



BATALLAS IV

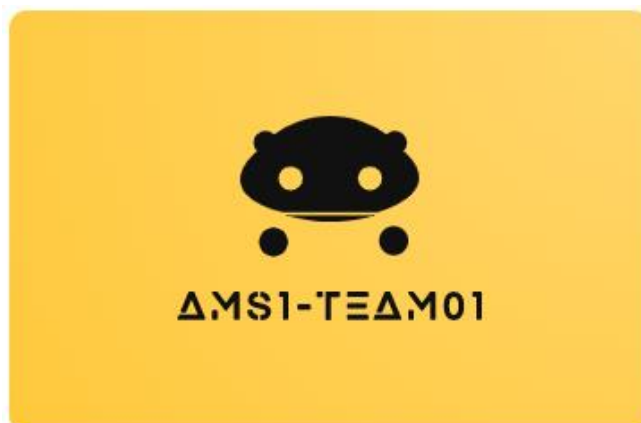
PROYECTO AMS1

by TEAM01 AMS1 | info@ams1team01.com



INDICE

Resumen	3
Tecnologia Aplicada	4
Incidencias técnicas y resolución.....	5
Clases contenedor.....	5
Insert & Update BBDD.....	5
Método de defensa.....	6
Sistema de puntos.....	6
Sistema de turnos.....	7
Interfaz gráfica.....	7
Restricciones de red.....	7
¿Qué es lo siguiente?.....	8
Valoración personal del Proyecto.....	9



RESUMEN

GENERAL

Este proyecto se ha realizado como proyecto de fin de curso de AMS1, basado en unas especificaciones mínimas requeridas.

Se trata de un juego de lucha con GUI estilo máquina de arcade, donde el usuario podrá elegir entre un plantel de 9 personajes clasificados en 3 razas, con estadísticas diferenciadas, y sus armas disponibles. El usuario jugará tantos combates como quiera, hasta que pierda o salga de la partida, y generará un registro en el ranking.

Los datos de Personajes, Arma y Razas se pre-cargan en la aplicación desde una base de datos, y la información de Jugador, Batallas y Ranking se cargan a la base de datos durante la ejecución del juego, permitiendo acceder a esa información de forma persistente y guardar el ranking entre partidas y sesiones de juego.

AMS1 - TEAM01

Es un equipo de programadores que se han unido bajo la bandera de AMS1 - Team01 para desarrollar aplicaciones y aprender de la experiencia de trabajar en equipo al gestionar proyectos conjuntos y complejos, que abarquen los diferentes módulos de aprendizaje que se cursan a lo largo del curso AMS1.

Está formado por:

Pol Tell	–	poltelljove@gmail.com https://www.linkedin.com/in/pol-tell-jove/ https://github.com/PolTellJove
Jose Rosillo	–	jrosillonieto@gmail.com https://www.linkedin.com/in/joserosillonieto/ https://github.com/jrosillonieto
Iwo Przybyszewski	–	iwoskater3@hotmail.com https://github.com/Arzzet https://www.linkedin.com/in/iwo-przybyszewski-a38433b6/



TECNOLOGIA APLICADA



DESARROLLO

- JavaSE-15 (IDE Eclipse)
- MySQL Workbench



DOCUMENTACION

- Microsoft Office
- Google Docs
- Draw.io



VERSION CONTROL

- GitHub
- Google Drive



draw.io

TEAM COORDINATION

- Trello
- Discord



INCIDENCIAS TÉCNICAS Y RESOLUCIÓN

CLASES CONTENEDOR

En un principio nos había resultado difícil crear las clases contenedoras de Guerreros y Armas que aparecieran en el juego. El concepto de pasar una clase como parámetro ha sido un reto para que salga bien y funcione como se pretendía, pero nos ha permitido crear un método para importar la información de los atributos de los Guerreros y Armas desde la BBDD, y crear los ArrayList de Objetos ya listos para ser usados en el juego.

