

Materiales Cambio de mundos

Mientras que algunos objetos, como los arboles, tienen una animación para el cambio de mundos otros utilizan una transición entre los materiales de sueños y pesadillas. Para definir como se hará esta transición se ha creado los materiales de cambio de mundos. Esta transición utiliza los materiales de cada mundo y una textura para definir como se hará esta transición.

Utilización de la clase ChangeWorldMaterial

Inicialización

Los materiales de cambio de mundos están integrados en la RenderComponentEntity y para utilizarlos primero hay que inicializarlos con la siguiente función, en la GameObjectFactory.

```
void RenderComponentEntity::initChangeWorldMaterials(  
TChangeWorldMaterialParameters  
tChangeWorldMaterialParameters, RenderComponentEntityPtr  
pOtherComponentEntity);
```

Hay que pasarle la entidad a la cual se cambiará y los parámetros de cambio de mundos. En el caso de que el tipo de cambio de mundos no fuera necesaria la segunda entidad (por ejemplo con el tipo CW_EROSION_TRANSPARENT), se ignorará el parámetro.

Utilización

```
void RenderComponentEntity::setOriginalMaterials();
```

Pone los materiales originales a la entidad.

Nota Importante: Cuidado si tiene animación por coordenadas de textura ya que no se coordina bien con la de los materiales de cambio de mundos.

```
void RenderComponentEntity::setChangeWorldMaterials();
```

Pone los materiales de cambio de mundos a la entidad.

```
void RenderComponentEntity::setChangeWorldFactor(double factor);
```

Factor tiene que ir de 0 a 1. A 0 solo se mostrará el material original de la entity, a 1 se muestra el de la OtherEntity y en medio la transición.

```
void RenderComponentEntity::randomizeChangeWorldMaterials();
```

Da un valor aleatorio a las coordenadas de textura base de la textura utilizada para hacer la transición. Útil para hacer que cada cambio de mundos sea parecido pero no identico.

Parámetros del material de cambio de mundos

Se pueden definir los siguientes parámetros para cada material de cambio de mundos:

Type: Tipo de la transición

- CW_BLENDING hace blending básico
- CW_EROSION utiliza la textura y los parámetros que definiremos a continuación para hacer la transición entre los 2 materiales.
- CW_EROSION_TRANSPARENT lo mismo que la anterior pero en vez de un segundo material se hace transparente.

Blending Texture: Textura en escala de grises utilizada para hacer el blending. Se utilizará el color de esta textura para definir que partes cambian antes y que partes después. Para valores oscuros la transición se hará antes que para valores claros.

Blending Amount: Cantidad de blending entre los dos materiales, a 0 es completamente brusco y a 1 es el caso del blending básico.

Tiling: Tiling de la blending texture.

Randomize: Indica si se seleccionaran aleatoriamente unas coordenadas base para cada cambio de mundos o siempre se utilizaran las mismas. A falso anula el efecto de randomizeChangeWorldMaterials.

Scroll Animation: Animación del material. X y Y indican la velocidad animación de scroll. Z indica la velocidad animación de rotación en grados. Si lo ponemos todo a 0 no habrá animación.

Scroll Blending: Animación de la textura utilizada para el blending. X y Y indican la velocidad animación de scroll. Z indica la velocidad animación de rotación en grados. Si lo ponemos todo a 0 no habrá animación.

Asignación de parámetros de cambio de mundos

Por defecto:

Podemos asignar los parámetros por defecto para todos los materiales en el fichero Config/gameworldmanager-cfg.xml.

Desde Ogitor:

Se pueden especificar los parámetros para un game object concreto en ogitor asignándole los custom parameters correspondientes. Para hacerlo es necesario ir a la pestaña de custom parameters y hacer "Import in" del fichero Resources\Levels\Templates\ChangeWorldMaterial.ctp.

Otros:

Para algunos game object, como por ejemplo el cielo, hay que asignar los parámetros desde el fichero del game object en la carpeta Resources\Levels\GameObjects.

- **Cielo:** Para el cielo hay que modificar el fichero SceneManager.ctp

Funciones cambio de mundos

Durante el cambio de mundos la clase GameWorldManager y GameObject se encargan de gestionar el tiempo y llamadas a las funciones para gestionar el cambio de mundos, que veremos a continuación. Para definir el comportamiento específico de cada tipo de gameobject se heredan las funciones de activación, transición y final de cambio de mundos.

Ciclo de cambio de mundos

Activación

```
virtual void changeWorldStarted(int newWorld);
```

Esta función se llama al activar el cambio de mundos una sola vez, el parametro indica el mundo al que se esta cambiando. No tiene porque coincidir con el inicio de cambio de mundos general, es local del game object.

Nota Importante: si un objeto está cambiando de mundos y el cambio de mundos se vuelve a activar NO se volverá a llamar la función de activación.

Durante el cambio de mundos

```
virtual void changeToWorld(int newWorld, double perc);
```

```
//perc va de 0 a 1
```

Esta función se llama cada frame durante el cambio de mundos del game object, el parametro indica el mundo al que esta cambiando.

El porcentaje indica cuanto se ha completado del cambio de mundos al nuevo mundo, por ejemplo si perc=0.8 y newWorld es DREAMS estaremos a punto de terminar el cambio a sueños. Si en ese punto cancelamos el cambio la proxima llamada sera perc=0.2 y newWorld NIGHTMARES indicando que ahora cambiamos a pesadillas y solo llevamos un 20%. El porcentaje no tiene porque coincidir con porcentaje de mundos general, es local del game object.

Final

```
virtual void changeWorldFinished(int newWorld);
```

Esta función se llama al finalizar el cambio de mundos una sola vez, el parametro indica el mundo al que acaba de cambiar. No tiene porque coincidir con el fin de cambio de mundos general, es local del game object.

Cambio de mundos inicio de nivel y muerte

En estos casos se usará la versión rapida del cambio de mundos, donde solo se llamará a la funcion de final.

Opciones temporales cambio de mundos

Retraso de la activación

```
virtual void calculateChangeWorldDelay(double totalElapsedTime, double  
totalTime, int newWorld, double random);
```

Permite indicar un retraso de tiempo respecto al inicio y fin general del cambio de mundos para el game object. Útil para hacer los árboles por ejemplo. Random va de 0 a 1.

Duración del cambio de mundos

```
virtual void calculateChangeWorldTotalTime(double  
changeWorldTotalTime);
```

Indica el total de tiempo de cambio de mundos local del game object.