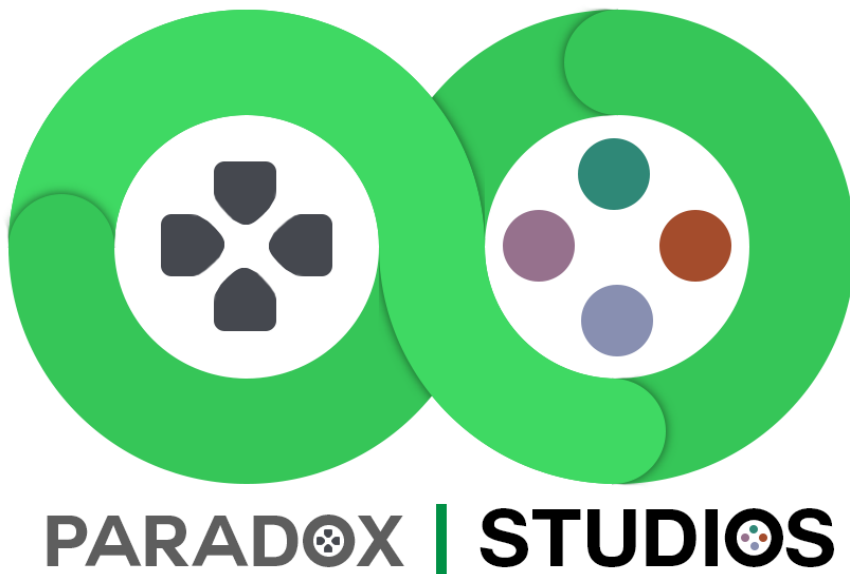


Entregables de **Videojuegos 2 en el** **prototipo (hito 1):**



Paradox Studios:

Moltó Ferré, Enrique

Muñoz Periñán, José Luis

Pérez Cristo, Rubén

Rebollo Berná, Antonio

Zamora Pastor, Julio

Entregables en el prototipo:

- **Control del player por motor de físicas 3D kinematic**

Utilizamos el kinematicPlayer de la última versión de bullet3D, sin embargo, hemos realizado algún cambio para ajustarlo más a nuestro juego (como la posibilidad de hacer un doble salto o que personaje acelere y desacelere al moverse para tener un movimiento algo más suave).

La clase que se encarga de utilizar esta clase de bullet es player que contiene un objeto kinematicPlayer.

- **Utilización de propiedades físicas en las mecánicas jugables**

Se usan las físicas para las mecánicas de manera constante. El personaje es capaz de colisionar con los objetos del mundo que son rigid bodies. Además, también puede saltar afectándole la gravedad que bullet nos proporciona. También tenemos ghostObjects que detectan cuando colisiona, pero los objetos rígidos son capaces de atravesarlos (esto sirve para coger los paquetes de armas o las vidas)

- **Uso de trazado de rayos y otros tests de las físicas**

El trazado de rayos esta implementado en la parte de las balas, la lógica de las balas es un raycast de bullet. Este rayo se detendrá en el primer objeto que detecte y si es un enemigo le restará vida.

- **Sistema de depuración visual de las físicas**

Hemos hecho un sistema de depuración visual, para ello cuando renderizamos llamamos a debugPhysics del mundo que tiene bullet pero está sin implementar (es virtual).

Hemos implementado ese método para que pueda hacer debug a las físicas en Irrlicht (habrá que cambiarlo cuando esté nuestro motor gráfico)

- **Mecánicas de acción**

Las mecánicas de acción están presentes en el salto, el movimiento, en el disparo de armas normales o en el disparo de lanzacohetes

- **Power-ups y elementos adicionales**

Como power-ups tenemos cajas que son ghost objects que te permiten obtener un arma o, en caso de que ya la tengas reponer su munición, también podemos recuperar vida con los lifeObjects que hay por el mapa.

- **Implementación de cámara FPS**

La cámara FPS que tenemos es la que nos proporciona irlicht y sigue a la posición del jugador en todo momento, así como el modelo de arma que tiene. Seteamos la posición de la cámara según la posición de nuestro jugador y la dirección de este al moverse viene determinado por la rotación de la cámara.

- **Creación de un cargador de niveles (Hito 2 pero ya hecho)**

El cargador de niveles es un lector de JSON (para leer el JSON utilizamos un parser de este que es nlohmann json). Aquí se lee el tipo de objeto que contiene y va creando las entities según le indica su posición, rotación y escalado

- **Creación de un editor de niveles (Hito 2 pero ya hecho)**

El editor de niveles que utilizamos es Unity y con un script se pueden leer todos los objetos de la escena y convertirlos en JSON.

Tiene que tener una estructura determinada el nivel para que esto funcione, todos los gameObject tiene que estar dentro de uno que se llame “Script”, que, como su nombre indica tiene el script, además a cada objeto hay que ponerle un tag para después identificar en nuestro juego el tipo de objeto que es.

Controles e instrucciones del prototipo:

Los controles del juego son los siguientes:

- WASD: movimiento del jugador
- Botón izquierdo del ratón: disparar
- Botón derecho del ratón: lanzar granada
- 3: retroceder arma
- 4: avanzar arma
- Barra espaciadora: saltar
- 1: debug de las físicas
- 2: separar cámara del player

Algunas cosas que se puede hacer en el prototipo:

- Disparar.
- Lanzar granadas.
- Coger vida y más armas.
- RocketJump: puedes saltar más que con un salto si te lanzas un misil del rocket contra el suelo.
- Doble salto: una vez saltas, en el aire puedes darle una vez más al espacio y saltar otra vez.

Instrucciones para ejecutar:

- Está preparado actualmente para ejecutar una versión del juego en Release, para ello descomprimir



el archivo LastBulletAlphaRelease.zip y en la carpeta Release ejecutar el archivo “BulletTest.exe”.

- Para compilar ir a la rama master. Se puede compilar un ejecutable en Debug ya que en el repositorio están las librerías necesarias para ello.
En caso de querer compilar en Release, primero habría que ir a la carpeta lib/irrlicht y descomprimir el archivo IrrlichtStatic_RELEASE.zip, ya que, sin comprimirse, el .lib supera el tamaño máximo por fichero en github.