



GUÍA BÁSICA DE EXPORTACIÓN DE OBJETOS AL RAILWORKS

Versión 1.0 – 18-05-2010

1.- Introducción:

Este documento ha sido creado usando la experiencia y recopilación de información encontrada por Internet sobre el proceso de exportación de objetos para rutas en RailWorks.

Se ha usado el 3DS Max 2008 para la exportación y modelado de los diversos objetos que deseamos exportar. Se puede usar cualquier otra versión siempre que se disponga de los plugins correctamente instalados y operativos.

A continuación se describirán los siguientes puntos:

- 1- Estructura de directorios
- 2- Tipos de shaders
- 3- Forma de nombrar los objetos
- 4- Tipos de objetos
 - 4.1 Estáticos
 - 4.2 Animados
 - 4.3 Procedurales
- 5- Texturas estacionales
- 6- Texturas nocturnas
- 7- Sombras
 - 7.1 Sombras dinámicas
 - 7.2 Sombras estáticas
- 8- Cómo exportar las texturas a formato ACE
- 9- Tipos de blueprints de configuración de objetos
- 10- Limitaciones

1.- Estructura de directorios:

En Railworks hay una estructura de directorios que se debería mantener para hacer más fácil de organizar los objetos que creemos. Y para que en el Asset Editor en ciertas carpetas muestre un icono sobre que se trata.

Lo primero que hay que crear si no se tiene es una carpeta llamada Source en la ruta que se instala por defecto Railworks (C:\Archivos de programa\Steam\steamapps\common\railworks).

Esta carpeta contendrá todos los Providers y sus Products que creemos.

Dentro de la carpeta Source debemos crear nuestro Provider que es como se identificarán nuestros objetos o lo que creemos (Es el mismo principio que el KUID de TRS). En mi caso por ejemplo mi Provider sería Jjlor.

Dentro de un Provider puede haber varios Products que por ejemplo englobasen todos los objetos creados referentes a Renfe, Feve, FGV, etc por estandarizar un poco y ayudar al usuario a la hora de activar objetos en la ventana del editor de rutas del railworks.

Una vez creado el Provider y el Product, en el caso de un objeto como puede ser una caseta, la ruta quedaría como la siguiente:

...\Source\Jjlor\Renfe\Scenery\Buildings\ES_RNF_Caseta

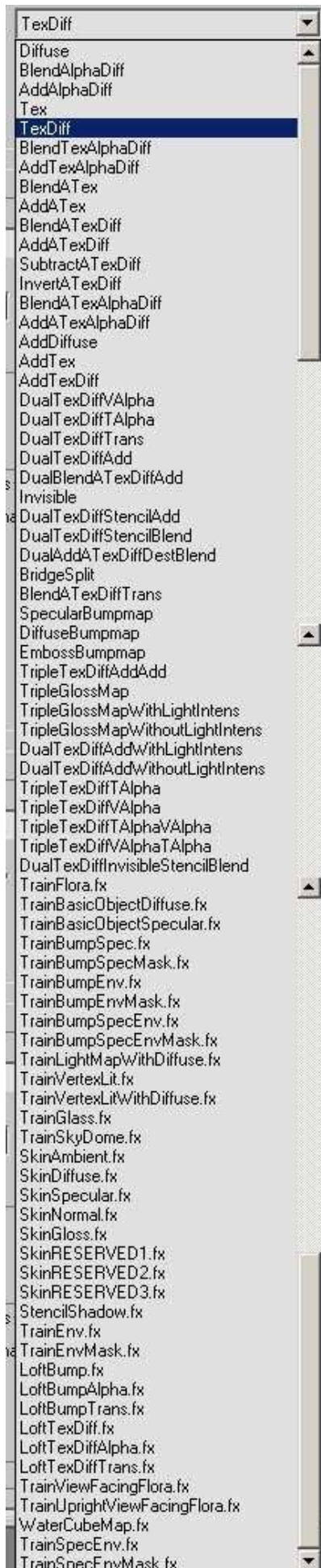
Poniendo la carpeta Scenery veremos en el árbol de directorios del Asset Editor como coloca un icono identificativos a esta categoría.

