

EXPORTER UNE LOCO DE 3D CANVAS VERS RAILSIMULATOR

par Kevin Martin et Richard Scott

Mai 2008

Version française par Fildefer

mars 2010



Préambule

Cette partie comporte les observations du traducteur.

Ce tutoriel a été fait pour RailSimulator, mais il peut s'appliquer à RailWorks.

Pour plus de clarté, j'ai conservé certains noms en anglais car c'est ceux-ci que vous rencontrerez au cours de votre travail.

Dans RailWorks, il n'y a pas les exemples de RSDL, aussi, vous devrez les copier de RailSimulator.

Attention, vous devrez créer le dossier "Source" dans RailWorks.

J'ai ajouté quelques commentaires que je pense utiles en espérant que les auteurs ne m'en voudrons pas.

Ce tutoriel est conçu pour une locomotive à vapeur, mais peut aisément s'appliquer à d'autres modèles comme les wagons, les voitures et, bien sûr, les engins mus par d'autres sources d'énergie.

En dehors de l'aspect purement tutoriel de création, celui-ci fourmille de choses intéressantes quand aux blueprint en particulier!

Ce tutoriel a été conçu pour l'exportation à partir de 3D Canvas mais il apporte des conseils applicables à d'autres situations.

Personnellement, il m'a permis de comprendre bien des aspects obscurs comme la numérotation automatique.

J'espère que sa traduction vous donnera envie de vous lancer dans la conception de modèles ou vous apportera des réponses utiles.

Le document original date de 2008, certains points peuvent se révéler différents.

Philippe dit Fildefer

Cette documentation doit être utilisée conjointement à la "Railsimulator dev docs 2.06 Engine Blueprint et 2.07 Wagon Blueprint. Ceux-ci fournissent une documentation plus approfondie de la manière d'utiliser les "blueprints" correctement.

Introduction

Le but de ce tutoriel est de tenter de vous apporter le maximum d'aide possible pour exporter une locomotive de 3D Canvas vers Rail Simulator. Il ne vous apprendra pas à utiliser 3D Canvas, il est supposé que vous l'avez déjà utilisé. Nous n'approfondirons ses fonctions que quand elles sont liés au processus d'exportation. De même, nous n'approfondirons pas l'utilisation des blueprints et leurs aspects techniques. Encore une fois, l'idée est juste de vous aider par rapport à Rail Simulator. Un fois que vous aurez réussi, vous pourrez nous en apprendre plus! De même, ce tutoriel ne vous apprendra pas à réaliser une cabine, il utilise les "alias" des cabines, des sons, etc... de RSDL pour réaliser une locomotive. Nous sommes sûr d'avoir oublié un grand nombre de choses, mais c'est un début.

Quelques conseils avant de commencer: Derek Siddle de RS recommande de commencer à exporter vers RS le plus tôt possible. C'est un très bon conseil! Si vous avez commencé à modéliser et que vous avez appliqué une texture de base, vous pouvez essayer d'exporter vers Rail Simulator directement. Vous aurez plus de chances de réussir avec un modèle simple. Ensuite, au fur et à mesure que vous ajouterez des détails et de meilleure textures, vous pourrez faire d'autres exports. En utilisant cette méthode, si un problème survient, vous aurez une meilleure idée de ce qui le provoque.

Kevin et moi, nous vous conseillons d'utiliser le "RS Asset Editor" le moins souvent possible. L' Asset Editor est utile pour positionner les attelages, les fumées, les échappements et les orifices d'approvisionnement. Ceci requière parfois l'usage du "Gizmo". Ceci dit, vous pouvez choisir la solution du bambou sous les ongles ou utiliser le Gizmo, c'est à vous de voir. Mais si vous choisissez la torture par la nourriture du panda chaque fois, c'est plus douloureux. Malheureusement, vous devrez utiliser le Gizmo pour positionner les attelages et les fumées. Donc utilisez-le quand c'est absolument nécessaire. Nous avons remarqué que l'Asset Editor réinitialise le positionnement. Ceci semble se produire lorsque vous utilisez un élément qui se trouve en dehors de votre dossier de développeur. De ce fait, nous vous conseillons de ne placer les effets de particules (fumées et échappements) que quand votre modèle est terminé. En dehors de l'usage particulier de l'Asset Editor, il est recommandé d'utiliser le Blueprint Editor à la place car c'est un programme beaucoup plus stable.

"Program Files\Rail Simulator\Blueprint Editor.exe".

Quand vous utilisez l'**Asset Editor**, **Rail Simulator** doit être **en mode fenêtre** sinon vous ne verrez pas l'écran de prévisualisation (preview screen). Néanmoins, quand

vous utilisez le **Blueprint Editor**, **Rail Simulator** peut être en **plein écran** et vous pouvez utiliser **alt + tab** pour naviguer entre les deux. Ou bien, vous pouvez avoir Railsim (en plein écran ou en fenêtre) et 3D Canvas ouverts en même temps. Ceci rend l'ajustement du modèle très facile car vous pouvez exporter un modèle et voir aussitôt le résultat dans Railsim.

Notre feuille de route pourrait être:

- Commencer la modélisation dans 3D Canvas
- Utiliser l' Asset Editor pour un premier export et positionner les attelages
- Continuer la modélisation et utiliser le Blueprint Editor pour tester les exports
- Utiliser l' Asset Editor pour positionner les effets de particules et les sons
- Ajouter les cabines, conducteurs, etc... à votre loco (si vous ne les faites pas vous même). Il y a deux méthodes pour cela:

- Utiliser **RS Bin Editor** pour éditer les fichiers .bin
- Ou utiliser Notepad2 pour éditer les fichiers .xml. Ensuite, vous devrez les convertir en .bin avec serz.exe. Cette façon de faire est sensiblement plus longue mais a l'avantage de permettre l'utilisation d'outils d'édition et de recherche (ce qui peut être utile).

(n.d.t: ou utiliser RS Tools de Mike Simpson)

Notez, s'il vous plait, que ce tutoriel est une première approche. Il n'est pas une méthode pas à pas, juste un guide. Nous restons ouvert à toutes vos suggestions et nous pourrions les ajouter - à la manière d'un "wiki". Nous voulons seulement plus de modèles dans notre simulateur, alors allez-y!

Exporter depuis 3D Canvas

- Premièrement, vous devez créer une série de fichiers correctement nommés dans le répertoire **Source** de Railsim. La manière la plus simple est de copier le **répertoire developer** et de le renommer à votre propre nom.
 - la seconde étape dépend du type de modèle que vous exportez. Si, par exemple, vous exportez une loco vapeur appelée (pour les besoins de ce document) tankloco, copiez le répertoire "exemple" situé dans addon\rail vehicles\ steam. Renommez-le "tankloco".
 - Changez dans tous les répertoires de "tankloco" les références "exemple" en "tankloco".
 - Utilisez l'option "rechercher et remplacer" de Notepad2, ouvrez chaque fichier .xml du répertoire "tankloco" et renommez chaque occurrence "exemple" en "tankloco". Procédez ainsi pour renommer "developer" avec le nom que vous avez choisi dans le répertoire principal "source".

Cela semble un peu compliqué, mais cela ne prend que quelques minutes et vous permet d'avoir un ensemble de blueprints qui, s'ils ne sont pas adaptés parfaitement à votre modèle, vous permet tout au moins de démarrer. Nous pourrions affiner plus tard.

Nous supposons que vous avez une version à jour et avec tous les patches (pour Rail Simulator) et tous les "example". Ceci vous permet d'avoir tous les fichiers nécessaires à l'exportation ou utilisable comme référence en attendant d'avoir les références correctes. Par exemple, l'"example smoke" que vous pouvez ajouter dans l'Asset Editor et modifier ensuite pour l'améliorer.

Assurez-vous que vous avez les répertoires de destination nécessaires dans le répertoire Rai Simulator pour recevoir vos fichiers. Idéalement, ce pourrait être:

- un répertoire portant votre nom ou votre alias de créateur dans le répertoire "Source".
- dans ce répertoire, un répertoire "Addon"
- dans celui-ci, les répertoires pour le type de modèle que vous créez (dans cet exemple "RailVehicles"). La méthode la plus simple est de copier tous les répertoires de Developer\Addon\ "typedevéhicule" dans votre répertoire. Vous devrez aussi créer dans le dernier répertoire un répertoire pour votre loco (ici: Tankloco) qui comprendra tous les répertoires nécessaires.

n.d.t: par exemple, pour tankloco

Source

 "nomdedeveloppeur"

 Addon

 RailVehicles

 Steam

 Tankloco

 Default

 Bogies

 Cabview

 Driver

 Engine

 Textures

 Simulation

 Tender (si besoin)

Fichiers .igs

Ce sont les fichiers "formes" qui constituent votre modèle. Créez votre modèle en un seul fichier. Autrement dit, incluez les bogies et les roues comme sous-objets de votre objet principal (Main). Quand vous exporterez, ils seront placés correctement. Bien que 3D Canvas les renomme pour vous, il est plus simple d'utiliser les conventions de Railsim dès le départ. Ainsi, nommez votre premier bogie bo01 (ou si vous n'utilisez pas le système de LOD automatique: 1_0050_bo01 (par exemple et en fonction des caractéristiques du LOD)), votre second bogie bo02, etc... Les roues seront nommées wh01, wh02, etc... Ceci vous aidera dans le bogie blueprint. Rappelez-vous que les roues motrices (d'une loco vapeur) n'ont pas besoin de bogie mais elles devront être animées alors que celles des bogies le seront automatiquement.

