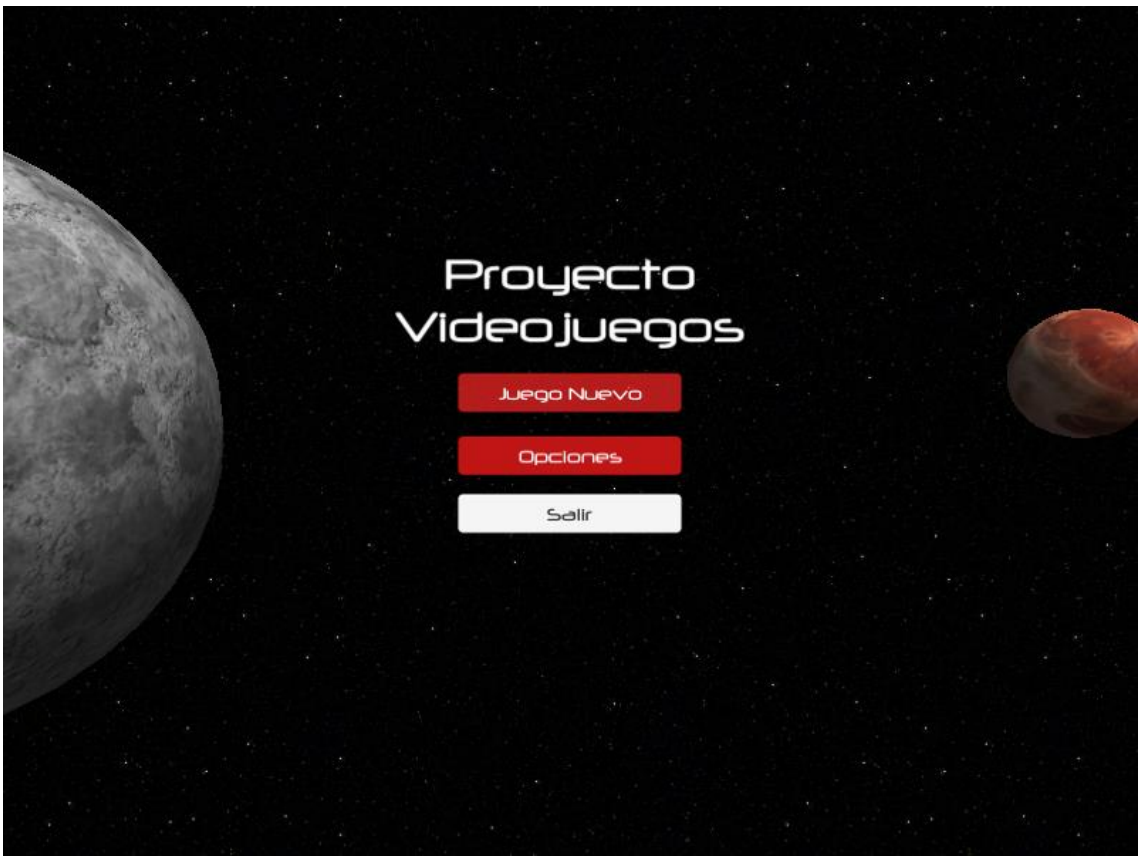

PROYECTO DE VIDEOJUEGO

Para la asignatura Desarrollo de Videojuegos

David Abad Vich
Universidad de Jaén – Curso 2014/15

Contenidos

Diseño del juego

Estilo del juego: Shoot'em up, Aventuras

Propósito y audiencia: Casual. Entretener a un público de todas las edades. Un jugador.

Mecánica de juego:

- Conjunto de reglas
- Objetivos de los jugadores:
    - En el primer nivel, hay que llegar al final del escenario sin ser tocado 3 veces por los enemigos (a la segunda vez, la nave empieza a humear) y derrotar al jefe final que aparece ahí. Hay que tener en cuenta que los puntos mejoran la velocidad de disparo.
    - En el segundo nivel, el objetivo es encontrar la casa y meterse dentro, sin caer en el agua.
    - La dificultad del juego es sencilla por lo que se requieren pocas habilidades.
    - Como entidad no jugadora cuenta con los aviones que salen para atacar al protagonista en el primer nivel.
- Interfaz gráfica: Indica los puntos que lleva en el primer nivel. Además, al ser tocado 2 veces por los aviones enemigos, la nave empieza a soltar humo como indicación de que le queda poca vida.
- Efectos de sonido:
    - Disparar con la nave.
    - "Acabar" con un enemigo.
    - Finalizar un nivel.
    - Música de fondo.
- Efectos visuales:
    - Humo en la nave al recibir varios impactos.
    - Propulsión de la nave.
- Otros comentarios:

Enlace a Google Drive con el juego:

https://drive.google.com/file/d/0B_iZoSOpFgZsYXJXV2IzX2tNY3M/view?usp=sharing

Enlace a Google Drive con el vídeo:

https://drive.google.com/file/d/0B_iZoSOpFgZsYVRaM1dMZXN1TFk/view?usp=sharing

Para el desarrollador

Para el desarrollo de los scripts, me he querido centrar en un script de control del juego de manera que si algo tiene que afectar al mundo o hacer algún efecto, mande mensaje o avise a este mismo para que ejecute las acciones pertinentes.

