



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Máster Profesional en Ingeniería Informática**

Curso 2020/2021

## **PRÁCTICA 4**

Entornos Virtuales

### **Breve descripción**

Programación de entornos virtuales

### **Autor**

Álvaro de la Flor Bonilla (alvdebon@correo.ugr.es) 15408846-L

### **Propiedad Intelectual**

Universidad de Granada



## **RESUMEN**

El objetivo de este documento es aclarar alguna de las decisiones tomada para la realización de la tercera práctica.



**ÍNDICE DEL PROYECTO**

**Resumen ..... 1**

**No se encuentran elementos de tabla de ilustraciones.¡Error! Marcador no definido.**

**1   Ejercicio calificable P4.A..... 4**

**2   Ejercicio calificable P4.B..... 4**

      2.1   Mejoras..... 5

**3   Ejercicio calificable P4.C..... 5**

**4   Conclusiones ..... 7**



# ÍNDICE DE ILUSTRACIONES

Ilustración 1 – Escalera unión ..... 4

Ilustración 2 – Personaje avatar y cámara ..... 4

Ilustración 3 – Etiquetas de ayuda ..... 5

Ilustración 4 – Configuración de variables..... 5

Ilustración 5 – Bloqueo del movimiento de la cámara ..... 6

## 1 EJERCICIO CALIFICABLE P4.A.

### ¿Qué he hecho para este ejercicio?

Pues en primer lugar he corregido algunos de los errores de iluminación que encontré de la práctica anterior (P3).

En cuanto a mi pasillo que conecta dos escenas en mi caso será una escalera que conecta las dos plantas de un edificio.

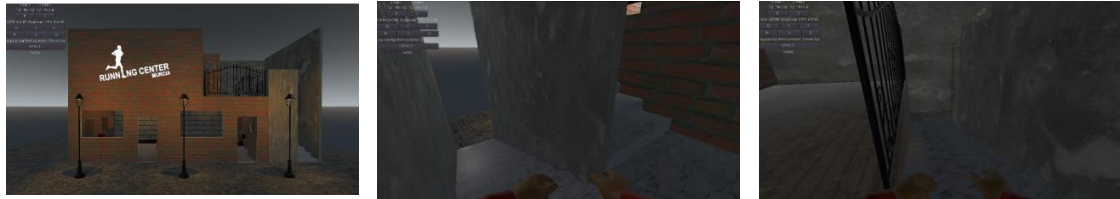


Ilustración 1 – Escalera unión

En la imagen anterior mostramos una sucesión de tres fotogramas claves de la composición de nuestro “pasillo”. Básicamente está compuesto por una escalera típica de peldaños a la derecha del edificio que en ningún momento permite la visión simultánea de los dos pasillos tal y como se indica en el guion de prácticas.

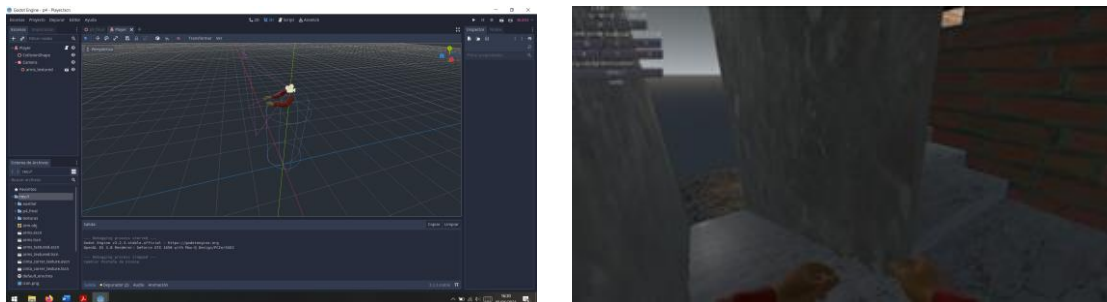


Ilustración 2 – Personaje avatar y cámara

Además, tal y como se puede ver en la ilustración anterior también hemos incluido una cámara que acompaña al avatar de nuestro “juego”. En esta ocasión hemos optado por que nuestro avatar simplemente sean un par de brazos. En una futura mejora hemos pensado que sea un cuerpo completo al que también se le aplique movimiento.

A este avatar se le ha dotado de una capa de colisión para que pueda impactar con los objetos tales como paredes, las sillas, las máquinas, etc.

## 2 EJERCICIO CALIFICABLE P4.B.

### ¿Qué he hecho para este ejercicio?

Para este ejercicio, he decidido aprovechar mi “Ejercicio teórico-práctico 2 Estudio del funcionamiento de una cámara FPS y salto” ya que opté por implementarlo en lugar de realizarlo de forma completamente teórica.

