PROYECTO FINAL SIMULACION FÍSICA

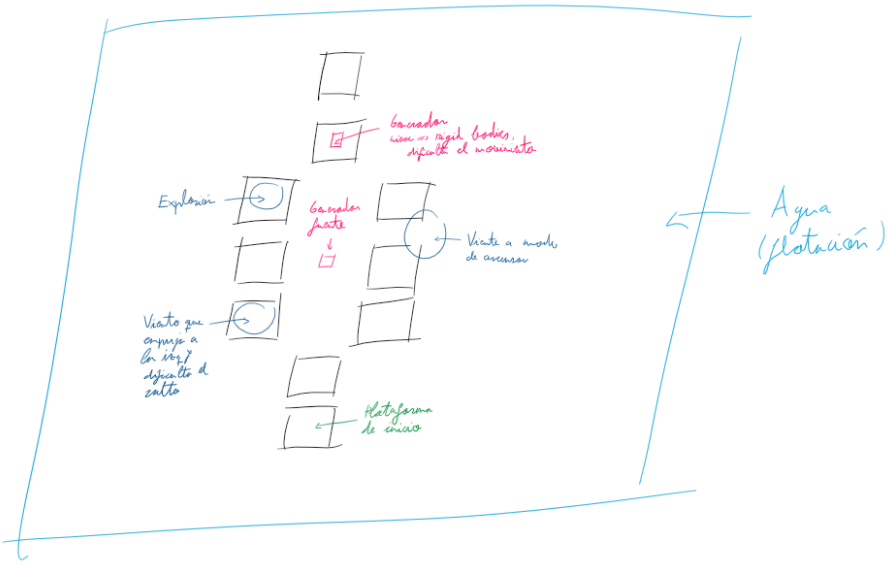
ÍNDICE

1. DESCRIPCIÓN

Se trata de un juego de plataformas en primera persona que tiene como objetivo pasar por cada una de las 10 plataformas existentes. Cada plataforma presenta un reto, de modo que están a diferentes alturas y distancia, hay fuerzas en ellas o que las conectan… Además, se muestra arriba a la izquierda un contador indicando el número de plataformas restantes por las que pasar.

El modo de perder es caerse al mar que hay debajo de las plataformas. Entonces, se restablecerá el juego y se le dará una nueva oportunidad al jugador.

a. ESTRUCTURA NIVEL



También hay una lluvia constante en la zona delantera del área de juego. La lluvia y la fuente sirven para que el jugador sea consciente donde se aplican esas fuerzas y pueda actuar en consecuencia.

Por ejemplo, en el viento que funciona como ascensor, el jugador se daría cuenta que no puede llegar a la plataforma siguiente porque está muy alta, pero que en la parte derecha hay un viento que sí le permite alcanzarla.

