



Práctica 5

Simulación física

Entornos Virtuales

Pedro Manuel Gómez-Portillo López

gomezportillo@correo.ugr.es

14 de abril de 2018

Índice

1. Introducción.....3

2. Práctica.....3

 2.1. Restricción.....6

1. Introducción

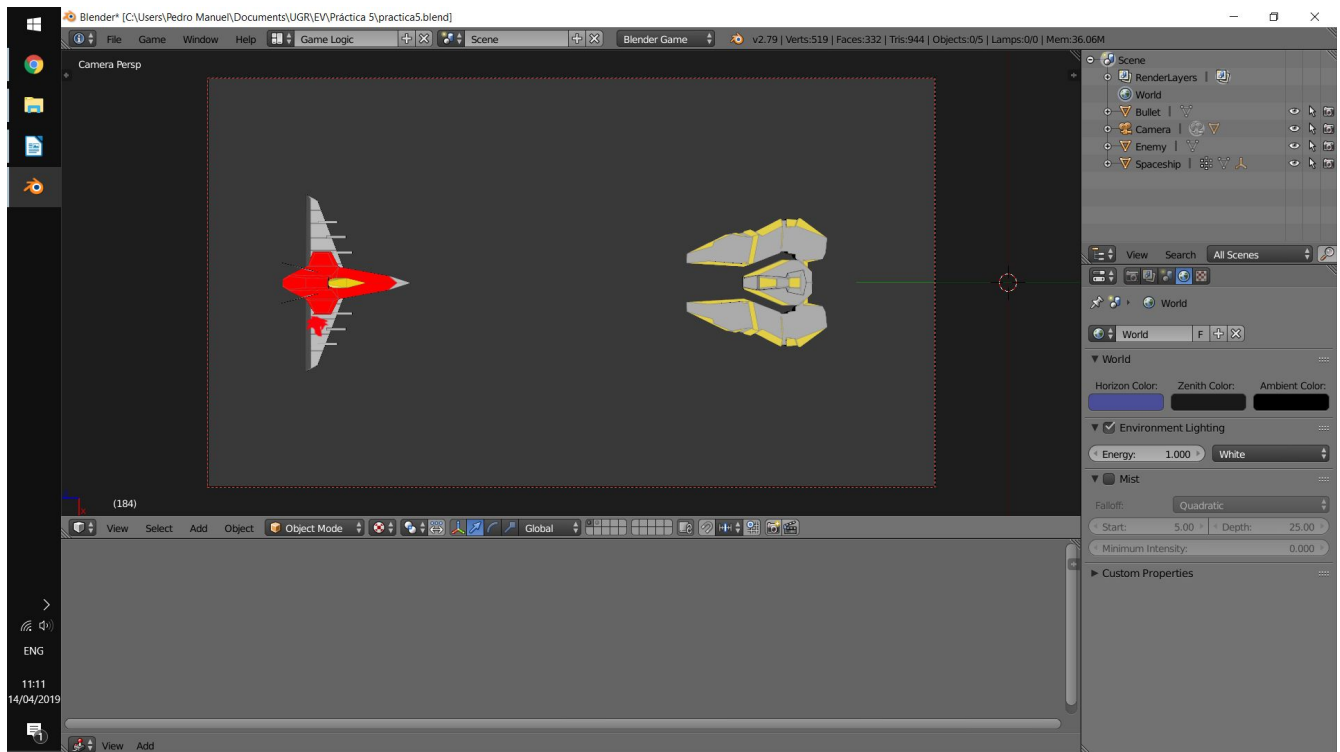
Blender Game Engine (BGE a partir de ahora) es un componente de Blender que se utiliza para crear contenido interactivo en tiempo real. El motor del juego fue escrito desde cero en C ++ como un componente mayoritariamente independiente e introducido en la versión 2.37. Incluye soporte para funciones como las secuencias de comandos Python o el motor de físicas Bullet¹.

2. Práctica

En esta práctica vamos a añadir comportamiento físico al modelo que tenemos de momento, lo que será un hito previo para el sistema interactivo de la práctica final, una versión del *Space Invaders*. Para ello, crearemos una escena con una nave enemiga, la nave espacial del jugador y un emisor de partículas con físicas justo enfrente para dar la sensación de que son balas disparadas.