Este script, "*GameController*", lo posee la cámara principal del juego. Por tener diferentes elementos en los dos niveles, he requerido de un controlador secundario para el siguiente nivel.

Las pantallas de *prenivel* (aquellas que se lanzan antes de los niveles) poseen también un pequeño controlador para manejarlas y al final cada escena ha tenido que quedarse con su único controlador, pero siendo éste encargado de controlar el juego.

Nivel 1

En este nivel, la cámara se mueve a lo largo del mundo, en un eje, y teniendo como hijo al objeto jugador (que es la nave) se tiene que la nave se mueve junto a la cámara, dando la sensación de movimiento a lo largo del juego.

También hay, como hijos de la cámara, cuatro cajas de colisión que representan los límites de pantalla, de manera que cualquier disparo o enemigo que sobresalga, sea destruido antes de que pueda permanecer ocupando espacio en el mundo de juego y sobrecargando la capacidad del ordenador.

La cámara es la que posee el script de control del mundo de juego, para facilitar el uso de los hijos como límites y tener el punto de reaparición de enemigos controlado.

También hay que destacar que al crear objetos de naves enemigas no se dotan de ningún movimiento ya que con el simple movimiento de la cámara, también se simula su movimiento hacia el lado opuesto de la pantalla.

Dentro del juego existe un límite de nivel, de manera que cuando el limitador (hijo de la cámara también) llega a ese límite, da comienzo la parte del jefe final.

Los sonidos y música de fondo forman parte de diferentes *assets* de la tienda de *Unity*.

En este primer nivel no se hace uso de la gravedad, al no ser un elemento necesario para la jugabilidad ya que todo es controlado por script.

Como funcionalidad extra, la cantidad de puntos influye directamente en la velocidad de disparo.

Nivel 2

Este nivel se asemeja a la idea primera con la que hice las dos primeras prácticas, de llegar de un punto inicial a uno final y la de hacer uso de un "laberinto" de montañas para dificultar la tarea al jugador. Usando también una *Skybox* para el cielo.

El controlador usado para manejar al jugador en tercera persona es el que incorpora *Unity* como *asset* gratuito.

Este nivel es más sencillo, consta de un punto inicial y otro final, situado en una casa creada por mí mediante *Google Sketch Up*, e importada. La música de fondo también ha sido creada usando *LMMS*, un software de creación de audio digital de uso gratuito y código abierto.

Scripts del juego:

Los scripts de los que se compone el juego, alfabéticamente, son:

- Ammo.cs: Utilizado para controlar cualquier munición disparada. Controla la velocidad y dirección de la misma, además de la duración (vida) que tiene, para evitar la creación infinita de clones.
- BossController.cs: Controlador de la IA del jefe final del primer nivel. Se encarga de moverlo y hacerle disparar, además de llevar la cuenta de la vida que tiene y la que le queda.
- ChangeOptions.cs: Controla la transición entre opciones del menú principal, para facilitar el uso de la misma.
- DestroyOnContact.cs: Destruye objetos en contacto, para mantener la cantidad de objetos en juego bajo control dentro de los límites del nivel 1.
- FinJuego.cs: Asignado al punto final del nivel 2, controla cuando entra el jugador para finalizar el juego.
- GameController.cs: Controla el juego durante el nivel 1: Interfaz de usuario, creación de oleadas enemigas e inicio de la parte del jefe final cuando corresponde.
- InitPosition.cs: Posición inicial del jugador en el segundo nivel.
- LimiteInferior.cs: Destruye objetos enemigos que sobresalen por el límite inferior (para evitar la sobrecarga).
- LimiteSuperior.cs: Como el inferior, pero éste destruye la munición que sobresale por el límite superior.
- MovingCamera.cs: Mueve la cámara para dar esa sensación de movimiento.
- Prenivel1.cs y Prenivel2.cs: Controlan las fases de prenivel.
- RestartOnFall.cs: Reinicia al jugador si cae en el agua en el nivel 2.
- RotationY.cs: Rota los planetas que se ven en el menú principal.
- ShipController.cs: Controla los eventos de la nave manejada por el jugador durante el primer nivel (movimiento y disparar, además de puntos).
- SiguienteNivel.cs: Cambia al siguiente nivel mostrando texto de que se ha completado el nivel actual.
- ThirdPersonCamera y Controller: Scripts incorporados por Unity para el manejo del jugador y la cámara en el nivel 2.
- UserInterface.cs : Usado para controlar la interfaz de usuario en el menú principal.
- UserInterface2.cs: Controla el menú de pausa durante el juego.