Como ya adelanté en el punto anterior, lo único que he tenido que realizar frente a la entrega de este ejercicio previo es añadirle al avatar y al resto de materiales de mi

escena las propiedades de colisión, evitando así que el avatar pudiera atravesarlas (al principio incluso atravesaba mi suelo y no sabía que ocurría).

La implementación ha sido simple, utilizo las teclas “W”, “A”, “S” y “D” para avanzar, ir a la izquierda, ir a la derecha y retroceder respectivamente. Además, también he incluido la opción de que el personaje sea capaz de saltar imitando el tiro parabólico.

En la próxima sección lo veremos en detalle, pero también se han añadido elementos para cambiar la cámara principal.



Ilustración 3 – Etiquetas de ayuda

Como puede ver en la imagen anterior, en la zona superior izquierda se han añadido “labels” a modo ayuda que identifica las funciones que puede realizar nuestra aplicación.

```

8 # Variables de la cámara
9 var minLookAngle : float = -25.0
10 var maxLookAngle : float = 25.0
11 export var lookSensitivity : float = 10.0 setget change_sensitivity
12
13 func change_sensitivity(new_sensitivity):
14     lookSensitivity = new_sensitivity
15

```

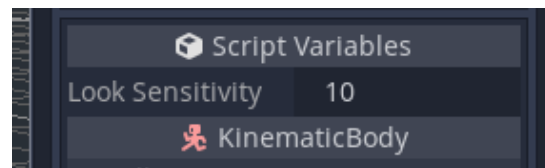


Ilustración 4 – Configuración de variables

Además, para finalizar esta parte de la práctica, se han restringido el movimiento vertical tanto hacia arriba como hacia debajo de la cámara, para evitar que de una vuelta completa.

## 2.1 Mejoras

Como añadido, se ha configurado una variable que puede ser modificada por el usuario fácilmente y establece la sensibilidad del movimiento de cámara-ratón que puede apreciar en la segunda imagen de la ilustración 4.

## 3 EJERCICIO CALIFICABLE P4.C.

En esta sección de la práctica lo que hemos hecho es habilitar el cambio de cámaras, simulando dos modos.

1. **Modo jugador.** En esta situación veremos una cámara “FPV” simple que seguirá los brazos de nuestro jugador allá donde vaya. Para activarla, tal y como se indica en la ayuda, pulsaremos la tecla “F”.
2. **Modo seguridad.** Este tipo de modo trata de simular que estamos viendo toda nuestra escena desde una cámara de seguridad. Para activarla tendremos que pulsar en esta ocasión la tecla “Q”.

Como mejora al caso anterior, **si y solo si** estamos dentro del “modo seguridad” si pulsamos sobre la tecla “R” cambiaremos a la cámara de seguridad que se encuentra en la zona “running”. Por el contrario, si pulsamos sobre la tecla “C” pasaremos a la cámara de seguridad de la zona del vestuario.

Todos estos últimos cambios de cámara disponen además de una animación de movimiento entre ellos.

Por último, para solucionar el problema que indica en la última parte del guion de prácticas se ha llevado acabo la acción que puede verse en la siguiente ilustración.

```
#####
#### PARTE 2 - MOVIMIENTO DE LA CAMARA CON EL RATON ####
#####
func _ready():
    # Ocultamos el cursor al iniciar
    Input.set_mouse_mode(Input.MOUSE_MODE_CAPTURED)

func _process(delta):
    if(camera.current==true):
        # rotate the camera along the x axis
        camera.rotation_degrees.x -= mouseDelta.y * lookSensitivity * delta
        # clamp camera x rotation axixs
        camera.rotation_degrees.x = clamp(camera.rotation_degrees.x, minLookAngle, maxLookAngle)
        # rotate the player along their y axixs
        rotation_degrees.y -= mouseDelta.x * lookSensitivity * delta
        # reset the mouse delta vector
        mouseDelta = Vector2()

func _input(event):
    if event.is(InputEventMouseButton):
        mouseDelta = event.relative
#####
#### PARTE 2 - MOVIMIENTO DE LA CAMARA CON EL RATON ####
#####
```

Ilustración 5 – Bloqueo del movimiento de la cámara

Lo único que he hecho es añadir una restricción que comprueba que **si y solo si** la cámara activa es la que persigue al avatar si habilita el movimiento con el ratón.

## 4 CONCLUSIONES

Una vez más creo en esta práctica he mejorado respecto a las anteriores, así como creo que en mi opinión he cumplido con todos los requisitos que se pedían en el guion.

En el caso de que existiera algún problema con la ejecución de la práctica, se envía en añadido una demo de la ejecución del script final con el nombre "*demo.mkv*".