La carpeta Buildings es por mantener las cosas ordenadas según el tipo de objeto creado, ya que podríamos crear vehículos bajo la carpeta Vehicles, o muros bajo Procedural, etc.

Por estandarización, el nombre de la carpeta incluye el país al que pertenece (ES), seguida de la compañía (RNF) y por último el nombre del objeto.

A la hora de nombrar los objetos también usaremos el nombre de su carpeta para facilitar su búsqueda en la lista de objetos en el editor de rutas.

2.- Tipos de shader:



Los shader o sombreadores son diferentes tipos de materiales que se le pueden aplicar a las diferentes partes de un objeto sea cual sea. Entre la extensa lista de shaders que dispone el RW, los hay básicos (TexDiff), shaders que no les afecta las condiciones de luz y por tanto se usan para dar efectos de luz o en texturas nocturnas (AddATex), shaders que permiten transparencias, BumpMaps, reflexiones o los específicos para objetos procedurales.

La lista de la izquierda muestra la lista completa de shaders disponibles así como una breve descripción debajo de la lista desplegable.

Los que más se suelen usar en los objetos para decoración son:

-**TexDiff:** Es simplemente una textura con aplicación de color difuso.

-**AddATex:** Los objetos con este shades no les afectará la luz, usa un canal alpha.

-**SubstactATexDiff:** Este shader es usado para colocar sombras debajo de un objeto eliminando la luz en las zonas de la textura que esté en color blanco y dejando transparente las zonas negras.

-**TrainBumpEnv:** Se usa para agregar una textura de normales Bump. Dispone de dos slots para dos texturas, una diffuse y otra para el bumpmap.

-**TrainSpecEnv:** Todo objeto metálico llevará este tipo de shader con dos slots, uno para la textura diffuse y otro para crear el efecto reflejo.

-**StencilShadow:** Es el shader necesario para la sombra dinámica del objeto.

3.- Forma de nombrar los objetos:

En Railworks todas las partes que compongan un objeto han de seguir unas normas básicas de nombrado. Para ello se deberá nombrar de la siguiente forma:

X_YYYY_Z

X: es el número de LOD empezando por 1.

YYYY: es la distancia a la que se aplicará el LOD. Si la distancia es inferior a 1000m se rellenará con ceros hasta completar los cuatro dígitos.

Z: es el nombre de la pieza. En ciertos casos va precedido de una palabra clave como puede ser **_shadow_**, **_fxnight_**, **_fxday**.

En caso de que una misma pieza tenga más de un LOD, un ejemplo sería el siguiente:

Nombre
1_0100_chimenea
2_0500_chimenea
3_1000_chimenea

4.- Tipos de objetos:

Básicamente existen tres tipos de objetos (hay alguno más pero en esta guía sólo veremos los más usados) que son objetos estáticos, objetos animados y objetos procedurales.

4.1 Objetos estáticos:

La mayor parte de los objetos de una ruta son de este tipo. Pueden tener transparencias, reflejos, BumpMap, o simplemente una textura. Su configuración es muy básica y ser describirá paso a paso más adelante.

Para crear el fichero **.IGS (Kuju Intermediate Geometry)**, se va al menú File y se selecciona la opción Export, se especifica el formato anteriormente citado y se da a exportar con el nombre deseado.

Aparecerá una ventana con diversas opciones, dejamos todas tal y como parecen salvo la de que genere los LODs de forma automática. Le damos a exportar y saldrá una lista de mensajes con las partes que contiene el objeto así como sus nombre y si hubo algún tipo de error.

4.2 Objetos animados:

Son idénticos a los anteriores salvo claro está que tienen algún tipo de animación.

Las animaciones permitidas en Railworks son de rotación y de traslación, no estando soportada la de escalamiento.

A la hora de nombrar las partes no tiene nada de especial y es igual que cualquier otro objeto y se exporta de la misma manera. Aparte de crear el fichero de geometría **.IGS (Kuju Intermediate Geometry)**, se ha de exportar si animación en formato **.IA (Kuju Intermediate Animation)**. Para ello, se ocultan todas aquellas piezas que NO tengan animación y desde el menú File se escoge la opción Export. En la lista de formatos seleccionamos el tipo de formato **.IA** anteriormente citado y le damos a guardar con un nombre que defina la acción de forma más descriptiva posible.