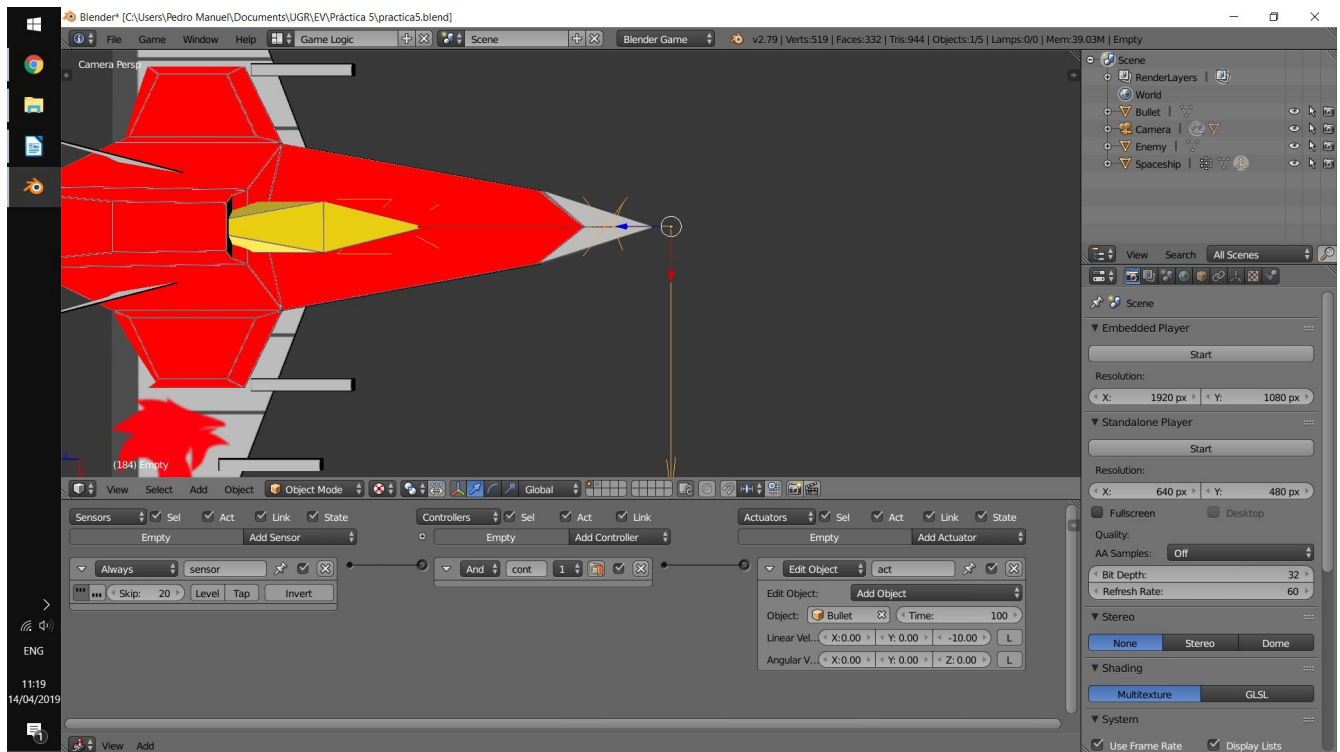
Lo primero que haremos será cambiar al motor BGE e ir a la vista *Game logic*, importar los modelos 3D de las dos naves espaciales, colocarlas una enfrente de la otra y definirlas como cuerpos estáticos. Los colocaremos de manera vertical para que las balas caigan por gravedad. Tras eso, colocamos la cámara a través de la que veremos el juego de manera perpendicular a las naves. Además, en la pestaña del mundo activaremos *Environmental lighting* para que todo tenga una iluminación uniforme.