b. INPUT

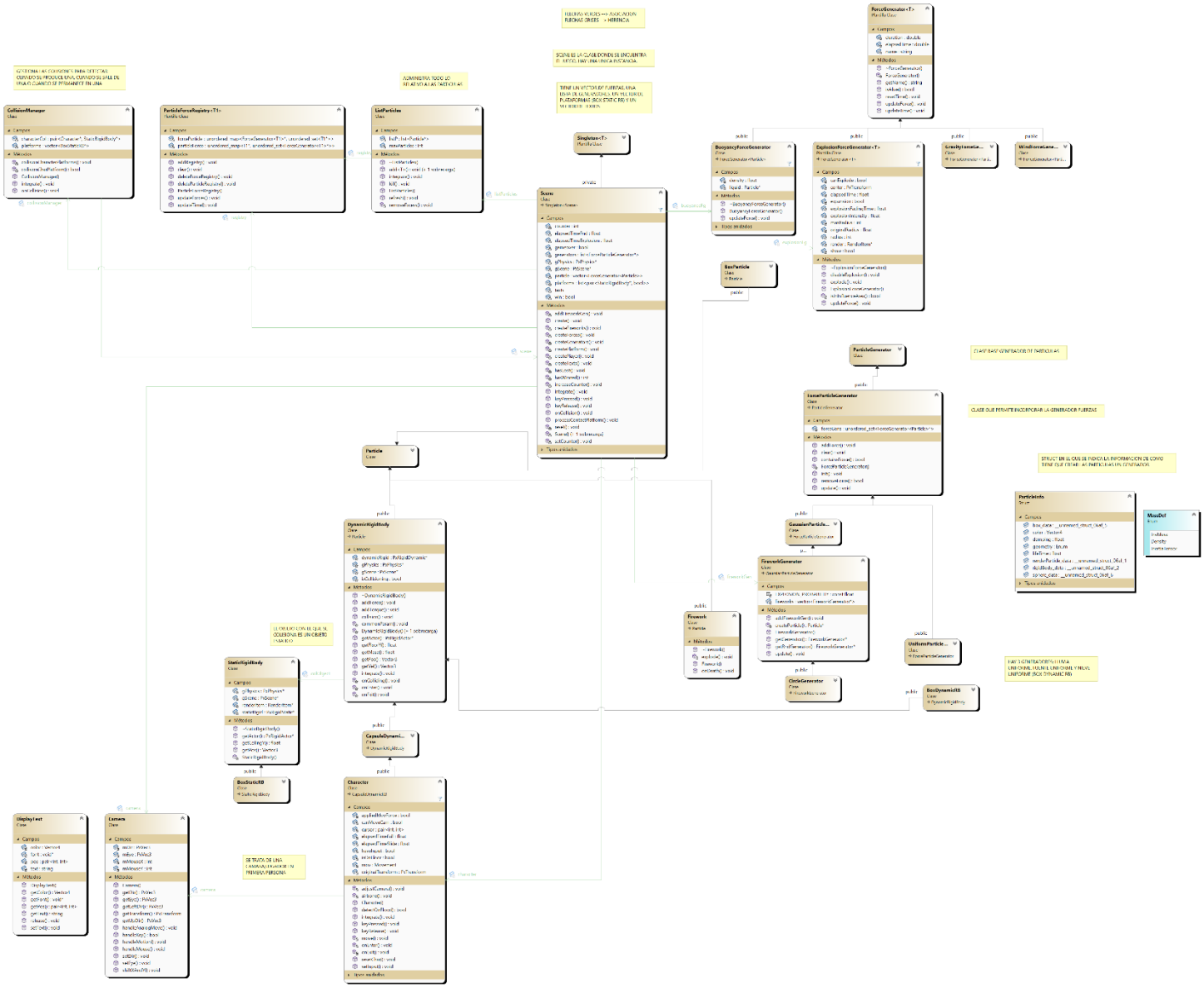
La cámara está en primera persona, de modo que sigue al puntero del ratón. El input es el siguiente:

- Se utilizan las teclas IJKL para desplazarse en todas las direcciones.
  El jugador solo puede desplazarse en la tierra, pero no mientras está en el aire.
- La acción de que la cámara siga al puntero del ratón se puede intercalar.
  Si se clica en la pantalla se parará esta acción si no lo estaba y si estaba parada, se activará.

- Se utiliza la barra espaciadora para saltar.
  Se puede saltar tanto en el sitio como en carrera.
- (Arrastrar la cámara y mover la cámara con WASD se han dejado por motivos de depuración, pero no conviene utilizarlos durante el juego).

2. DIAGRAM DE CLASES

a. UML



(Se puede encontrar también dentro del proyecto en Visual Studio y en el repositorio)

b. ELEMENTOS ESCENA

La clase Scene es la clase principal, es aquella que contiene todos los elementos del juego y los conecta. Estos elementos son:

- Jugador
- Generadores de partículas (todos los generadores funcionan con fuerzas)
    - Torbellino uniforme utilizando render items
    - Lluvia uniforme utilizando render items
    - Fuegos artificiales utilizando render items (1 lanzador inicial en forma de círculo y 3 generadores que sirven para propagar)
    - Nieve uniforme utilizando rigid bodies, de modo que afectan al movimiento del personaje
- Fuerzas
    - Gravedad
    - Viento que empuja hacia la izquierda
    - Viento a modo de ascensor
    - Explosión
    - Flotación
- Plataformas
- Textos que aparecen en la interfaz
- Lista de partículas
- Registro de fuerzas y partículas (también personaje)
- Collision manager

3. FUERZAS

a. GRAVEDAD

$$F_g = \frac{G m_1 m_2}{r^2}$$

Se parte de esta fórmula, sin embargo, se suponen masa y radio de la Tierra como constantes, por lo tanto, la fórmula se reduce a:

$$F_g = G * m$$

- siendo m la masa del objeto
- G es una constante, que en el caso de la tierra es 9.8, pero se ha aproximado a 10.

b. FLOTACIÓN

Cuando se introduce un objeto en un líquido hay un parte del líquido que queda fuera/desalojada. Entonces, esa parte aplicara una fuerza en sentido opuesto al peso del objeto con el fin de regresar a su posición inicial. Esta fuerza se conoce como fuerza de flotación y se basa en el peso del agua desalojada.

