

Memoria

Proyecto Final

Temática y objetivo del juego

Mi objetivo con este proyecto ha sido aprender sobre física en videojuegos y cómo implementarla. En mi caso, la ciencia que posiblemente más disfruto es la astronomía, así que decidí hacer un proyecto enfocado a eso, aunque la astronomía no es el único factor determinante que me ha llevado a este proyecto.

Sin duda alguna, jugar *Outer Wilds* (un juego basado en el espacio y el conocimiento) ha significado un antes y un después en mi vida, convirtiéndose rápidamente en uno de mis juegos favoritos y en lo que es para mí el mejor juego de aventura espacial que existe.

Inspirado por estos factores, decidí realizar un simulador de órbitas y el interior de un planeta acuático con fuertes corrientes y vientos.

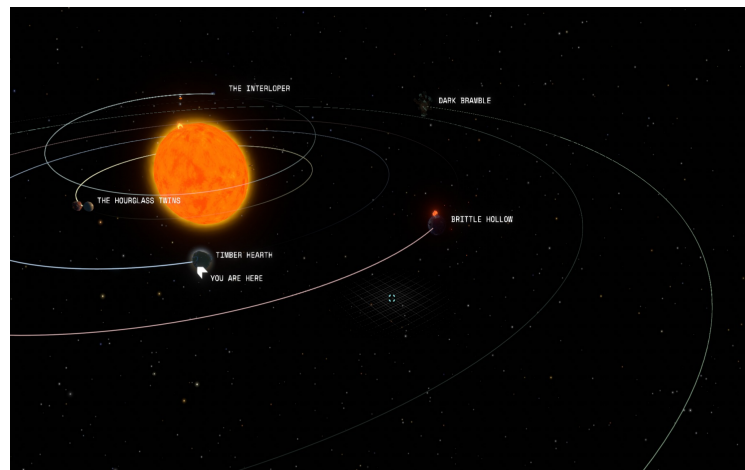
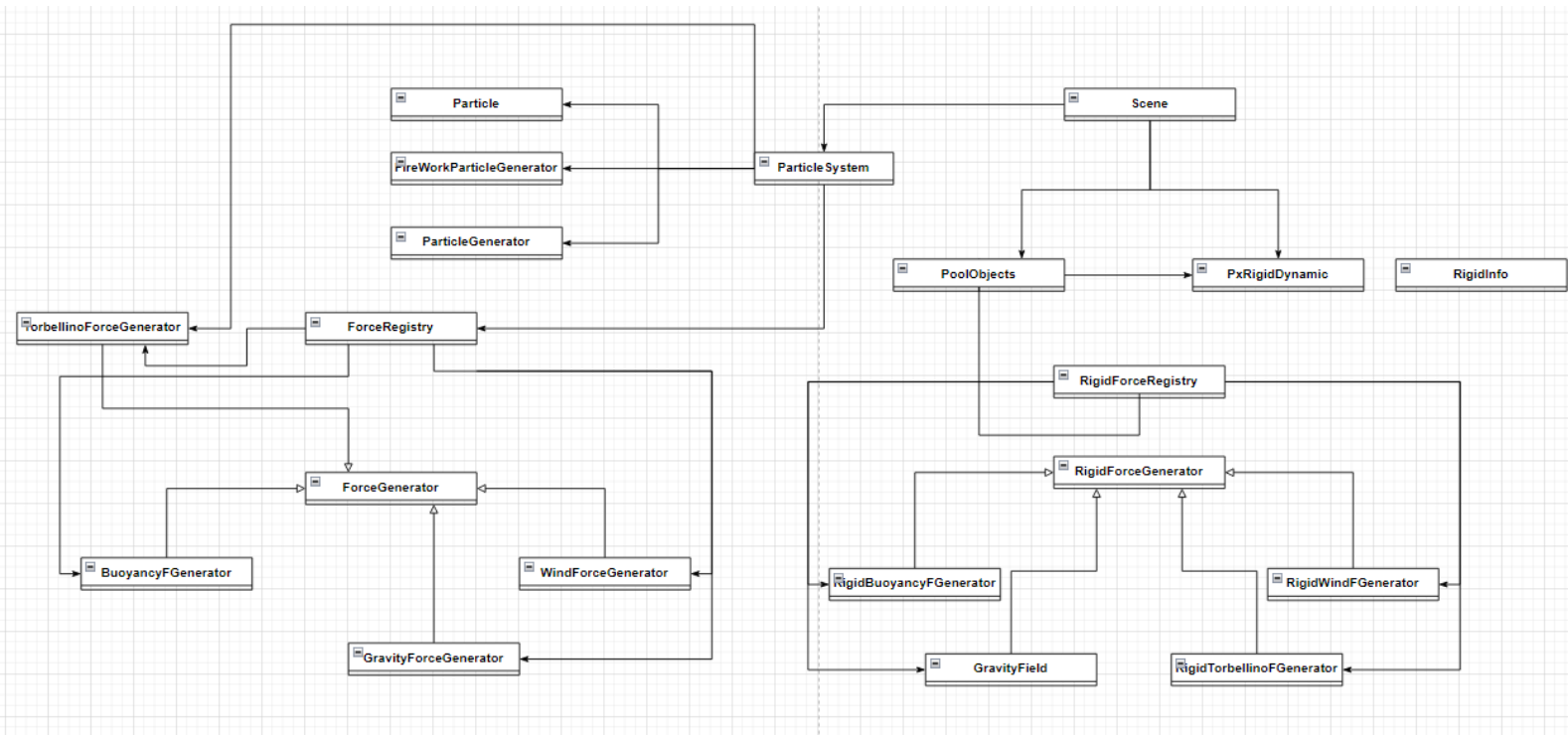