¹ https://en.wikipedia.org/wiki/Blender_Game_Engine



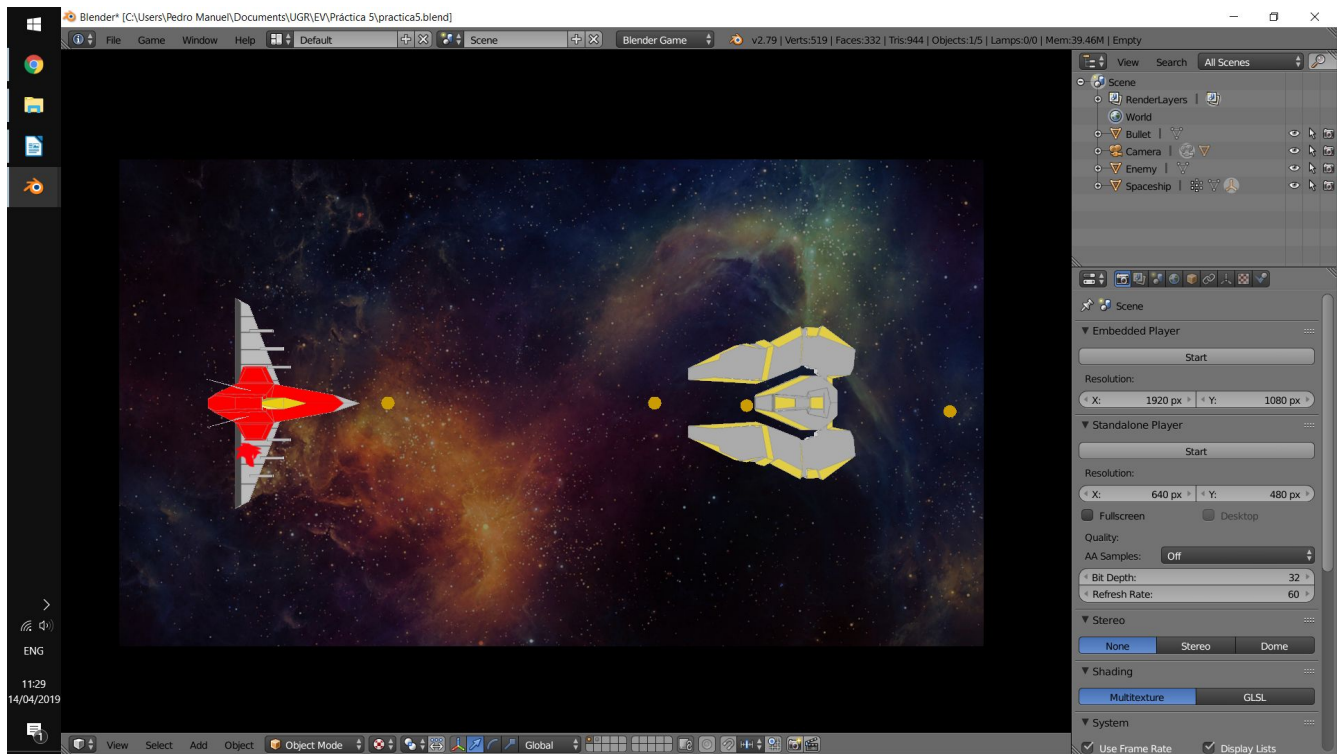
Ahora crearemos un objeto *empty* frente a la punta de la nave del jugador de la que aparecerán las balas. Este objeto lo emparentaremos con la nave para que cuando movamos ésta también se mueva el objeto vacío. Para las balas, crearemos una pequeña esfera amarilla en otra capa con físicas *Rigid Body*, que será la que emita nuestro objeto vacío.

Lo siguiente que tenemos que hacer es configurar nuestro objeto vacío para que funcione. Para ello, tras seleccionarla vamos a la vista *Logic Editor* y añadimos un sensor que emita una señal cada 20 frames. Luego, lo conectaremos a un controlador y a un actuador del tipo *Add object* seleccionando la esfera como objeto a emitir. Además, le añadiremos 10 unidades de velocidad lineal en el eje -Z para que parta con velocidad inicial.



Por último, usaremos el addon *Import images as planes* (tras activarlo en las preferencias de usuario) para importar la imagen del fondo como un plano, que colocaremos tras las naves de manera que tape por completo los espacios vacíos y que emparentaremos con la cámara, para que si en un futuro ésta se mueve lo haga con ella.