4.3 Objetos procedurales:

Los objetos de este tipo son los usados para crear vías, andenes, vallas y todo tipo de objetos lineales. Deben tener su centro siempre en las coordenadas (0,0,0) y se el Railworks los crea mediante extrusión.

En el 3DS Max debemos darle una longitud al objeto para que nos permita asignar una textura que es la que se repetirá cuando se coloque el objeto en la ruta.

Este tipo de objetos puede tener tres tipos de geometrías asociadas que se especifican en el blueprint de configuración que veremos más adelante, y son una geometría inicial, una intermedia y una final.
 Los objetos procedurales deben tener un shader específico que viene indicado con la palabra Loft. Permite transparencias y BumpMaps.

5.- Texturas estacionales:

RailWorks permite implementar texturas estacionales tanto en objetos como en material rodante. Sólo hay que seguir una convención de sufijos a la hora de nombrar las texturas.

Por defecto la textura que coloca el RailWorks es la de verano y en el 3ds max deberemos texturar siempre con la textura por defecto.

Ejemplo:

Nombre	Estación
Caseta.ace	Defecto
Caseta_sp.ace	Primavera
Caseta_su.ace	Verano
Caseta_au.ace	Otoño
Caseta_wi.ace	Invierno

Si no se quiere poner texturas estacionales no es necesario poner ningún sufijo. Al exportar un objeto desde el Asset Editor, el programa mirará si existen texturas con sufijo en la carpeta de texturas, y en caso afirmativo las exportará automáticamente.

6.- Texturas nocturnas:

Para dar más realismo a los edificios y en general a los objetos que por la noche deben estar iluminados, se usa un efecto para simular la luz o que un determinado polígono no se vea afectado por la variación de iluminación.

Este efecto se crea desde el modelo en el 3ds max añadiendo al nombre del objeto fx_night. Esto hará que sólo sea visible en tiempo de noche.

Nombre	Tiempo	Descripción
fx_night	Noche	Sólo se renderizará de noche.
fx_day	Día	Sólo será visible por el día.

Para el efecto de luz en farolas por ejemplo, usaremos el shader AddATex y aplicaremos una textura como la siguiente.



glow_lens_nm.ace



glow_floor_nm.ace

7.- Sombras:

Normalmente los objetos como vehículos, puentes, marquesinas o el material rodante llevan dos tipos de sombras. Son las que veremos a continuación:

7.1 Sombra dinámica:

La sombra dinámica está formada por un objeto algo más grande y con menos detalles que el objeto al que se le quiere colocar la sombra dinámica, y tendrá una textura en un color gris medio al que no es necesario hacer un unwrap, simplemente se aplica al objeto.



El objeto ha de contener en el nombre una palabra reservada para que RailWorks detecte que es un objeto de sombra:

Nombre	Descripción
1_0350_caseta	Objeto al que queremos colocar la sombra.
1_0075_shadow_caseta	Objeto de sombra del objeto principal al que queremos añadir la sombra dinámica.

Al ser un objeto especial también usa un shader especial y con ciertas características:

- El shader que se debe usar es **StencilShadow**.
- El nombre del material debe ser según el ejemplo anterior, **shadow_caseta**.

7.2 Sombra estática:

La sombra estática es bastante más simple que la anterior ya que es sólo un polígono con el shader **SubstractATexDiff**.

En la configuración del shader hay que establecer la propiedad Z-Buffer Mode en TEST ONLY.

La textura es de 24bits por lo que no necesita canal alpha para la transparencia, será negra en donde sea transparente por completo y gris o blanco según el grado de transparencia que se desee. Esto es idéntico a como se haría un canal alpha.



