

# **MEMORIA PROYECTO FINAL**

Pablo González Arroyo  
Simulación Física para Videojuegos  
(3°GV)

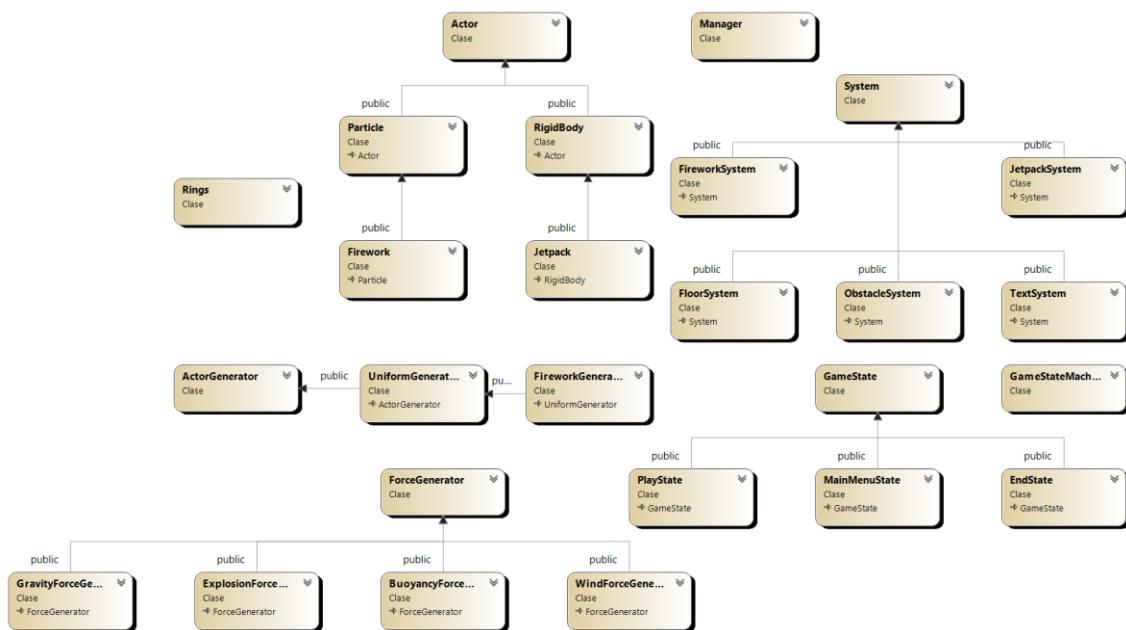
# 1. Temática

El objetivo en este juego es el de, tras ser disparado hacia delante, llegar lo más lejos posible sin caer al suelo, manejando correctamente el uso del combustible y pasando a través de unos aros para recuperar repostaje.

Esto es porque, técnicamente, el jugador cuenta con un jetpack, que tras ser lanzado podrá activarlo para subir o bajar en el eje Y lo cual le permitirá maniobrar en el aire.

No hay niveles, si no que cuenta con una generación infinita de los objetos que conforman el nivel por lo que siempre hay distancia que recorrer.

## 2. Diagrama de clases



## 3. Ecuaciones físicas usadas

Para la realización de este proyecto se han usado las siguientes físicas:

- Fuerza de la gravedad. Esta sigue la siguiente fórmula:

$$F_g = g * m$$

Donde  $F_g$  es la fuerza resultante de multiplicar la aceleración de la gravedad por la masa del objeto.

Esta fórmula se aplica, dentro de los generadores de gravedad, en las partículas puesto que estas no cuentan con la propia implementación que sí tiene Physx para los sólidos rígidos.

- Fuerza del viento. Cuenta con la siguiente fórmula:

$$F_v = k_1(v_v - v) + k_2|v_v - v|$$

Donde  $F_v$  es la fuerza resultante de sumar la multiplicación de las constantes de rozamiento por la diferencia entre la velocidad del viento y la del objeto.

Esta fórmula se aplica tanto a sólidos rígidos como a partículas a través de los generadores de viento. En el código se ven tal que:

```
WindForceGenerator* wfgUp = new WindForceGenerator(jetpack->getPos(), Vector3(CONSTANT_X_VEL, 20, 0), 0, colors[RED], 1);
WindForceGenerator* wfgDown = new WindForceGenerator(jetpack->getPos(), Vector3(CONSTANT_X_VEL, -20, 0), 0, colors[RED], 1);
```

Siendo importantes la velocidad que se pasa (segundo valor) y la constante k1 (último valor, siendo k2 un parámetro determinado).