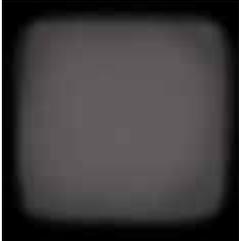
Ne tenez pas compte de la documentation de RS pour le nombre de séquences d'animation. Comme pour MSTs, utilisez 8 ou 16 séquences.

Si il faut un tender, créez-le et exportez le comme un modèle séparé.

Les ombres doivent être ajoutées au groupe principal.

Pour les ombres au sol (ground soft shadow)

Créez un rectangle plat (gardez juste le sommet d'un cube) et placez-le sous votre modèle avec Y un peu au-dessous de 0.



Utilisez le nom approprié: pour cela, clic droit sur l'objet et entrez **softshadow** dans la case "custom".

Utilisez la texture appropriée: pour cela, créez un simple carré de couleur noire avec une bordure de blanc "au spray", comme dans l'exemple ci-dessus (nommez-la comme vous voulez). Entrez le nom d'usage pour cette ombre **SubtractATexdiff**. Vous n'avez pas besoin de créer une couche alpha pour avoir un effet de transparence autour. Blanc est totalement opaque, noir est totalement transparent et vous pouvez utiliser les gris entre les deux.

Pour l'ombre principale (main shadow)

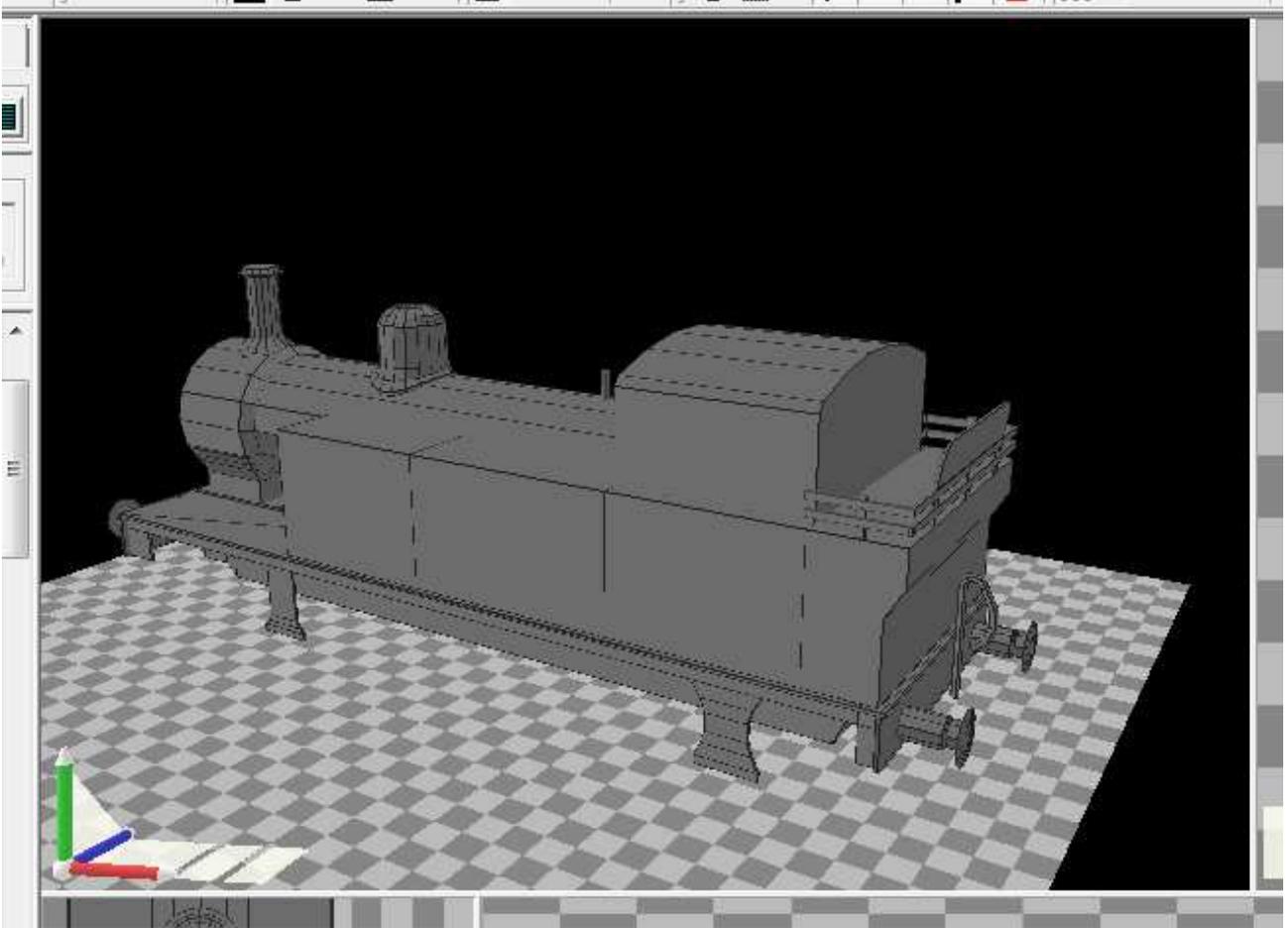
Créez une version très simplifiée de votre modèle (vraiment très simple). Positionnez là au mêmes coordonnées que votre loco et assurez-vous qu'elle soit légèrement plus petite. Si l'ombre de l'objet dépasse celui-ci, cela peut projeter l'ombre comme si elle se trouvait en dehors de l'objet.

Il semble parfois que faire les ombres soit un peu compliqué et qu'il soit facile de voir ce que RSDL appelle des artefacts (certaines parties de l'ombre semble se projeter à l'infini (et au delà!).

"Creasing" semble utile dans ce cas. Les objets ronds apparaissent mieux avec un "creasing" de 100 tandis qu'un "creasing" de 0 convient mieux aux objets à angles pointus. Ca peut marcher à cette occasion. Ainsi, les cubes allongés (par ex. les cotés d'un réservoir) semble bénéficier d'une division (voir les soutes du modèle de Jinty ci-dessous).

Vous pouvez utiliser la texture de la partie principale pour texturer l'ombre, mais il est préférable de sélectionner une couleur différente pour mieux éclairer l'ombre. De plus, si elle est au même endroit que la texture principale, il peut y avoir confusion! La chose importante à se rappeler est de donner le nom correct à la texture de l'ombre! Si vous créez le modèle pour l'ombre à part, cela contribuera certainement à éviter les artefacts lorsque vous ajouterez d'autres ombres.

Vous pouvez essayer de cloner une ou plusieurs parties de votre modèle pour les utiliser comme ombre pour accélérer votre travail. Ça semble bien fonctionner. Essayez d'utiliser la fonction "solidify" sur ces parties car la documentation (Dev Docs) recommande d'utiliser des objets sans trous.



Modèle pour l'ombre de la loco Fowler 3f

Utilisez les noms d'usage:

Clic droit sur l'ombre et entrez dans le champ "custom" **shadow_locomotive**.

n.d.t: "locomotive" est le nom pour l'exemple. Vous pouvez choisir le nom qui convient le mieux à votre objet.

Utilisez la couleur que vous souhaitez. Ce peut être un simple carré gris de 128X128 nommé comme vous voulez mais vous devez entrer le nom **StencilShadow.fx** pour le type de texture.

Pour les feux

Supposons que vous ayez deux feux à l'avant de la loco (une loco peut n'avoir qu'un seul ou trois feux à l'avant, bien sûr).

Pour l'avant, créez des carrés (juste assez grands pour la taille du verre de lampe) et nommez les:

lights_fwdhead et **lights2_fwdhead** ainsi que **lights_revtail** et **lights2_revtail**

Ces carrés seront situés au même endroit à l'avant du modèle (la simulation choisit ceux qui doivent être utilisés).

Si vous devez avoir des feux à l'arrière:

Même procédé, mais nommés

lights_fwdtail et **lights2_fwdtail** et **lights_revhead** et **lights2_revhead** et, bien sûr, situés à l'arrière du modèle.

*n.d.t: pour mieux comprendre les noms, fwd=forward=marche avant, rev=reverse=marche arrière, head=tête, tail=queue. Ce qui donne pour **lights_fwdhead**, feux avant en marche avant et **lights_revtail** feux avant en marche arrière (les feux avant se trouvant alors en queue de train).*

Les carrés peuvent être placés dans le même groupe et cela marche bien.

Les textures (une jaune ou blanche pour les feux avant et une rouge pour les feux arrière) sont créés en 64x64 ou 32x32 au format .tga, avec une couche alpha pour la transparence (translucency ne semble pas marcher).. Nommez-les comme vous voulez mais assurez-vous de les appliquer comme **AddTex**.