INSERT & UPDATE BBDD

En la fase media del desarrollo de esta Alfa, una vez creado el loop de juego, dado que es vital que se almacene la información de las batallas, se ha tenido que plantear cómo y en qué momento iba a ocurrir la inserción de datos. Considerando que era poco eficiente guardar datos e irlos actualizando durante la ejecución del juego para cargarlos todos de golpe posteriormente, puesto que habría que crear variables para ello, y posiblemente, si se acumulaban muchas partidas, un insert masivo podría provocar errores y un consumo de recursos mayor, además de que los datos no serían actualizados en “tiempo real”, se ha optado por la solución de realizar un insert con los datos tras cada lucha. Si bien es cierto que se podría actualizar esa información tras cada turno para que el concepto de actualización en tiempo real fuera más fiel a su significado, el uso de recursos y procedimientos se multiplicaba. Es por eso que la siempre elegante e intermedia opción ha sido la elegida en este caso. Tan solo era necesario crear una variable que guardase los puntos totales y sumase los puntos de batalla en cada ocasión, para reflejar la puntuación final en el ranking.

Sin embargo esto planteaba un pequeño problema, ya que tras cada lucha se añadía un nuevo registro con la puntuación actualizada. Era necesario entonces, añadir una función update a la Clase que contenía los métodos de conexión. Establecimos una condición que determina que tras la primera lucha se haga un insert, y a partir de la segunda, un update recuperando el ID de jugador actual.

MÉTODO DE DEFENSA

Si bien el juego tiene un punto de aleatoriedad suficiente al poder fallar el propio ataque, poder defenderse del ataque enemigo e incluso poder atacar de nuevo si el personaje atacante tiene más velocidad, el hecho que el daño sea siempre el mismo hacia la resolución del combate algo aburrida y dejaba en desventaja a aquellos personajes que podían hacer menos daño contra los que tenían más defensa.

Hemos hecho un pequeño ajuste en la característica de defensa de manera que lo que antes era un número fijo, pasa ahora a ser un máximo de rango. De esta manera un personaje que antes defendía siempre 5 puntos de vida, ahora defenderá hasta 5 puntos de vida, siendo el número exacto calculado de manera aleatoria en cada ataque.

Este cambio, a priori hace que las luchas sean un poco más cortas, dado que a menudo se hará un poco más de daño que el que se haría con el valor completo, pero permite que los personajes que hacen poco daño tengan más opciones en proporción a los que hacen mucho, que se beneficiaran de este cambio pero en menor medida y haciendo que el seguimiento de combate y los puntos de daño realizados sean menos monótonos.

SISTEMA DE PUNTOS

Del mismo modo, el problema anterior se veía reflejado en el sistema de puntos, permitiendo que los personajes que hacían más daño y tenían mayor defensa, ocupasen los puestos más altos del ranking.

Para acompañar el cambio anterior, se ha reescrito el sistema de puntos de la siguiente manera: Si antes los puntos ganados tras la lucha correspondían a la vida restante, ahora los puntos ganados tras la batalla se calcularían sumando la vida restante y el daño infligido.

De esta manera, también se aborda otro problema de guardar muchos registros con 0 puntos en el ranking si se perdía la primera lucha.

Por razones de diseño y dar un poco de variedad a la puntuación se multiplica la puntuación por 100 para obtener cifras más altas, y se resta el daño recibido para que las puntuaciones no sean siempre números redondeados a la centésima, minimizando registros de empate.

Por tanto la puntuación de batalla actual se calcula así:

$((\text{vida restante} + \text{daño infligido}) \times 100) - \text{daño recibido}$

A
M
S
1
-
T
I
-
-
A
M
O
1

SISTEMA DE TURNOS

En un inicio se había planteado un sistema de turnos de intercambio de roles de atacante y defensor, pero rápidamente detectamos que no era un planteamiento correcto, ya que generaba mucha confusión al intercambiar las posiciones de los prints por consola de vida restante, por poner un ejemplo.

Además había que tener en cuenta que el orden del ataque y la defensa tampoco podía corresponder a un turno, ya que los guerreros tienen la opción de volver a atacar de nuevo en el siguiente turno.

Es por eso que se ha vinculado el método de cambio de atacante y defensor a un ORDEN que es independiente del turno. Así, el turno avanza cada vez que se intenta un ataque, pero el orden solo avanza cuando se cumple la condición de cambio.

