Descripcion de la practica:

Este documento corresponde a la practica 1 de la asignatura Multimedia y Movil.

La practica consiste en el desarrollo de una aplicación tipo videojuego, de temática espacial, donde se aplican todos los conocimientos aprendidos durante la primera parte del curso.

El videojuego, diseñado en un entorno 2D presenta 3 niveles de dificultad, según las características implementadas.

Yo he realizado las tareas correspondientes a un videojuego de nivel 2, donde se implementan los sprites de las naves y los scripts, los movimientos de cada una de las naves, dentro de los limites del juego, los disparos, sonidos y explosiones.

El videojuego es perfectamente jugable y puedo decir que, aunque toda la creación del mismo es por momentos frustrante, además de que se necesita lidiar con la interfaz de usuario compleja d eUNITY, además de manejar múltiples simples en un entorno C#. Decía que el resultado es una aplicación interactiva que engancha al usuario desde el primer momento, donde lo mas asombroso es la capacidad casi infinita de personalizar tu propio juego.

Quiero hacer hincapié en dos aspectos relacionados con este unltimo pensamiento.

1) En primer lugar, he escogido utilizar los sprites, animaciones y sonidos que me ha proporcionado el maestro.

Esta ultima decisión está basada en la numerosa cantidad de recursos que existen hoy en día en internet. He probado diferentes spirets de naves, barcos, fondos de pantalla, mas o menos opacos, sonidos de móvil, de disparo o incluso de ambiente espacial.

Hasta incluso se puede cambiar todo el concepto del videojuego, cambiando naves por sprites d elibros y disparos por sprites de lápices(¡)

Uno de los puntos positivos de esta practica es darse cuenta que no existen, realmente, limites a la hora de diseñar un videojuego, y es mas, no existen límites a la hora d ediseñar una interacción digital entre usuario-máquina-usuario.

Esta practica me ha ayudado a entender las posibilidades de las interacciones y mas concretamente, el de un videojuego clásico.

2) Y así paso al segundo punto donde me he dado cuenta que, hoy en día, las personas que aun no han cumplido 30 años y algunas mas mayorcitas, contamos con muchísimas referencias hacia lo que debe ser un videojuego. Quizás esto se deba a mis experiencias personasles con los videojuegos desde bien pequeño. Pero quiero decir que videojuegos espaciales se han hecho desde los primeros momentos, pero el Consciente Colectivo de como debe ser un videojuego espacial está terriblemente asentado.

La parte positiva es que todos tenemos referencia hacia lo que debe ser un videojuego de naves 2D. La parte aun mas positiva, es que modificar el videojuego para hacerlo mas novedoso y transgresivo consiste en saber actuar sobre los elementos de interacción del Player. Juguemos!

NOTAS SOBRE LA INTERFAZ

Mi primera impresión sobre la interfaz Unity es pensar que cada variable de nuestro juego, está asociado a una opcion de us interfaz.

Esto no es del todo cierto porque existen los scripts con odigo fuente en C# que es el verdadero funcionamiento del videojuego.

De algún modo, la primera vez que te enfrentas a la interfaz Unity, te sientes abrumado por la cantidad de opciones. Lo primero que se aprende es que hay ciertas opciones que hay que tener bien localizadas. Por ejemplo el order in layer que establece el orden de las caparas. De esta forma si el background tiene un orden 5, pues seguramente no veremos otra cosa que no sea el fondo de l a pantalla.

También son importantes las coordenadas del objeto y su escala.

De lo primero que se aprende es que podemos importar las imágenes de los objetos, hacia una carpeta llamada sprites. Desde aquí lo añades al escenario y es entonces cuando modificas sus opciones. El trabajo queda completo cuando estamos contentos con nuestro objeto y lo añadimos a la carpeta prefab de objetos, que son objetos con características determinadas por el diseñador.

Otro factor importante es la correcta creación de scripts y lo que es mas importante, conocer la estructura de la aplicación para dotar a nuestros objetos del adecuado scrip que lo hace funcionar tal y como queremos.