- Explosión. Se usa la siguiente fórmula:

$$F_e(x, y, t) = \frac{K}{r^2} \frac{x - x_c}{z - z_c} [y - y_c] e^{-\frac{t}{\tau}}$$

En las explosiones,  $F_e$  será el resultado de multiplicar la intensidad de la explosión  $K$  entre el radio al cuadrado de esta por la distancia a la que se encuentra el objeto del centro por  $e$  elevado al tiempo entre tau. A mayor distancia, menor será la fuerza recibida; y a mayor tau más tiempo tardará en desvanecerse la explosión.

En el código se aplica de esta fórmula tras hacer previamente una comprobación de que el objeto está dentro del radio de explosión, y se ve tal que:

```
ExplosionForceGenerator* efg = new ExplosionForceGenerator(pos, rad * 20, 4000, 1);
```

Aquí el radio será 20 veces el introducido (puesto que en la misma función se crea la mitad de una esfera para hacer un efecto de empuje), la intensidad será muy alta y la explosión se disparará relativamente rápido.

- Flotación. Por último, se usará la flotación tal que:

$$F_f = V_s * d_l * |g|$$

Donde  $F_f$  será el resultado del producto entre el volumen del objeto que se sumerge, la densidad del líquido al que se le sumerge y la aceleración de la gravedad. En el código, se deben hacer cálculos previos a la aplicación de la fuerza para ver cuán sumergido está la masa en el líquido.

```
float h = actor->getPos().y;
float h0 = liquidParticle->getPos().y;

float inmersed = 0.0;
if (h - h0 > height * 0.5) inmersed = 0.0;
else if (h0 - h > height * 0.5) inmersed = 1.0;
else inmersed = (h0 - h) / height + 0.5;

Vector3 force = Vector3(0, liquidDensity * volume * inmersed * gravity, 0);
actor->addForce(force);
```

En el código se traduce a esta función:

```
BuoyancyForceGenerator* bfg = new BuoyancyForceGenerator(posBg + Vector3(x, height, 0), 20, 10, 1);
```

Los 3 últimos valores son, según aparecen, la altura del líquido, el volumen del objeto introducido y la densidad del líquido.

- Integrate. Además de las fuerzas, otra fórmula a usar es la que permite que se hagan efectivas todas las transformaciones, que se apliquen las fuerzas, etc. Se llevan a cabo dentro del método *integrate*, el cual usará las funciones:

$$v = v_0 * a * t \quad y \quad x = x_0 * v * t$$

## 4. Efectos incorporados

- Lanzamiento. Al iniciar el juego y tal y como se ha mencionado en el apartado uno, lo primero que debe hacer el jugador es lanzar el jetpack. Para ello, se le aplica una velocidad inicial al sólido rígido que lo lanza en un tiro parabólico.
- Movimiento en el eje y. Como se ha mencionado en el apartado uno, se ha implementado el movimiento en el eje Y que se lleva a cabo a través del empuje

generado por un generador de viento. Hay dos: uno para moverse hacia arriba y otro hacia abajo (véase imagen de la fuerza del viento del apartado 3).

- Explosiones a modo de feedback. Para dar aun más a entender al jugador que se ha utilizado el jetpack, por cada movimiento realizado, ya sea hacia arriba o hacia abajo, se generará una explosión en la posición del jugador. Esta explosión hará efecto en un semicírculo de partículas multicolor que se generará detrás del jugador.
- Fondo y frente. En el fondo del escenario y en el frente se generarán partículas para dar la sensación de profundidad y de movimiento al jugador. Pero además las partículas del fondo cuentan con generadores de flotación, lo cual hará que unos sólidos rígidos que se generan a la vez que ellas reboten según va jugando el jugador.
- Fuegos artificiales. Al terminar el juego por haber chocado, saldrá la pantalla final donde se pueden apreciar fuegos artificiales generándose al fondo. Estos son partículas que se ven afectadas por la gravedad del generador explicado en el apartado 3.

## 5. Manual de usuario

### Descripción

Un jetpack. El cielo. Tú y unos anillos contra el límite de combustible. ¿Qué puede salir mal? Pues que te estampes contra el suelo, ¡claro está! ¿Crees que tienes lo que vale para llegar lo más lejos posible y aprovechar al máximo el combustible de este precioso jetpack? ¡Prueba prueba!

### Controles

Para lanzar el jetpack deberás:

- 1- Esperar a que el jetpack esté a la mitad del indicador verde
- 2- ¡Pulsar el espacio en el momento adecuado!

En el aire podrás:

- Subir pulsando la tecla 'I'
- Bajar pulsando la tecla 'K'

Y para pasar de menú, pulsa espacio. ¡Ya lo sabes todo para surcar los cielos!

## 6. Enlaces útiles

**Repositorio GitHub:** [PabloGonzalezArroyoo/SimulacionFisicaVideojuegos: Repositorio de las prácticas de la asignatura de Simulación Física de Videojuegos \(github.com\)](https://github.com/PabloGonzalezArroyoo/SimulacionFisicaVideojuegos)