Les feux s'allument avec "H". Une fois pour le premier niveau de luminosité, deux fois pour le niveau supérieur. Ils s'éteignent avec "H+majuscule". La simulation est capable de distinguer si la loco est couplée ou non pour l'allumage des feux situés en arrière de la loco.

LOD'S

Pour les lod's, vous avez deux choix possibles:

- Soit nommer vous même les parties en respectant les conventions d'usage de RS. Par exemple, pour un premier bogie en lod 1, visible de 0 à 32m: 1_0032_bo01, en lod2 visible de 0 à 100 m: 2_0100_bo01, etc... Le premier chiffre est le numéro d'ordre du lod, le second (toujours à 4 chiffres), la distance de 0 à ### m durant laquelle l'objet reste affiché par le simulateur, les derniers caractères étant le nom d'usage à employer dans RS.

- Soit utiliser 3D Canvas pour créer automatiquement les LOD'S. Ceci vous laisse moins de possibilité de contrôle, mais est la façon la plus simple de procéder. Canvas est assez doué pour faire tout le "renommage" pour vous.

Pour exporter le modèle

Utilisez le Train Simulator Wizard (voir les tutoriaux d'Amabilis pour cela) ou sélectionnez l'objet principal de votre modèle (cela suppose que vous ayez hiérarchisé les éléments auparavant), allez dans le menu File - Export et sélectionnez Trainworks. Assurez-vous de bien choisir Rail Simulator Intermediary Geometry Format (.igs) et exporter vers le dossier correct.

Pour cet exemple:

Source*votre dossier*\Addon\RailVehicles\Steam*nom de votre modèle*\Default\Engine\

Le dossier Default est utilisé pour votre première et principale version d'un modèle. Vous pouvez par la suite avoir d'autres dossiers pour les différentes versions.

Cela permet de référencer le modèle par défaut dans votre blueprint pour éviter la répétition.

Pour les animations (par ex. les bielles), faite votre animation comme d'habitude dans 3D Canvas. Sélectionnez l'objet principal de l'animation (par ex. la roue motrice principale. Allez dans le menu File - Export et sélectionnez Trainworks. Assurez-vous de bien choisir Rail Simulator Intermediary Animation Format (.ia) et exportez vers le dossier correct. C'est le même que pour votre fichier .igs.

Note: Le Train Simulator Wizard ne placera pas automatiquement les fichiers dans le répertoire adéquate. Cependant, l'utiliser une fois sera utile pour créer la hiérarchie dont vous aurez besoin.

Autres animations que les bielles

Vous aurez besoin d'animer les autres bielles, les portes, les essuie-glaces, etc... Créez vos animations dans 3D Canvas comme d'habitude. Ignorez la documentation de RS. Sélectionnez l'objet principal et exportez le comme Trainworks - Rail simulator Intermediary Animation (.ia) - nom du fichier.ia (par ex. wipers.ia). Placez le fichier dans le même répertoire que le fichier .igs. Nous aurons besoin de référencer ces fichiers dans les blueprints - nous verrons cela plus loin.

Comme les animations sont exportées séparément, elles peuvent avoir un nombre différent de clés d'animation. Ceci est très utile car les essuie-glaces ont seulement besoin de 4 clés, alors que les bielles ou la distribution en nécessitent 8 ou 16.

Pour les loco vapeurs cependant, vous sélectionnerez la partie principale (après avoir fait les animations) et exporterez. Si vous avez d'autres parties animées, elle seront exportées en même temps que l'animation des bielles et le mouvement des roues motrices.

Textures

Pour avoir les mêmes effets d'environnement (par ex. le métal brillant ou mat) que ceux de RS, vous aurez besoin d'utiliser des textures spécifiques ou des "shaders". Vous devrez aussi utiliser des noms spécifiques pour rendre les transparences. Les principaux "shaders" dont vous aurez besoin sont:

TexDiff - la texture de base. Quand vous débutez votre modèle, utilisez cette texture de base pour texturer votre modèle et l'exporter pour essais. Ceci mettra en évidence les problèmes que vous pourriez avoir liés à l'application des textures ou autres. Canvas exportera automatiquement vos textures sans que vous ayez besoin de spécifier autre chose.

BlendATexDiff - Utilisez celle-ci lorsque vous avez besoin de transparence. Par exemple, les fenêtres de cabine ou de voiture. Rappelez-vous, votre texture doit avoir un canal alpha qui spécifie le degré de transparence.

Tex - Utilisée pour rendre l'éclairage de nuit (intérieur des cabines ou des voitures, etc...).

TrainSpecEnvMask.fx - Celle-ci donne à vos modèles l'aspect du métal brillant. Elle requière une "dummy texture" comme montré plus loin.

TrainBumpSpecEnvMask.fx - Celle-ci donne un aspect en trois dimensions aux textures (comme les rivets) et un effet de métal forgé sans nécessiter de polygones supplémentaires. Ceci nécessite une texture normale imitant le métal forgé dans la case "bump texture" de Canvas et une petite texture env dans la case "environmental".

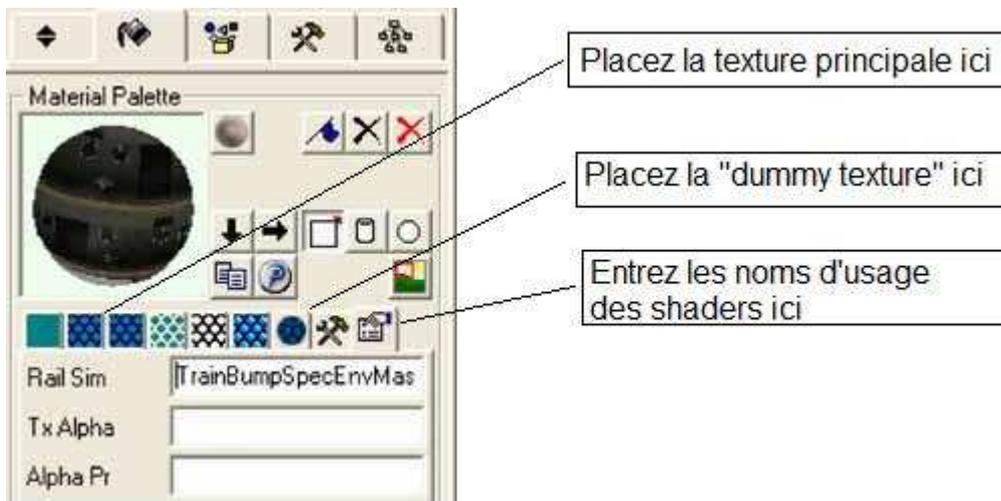
StencilShadow.fx - A utiliser pour l'ombre du modèle principal.

SubtractATexDiff - A utiliser pour l'ombre au sol.

Si vous avez besoin de textures spécifiques, vous devez procéder ainsi dans Canvas. Evidement, commencer par créer votre texture. J'utilise habituellement une texture de 1024x1024 pour l'objet principal. Les autres textures peuvent être de tailles variables (128x128, 256x256, 512x512, etc... Assurez vous que cette texture soit entrée comme votre "primary texture" dans la palette "Material". Si vous utilisez certains des shaders se terminant par .fx, vous devrez aussi créer une "dummy texture". Ce sera un carré noir de 128x128 en .tga. Cette texture doit être noire car elle influence l'apparence du modèle dans Canvas. Elle doit être entrée dans l'"environnement map" de Canvas. Vous devrez ensuite entrer le nom spécifique dans le champ "custom fields" (n'oubliez pas le .fx si nécessaire).

Il est suggéré que, à moins que vous utilisiez une texture avec un effet de transparence, vous texturiez en .bmp. Ceci afin de rendre plus facile l'application des effets de brillance spéculaire plus tard. Ces effets utilisent le canal alpha des textures .tga. Si vous utilisez une texture .tga, Canvas interprétera le canal alpha de votre effet brillant comme une transparence et une partie de votre modèle disparaîtra quand vous le regarderez dans Canvas. Ceci rend le modèle un peu bizarre !!

Il a été suggéré que votre modèle soit texturé en .bmp. Faites un copie du même nom en .tga. Editer le canal alpha de la texture en éclaircissant ou en assombrissant les différents parties si nécessaire. Le blanc donne un brillant maximum alors que le noir supprime la brillant. Vous pouvez utiliser toutes les nuances de gris intermédiaires pour obtenir différents niveaux de brillance. C'est ce fichier .tga que vous convertirez en fichier .ace et non le fichier .bmp. Utiliser la brillance spéculaire fait briller la texture considérablement et vous devrez assombrir et ajuster de manière importante avant que son aspect ne soit correct dans le jeux. Si vous n'utilisez pas de mappage en brillance spéculaire ou en effet forgé, vous pouvez ignorer ce qui précède et convertir la texture que vous avez utilisée dans Canvas.



Vous pouvez alors appliquer vos textures avec ces données. Rappelez-vous que si vous changez pour texturer une partie avec un effet de transparence, vous devez changer votre texture, enlever la "dummy texture" (sectionnez la et cliquez sur la croix noire) et changez le nom d'usage dans le champ approprié. De la même manière, si vous revenez texturer votre partie principale, vous devrez ré-entrer la "dummy texture" et le nom du "shader". Si vous ne le faites pas, vous aurez des messages d'erreur quand vous exporterez.

