



Práctica 1: Entorno de prácticas. ¡Nuestra primera partícula!

Introducción al entorno de prácticas. Repositorio de la asignatura

El código que vamos a usar a lo largo del curso va a partir de nuestro repositorio GitHub de la asignatura. Para acceder a él hay que visitar la siguiente dirección:

<https://github.com/UCM-237/SimulacionFisicaVideojuegos>

Tal y como se ha visto en clase, nuestro repositorio incluirá el código realizado por nosotros y la librería Physx de NVidia.

Actividad 0a: Cada alumno debe de copiar (hacer un fork) el repositorio de la asignatura para tener una réplica en su usuario.

Una vez realizada la copia del repositorio, vais a iniciar un proyecto dentro del entorno de desarrollo Visual Studio¹ a partir de dicha réplica. La idea va a ser que todo el código que vayáis desarrollando a lo largo de esta asignatura quede registrado en el repositorio del usuario. De esta forma podremos ver vuestra evolución e incluso podremos marcar entregas mediante etiquetas (tags) dentro del repositorio.

Actividad 0b: Cread un proyecto con Visual Studio a partir de su copia del repositorio.

1. Usad el menú Archivo Clonar repositorio...
2. En el cuadro de diálogo, escribid la ubicación de vuestro repositorio.

Ejecutad el proyecto atendiendo a la explicación de profesor.



¹ Nótese que dicho entorno está disponible con gratuitamente con licencia estudiante para todo el alumnado de la UCM.

Actividad 1a: Al ver el mundo que se genera al renderizar, habréis notado que se queda algo triste: no está poblado ni siquiera por una mísera partícula.

En esta actividad diseñaréis vuestra primera clase partícula en C++ y la dotaréis de lo necesario para ser representada (un objeto *RenderItem*). El aspecto que tendrá la partícula es algo que podéis decidir vosotros. Dentro del esqueleto, como se ha comentado en clase, tenéis varias opciones para pintar objetos en pantalla. Cada uno de ellos tiene parámetros para poder personalizarlo como queráis.

Además, haréis que actualice su posición implementando el método de integración "*integrate*". Haréis que dicha partícula tenga velocidad constante, definida como un *Vector3*.

Aquí tenéis un ejemplo de definición de partícula con velocidad. Como último apunte, recordad que tendréis que *deregistrar* el objeto *RenderItem* de la escena en el destructor de la partícula.

```
class Particle
{
public:
    Particle(Vector3 Pos, Vector3 Vel);
    ~Particle();

    void integrate(double t);

private:
    Vector3 vel;
    physx::PxTransform pose; // A render item le pasaremos la direccion de este pose, para que se actualice automaticamente
    RenderItem* renderItem;
};
```

Actividad 1b: Una vez realizada dicha partícula, la instanciareis en el archivo principal (main.cpp), llamaréis al método *update* en la función de actualización y tendréis cuidado de liberar la memoria reservada a dicha partícula en la función "*cleanup*" si habéis hecho una reserva dinámica con el operador "new".

Al realizar esta práctica, se debe mostrar por pantalla una partícula moviéndose a velocidad constante por la pantalla.