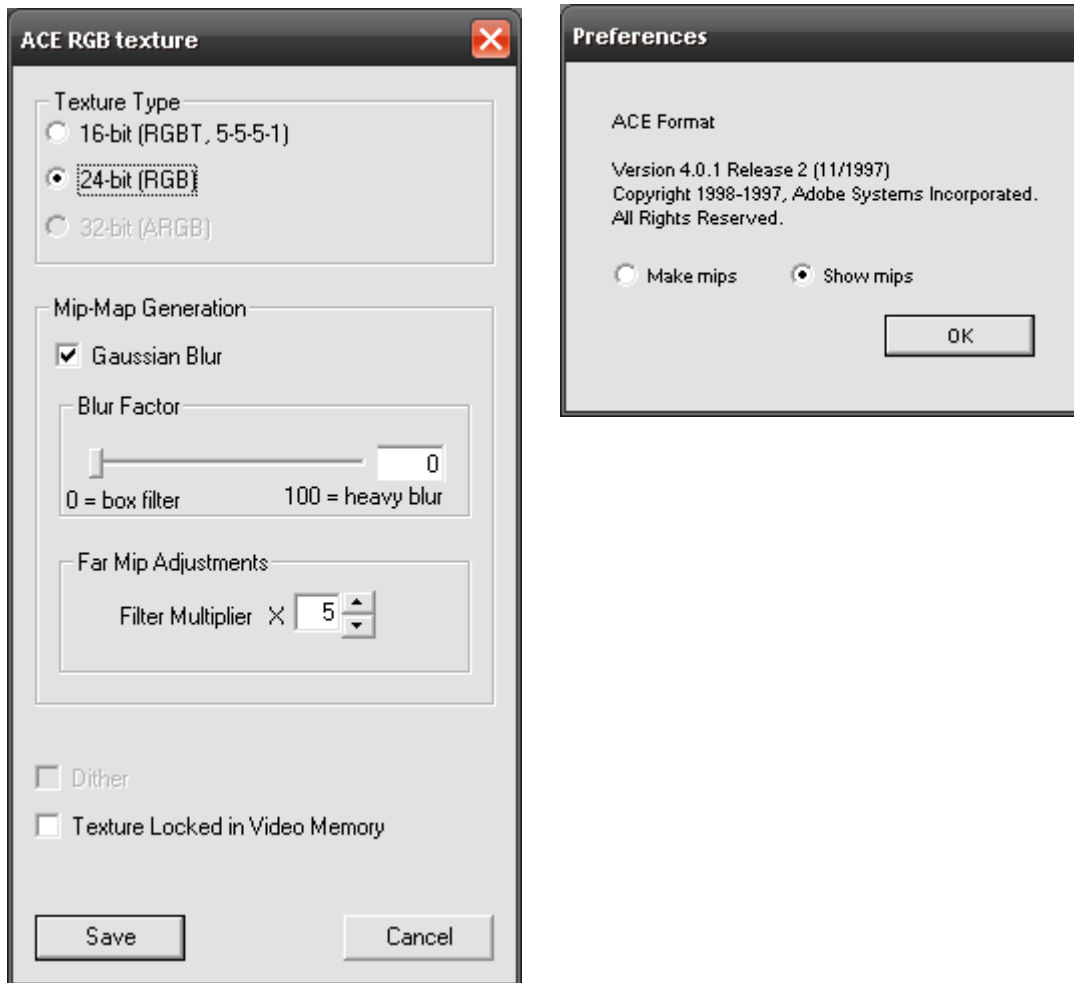
8.- Cómo exportar las texturas a formato ACE:

Junto a los plugins de exportación para diversas versiones del 3ds max, viene también un fichero comprimido con los plugins para manipular texturas en formato **.ACE**.

Una vez instalados en sus correspondientes carpetas los dos ficheros que vienen, al abrir el PhotoShop veremos que tenemos un nuevo formato de fichero tanto al abrir como al guardar.

Una vez tengamos nuestra textura finalizada, sólo tenemos que seleccionar en el menú archivo **guardar como...** e indicar el tipo de formato **.ACE**. Debe llevar el mismo nombre que la textura que hayamos aplicado al modelo, por ejemplo si nuestra textura es **Caseta.psd**, debemos guardar como **Caseta.ace**.

Al guardar aparecerá la siguiente ventana:



Si la textura tiene canal alpha, directamente estará seleccionado el tipo de textura en 32bits. La opción de generar Mip-Maps yo siempre desactivo el check de Gaussian Blur porque he experimentado que con la distancia la textura empieza a verse borrosa dando un efecto feo, por lo que nunca lo uso.

Respecto a la ventana principal no se cambia ninguna opción más.