Note: Si le champ des noms d'usage de Rail Simulator n'apparaît pas, lancez le Train Simulator Wizard (même si ce n'est pas pour sauvegarder un modèle) ou allez dans "Properties" et entrez le nom comme dans le champ utilisateur (essayez ça).

Exporter les textures

Contrairement à MSTS, les fichiers .ace ne sont pas la forme finale des graphismes dans RS. Vous créez les fichiers .ace pour exporter avec vos fichiers .igs et .ia et l'Asset Editor prend le tout pour faire les fichiers dont vous aurez besoin dans RS. Utilisez To Ace que vous trouverez avec les "Developer Tools" pour convertir les fichiers .tga en .ace et exportez ainsi:

- Toujours convertir en utilisant l'option RS32Bit (oui, même si vous utilisez des transparences, ce sera ok).
- placer les dans le répertoire Textures qui doit se trouver dans:
Source*votredossier*\Addon\RailVehicles\Steam(ou diesel)*nomdevotremodèle*\Default\Engine\

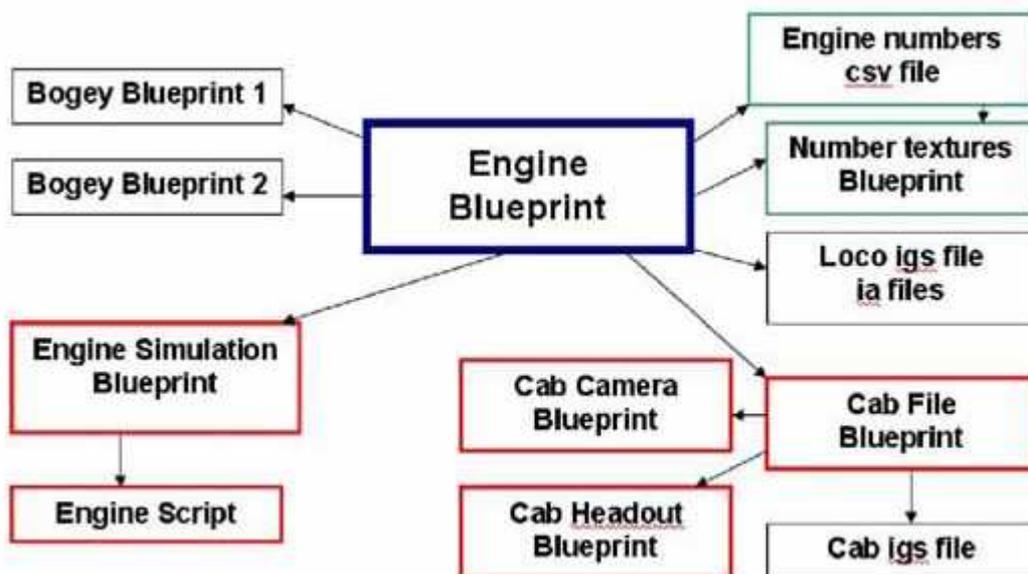
Ceci créera les fichiers dont vous aurez besoin pour l'étape suivante qui est l'utilisation de l'"Asset Editor" pour créer les blueprints dont vous avez besoin.

Si vous utilisez Adobe Photoshop ou Photoshop Elements, vous pouvez télécharger un exporteur de fichiers .ace qui marchera très bien.

Entrer dans le simulateur

Nous avons maintenant besoin de créer les fichiers qui permettront au modèle d'être exporté et de fonctionner dans le simulateur. C'est l'enfer des blueprints, si j'ose dire! Il y a un ensemble de documents conjoints qui indique où le modèle doit être placé, comment il fonctionne et comment il s'utilise. Le blueprint principal est l' **Engine Blueprints**, mais il se référera à votre **Bogie Blueprints**, **Cab Blueprints** et à bien d'autres petits fichiers nécessaires. C'est bien évidemment une source d'erreurs supplémentaires c'est pourquoi, au départ, il est préférable d'utiliser au maximum les fichiers d'exemples. Vous pourrez ensuite les modifier point par point, au fur et à mesure du développement de votre modèle. Certaines parties, comme les attelages et les fumées, ont des blueprints qui définissent comment ils doivent apparaître. Ca peut paraître compliqué, et, à vrai dire, ça l'est! Cependant, au bout d'un certain temps, une certaine logique commence à apparaître dans cette folie. Vous pourriez finir pas aimer ça, ce qui, après tout, est juste une façon différente de travailler que celle de MSTS (qui est un simulateur beaucoup plus simple).

Le diagramme suivant donne un aperçu sommaire de la configuration des blueprints. C'est très simplifié, mais cela montre les bases de ce que vous devez avoir pour travailler. Et n'oubliez pas, beaucoup de ça peut être copié des fichiers d'exemples de RSDL. En fait, la manière la plus simple de travailler avec les blueprints est de les copier dans votre répertoire et de changer les entrées pour les adapter à votre modèle.



Mais, comme je le dis plus haut, il y a d'autres choses qui peuvent s'ajouter à ça, comme les fumées (et une loco vapeur peut avoir 3 Bogey Blueprints). Mais la plus part du temps, ils peuvent être copiés à partir de fichiers existants.

Les Bogey Blueprints (Blueprints des bogies)

Bien que l'**Engine Blueprints** soit notre premier élément, c'est aussi bien de créer votre **Bogey Blueprints** en premier. Vous ne pourrez pas exporter une loco, une voiture ou un wagon sans le **Bogey Blueprints** en place. Si quelque chose est erroné ou oublié dans celui-ci, votre modèle ne sera pas exporté. Il est aussi important de comprendre que ce blueprint fait référence aux composants des bogies et des roues que vous avez exporté en tant que composant de votre modèle.

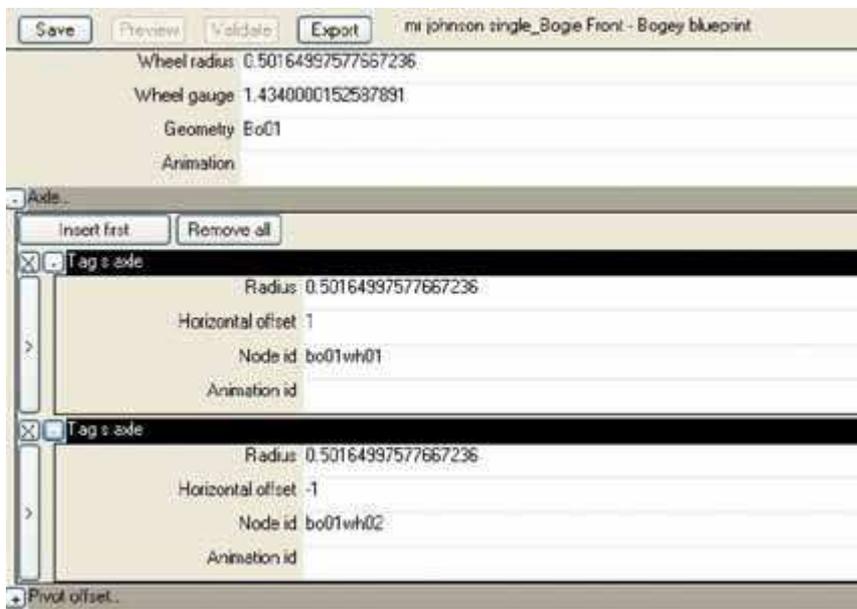
Les informations initiales semblent suggérer que les bogies sont dans des fichiers séparés et peuvent être liés à un autre modèle et qu'un seul est nécessaire qui peut être dupliqué.

Nous pensons que c'était l'idée initiale (comme dans Trainz) mais cela a été abandonné et c'est l'école de modélisation de MSTS qui a été adoptée.

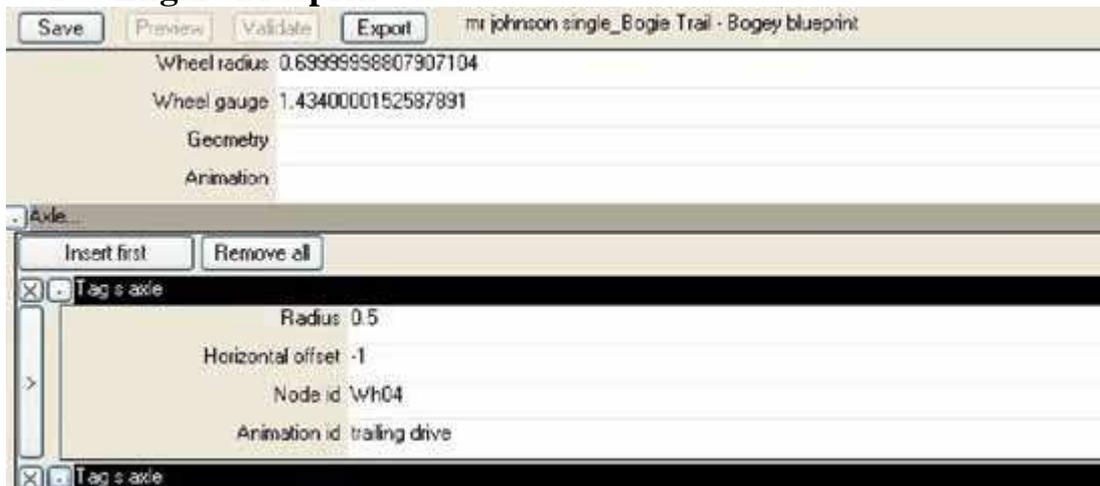
Ainsi, il semble que (certainement du point de vue de Canvas) vous deviez modéliser tous les bogies dont vous avez besoin et les placer chacun dans un blueprint différent. Utilisez le format standard pour les bogies et les roues, c'est à dire:
bo01 pour le premier bogie, bo02 pour le second, etc...
wh01 pour le premier essieu, wh02 pour le second, etc...