Aquí es donde mas problemas he encontrado, porque añadir un cscript de nave al objeto de la nave del personaje principal es fácil. Lo difícil es conocer la aplicación que estamos diseñando para saber don decolocar los scripts que controlas las rutinas del juego.

A pesar de todo el tiempo que le he dedicado a la practica, aun encuentro problemas eventuales relacionados con objetos que no realizan las tareas que se le especifican en el código del script. La gran parte de las veces se debe a un fallo en la interfaz, normalmente si no enlazamos el script correspondiente al atributo correspondiente en el inspector de objeto, el objeto no funcionará.

Para terminar, puedo deir que el 30% de la aplicación se soporta en las opciones d interfaz de UNITY. Order in layer, bounding box is trigger o gravity de la rigid body son algunos ejemplos. Solo la experiencia del diseñador asi como el tiempo que le hemos dedicado al programa tiene la solución de los fallos habituales de nuestro programa.

Incluso personalmente hablando me he encontrado con problemas, que a pesar de la aplicación de ChatGpt o el foro de código fuente Stack Overflow no ha habido forma humana de resolverlo y solo con la reformulación del problema, reformulando los objetos y el código y resolver el problema de una forma con un enfoque totalmente distinto al usual. Cada minuto dedicado a este juego ha sido de oro. Desde luego.

NOTAS SOBRE EL CODIGO FUENTE

A continuación voy a intentar explicar como he manejado los archivos de script de lvideojuego espacial.

Lo aprendido en clase se sencillo. Creamos un script, basado en el lenguaje de programación de C#, y lo enlazamos con el objeto deseado.

El script viene por defecto con dos funciones, Start() y Update() que realiza para cada actualización de cada frame.

Al principio he seguido las instrucciones en la teoría para crear los scripts. Nave principal, naves enemigas, disparos.

Con el tiempo he descubierto qie mantener el orden con los scripts es necesario. De repente me encuentro con una carpeta scripts con al menos 7 scripts distintos para manejas hasta 5 rutinas d ejuego .

Al principio, para cada rutina tenia un script. Puedo contar algunas de las características de mi código. En algún momento se debe contar el numero de vidas del player. La solución consistía en mantener una variable llamada PlayerLives en la nave del Player y cada vez que ocurra un collide con un misil enemigo, se resta una vida. Por desgracia esto no he conseguido hacerlo funcionar aunque pienso que esta relacionado con la instanciación de objetos y su destrucción, quizás antes que el script cumpla su función. Sin embargo sí he conseguiod crear una nave fantasma del Player que no realiza la colisión con misilies enemigos y por tanto nunca se destruye. De esta forma puedo contar el numero de vidas.

CONCLUSIONES

La conclusión principal es algo que he comentado en la introducción. Cada hora que se le dedica a UNITY cuenta, y la experiencia de conocer es un grado en UNITY. De hecho, tener la experiencia de hacer funcionar el código puede ahorrarte horas de búsqueda en foros y paginas web.

Claro, la recompensa de este trabajo es un entorno, en nuestro caso, en 2D donde tenemos el control sobre todos los aspectos del programa. La diferencia de esta disciplina, es que el material digital que se crea es una interacción con el/los usuarios. Dicho de otro modo, es una película donde el creador define preguntas y respuestas para cada frame de película, dotándola de múltiples cuerdas de decisiones, que llevan a distintos efectos que tienen finalmente implicaciones en la experiencia del usuario. Al final el éxito o fracaso de una Realidd Digital consiste en el sistema de retos/recompensas que mantiene o no a nuestro usuario inmerso en una realidad digital a medida.

Mas que pensar en la nota de este ejercicio, con el trabajo que he realizado he aprendido los conceptos mas básicos del 2D, pero por encima de todo he dado el primer salto en el entorno de los videojuegos y he perdido el mierdo a la interfaz UNITY. Para crear un videojuego, no hace falta una Realidad complicada, tan solo una idea rebelde y un comportamiento capaz de llevar a cabo dicha idea.