Diagrama de clases del sistema de partículas y sólido rígido



Ecuaciones físicas usadas

Ley de Gravitación Universal: $F = G \cdot \frac{m_1 m_2}{r^2}$

Usada para realizar la simulación de las órbitas.

La ley de gravitación universal es una ley física clásica que describe la fuerza o interacción gravitatoria entre distintos cuerpos con masa, fue formulada por *Isaac Newton*.

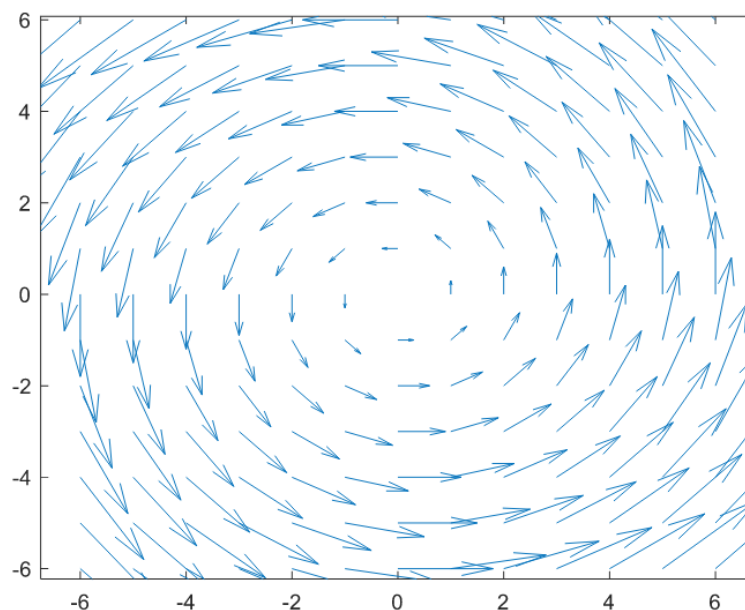
En mi caso y a la hora de implementar este sistema, la constante de gravitación tiene un valor de '30' y las masas de los planetas y estrellas están reducidas para balancear la ecuación, además que las distancias entre ellos también es mucho menor.

Por ejemplo, el Sol tiene una masa de 200-300 y el planeta azul (el más grande en la simulación) tiene una masa de 10.

Ecuación de los torbellinos:
$$\vec{v}_{torbellino}(x, y) = K \begin{bmatrix} -(z - z_c) \\ 50 - (y - y_c) \\ x - x_c \end{bmatrix}$$

Usada para realizar la simulación de los tornados.