Vous n'avez pas besoin de ces noms d'usage dans Canvas (ils seront appliqués automatiquement à l'exportation), mais vous pouvez le faire pour plus de clarté.

Pour une loco vapeur, vous pouvez utiliser cette dénomination (une 231 (4-6-2 pour les anglo-saxons) a trois bogies), toute fois les roues motrices n'ont pas de bogie modélisé. Les roues peuvent **avoir une animation automatique avec ces noms quand elles sont liées à un bogie**. Les roues motrices, cependant, doivent avoir un fichier d'animation exporté (voir plus haut) et référencé comme partie intégrée au bogey blueprint. Ci-dessous, vous trouverez le bogey blueprint type d'un bogie. Assurez-vous que vous avez tous vos fichiers au bon endroit, ouvrez l'Asset Editor et allez dans le répertoire principal de votre modèle. Clic droit **sur le fichier** et choisissez Bogey Blueprint. L'image ci-dessous montre les entrées type d'un bogie avant de locomotive (ou d'un wagon). Vous devrez remplir un "tag axle" pour chacun de vos essieux.



Si le bogie concerne des roue motrices, vous devrez inclure la référence à leur fichier d'animation que vous avez exporté. Donc, notez l'"**animation id**" qui se trouve dans votre "**Engine Blueprint**".



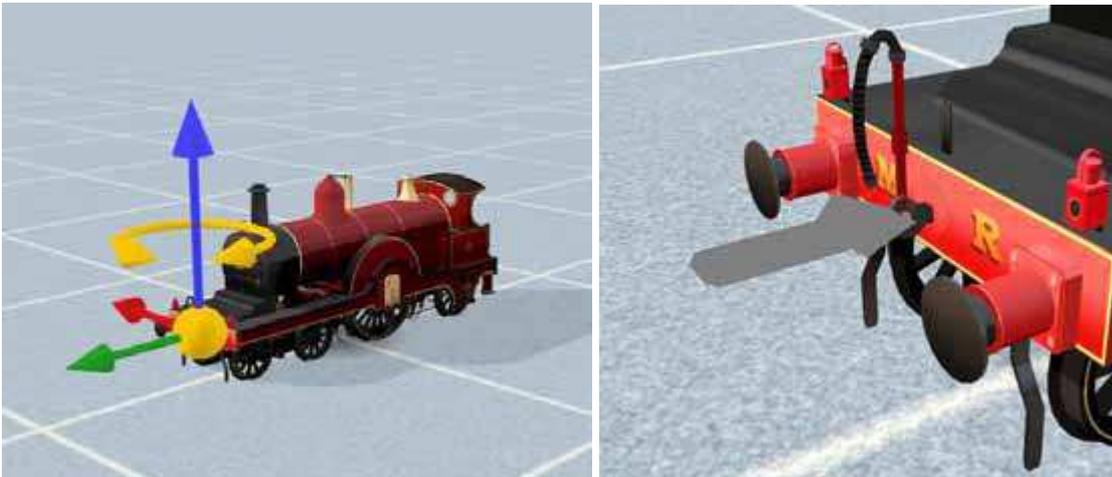
Dans l'Engine Blueprint, les références des bogies doivent être entrées dans l'ordre où ils se trouvent sur le modèle.

Engine Blueprint

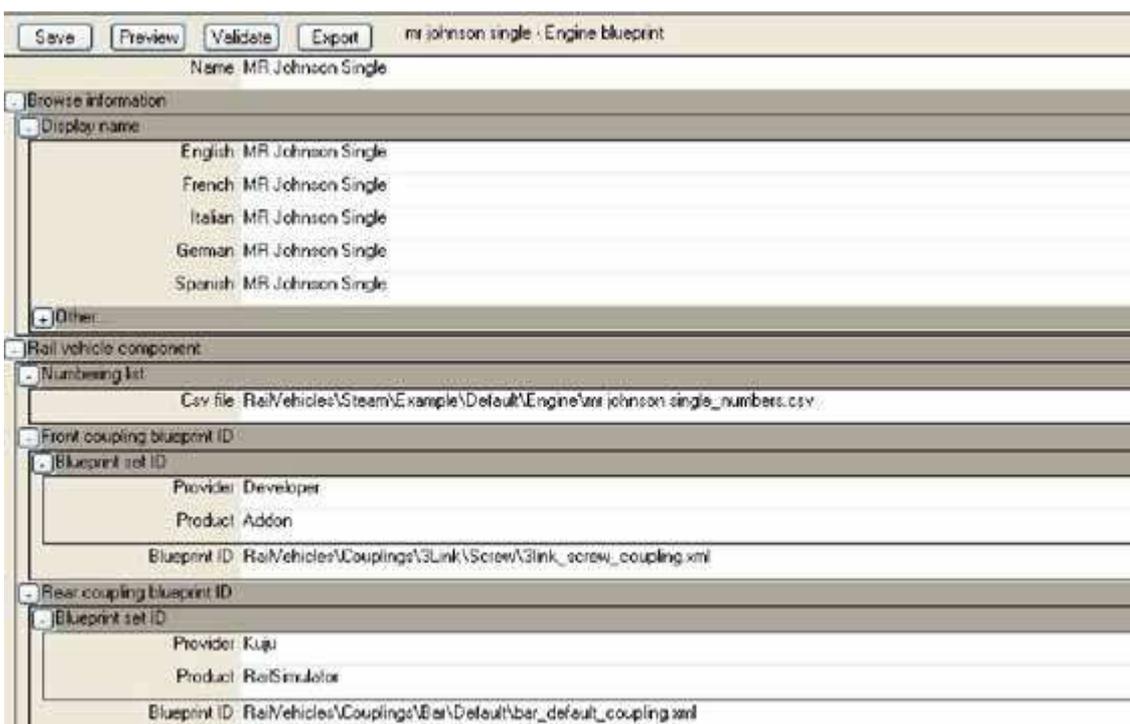
La meilleure manière d'aborder ce document compliqué est de copier l'Engine Blueprint de l'**Example** (copiez le fichier example engine .xml du dossier Developer dans votre propre dossier et renommez-le). Changez le nom de l'engin en haut (*n.d.t: dans les langues autres que l'anglais si vous voulez*) et commencez à entrer les informations propres à votre engin. Si vous utilisez des "alias" d'autres créations de RSDL/Kuju (par ex. des cabines ou des conducteurs), vous devrez utiliser un éditeur de fichier .xml, RS Bin Editor ou avec RS Tools (RW Tools) pour copier ces fichiers depuis les fichiers des autres modèles. Cependant, c'est une bonne idée de faire par vous même le maximum d'entrées dans le blueprint d'abord. Vous rentrerez le nom de votre loco, plus certains paramètres comme le poids (*n.d.t: attention, dans les blueprints, il vous faudra convertir les poids en unité*), la "collision length" et la hauteur (comme la "bounding box" de MSTs). Beaucoup de choses seront à rajouter, ici ce n'est qu'un exemple de blueprint. Nous voulons juste faire fonctionner une loco.

Ajouter les attelages

Ceci se fait en utilisant le "Gizmo". Ian Morgan a décrit en détail ce processus dans son tutoriel pour créer un wagon. Aussi, nous ne ferons que survoler cette partie. Le Gizmo utilise le "preview" de l'Asset Editor". Mais pour voir apparaître l'objet, il doit exister dans un blueprint. Les attelages sont dans ce cas. Notez que dans le cas d'une loco avec un tender, l'attelage arrière est un attelage par barre. Les diesels ont un attelage à choquelle (3Link) ou à boucle (buckeye) à chaque extrémité.



