

TP en grupo

Cachuflito Games Inc está al borde de la ruina! (de nuevo...)

Aún así el CEO ha logrado conseguir financiación de un inversor caritativo (su madre) para realizar un nuevo proyecto que ponga a Cachuflito Games Inc de una vez por todas a flote. Teniendo en cuenta que, a esta altura, en la compañía solo quedan unos pocos programadores al CEO se le ocurrió reusar proyectos anteriores para acelerar el desarrollo. Para esto ha decidido rescatar del polvo un proyecto de StarMap que la empresa había hecho unos meses atrás.

Como no quedan game designers en la empresa el CEO ha decidido que desarrollemos el juego que más nos guste pero respetando ciertas delineaciones:

Delineaciones Principales

Splash Screens

Debe haber un splash screen del logo/nombre de la compañía seguido de un splash screen del logo/nombre del juego.

Menu Inicial

Debe haber un menú inicial con la posibilidad de empezar el juego, controlar el estado del sonido y la música y mostrar los créditos. Los créditos deben mostrar los integrantes del equipo y además referencias al arte usado en el juego.

Starmap (selector de nivel)

El CEO ha decidido darnos un video de referencia de **más o menos** (sus palabras) la funcionalidad que quiere que tenga el starmap.

<https://www.youtube.com/watch?v=glT0zrSed1A>

Debemos reutilizar lo que podamos del starmap mencionado pero este debe tener las siguientes funcionalidades:

Universo random

El universo se crea randomicamente al empezar el juego.

Nave

La nave se mueve con los controles del teclado. Esta tiene una cantidad de combustible. El CEO ha dejado a nuestra disposición ver qué hacer cuando el combustible se agota...

Zoom al planeta

Cuando la nave se acerca a un planeta tenemos la opción de hacer zoom al este y mostrar información del planeta. Esta pantalla debe tener un botón para aterrizar en el planeta dependiendo de si el planeta es aterrizable o no.

Aterrizar en el planeta

Cuando se aterriza en un planeta se carga nivel de juego referido a ese planeta.

Nuestro CEO de alguna manera se ha referido al Starmap como un **selector de nivel**.

Tipo de juego al aterrizar al planeta

El Juego debe ser del estilo de juegos como Diablo, Final Fantasy. Básicamente un juego en tercera persona con la cámara alejada.

Ejemplos dados por el CEO:

Diablo (Action RPG): el juego es de acción RPG. Se controla con el mouse en un estilo point and click.

Final Fantasy (Turn-based battle system): se encuentra a un enemigo y pasa a una pantalla de pelea por turnos.

God Of War(hack and slash): juego plagado de acción con muchos enemigos recibiendo golpes de nuestro personaje.

Cada planeta al que podamos aterrizar tendrá su terreno diferente dependiendo del estilo del planeta.

El juego

El tipo de juego elegido debe contar con los siguientes sistemas:

Personaje

El personaje debe tener ciertas stats que modificarán sus capacidades: velocidad, daño, etc.

El personaje debe tener las animaciones pertinentes a sus acciones.

Sistema de combate

Sistema similar a los ejemplos de juegos citados.

Sistema de Items e Inventario

El juego debe contar con un sistema de Inventario. Estos ítems influirán en los stats de nuestro personaje (ataque, daño recibido, etc).

El inventario se accederá a través de la tecla "I" del teclado y debe mostrar items equipados e ítems que estemos cargando con nosotros que no estén equipados.

Los items se consiguen al eliminar enemigos. Caerán al piso y los recogeremos con un click del mouse.

Los items se generarán randomicamente.

Enemigos

El juego debe contar con 3 tipos de enemigos diferentes con una AI básica dependiendo del tipo de juego elegido.

Los enemigos deben tener las animaciones pertinentes a sus acciones.

La IA de los enemigos debe estar programada siguiendo los principios del patrón FSM.

Menu de Pausa

Este menu debe poder:

- al abrirse el menú este debe pausar el juego de alguna manera
- modificar opciones de sonido
- salir del juego

Este menú se debe poder acceder de un botón en alguna parte de la pantalla.

Loader

El juego deberá tener una pantalla de Loading entre situaciones que cambien el estado del juego.

Ejemplos: al iniciar el juego, al aterrizar en un planeta, al entrar en modo combate si el juego es estilo Final Fantasy, etc.

Extras

El CEO ha definido ciertos features que podrían hacer que al juego le vaya mejor en ventas. Estos son:

Cambio de personajes

Nuestro personaje forma parte de una party de la cual podemos elegir con qué personaje jugar.

Cada personaje tendrá funcionalidades diferentes. Ejemplo: mago, guerrero, etc.

Cada personaje tendrá sus stats e inventario propios.

Sistema de experiencia

Nuestros personajes podrán subir de nivel dependiendo de los enemigos que eliminen. Esta subida de nivel modificará sus stats de alguna manera.

Save/Load

El juego tiene un sistema de loading y saving. El loading se dará desde la pantalla inicial del juego. El save se da en momentos particulares decididos por nosotros.

Se guardarán los datos de:

-El Universo: dónde y qué tipo de planetas están en que parte.

-PlayerData: stats, nivel de cada personaje, ítems que tengan.

Boss Fight

Algún planeta tiene un boss. "El boss debe ser impresionante" fueron las palabras del CEO.

Nota: Debe ser fácil de testear. (Que no haya que matar a todos los enemigos para que aparezca)

Código y orden del proyecto de Unity

Si al juego le va bien, nuestro CEO planea reutilizar grandes partes del proyecto y el código para futuros proyecto. Para lo cual es fundamental mantener un código legible con nombres de scripts, métodos y variables representativos de la funcionalidad de la que forman parte. Además mantener una estructura coherente del proyecto de unity.

Pulido

Fundamental para que el juego tenga buena recepción entre el publico.

Detalles y cuestiones a tener en cuenta

Van a tener a disposición ciertos assets para hacer más fácil el desarrollo del proyecto.

Son varios modelos de personajes con sus animaciones.

Estos ya están depurados para que los puedan usar y no pierdan tiempo bajando personajes y animaciones.

Los assets serán entregados en clase o a pedido del alumno por IWALP o email.

Si bien el trabajo es en grupo se debe especificar qué parte hizo cada alumno. Además, todo alumno debe saber línea por línea que hace el código escrito por sus compañeros.

Sí o sí, el código debe estar mínimamente comentado en las partes de código no interpretables a simple vista.

Cada script debe tener un comentario describiendo en qué escena y gameobject se usa.

El código no debe ser copy/paste de los ejemplos de la materia.

Debe entregarse junto con el proyecto un archivo de texto o pdf con las decisiones de diseño y el tipo de juego para asegurar la completa corrección del TP.

Texto no muy largo. Una carilla como mucho.

El proyecto debe entregarse de alguna de las siguientes formas:

- Link a repositorio Git
- Link a archivo .rar para descargar

Además, en itch.io el juego ya buildeado, para plataforma PC

El proyecto no debe contener errores de Unity al compilar, correr en editor o buildear.