INTERFAZ GRÁFICA

Se han generado problemas a la hora de juntar código de ambas partes, la lógica y la encargada de gestionar la interfaz, llegándose a ejecutar código comentado he incluso eliminado, y/o teniendo que rehacer el código de nuevo en otro proyecto.

Algunas definiciones de atributos como las de player o bot se han tenido que insertar en los Listeners, porque al ejecutar el programa no encontraba estos registros, ya que aún no se habían declarado. Por lo tanto se han tenido que crear al ejecutarse estos eventos al momento de clicar un botón, por ejemplo. Para evitar que el usuario introdujera un tipo de argumento en el panel del nombre de usuario, se ha utilizado regex para un control menos permisivo.

Se ha controlado que al cerrar la ventana del JOOptionPane al finalizar un combate no se rompiera el programa, sino que nos enviara a la pantalla de inicio. La introducción en las tablas de ranking sólo se podía realizar con `Object[][]`, me ha generado complicaciones para poder almacenarlo en este formato. Finalmente, se ha utilizado un Resultset con un foreach y se han añadido individualmente cada fila con un `new Object[]`.

RESTRICCIONES DE RED

Un pequeño inconveniente pero muy recurrente ha sido la dificultad para consultar información y buscar ayuda en la red desde el instituto, debido a las restricciones de red. En ocasiones se ha tenido que tirar de datos o espear a consultar cosas en casa.

Nuestro código no cierra puertas a futuras adiciones.

El código fuente es modular para facilitar la integración de nuevas funciones.



¿QUÉ ES LO SIGUIENTE?

La aplicación se ha desarrollado contemplando futuras ampliaciones de funcionalidades, contenido y opciones.

En esta fase temprana del desarrollo, se incluyen las funciones básicas requeridas para su correcto funcionamiento, pero el source code ya se ha planteado de manera que se puedan implementar fácilmente las nuevas funcionalidades en próximas versiones, teniendo incluso código listo para cuando haya que incluir nuevas opciones.

La versión Beta es el siguiente paso, y se espera que contenga las siguientes funcionalidades.

- ✓ Usuarios persistentes y log-in
- ✓ Botón para pasar varias rondas
- ✓ Navegar por el ranking
- ✓ Filtrar ranking por nombre/usuario
- ✓ Mejora del sistema de combate
- ✓ Opciones de ataque / críticos
- ✓ Mejoras en el diseño gráfico
- ✓ Jefes finales
- ✓ Nuevas Razas/personajes/armas
- ✓ Instalador
- ✓ Página web del producto.
- ✓ Integración de sonido
- ✓ Estamos promocionando la aplicación para que este disponible en las principales plataformas tales como AppStore y GooglePlay



VALORACIÓN PERSONAL DEL PROYECTO

Este proyecto ha sido una muy buena experiencia en comparación con el anterior. Ha estado todo bien definido desde el primer momento, lo que nos ha permitido plantearnos su funcionamiento y organizarnos mejor. También hemos podido conectar lo que hemos aprendido en los distintos módulos del curso, y hacer un caso de uso real de una base de datos en una aplicación.

Ha sido muy empocionante también poder empezar a trabajar en un proyecto fuera de la consola, usando interfaz gráfica. Si bien hubiésemos agradecido poder tener 2 semanas completas en lugar de 10-11 días, ha sido una experiencia muy positiva que nos ha permitido crear las cosas a nuestra manera y plantearnos diversas soluciones frente a los problemas que han ido surgiendo, los requisitos y nuestros propios objetivos.

Nos hemos encontrado buscando soluciones y herramientas por nuestra cuenta para aprender más o mejorar en las cosas que ya habíamos aprendido y cada paso que hemos dado en la dirección correcta ha sido muy satisfactorio.

Por último, hemos tenido más herramientas a nuestra disposición en cuanto a aprendizaje y menos limitaciones en cuanto a conocimientos.

Todo esto, nos ha dado la oportunidad de disfrutar más de este proyecto.

Valoracion media del grupo

Proyecto 01

Proyecto 03

Planificación	2.1	3.9
Aprendizaje	3.6	4.4
Herramientas disponibles	2.6	4.2
Cohesión entre módulos	2	4.5
Libre albedrio	1.9	4
Disfrute	3.2	4
Dificultad	4.3	3.4