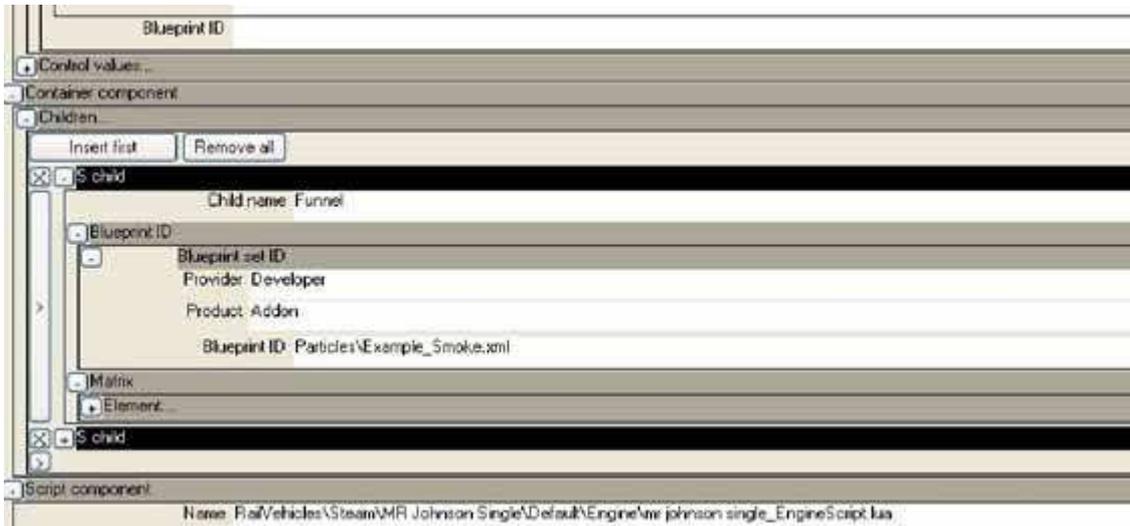
Notez aussi que le Gizmo et ses flèches ne sont pas situés au bon endroit. Le Gizmo se déplace en fonction de votre point de vue et doit être placé pour que la position donne la flèche de l'image de droite (elle devient rouge lors du déplacement). Déplacez les flèches avec la souris, mais doucement. Déplacez-la trop vite et votre attelage se retrouve à plusieurs kilomètres! Notez aussi qu'il est possible de verrouiller la position de la caméra (en haut à gauche de la fenêtre de l'Asset Editor). Ca peut vous aider dans l'utilisation du Gizmo.



Cet attelage de référence vous aide à positionner celui de votre modèle. Il est orange brillant! Quand vous êtes satisfait de l'emplacement de votre attelage, si vous changez Developer pour Kuju et Addon pour RailSimulator, quand la loco sera dans le jeu, ils apparaîtront comme de vrais attelages rouillés.

Ajouter les Fumées

Les fumées sont ajoutées comme "Child", déroulez le blueprint jusqu'à "Container Component" et cliquez sur Insert first. Le blueprint s'ouvre:



Dans le preview, vous pouvez voir la fumée comme si elle était faite de balles de golf - ceci est juste l'exemple de fumée. Comme avec les attelages, une fois que vous êtes satisfait de leurs positionnement, vous pouvez changer Developer pour Kuju et Addon pour RailSimulator et le blueprint id en Particles\smoke-stack01.xml. Dans le jeu, elle apparaîtra comme une vraie fumée "fumante".

Il est recommandé de renommer l'effet de fumée en référence aux effets de particules de Kuju en attendant que vous soyez tout à fait satisfait du fichier Engine .xml.

Les Fichiers Cab (cabine)

Ce tutoriel ne rentrera pas dans le processus de construction de votre cabine. Nous allons "aliaser" les cabines à partir de celles des locos RSDL/Kuju pour l'instant. La meilleure façon de procéder est de simplement copier le script de l'une des locos existantes dans vos fichiers. Il y a deux façons de faire, aussi valide l'une que l'autre et vous pourrez utiliser celle qui vous conviendra le mieux. Soit avec un éditeur de fichier .xml, soit avec Notepad2.

- Avec Notepad2

<http://www.flos-freeware.ch/notepad2.html>

pour modifier les fichiers .xml du Blueprint Editor. Vous devrez ensuite utiliser Serz.exe (il fait partie des Dev Tools de RSDL) pour les convertir en fichier .bin. La recherche et le remplacement des mots est facile.

- Avec RW Tools

<http://www.rstools.info/index.html>

Avec cela, vous pouvez ouvrir les fichiers .bin directement (à partir du menu ->Open Railsim File) et les modifier. Ainsi, vous pouvez ouvrir le fichier RSDL S&D 7F.bin, copier la partie du texte qui concerne la cabine, les contrôles, le conducteur, etc... et les coller au bon endroit dans le fichier .bin de votre loco (vous devrez l'importer du Blueprint Editor pour ce faire, et ceci créera tous les fichiers et les dossiers nécessaires pour votre loco. Prenez garde au fait que cela modifiera aussi toutes les données concernant la fumée, le conducteur, etc... que vous aviez créé dans l'Asset Editor.

Ainsi, par exemple:

- Faites une copie du fichier .bin de la Black 5

C:\ProgramFiles\RailSimulator\Assets\Kuju\RailSimulator\RailVehicles\Steam\5MT_Black5\Default\Engine

- Placez la dans un dossier temporaire.

Si vous utilisez Notepad2, il faut le convertir au format .xml. Vous pouvez le faire en le déplaçant sur l'icone du fichier Serz.exe. cela créera un fichier .xml dans le même répertoire. Ouvrez le fichier .xml

Si vous utilisez RS Tools, ouvrez directement le fichier .bin
Descendez jusqu'à la ligne

```
<InteriorGeometryID  
d:type="cDeltaString">Kuju\RailSimulator\RailVehicles\Steam\5MT_Black5\Default\CabView\[00]5mt_black5_cab</InteriorGeometryID>
```

Copiez la totalité des lignes jusqu'à la ligne

```
<cEntityContainerBlueprint-sChild d:id="21417320"> comprise (le chiffre entre parenthèse pourrait être différent)
```

Cette ligne est immédiatement en dessus de la clé "main smoke stack"
Cela fait environ 800 lignes de code!

-Ouvrez votre propre fichier "Engine" (comme .xml ou .bin) dans le répertoire "Asset " et remplacez les lignes de code correspondantes par celles copiées du fichier de la Black 5. Notez que votre code n'incluera pas toute fois une référence pour la cabine, aussi, elle devra ressembler à ça:

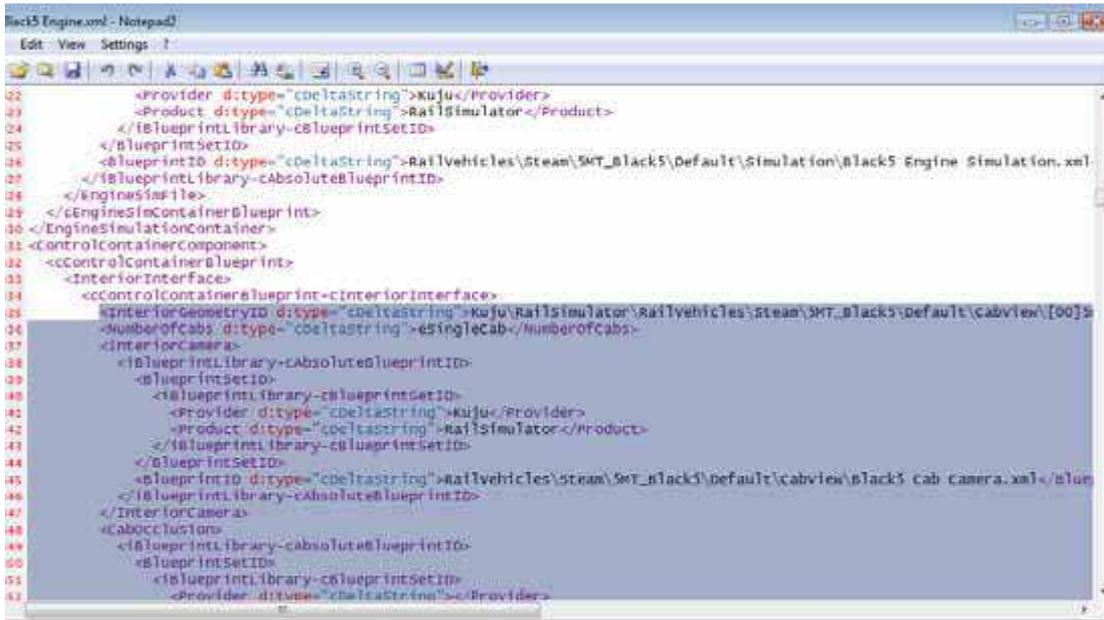
```
<InteriorGeometryID d:type="cDeltaString"></InteriorGeometryID>
```

Enregistrez votre fichier .xml et déplacez le sur l'icone de Serz.exe pour le reconvertir en fichier .bin.