$$F_{flotación} = P_{líquido\ desalojado} = m * G$$

$$m = V_{líquido\ desalojado} * d$$

Si el líquido se trata de agua, su densidad (d) es 1000 kg/m^3. El volumen del líquido desalojado corresponde con el volumen del objeto que hay dentro del líquido:

$$V_{líquido\ desalojado} = V_{objeto\ sumergido} * \text{porcentaje de inmersión}$$

La fórmula resultante quedaría así:

$$F_{flotación} = V_{objeto\ sumergido} * \text{porcentaje de inmersión} * d * G$$

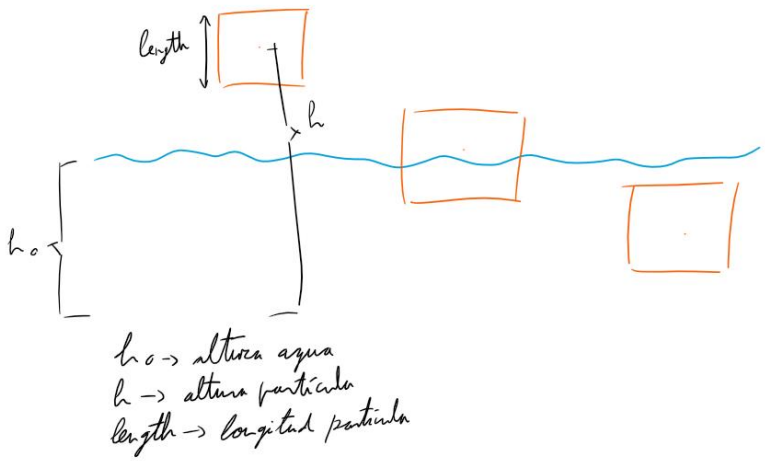
Esta fórmula es muy parecida a la ley de Hooke para los muelles.

$$F_{flotación} = V_{objeto\ sumergido} * d * G * \text{porcentaje de inmersión}$$

$$F = -k * x$$

El volumen del objeto, la densidad y la gravedad son constante que no van a variar, al igual que la constante elástica de un muelle. En cambio, el porcentaje de inmersión se asemeja a la longitud de deformación y sí va a variar.

Hay tres casos de inmersión: no está en el agua, está parcialmente sumergido o está completamente.



4. EXPLICACIONES ADICIONALES

a. JUGADOR

El jugador se trata de una rigid body con forma de cápsula, lo que permite que tenga colisiones. Además, sirve de punto de referencia para realizar el efecto de primera persona, pues es donde se coloca la cámara y a partir de donde se gira.

Para que el jugador pueda moverse de forma adecuada se le aplican varios efectos y fuerzas:

- Fuerzas continuas en los sentidos que quiere moverse hasta dejar de pulsar la tecla. Para detectar cuando se deja de pulsar una tecla se ha implementado el callback glutKeyboardUpFun(…).
- Frenado en seco cuando no está realizando ningún movimiento para evitar el deslizamiento.
- Fuerza hacia abajo para evitar el deslizamiento cuando se salta en carrera.
- Fuerza hacia arriba cuando se quiere saltar.
- Fuerza hacia abajo muy débil pero constante para que cuando descienda no parezca que está flotando.

b. CÁMARA

Se ha modificado para que se pueda hacer que gire una distancia dada (método shifXAndY). Además, para poder realizar este giro y para poder crear el fuego artificial en forma de círculo en el plano apropiado, se han creado métodos que devuelven direcciones perpendiculares a dirección original. Para ello se han usado quaternions.

También se ha implementado el callback glutPassiveMotionFunc(…) para poder guardar la posición del cursor del ratón en todo momento.

c. SISTEMA DE FUEGOS ARTIFICIALES

El sistema de fuegos artificiales funciona con generadores y es infinito. Hay un generador que se elige como lanzador y es el que inicia todo el proceso. Luego, hay otros generadores que se suscriben a este sistema y son los que van propagando, es decir, son los que crean más partículas cuando se elimina una.

d. LISTA DE PARTÍCULAS

Se encarga de toda la administración de partículas que crean los generadores (update, gestionar la muerte…). Tiene un cupo máximo de partículas que puede contener, de modo que cuando se ha superado y se crean nuevas partículas, se eliminan las más antiguas para dejar hueco a las más nuevas.

e. COLLISION MANAGER

Se encarga de identificar a partir del sistema de colisiones de Physx cuando se ha producido una colisión, cuando se está en una colisión y cuando se ha terminado una colisión. Se utiliza esta información para indicar al jugador cuando puede saltar y cuando no.

IMPORTANTE: Se puede alternar entre el modo DEBUG (se muestra partícula del personaje y áreas de fuerzas) y no, cambiando el valor de la variable constante DEBUG en la clase Scene.