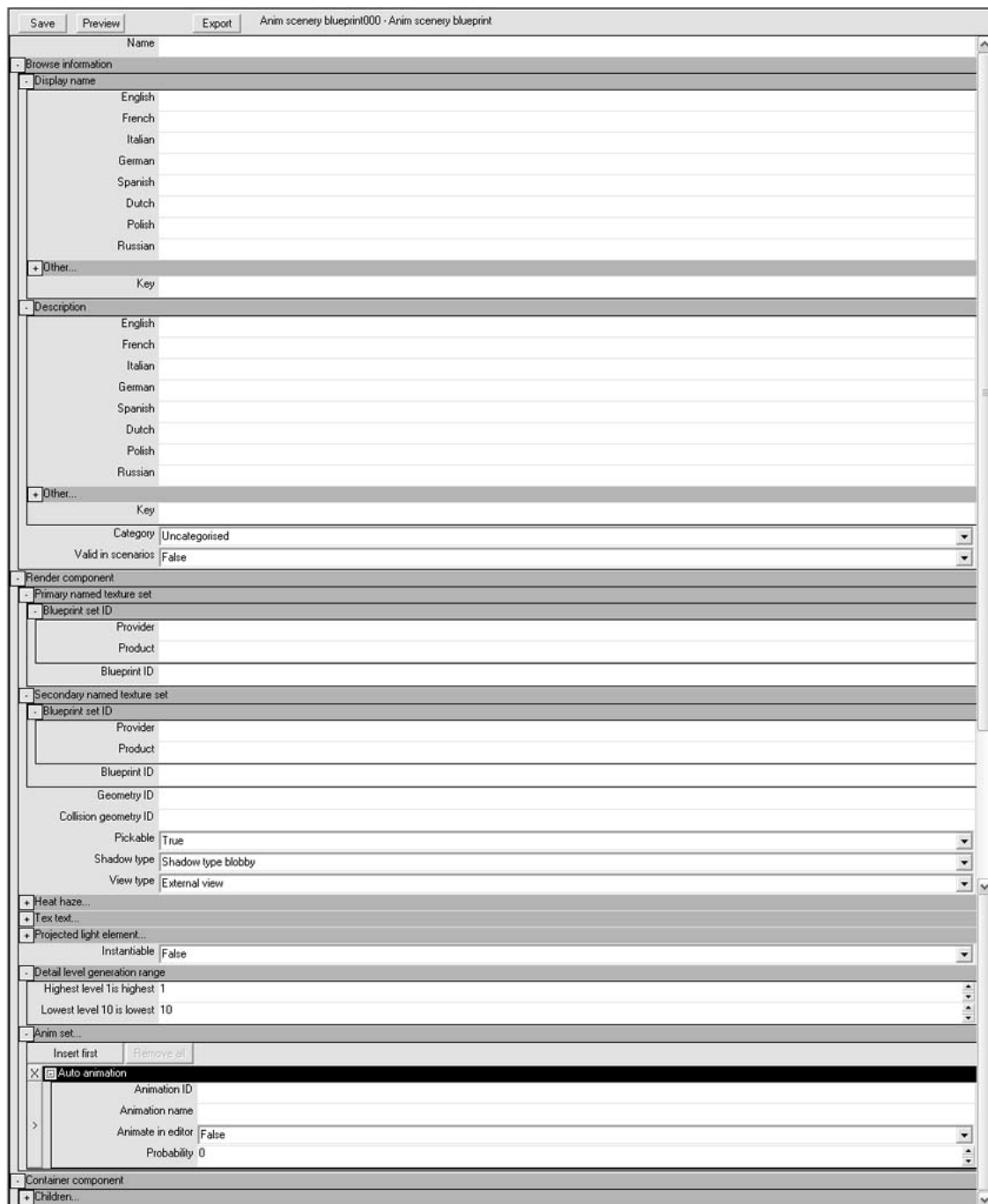
En la segunda ventana que aparece, se puede elegir entre la opción de crear MipMaps para que se vaya reduciendo el tamaño de la textura según la distancia o dejar la opción Show Mips para que no los genere al salvar la textura.

En caso de abrir una textura que contenta MipMaps, los mostrará automáticamente.