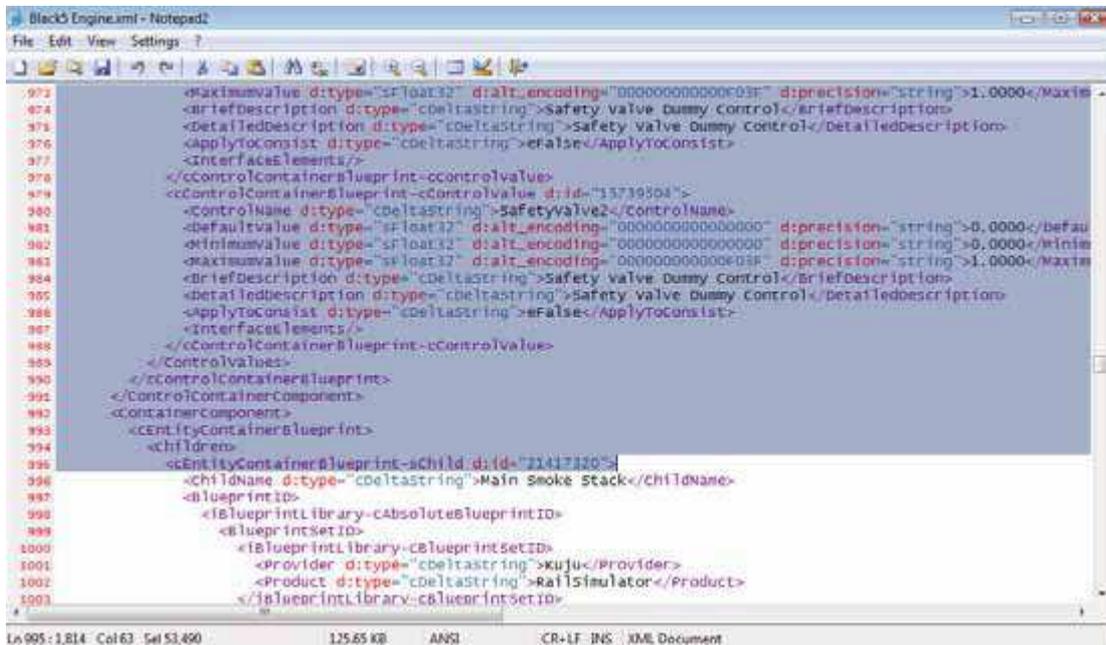
Avec RS Tools, enregistrez le comme un fichier Railsim

Ensuite, allez dans votre dossier Addon et supprimez le fichier blueprint.pak pour que vos changements apparaissent. Comme d'habitude, faite une sauvegarde de vos fichiers avant de faire ces manœuvres.

Ci-dessous, deux capture d'écran (de fichiers .xml édités) montrant le début et la fin des lignes concernées



```
22 <Provider d:type="cDeltaString">Kuju/Provider</Provider>
23 <Product d:type="cDeltaString">Rail5Simulator</Product>
24 </BlueprintIntLibrary-cBlueprintSetID>
25 </BlueprintSetID>
26 <BlueprintID d:type="cDeltaString">Railvehicles\Steam\SMT_Black5\Default\Simulation\Black5 Engine Simulation.xml</BlueprintID>
27 </BlueprintIntLibrary-cAbsoluteBlueprintID>
28 </EngineSimFile>
29 </EngineSimContainerBlueprint>
30 </EngineSimulationContainer>
31 <ControlContainerComponent>
32 <ControlContainerBlueprint>
33 <InteriorInterface>
34 <<ControlContainerBlueprint-cInteriorInterface>
35 <InteriorGeometryID d:type="cDeltaString">Kuju/RailSimulator/Railvehicles\Steam\SMT_Black5\Default\CabView\0003</InteriorGeometryID>
36 <NumberOfCabs d:type="cDeltaString">eSingleCab</NumberOfCabs>
37 <InteriorCamera>
38 <BlueprintIntLibrary-cAbsoluteBlueprintID>
39 <BlueprintSetID>
40 <BlueprintIntLibrary-cBlueprintSetID>
41 <Provider d:type="cDeltaString">Kuju/Provider</Provider>
42 <Product d:type="cDeltaString">Rail5Simulator</Product>
43 </BlueprintIntLibrary-cBlueprintSetID>
44 </BlueprintSetID>
45 <BlueprintID d:type="cDeltaString">Railvehicles\Steam\SMT_Black5\Default\CabView\Black5 Cab Camera.xml</BlueprintID>
46 </BlueprintIntLibrary-cAbsoluteBlueprintID>
47 </InteriorCamera>
48 <CabOcclusions>
49 <BlueprintIntLibrary-cAbsoluteBlueprintID>
50 <BlueprintSetID>
51 <BlueprintIntLibrary-cBlueprintSetID>
52 <Provider d:type="cDeltaString">c/Providers</Provider>
```



```
972 <MaximumValue d:type="float32" d:alt_encoding="000000000000003F" d:precision="string">1.0000</MaximumValue>
973 <BriefDescription d:type="cDeltaString">Safety valve Dummy Control</BriefDescription>
974 <DetailedDescription d:type="cDeltaString">Safety valve Dummy Control</DetailedDescription>
975 <ApplyToConsist d:type="cDeltaString">efalse</ApplyToConsist>
976 <InterfaceElements>
977 </InterfaceElements>
978 </ControlContainerBlueprint-cControlValue>
979 <<ControlContainerBlueprint-cControlValue d:id="15739504">
980 <ControlName d:type="cDeltaString">SafetyValve2</ControlName>
981 <DefaultValue d:type="float32" d:alt_encoding="0000000000000000" d:precision="string">0.0000</DefaultValue>
982 <MinimumValue d:type="float32" d:alt_encoding="0000000000000000" d:precision="string">0.0000</MinimumValue>
983 <MaximumValue d:type="float32" d:alt_encoding="000000000000003F" d:precision="string">1.0000</MaximumValue>
984 <BriefDescription d:type="cDeltaString">Safety valve Dummy Control</BriefDescription>
985 <DetailedDescription d:type="cDeltaString">Safety valve Dummy Control</DetailedDescription>
986 <ApplyToConsist d:type="cDeltaString">efalse</ApplyToConsist>
987 <InterfaceElements>
988 </InterfaceElements>
989 </ControlContainerBlueprint-cControlValue>
990 </ControlValues>
991 </ControlContainerBlueprint>
992 </ControlContainerComponent>
993 <ContainerComponent>
994 <EntityContainerBlueprint>
995 <Children>
996 <<EntityContainerBlueprint-child d:id="21413320">
997 <ChildName d:type="cDeltaString">Main Smoke Stack</ChildName>
998 <BlueprintID>
999 <BlueprintIntLibrary-cAbsoluteBlueprintID>
1000 <BlueprintSetID>
1001 <BlueprintIntLibrary-cBlueprintSetID>
1002 <Provider d:type="cDeltaString">Kuju</Provider>
1003 <Product d:type="cDeltaString">Rail5Simulator</Product>
1004 </BlueprintIntLibrary-cBlueprintSetID>
```

Ajouter un ou des personnages

A ce stade, jusqu'à ce que vous ayez créé vos propres personnages, nous copierons les codes existants des modèles RSDL/Kuju. Vous pouvez aller voir dans les fichiers .bin ou .xml d'un des exemples de locos et utiliser la même méthode que ci-dessus pour les copier/coller. Chauffeurs et mécaniciens sont référencés vers la fin des fichiers "engine", juste avant les fumées.

"Aliaser" les sons

Les sons peuvent être ajoutés au blueprint comme "child" (enfant d'une hiérarchie où le père est l'objet principal ("Main")) dans la section "container component" vers la fin de l'"engine blueprint", juste après les effets de particules (fumées). Vous pouvez y mettre des fichiers déjà utilisés dans Rail Simulator.

Ci-dessous, un exemple de la façon dont sont créés les sons d'une loco vapeur :



La numérotation

Cette méthode s'applique à un engin dont la numérotation se fait sur six chiffres.

Changez juste 6 pour le nombre de chiffres qui convient à votre loco.

Faites un rectangle plat composé de six carrés. Nommez cet objet

"1_128_primarydigits_6" (6 car il y aura 6 chiffres et un carré n'est pas "lodable"

au delà de 128m). Il vous faudra autant de fois cet objet que vous aurez à utiliser celui-ci. Sur un diesel, vous pouvez en avoir besoin quatre fois, mais parfois moins.

Faites six textures de 32x32 avec un gris uni (la couleur exacte n'a pas d'importance).

Ces textures serviront à indiquer au simulateur où placer les chiffres de la

numérotation. Nommez-les "primarynumber_0 à "primarynumber_6 (pour un

numéro à 6 chiffres) et enregistrez-les sous forme de fichiers .bmp dans le répertoire

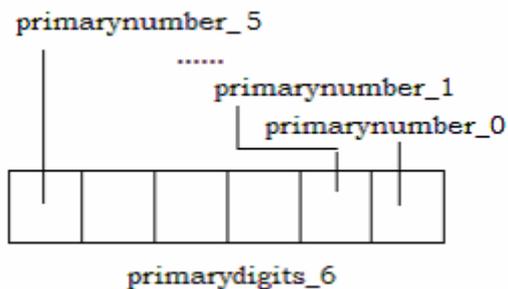
de votre modèle (dans Canvas). Convertissez-les en fichiers .ace et placez ces fichiers dans le sous-répertoire "Textures" de votre répertoire "engine" dans RS.

Texturez votre objet (primarydigit_##) en commençant par la première face à droite

avec ""primarynumber_0.bmp" puis la deuxième en allant vers la gauche avec

"primarynumber_1", puis la troisième avec "primarynumber_2, etc... jusqu'à ce que

toutes les face soient texturées.

