelante.
- Iker: muchos de los casos extremos han sido identificados por mí, y gracias a eso se ha podido corregir el diseño/implementación de manera adecuada. Además, he aprendido ciertas cosas que no sabía que se podían hacer y me vienen bien de cara a los exámenes. Eso sí, he sido bastante pasota con el proyecto, y mis compañeros han tenido que tirar de mí para que participara.

Si es cierto que hay cosas a mejorar. Principalmente que lo hemos dejado bastante de lado y lo hemos hecho todo al final, provocando así que no pudiéramos recibir buena ayuda en las entregas de diseño anteriores. También deberíamos añadir que a la hora de repartir el trabajo ha recaído más trabajo sobre uno que sobre otros, además cuando ha habido problemas a la hora de diseñar/programar había participantes que no se interesaban por corregirlo. De todas maneras, al final conseguimos ponernos de acuerdo para sacar todo adelante.

Creemos que esos problemas que tuvimos debieron haber sido abordados de otra manera; como por ejemplo solucionar las cosas al momento para no provocar arrastrar todo el trabajo y atrasar las entregas.

Tras la defensa de nuestra presentación nos damos cuenta de varias cosas. Primero, comprendemos que el diseño es considerablemente más importante que la implementación, y hemos pecado de utilizarla. También vemos que hay que revisar varias veces el video antes de entregarlo para que no pase algo que no debería pasar. Aun así, por lo que vimos quedó bastante claro el funcionamiento, ya que las preguntas fueron más acerca sobre situaciones extremas o errores a la hora de redactar la leyenda del diagrama de secuencia.

Si pudiéramos corregir algo acerca del diseño serían dos cosas principalmente. La primera es implementar métodos que llenen las listas MAE ellos mismos, en vez de que lo haga el método `jugarUnaPartida()`. Por último, podríamos haberlo diseñado de otra manera. Ya que la clase villano solo tiene una clase hija, no era necesario hacerla, sino que directamente con hacer la clase villano bastaba. Esto ha quedado así por si en algún futuro se quisiera actualizar con algún villano nuevo.