

Proceso de producción de un videojuego: Berserk 2D

Iker Roa Ameigidez

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

La creación y desarrollo de un videojuego es capaz de juntar el arte en forma de animación, escenarios y paisajes creados a mano, con sistemas tan complejos como la Inteligencia Artificial de ciertos entes, funcionalidades, y leyes lógicas del entorno.

El concepto del proyecto es trasladar el primer capítulo del manga Berserk, de Kentaro Miura, titulado "El Guerrero Negro" 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".

Gracias a la ambientación de fantasía oscura y gran diseño de personajes del material original, el videojuego se verá beneficiado estéticamente, teniendo ya una identidad visual básica, y con una estructura narrativa del nivel ya definida. Además, se añaden referencias estéticas de otros juegos, tales como Blasphemous, un videojuego español del mismo género y estética que Berserk 2D, este mismo, basado en el folklore derivado de la religión y festividades en España, u otros que han sido tomados como pautas a la hora de diseñar las mecánicas de juego como Bloodborne o Dark Souls, ambos juegos desarrollados por FromSoftware, en el que el videojuego gira en torno a la recompensa de un combate exigente y meticulosamente bien diseñado.

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.

A la hora de decidir en que motor se trabajaría, había muchas posibles opciones de entrada: Unreal Engine, el motor desarrollado por Epic, se encuentra ya en su quinta iteración, siendo, probablemente a día de hoy, el motor más utilizado dentro del desarrollo de videojuegos de gran producción en 3D, especialmente. Entrando ya en los

motores especialmente centrados y populares en el desarrollo en 2D, encontramos 3 motores: GameMaker, Godot y Unity. Aunque con los 3 motores las posibilidades de entregar un gran producto son elevadas, hay algo con lo que ninguno de los dos primeros motores puede competir contra Unity: Comunidad. Ya sea la que forman desarrolladores, que han ido entregando grandes juegos como Hollow Knight, Cuphead, o el propio mencionado Blasphemous, también existe una gran comunidad "amateur" o centrada en la educación y formación del motor.

Es por esto, y por utilizar un lenguaje tan amigable como es C# a la hora de programar la lógica del nivel, que se realizará el desarrollo en el motor gráfico Unity.

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.

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 OBJETIVOS DEL PROYECTO

	TAREA	SPRINT
1	Investigación y documentación sobre productos similares e inspiraciones externas	1
2	Formación sobre Unity / C# / IA	1
3	Formación Animación / Sprites y Tiles 2D	1
4	Realizar Sprites Jugador	2
5	Realizar Sprites Boss Final	2
6	Pixelart (Tiles) del Escenario y fondo	2
7	Sprites Enemigo Base y Arquero	3
8	Diseñar nivel	3
9	Programar movimiento del jugador (despl. horizontal, salto, rodar)	4
10	Programar ataques del Jugador (ataque débil, fuerte y especial)	4
11	Programar IA Enemigos: Base, Arquero y Final	5
12	Diseñar interfaz	5
13	Añadir partículas al nivel (sangre, iluminación, tierra...)	6

14	Añadir música y efectos de sonido al nivel	6
15	Diseñar menú inicial	6
16	Animar cinemática Inicial 2D	7
17	Diseñar pantalla Final Partida (Victoria/Derrota)	7

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.

3.1 Jira: La herramienta para la planificación/seguimiento del proyecto

Para planificar los sprints y tener controladas las tareas realizadas, realizándose y próximas a ser realizadas se utilizará la herramienta Jira

Jira es una herramienta de gestión de proyectos flexible y poderosa que ofrece seguimiento visual del progreso, colaboración efectiva, integraciones con herramientas populares, gestión avanzada de problemas y escalabilidad

para adaptarse a cualquier tamaño de equipo o proyecto. Y es que Jira ofrece una gran variedad de formas de representar/organizar las tareas:

En forma de cronograma o diagrama de Gantt, Backlog, con etiquetas de cada tarea/objetivo según la prioridad o un tablero que muestra las tareas por hacer, las que están en curso, y las finalizadas.

4 PASOS DE IMPLEMENTACIÓN

4.X Formacion

4.Inpiracion/referencias y productos similares

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 120 segundos.

BIBLIOGRAFÍA

- [1] "BEST AGILE PRACTICES IN GAME DEVELOPMENT", STARLOOP STUDIOS A MAGIC MEDIA COMPANY, <https://starloopstudios.com/best-agile-practices-in-game-development/>
- [2] MIKE COHN, "AGILE AND SCRUM FOR VIDEO GAME DEVELOPMENT", MOUNTAIN GOAT SOFTWARE, <https://www.mountangoatsoftware.com/presentations/agile-and-scrum-for-video-game-development>, 16 DE JUNIO DE 2012
- [3] ZAFER ELCIK, "APPLYING THE LEAN METHODOLOGY TO GAME DEVELOPMENT", <https://www.linkedin.com/pulse/applying-lean-methodology-game-development-zafer-elcik/>, 18 DE MARZO DE 2019
- [4] RICK DAVIDSON, GARY PETTIE, "COMPLETE C# UNITY GAME DEVELOPER 2D", GAMEDEV.TV TEAM, <https://www.udemy.com/course/unitycourse/>, JUNIO DE 2023
- [5] "HECHO CON GAME MAKER", GAME MAKER™, <https://gamemaker.io/es/showcase>
- [6] "FUNCIONES DE GAME MAKER", GAME MAKER™, <https://gamemaker.io/es/features>
- [7] GODOT FOUNDATION, "WHY GODOT IS RIGHT FOR YOU" <https://godotengine.org/features/>

- **E-mail de contacte:** iker.roa14@gmail.com
- **Menció realitzada:** Enginyeria del Software
- **Treball tutoritzat per:** Coen Antens (Àrea de Ciències de la Computació i Intel·ligència Artificial)

ANEXO: PLANIFICACIÓN - DIAGRAMA DE GANTT