9.- Tipos de Blueprints de configuración de objetos:

Los **blueprints** son los ficheros de configuración que usa RailWorks. Hay muchos tipos de blueprints pero por ahora sólo nos interesan tres que hacen referencia a los tres tipos de objetos descritos anteriormente. Los nombres de los blueprints son los siguientes:

Nombre	Descripción
Anim Scenery blueprint	Se usa para objetos con animación.
Scenery blueprint	Se usa para la mayoría de objetos.
Loft Section blueprint	Para objetos procedurales.



Los blueprints son los ficheros de configuración que usa RailWorks donde aparecen los nombres, propiedades y características de un objeto así como referencias a otros blueprints externos.

Todos los blueprints tienen campos iguales salvo algunos específicos del tipo de objeto que son los que se detallarán a continuación.

En la mayoría de blueprints tienen un campo nombre que no aparecerá en el editor pero sirve poner un nombre claro al blueprint. A continuación aparece el campo **Display Name**, que es el texto con el que aparecerá en la lista de objetos en el editor de rutas en el idioma en que esté configurado el RailWorks. Debe rellenarse todos los campos de **display name** o de lo contrario no aparecerá ese objeto en la lista del idioma que esté vacío.

También en este apartado es usar la convención de nombres que ya se comentó en el primer punto para facilitar la búsqueda y agrupar objetos de un mismo país y compañía.

El campo **Description** es sólo significativo para el autor ya que no se mostrará en el editor.

Según el tipo de objeto que estemos creando, tenemos una serie de categorías en donde clasificarlo según sea más conveniente.

Por defecto aparece en **Uncategorised** (Sin categoría).

Si el objeto que hemos creado es parte de otro como sucede en las geometrías intermedias o de los extremos de los objetos procedurales, si no queremos que sean seleccionables la categoría será **Exclude from Browne list** (Excluir de la lista).

Los campos siguientes que nos interesan son **Geometry ID** y **Collision geometry ID** que hacen referencia al fichero .IGS que exportamos desde el 3DS Max y que en ambos casos es el mismo fichero.

Para evitar problemas de escribir la ruta mal, en el árbol de directorios de la izquierda localizamos el icono con cubo azul con dos flechitas amarillas a los lados que es nuestro fichero .IGS. Con el botón derecho del ratón hacemos clic sobre el y en el menú que aparece seleccionamos **Copy filename** y después con **CTRL + v** lo pegamos en la casilla del campo Geometry ID y Collision geometry ID.

Si nuestro objeto no tiene sombra dinámica o ningún tipo de sombra, el campo **Shadow type** lo dejaremos en **Shadow type blobby**, que creará una sombra redonda bajo nuestro objeto. En caso contrario de que si tenga sombra dinámica, cambiaremos esta opción a **Shadow type none**.

Detail level generation range especifica con que nivel de detalle será visible según la configuración que tenga el usuario final.

Esto ya depende de cada uno cuando quiere que se vea su objeto en una ruta que normalmente será siempre. Si se configuran valores muy altos, en usuarios que tengan máquinas más lentas y por lo tanto no tengan el máximo nivel de detalle activado, puede que no le aparezca el objeto.

El primer campo específico de objetos animados es **Anim set**, aunque este lo veremos en más de un blueprint pero no se usa.

En este campo es donde se configura el nombre y el fichero .ia de la animación.

Hay animaciones que deben llevar un nombre reservado para que funcionen en el simulador pero para una animación simple de rotación por ejemplo se puede usar cualquiera.

Para especificar el fichero .ia seguimos el mismo método que usamos para los de geometría, identificamos en el árbol de directorio el icono que tiene una rueda dentada azul y pegamos la ruta con CTRL + v.

El nombre que hayamos especificado es el que aparecerá en la ventana intermedia en el **Asset Editor** cuando lo hemos abierto desde la ventana principal del simulador y le

hemos dado a **Preview** que es desde donde podemos comprobar que todo es correcto en nuestro objeto sin necesidad de entrar en el simulador y colocar el objeto en una ruta.

En el caso de que queramos añadir un objeto externo realizado por un tercero u otro que no esté contenido en la geometría principal del objeto que estamos realizando, se especificará en el campo **Children**.