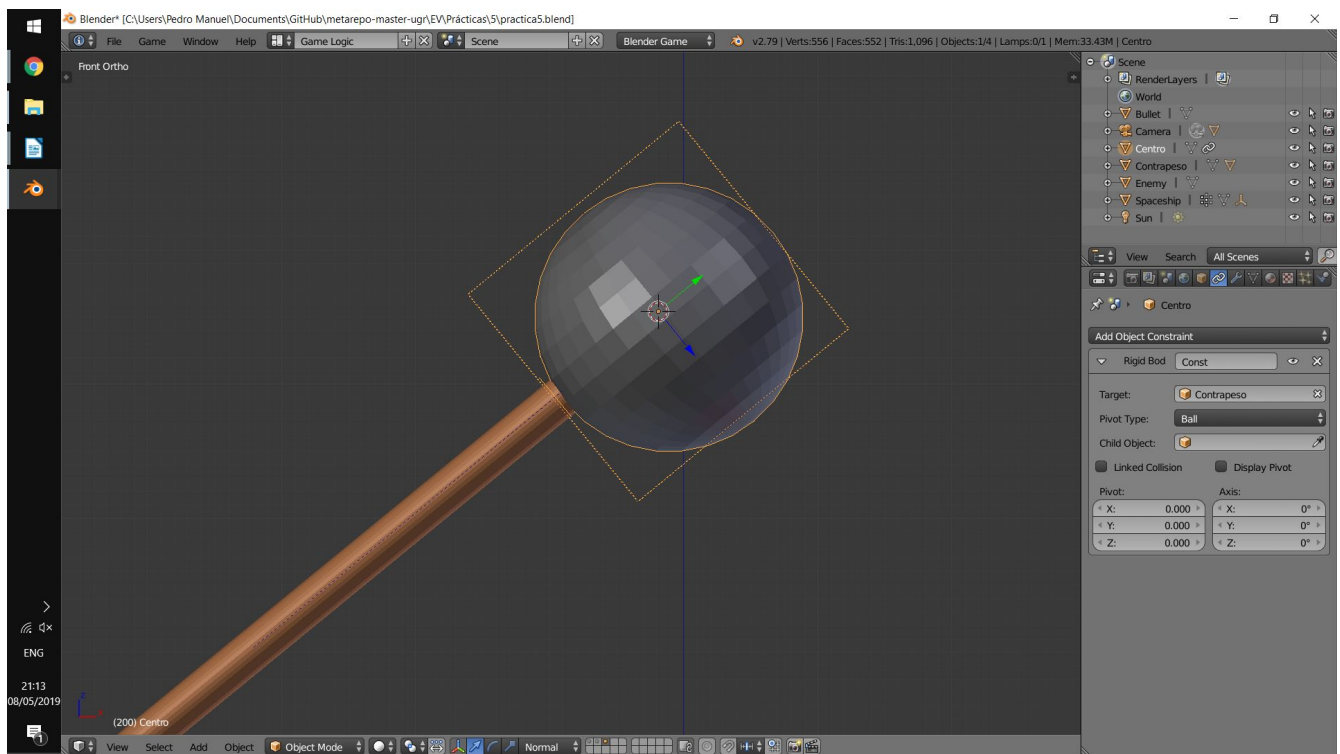
Ahora, tras seleccionar Game>Start game engine podemos ver cómo impactan las balas de la nave espacial sobre el enemigo.



2.1. Restricción

Como el enunciado pedía la implementación de una restricción pero no se me ocurrió ninguna ambientada en mi proyecto, he utilizado la tercera capa del archivo para incluir un pequeño ejemplo de una rotación con restricción tipo bola. Para ello, he añadido dos elementos,

- Una bola que actuará como centro y a la que se le ha añadido la restricción que se muestra a continuación.
- Un palo con un cilindro al final que será lo que rote y que se ha seleccionado como *Target*.



Ahora al iniciar la simulación podemos ver cómo el contrapeso rota alrededor del centro.