Práctica 4 – Sistema de partículas

Están incluidos dos sistemas de partículas en la nave del primer nivel:

- El primer sistema es el que forma la propulsión base, con un *asset* de la *asset store* modificado para el tamaño y propulsión que quería.
- El segundo es el humo que sale de la nave al recibir dos impactos. Con una imagen *jpg* descargada de internet para la textura a aplicar al sistema, siguiendo un tutorial he editado el humo para que empiece poco a poco y aumente en intensidad, llegando al límite puesto, creciendo en tamaño y transparentando el color (para dar ese aspecto de que se desvanece conforme se aleja del foco).

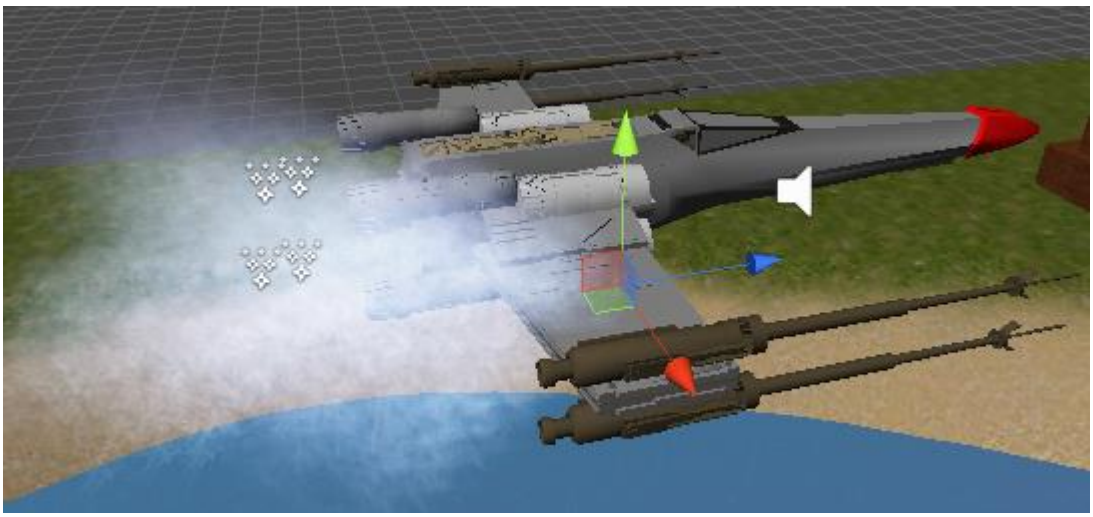

Figure 1 - Simulando humo

Práctica 5 – MakeHuman

He incluido para el segundo nivel, como personaje jugador, el modelo creado con la aplicación MakeHuman.

Para darle animación, he descargado de la *asset store* un paquete de animaciones para humanoides y he seleccionado las que he querido para el *animator* del jugador (editando algunas antes para que se repitieran o en el caso de correr, para coger la parte concreta en la que empieza a correr y corre).

Ha sido sencillo incorporar el personaje, ya que al importar el modelo se puede añadir un "*Rig*" humanoide (un esqueleto básicamente) que es la pieza clave para realizar los movimientos.

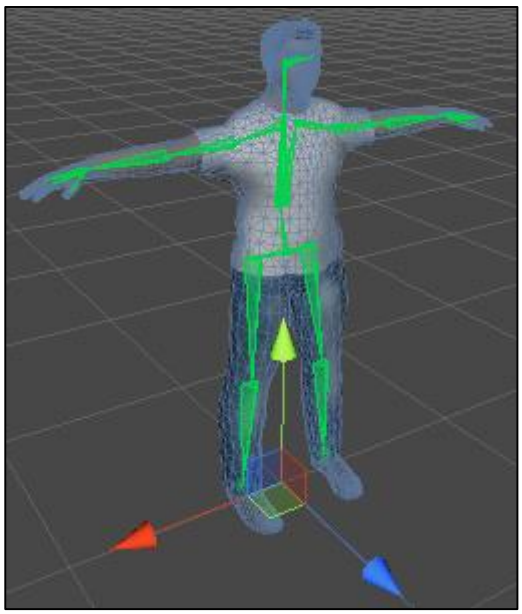

Figure 2 - Esqueleto de MakeHuman

Problemas:

- Intenté realizar el envío y recepción de mensajes, pero debido a la mala gestión del mismo, he acabado utilizando funciones públicas del script de control de juego para gestionar los eventos que ocurren.

Bibliografía y programas usados

http://docs.unity3d.com/Manual/index.html - Documentación de Unity como principal fuente de información.

https://lmms.io/ - LMMS programa de creación de audio digital.

http://www.sketchup.com/es - Herramienta gratuita de Google para la creación de modelos de edificios en 3D.

http://www.makehuman.org/ - Herramienta gratuita de modelado de personajes humanos, utilizada principalmente para realizar la práctica 5.

http://camstudio.org/ - Grabación del vídeo necesario para el proyecto.