Desplegamos la lista y agregamos un children (hijo) haciendo clic en **Insert first**.

Se creará un campo llamado **S Child** que es el campo que contiene la configuración de ese hijo con todos los parámetros necesarios.

En primer lugar hay que especificar un nombre que aparecerá en la ventana de preview en el Asset editor.

A continuación hay que especificar el Provider y Product del objeto que queremos referenciar al nuestro. En el caso de que fuese por ejemplo una persona que está sentada y que viene con el Railworks, el Provider sería **Kuju** y el Product **RailSimulator**. Estos parámetros depende de quién es el creador del objeto externo.

Para conocer el **Blueprint ID** navegamos hasta la carpeta Assets de nuestro RailWorks y entramos en la carpeta del Provider y Product que nos interesa. En el ejemplo citado unas líneas atrás la ruta para una mujer que aparece sentada sería la siguiente:

scenery\Characters\Female_Sitting_01.xml

Los ficheros **xml** compilados tendrán una extensión **.bin** con lo que habrá que buscar el nombre del fichero con esta extensión pero al escribir la ruta lo haremos poniendo **.xml**.

Para ajustar un Children lo haremos desde la vista previa dentro del Asset Editor. Es algo inexacta la colocación a mano ya que con muy poco nos podemos y a cientos de metros lejos de nuestro objeto. Lo mejor es hacerlo a mano abriendo el xml con el Notepad o Wordpad.

Una vez abierto el xml, buscamos por el nombre que le dimos al children y cuando lo tengamos localizado, las siguientes líneas al provider, al product y a la ruta del xml aparecen un muchos valores que empiezan por 1.000000 y 0.000000. Bien si vamos hasta el final y en los últimos cuatro valores que aparecen escribimos las coordenadas del centro del objeto que aparecen como X, Y, Z. El último valor lo dejamos en 1.000000. En caso de que queramos rotar el objeto se hace variando los valores 1.000000 pasándolos a valor negativo si por ejemplo el conductor de una locomotora esta mirando hacia atrás, el primer valor y el tercero que aparecen con 1.000000 lo pondremos con -1.000000.

```
<sFloat32>
  <Element>
    <Value>0.000000</Value>
  </Element>
</sFloat32>
<sFloat32>
  <Element>
    <Value>0.800000</Value>
  </Element>
</sFloat32>
<sFloat32>
  <Element>
    <Value>-1.200000</Value>
  </Element>
</sFloat32>
```

X

Y

Z

Los objetos **procedurales** tienen campos específicos para este tipo de objetos y otros comunes a los del resto de blueprints. A continuación se explicarán los más importantes para que un objeto de este tipo se visualice correctamente.

El primer campo específico que nos encontramos es **Editor ground Offset** que especifica en que altura sobre el eje **Y** se colocara el objeto procedural tomando como referencia el **gizmo** del objeto. Por ejemplo, si nuestro objeto tiene el gizmo en el 0,0,0 y queremos que aparezca a 4 m de altura porque es un cable de alta tensión, en este campo introduciremos el valor 4. No se la razón pero el editor tiende a redondear el valor introducido la primera vez dando un valor aproximado con muchos decimales. Si volvemos a introducir el valor ya deja el que hemos escrito.

