

Object-oriented Graphics Rendering Engine

Práctica 2

2.1 - Animación.

Alberto Núñez

Departamento de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

- ❑ Para la animación contaremos con tres elementos
 - ❑ Un plano nuevo que representa el suelo (distinto del suelo del laberinto)
 - ❑ Sinbad (con y sin espadas) y Ogrehead
- ❑ **¡Hay un vídeo disponible en el C.V con la animación!**
 - ❑ Por un lado crearemos una animación en la que moveremos a Sinbad y ogrehead
 - ❑ Por otro lado animaremos a Sinbad para que corra y baile
- ❑ El diseño de los *keyframes* y los valores para la transformación es libre
 - ❑ El objetivo es que la animación se parezca al vídeo
- ❑ Algunos datos que pueden ser útiles
 - ❑ El tamaño del plano es de 150x300
 - ❑ La animación dura 21 segundos, luego se repite en bucle
 - ❑ Finaliza cuando empieza el juego, al pulsar una tecla (por ejemplo 's')
 - ❑ La animación de Sinbad puede realizarse con **9 *keyframes*** (estado inicial incluido)
 - ❑ La animación de ogrehead puede realizarse con **6 *keyframes*** (estado inicial incluido)

- ❑ Para crear la animación puede resultar útil
 - ❑ Crear un método que añada *keyframes* a un `NodeAnimationTrack`
 - ❑ Por ejemplo el método
 - ❑ `addKeyFrame (nodeAnimationTrack, giro, posición, más parámetros...);`
- ❑ De forma que en el personaje tendremos algo así:

```
// Keyframe 0
this->addKeyframe (track, Quaternion::IDENTITY, Vector3::ZERO, . . .);

// Keyframe 1
this->addKeyframe (track, Quaternion::IDENTITY, Vector3::ZERO, . . .);
```

- ❑ ¿Cómo sabemos cuando tiene que correr o bailar Sinbad?
 - ❑ Con los instantes de tiempo de los *keyframes*
 - ❑ Puede ayudar tenerlos como atributos en la clase
 - ❑ Cada uno indica el instante en el que acaba el keyframe

- ❑ ¿Y cómo sabemos cuándo hemos llegado a ese instante?
 - ❑ Llevando la cuenta del tiempo. Dos opciones:
 - ❑ 1) Utilizando un Timer
 - ❑ Su uso es sencillo. Se debe incluir:

```
#include <OgreTimer.h>
```
 - ❑ Se define y se crea:

```
Ogre::Timer* timer; y timer = new Ogre::Timer();
```
 - ❑ Se reinicia:

```
timer->reset();
```
 - ❑ El tiempo transcurrido (en ms) se obtiene con:

```
timer->getMilliseconds();
```
 - ❑ 2) Utilizando un contador y sumando el tiempo transcurrido en `frameRendered`
 - ❑ `evt.timeSinceLastFrame` indica el tiempo transcurriendo desde que se renderizó el ultimo frame.
- ❑ En el método `frameRendered` calcularemos cuándo
 - ❑ Tiene que bailar
 - ❑ Tiene que correr
 - ❑ Tiene que llevar (o no llevar) las espadas