IES Francisco de los Ríos

Ciclo Formativo de Grado Superior

Desarrollo de Aplicaciones Multiplataforma

Proyecto Integrado

Endless Dungeon

Autor: José Ángel Marín Moreno

Tutor: Carlos Serrano

Abril, 2020

Contenido

[1. Descripción del problema 3](#_Toc99452467)

[2. Objetivos del proyecto 3](#_Toc99452468)

[3. Recursos necesarios 3](#_Toc99452469)

[3.1. Recursos de desarrollo 3](#_Toc99452470)

[3.2. Recursos de producción 3](#_Toc99452471)

[4. Planificación temporal 3](#_Toc99452472)

[5. Desarrollo del proyecto 4](#_Toc99452473)

[5.1. Requisitos de la aplicación 4](#_Toc99452474)

[5.2. Diseño de la aplicación 4](#_Toc99452475)

[5.2.1. Diseño de la arquitectura 4](#_Toc99452476)

[5.2.2. Diseño de datos 4](#_Toc99452477)

[6. Manual de instalación 5](#_Toc99452478)

[7. Manual de usuario 5](#_Toc99452479)

[8. Valoraciones y conclusión 5](#_Toc99452480)

[8.1. Evaluación del grado de cumplimento de los objetivos y finalidad 5](#_Toc99452481)

[8.2. Evaluación de la planificación temporal y de la toma de decisiones 5](#_Toc99452482)

[8.3. Posibles mejoras a la solución 5](#_Toc99452483)

[9. Bibliografía 5](#_Toc99452484)

[10. Anexos 5](#_Toc99452485)

[10.1. Código fuente de la aplicación 5](#_Toc99452486)

[10.2. Otros 5](#_Toc99452487)

# 1. Descripción del problema

El proyecto consiste en el desarrollo de un juego rpg donde se mezclaran las mecánicas del combate por turnos y juegos de cartas.

# 2. Objetivos del proyecto

-Desarrollar 3 niveles divididos cada uno en 3 etapas

-Implementar la mecánica de combate por turnos en equipo

-Implementar la mecánica de cartas aleatorias para cada héroe

# 3. Recursos necesarios

# 3.1. Recursos de desarrollo

Se va a utilizar el lenguaje C# para desarrollar los scripts necesarios para que funcione el programa.

# 3.2. Recursos de producción

Para desarrollar el juego se usara el programa de Unity y visual estudio core para desarrollar el código

# 4. Planificación temporal

|  |  |
| --- | --- |
| Semana | Objetivos |
| 21/03/2022----27/03/2022 | Diseño de sprites & desarrollo de mecánica de combate |
| 28/03/2022----01/04/2022 | Desarrollo de combate sin cartas |
| 04/04/2022----10/04/2022 | Implementación de cartas |
| 11/04/2022----17/04/2022 | Implementación de cartas |
| 18/04/2022----24/04/2022 | Desarrollo del nivel 1 |
| 25/04/2022----01/05/2022 | || |
| 02/05/2022----08/05/2022 | Desarrollo del nivel 2 |
| 09/05/2022----15/05/2022 | || |
| 16/05/2022----22/05/2022 | Desarrollo del nivel 3 |
| 23/05/2022----29/05/2022 | || |
| 30/05/2022----05/06/2022 | Corrección de bugs |
| 06/06/2022----12/06/2022 | Documentación proyecto |

# 5. Desarrollo del proyecto

-Desarrollar 3 niveles divididos cada uno en 3 etapas:

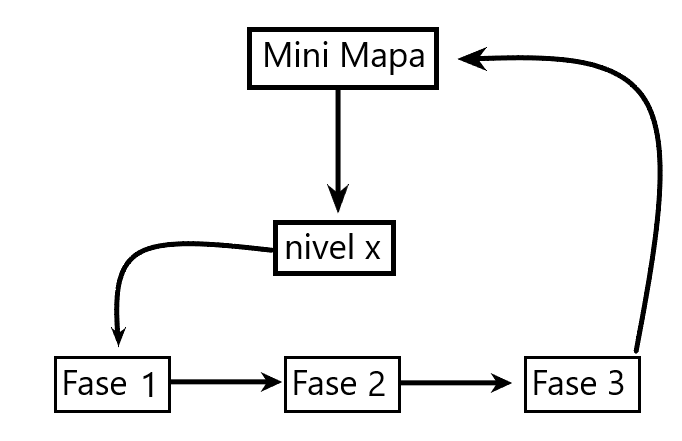
-Para desarrollar los 3 niveles que conformaran la demo del producto final primero se desarrollara un prototipo de combate en donde se aplicaran las mecánicas de combate por turnos y las mecánicas de cartas de habilidad que substituirán los movimientos de los héroes en los juegos clásicos de combate por turno.

-Implementar la mecánica de combate por turnos en equipo:

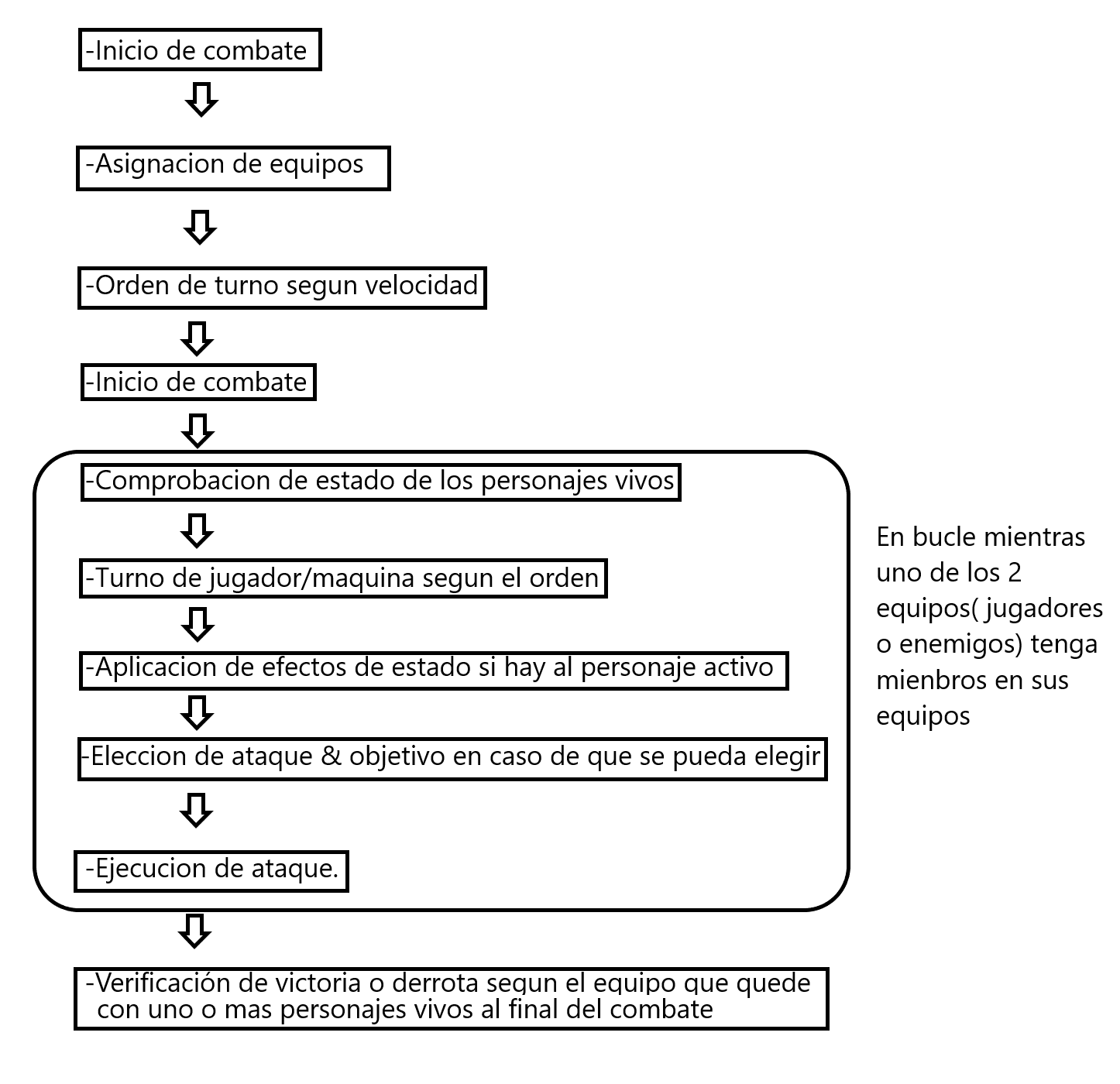
-Implementar la mecánica de cartas aleatorias para cada héroe

# 5.1. Requisitos de la aplicación

-**Desarrollo de nivel:** Los niveles están diseñados cada uno como una combinación de 3 escenas las cuales una vez eliminados todos los enemigos de una escena se cambia de escena y cargando los enemigos de la escena, siguiendo el siguiente patrón:



-**Mecánica de combate por turno:**



-**Mecánica de cartas aleatorias:** Cada jugador tiene una lista de objetos los cuales tienen distintos scripts los cuales contienen las características del tipo de ataque se realiza, desde ataques de daño simple o ataques que solo mejoran o bajan estadísticas, hasta ataques que hacen que una entidad pierda turnos o sufra daño cada inicio de turno. Cada script tiene en común un nombre, descripción y una imagen de fondo, las cuales se usan para aplicar a los botones de acción el fondo, nombre y descripción de la carta, simulando que son cartas reales. Para aplicar la mecánica aleatoria existe en código dos controladores para que en caso de tener mas de 4 ataques se elijan aleatoriamente cada inicio de turno, pudiendo repetirse el mismo ataque para emular que dentro de la baraja de cada jugador existen varias copias de la misma carta, como sucede en mazos de juegos de cartas reales, los cuales han servido de inspiración. (Véase magic the gathering, munchkin, etc)

# 5.2. Diseño de la aplicación

# 5.2.1. Diseño de la arquitectura

# 5.2.2. Diseño de datos

En el proyecto se usaran distintos tipos de carpetas para guardar los distintos elementos:

-Assets: Guardara las siguientes carpetas:

-Prefabs: En esta carpeta se guardaran todos los objetos usados varias veces en el programa.

-Scenes: En esta carpeta se guardaran las distintas pantallas que se usaran en el juego.

-Scripts: En esta carpeta se guardaran todos los scripts de codificación del juego, dividiéndose entre los destinados al combate, peleadores, habilidades, estados de condición y controladores de la pantalla.

-Sprites: En esta carpeta se guardaran los sprites de los objetos usados en la carpeta prefab.

-Anim: En esta carpeta se guardan las animaciones de los objetos usados en la carpeta prefabs, dividiéndose según son enemigos o heroes

# 6. Manual de instalación

# 7. Manual de usuario

# 8. Valoraciones y conclusión

# 8.1. Evaluación del grado de cumplimento de los objetivos y finalidad

# 8.2. Evaluación de la planificación temporal y de la toma de decisiones

# 8.3. Posibles mejoras a la solución

# 9. Bibliografía

# 10. Anexos

# 10.1. Código fuente de la aplicación

# 10.2. Otros