Los tornados son un elemento reutilizado de la anterior práctica que simulaba el campo de velocidades de un torbellino.



En este caso, los valores de la ecuación son a escala de las unidades del mundo de la simulación. Por ejemplo, el radio del torbellino viene dado por valores comprendidos entre 30 y 60 debido a las distancias que recorren las partículas, y la fuerza tiene un valor acorde para que la partícula no salga disparada ni se quede quieta.

Fuerza ejercida por el viento: $\vec{F}_v = k_1(\vec{v}_v - \vec{v}) + k_2\|\vec{v}_v - \vec{v}\|(\vec{v}_v - \vec{v})$

Usada para mover las partículas y los sólidos rígidos en el planeta de agua.

El viento también es un elemento reutilizado de una práctica anterior.

Al igual que en los torbellinos, los valores para la fuerza del viento son acordes al mundo de la simulación para que la partícula se mueva a una velocidad aceptable.

También, se ha expandido la clase **PxRigidDynamic** para añadir los coeficientes de fricción del viento para reutilizar la misma fórmula que usa el viento de las partículas.

Segunda Ley de Newton: $\vec{F} = m\vec{a}$.

Usada para simular la gravedad en el planeta de agua.

Siguiendo la segunda Ley de Newton, la fuerza de la gravedad es la masa del objeto por la aceleración de la gravedad.

En el caso de mi simulación, la aceleración de la gravedad es aproximadamente igual a la de la Tierra, y dependiendo de la masa del objeto la fuerza será mayor o menor.

Enumeración de los efectos incorporados.

La fuerza gravitatoria implementada con su respectiva fórmula, además de un entorno donde se puede observar cómo funciona.

Un generador de partículas que simulan estrellas en el espacio y un generador de “fuegos artificiales” usado para simular los restos de planeta que salen volando hacia el espacio después de un choque.

Generador de fuentes de partículas que se usan para simular los tornados.

Un planeta de agua donde se puede observar la flotabilidad, el viento, la gravedad y los tornados y cómo afectan todos estos elementos a partículas y sólidos rígidos con distintas características.

Por último, cumple todos los requisitos de la entrega, es decir, el proyecto cuenta con un gestor de escenas que se encarga de borrar todo al cambiar entre una u otra y todas las partículas tienen un tiempo de vida.

Manual del usuario

Cambio Escenas

Pulsa las para cambiar entre las escenas.

0: Simulación con órbitas estables

1: Simulación con órbitas inestables

2: Interior del planeta de agua, observando tornado ascendente.

3: Interior del planeta de agua, observando tornado descendente.

Interacción en las escenas

Algunas escenas permiten interacción:

‘Z’: Transición a la escena del planeta de agua

‘C’: Crear partícula con masa aleatoria en el tornado ascendente

‘R’: Crear sólido rígido con propiedades aleatorias en el tornado descendente

‘+’: Aumentar masa del Sol en la simulación de órbitas

‘-’: Disminuir masa del Sol en la simulación de órbitas

Efectos/experimentos extra incluidos

Cuando dos planetas o estrellas colisionan, sobrevive el que más masa tenga , aumenta su escala debido al choque y provoca que restos de ambos planetas salgan volando hacia el espacio.

Implementados tornados que se mueven en ambos sentidos (hacia arriba o hacia abajo).

En la escena del interior del planeta, los tornados afectan a las partículas/sólidos rígidos que entren dentro de su rango, elevando o sumergiendo el cuerpo dependiendo del sentido del tornado.

Implementada una interpolación lineal para la entrada al interior del planeta (simulando el efecto de zoom).

Añadido un fondo de agua para un mejor efecto visual del interior del planeta, además de un viento que mueve las partículas y los sólidos rígidos en dirección hacia los tornados.

Expandida la clase **PxRigidDynamic** para obtener el coeficiente de fricción con el viento y creación de un struct llamado **RigidInfo** que contiene datos como el tiempo de vida, la escala, y el estado de la partícula (viva o muerta) y se almacena en la variable **userData** aprovechándose del tipo *void**.