

# Proceso de producción de un videojuego: Berserk 2D

Iker Roa Ameigidez

**Resum**— El concepto del proyecto es trasladar el primer capítulo del manga Berserk, de Kentaro Miura, a un nivel en 2D con estética pixelart centrado en el combate “con un estilo soulslike” y metroidvania con una mentalidad concreta: “Que tenga un acabado lo más profesional posible para vivir en todos los aspectos, tanto los artísticos como la lógica, del proceso de desarrollo de un videojuego”.

Las secciones en las que se dividirá el nivel serán 3: La zona inicial que servirá para familiarizarte con los controles enfrentándote al enemigo que custodia la celda. La salida del castillo y llegada a la aldea en la que te enfrentarás a más enemigos de tipo base y arqueros. Y la última que será un enfrentamiento contra un jefe final. Cabe aclarar que la primera fase y la tercera son las más “inamovibles” en cuanto a la escala y lo que se quiere lograr, aunque las 3 fases crecerán de forma paulatina en complejidad hasta la fase final del desarrollo.

Para llevarlo a cabo, se realizaría el desarrollo en el motor gráfico Unity, que utiliza lenguaje C#, para ensamblar la lógica del nivel. Para crear todos los aspectos “artísticos” del nivel se hará uso de la aplicación Procreate disponible en iPad. Cabe mencionar que para agilizar algunos procesos, se hará uso de assets libres de derechos, pudiendo incluso modificarlos para las necesidades de la producción.

**Paraules clau**—Berserk, pixelart, 2D, C#, Unity, videojuego, soulslike, metroidvania

**Abstract**—The concept of the project is to transfer the first chapter of the manga Berserk, by Kentaro Miura, to a 2D level with pixelart aesthetics focused on combat “on a soulslike style” and metroidvania with a specific mentality: “To have the most professional finish possible to live in all aspects, both artistic and logical, of the development process of a videogame”.

The level will be divided into 3 sections: The initial area, which will serve to familiarise yourself with the controls by confronting the enemy guarding the cell. The exit from the castle and arrival at the village where you will face more base-type enemies and archers. Confrontation against a final boss. It should be noted that the first and third phases are the most “immovable” in terms of scale and what you want to achieve, although the 3 phases will gradually grow in complexity until the final phase of development.

To carry it out, the development would be done in the Unity graphic engine, which uses C# language, to assemble the logic of the level. To create all the “artistic” aspects of the level, use will be made of the Procreate application available on iPad. It is worth mentioning that in order to speed up some processes, royalty-free assets will be used, and they can even be modified for the needs of the production.

**Index Terms**—Berserk, pixelart, 2D, C#, Unity, videogame, soulslike, metroidvania

---

## 1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

Este proyecto surge de la idea de querer trasladar y adaptar a formato videojuego en 2D, de la forma más digna posible, el primer capítulo del manga realizado por Kentaro Miura: Berserk.

## 2 PROYECTO - LA SOLUCIÓN A IMPLEMENTAR

El objetivo, como ya se ha mencionado, es realizar un nivel al más alto nivel en el tiempo dedicado al proyecto (300 horas), este proceso de producción abarca cinemáticas animadas en nivel, diseños artísticos y a nivel de programación del jugador, enemigos y jefe final, y aspecto y comportamiento del mundo del nivel.

Se ha decidido desarrollar este videojuego en el motor Unity, por ser el ideal para el género y perspectiva 2D que tendrá el juego (el motor Unreal Engine de Epic se descartó por ser uno claramente enfocado al desarrollo en 3D), además de que dentro de este tipo de motores en 2D, es el que cuenta con una mayor comunidad a la hora de formarse y realizar consultas, además de que sus desarrolladores están constantemente realizando actualizaciones y facilidades a sus usuarios.

En consecuencia, a la hora de programar la lógica del nivel, se utilizará C# dentro del IDE Visual Studio Code, por la familiaridad que ya se tiene con este editor de código y la variedad de extensiones con las que cuenta para enriquecerlo aún más.