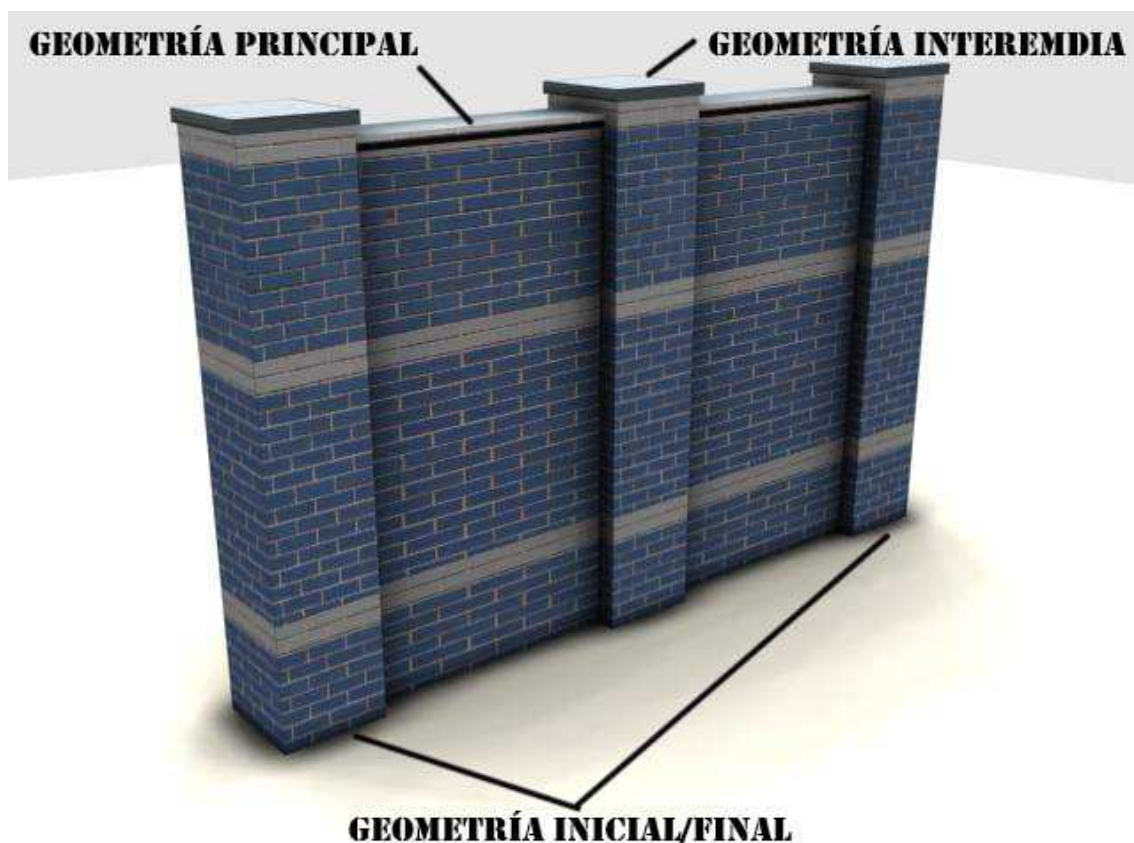
En la sección **Render** se especifican las geometrías principales del objeto procedural, del objeto en si y si se permiten bifurcaciones de la geometría secundaria que las permite. Estos campos son **Cross section ID** para el objeto principal y **Cross section CAP ID** para el objeto secundario si se especifica.

Los dos campos siguientes indican el ancho y alto del objeto en metro.

A continuación viene la referencia de tres objetos que pueden o no estar presentes. Son las geometrías iniciales, intermedias y finales de un objeto procedural. Estos objetos son como cualquier otro objeto salvo que en su categoría se utilizó la opción de excluir de la lista para no hacerlos seleccionables desde el editor.

Para mayor comodidad aconsejo que se creen antes los blueprints de estos objetos antes que el del procedural con el fin de hacer todo en un orden y no tener que estar abriendo y cerrando ficheros en el asset editor.

Para referenciar una de estas geometrías sólo hace falta indicar el Provider, el Product y la ruta del XML que en este caso como si lo hemos creado nosotros mismos si buscaremos el xml en el árbol de directorios de la izquierda buscando el icono del cubo marrón y haciendo clic con el botón derecho del ratón seleccionamos Copy filename y lo pegamos en el campo Blueprint ID.



El campo **Population frequency** se refiere a la geometría intermedia e indica cada cuántos metros debe repetirse ese objeto a lo largo del objeto procedural.

Por último comentar los campos **Minimum width** y **Maximum width** que son como su nombre indica, el ancho mínimo y máximo del objeto. Tiene una limitación de 100m de largo.

10.- Limitaciones:

Railworks sólo admite dos tipos de objetos cuando se está modelando, **Editable Mesh** y **Editable Poly**. **NURBS, Splines y Patches** no son soportados. Además tampoco soporta instancias, con lo que al clonar objetos hay que seleccionar la opción de **Copy**.

Los shaders que tienen más de un slot configurable, deben llevar siempre una textura seleccionada, de otra forma dará error al exportar el modelo.