### 3 METODOLOGIA

El desarrollo de un videojuego puede ser muy caótico: retrasos, horas extra, sofocar la creatividad del desarrollador por malas prácticas...

Tras investigar sobre metodologías de trabajo individuales que se adaptasen a la filosofía de este proyecto, se ha llegado a la conclusión de hacer uso de la metodología (adaptada al trabajo individual) Scrum.

Scrum permitirá, de forma iterativa, llegar antes y de una forma más sana al core del nivel y de cada uno de sus aspectos: las etapas tempranas cubrirían las necesidades principales de funcionalidad y aspecto, para con cada iteración y llegando a las últimas etapas, se pulirían y añadirían complementos para enriquecer la experiencia del usuario, es decir, la calidad del producto final.

Al comienzo del proyecto se realizará una lista de tareas (product backlog) a realizar que forman al proyecto, ordenadas por prioridad y con una descripción.

Se trabajará en planificar en sprint de 2 semanas de duración, en los que se especificará los aspectos en los que se trabajará en este periodo de esa lista. Al finalizar el sprint, se recogerá el trabajo realizado del planificado para analizar si se ha podido cumplir dicho sprint, y de forma satisfactoria. El trabajo faltante o revisable se desplazará al siguiente sprint.

### 4 PASOS DE IMPLEMENTACIÓN

#### 4.1 Análisis de requisitos

Detallado análisis de requisitos/funcionalidades que tendrá el producto, tanto las de aspecto como lógica del producto.

#### 4.2 Selección de tecnologías y recursos predefinidos

Escoger las tecnologías en las que se desarrollará el producto y la gran mayoría de recursos predefinidos para arrancar el proyecto (assets, efectos de sonido, plantillas de sprites... todo libre de derechos).

#### 4.3 Diseño del jugador

Desarrollar todas las funcionalidades del jugador: desplazamiento, combate, vida y resistencia. Esto comprende tanto la parte de código que las hace posibles, con su interacción con el resto de elementos del nivel (terreno, gravedad, enemigos), como la creación de los sprites necesarios para representar estas funcionalidades.

#### 4.4 Diseño de enemigos

Estos para este primer nivel tendrían dos variantes: base y arquero. Se disponen de estas dos variantes, ya que el diseño del nivel (posteriormente se volverá a explicar) sería enfocado a ser uno más de introducción a un hipotético juego completo, además de ser uno

especialmente enfocado en los "jefes finales". Se contempla también como diseño de enemigos lo mencionado en la sección del jugador a desarrollar.

#### 4.5 Diseño del nivel

Desarrollo de las tres secciones del nivel. Incluye la creación a boceto de este nivel y sus fases antes de implementarlo, creación y adquisición de todos los elementos visuales necesarios para representarlo, y desarrollo de la lógica y leyes del nivel. La complejidad del diseño del nivel será proporcional a la sección a la que pertenecería de un juego completo, en este caso es uno de introducción.

#### 4.6 Animación 2D

Realización de una cinemática previa hecha enteramente en 2D para sumergir al jugador en el contexto del nivel.

### 5 Despliegue

Compilación de todo el producto y realización de un simulacro y prueba completa del nivel.

### 6 TESTING Y VALIDACIÓN

#### 6.1 Pruebas y control de calidad

Pruebas unitarias y de integración para asegurar el cumplimiento de los requisitos. Feedback de terceros que prueben el nivel y del propio desarrollador.

#### 6.1 Pruebas de rendimiento

Los objetivos incluyen alcanzar mínimo los 60 fotogramas por segundo en equipos categorizados como gama media-baja, que el uso de recursos del equipo no superen el 85% y que la carga del nivel no supere los 10 segundos.

### BIBLIOGRAFÍA

- [1] <https://starloopstudios.com/best-agile-practices-in-game-development/>
- [2] <https://www.mountaingoatsoftware.com/presentations/agile-and-scrum-for-video-game-development>
- [3] <https://www.linkedin.com/pulse/applying-lean-methodology-game-development-zafer-elcik/>
- [4] <https://www.udemy.com/course/unitycourse/>