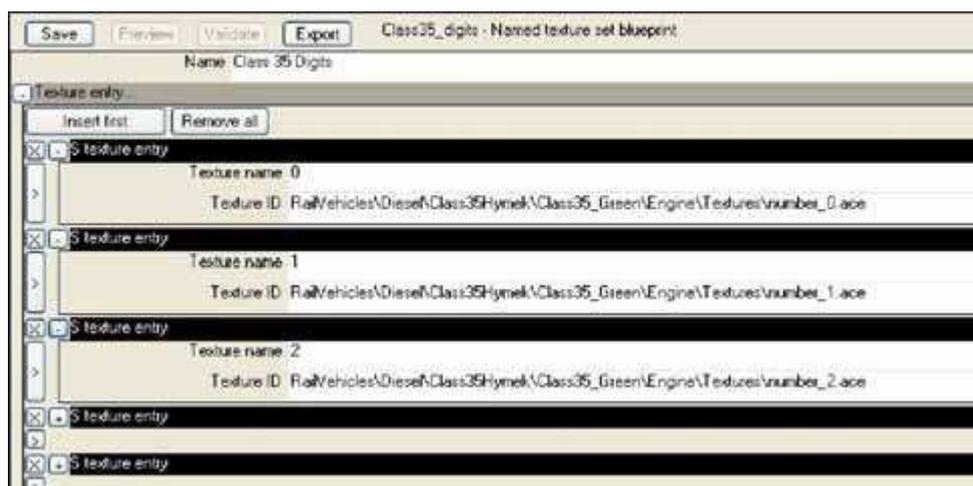


Utilisez le shader "**TrainBasicObjectDiffuse.fx**" pour ces textures.

Créez un ensemble de textures nommées **number_0** à **number_9** au format 32x64 (pour plus de détails, 64x64, si besoin). Utilisez un noir pur pour le fond et la couleur que vous souhaitez pour le chiffre. Le noir du fond sera utilisé en couche alpha pour "découper" le chiffre. Convertissez vos textures en fichiers .ace (Si votre chiffre doit être noir, modifiez la texture après conversion). Placez les fichiers dans le sous-répertoire textures de votre dossier dans RS.

Pour que la numérotation fonctionne, vous devez créer un fichier .csv que vous placerez dans le répertoire "engine" et qui contiendra la série de numéros que vous souhaitez voir afficher pour vos engins dans le jeu. La voie la plus simple est de recopier un fichier .csv à partir d'un exemple et de le modifier à votre convenance. Utilisez Excel pour modifier le fichier .csv et renommez-le à votre convenance (toujours sous forme de fichier .csv). Vous devrez référencer ce fichier dans le blueprint de votre engin.

Vous aurez aussi besoin d'un fichier blueprint dans votre sous-répertoire textures qui référence tous les fichiers de chiffres .ace. Nommez-le avec un nom en _digits (par ex. Myengine_digits). Ce blueprint est référencé à partir de l'engine blueprint. Il doit y être nommé.



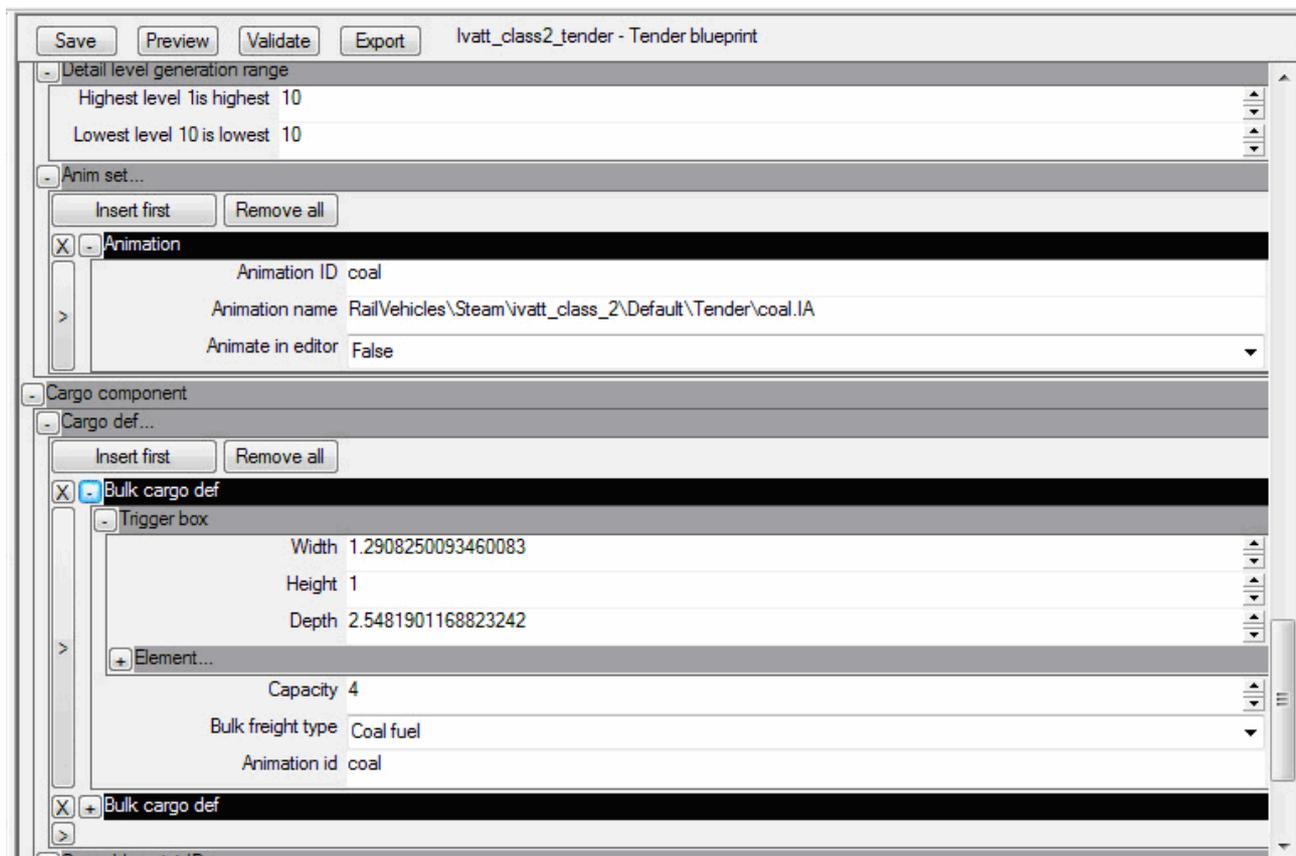
Le simulateur prendra les fichiers .ace dont il aura besoin (à partir du fichier .csv) et les placera sur votre objet dans l'ordre correct en se fondant sur l'emplacement des textures "primarynumber" que vous avez placées sur les objets "primarydigits". Merci à Gerrit (HondaWizard) sur le forum de Canvas pour avoir travaillé ça et l'avoir expliqué.

n.d.t: j'ai rajouté des points à la traduction là où cela me paraissait rendre les choses plus compréhensibles.

Tender Blueprint

Là encore, il est plus simple d'utiliser le blueprint d'un tender copié d'un exemple. Changez juste les noms. Exportez-le de la même manière que votre loco. Habituellement, un tender n'a besoin que d'un bogie (*n.d.t: pour les tenders anglais ou les petits tenders français*), probablement à trois essieux (les locomotives en voie étroite peuvent y faire exception). Placez la "partie" charbon à sa position vide (la plus basse) dans votre modèle dans Canvas. Ensuite, utilisez une animation en deux "images" (ou plus si vous voulez) pour passer de la situation "vide" à celle "plein". Contrairement à MSTs, le charbon n'a pas à être exporté séparément (mais l'animation, si!).

Exportez le modèle de la manière habituelle, puis exportez l'animation en sélectionnant la partie "charbon" et en sélectionnant l'option .ia. Donnez un nom significatif à cette animation (par ex. charbon). Vous devrez ajouter les données nécessaires dans le blueprint du tender. Voir l'exemple ci-dessous:



N'oubliez pas d'ajouter le nom de l'animation dans la partie "Tender Blueprint Anim" et l'animation .id dans la partie "Bulk cargo def". Si vous utilisez l'exemple de tender, vous pouvez utiliser le nom existant (coal). Votre fichier de tender .igs peut être dans le répertoire "engine" ou dans le répertoire "tender", à condition de référencer correctement dans le blueprint où il est situé. N'oubliez pas de rajouter "Water Fuel" et "Coal Fuel" (clic sur le signe + près de "Bulk cargo def"), en utilisant le menu

déroulant. Jetez un coup d'oeil à un fichier Kuju tender.bin pour savoir à quoi il doit ressembler.

Les choses qui peuvent mal se passer et comment les solutionner

Le bogey Blueprint

Dans l' "engine blueprint", les coordonnées du bogie doivent être entrées dans le même ordre qu'elle apparaissent sur la loco.

Si le "bogey blueprint" n'est pas correct, vous ne verrez généralement pas votre modèle dans le "preview" ou la simulation, **même si l'Asset Editor indique que tout va bien.**

Il peut y avoir un problème avec les bogie à un seul essieu (bissel). Il a été rapporté qu'il faille entrer un essieu "fantôme" dans le "bogey blueprint". Les bissel semble ne pas fonctionner dans RS. Vous devrez créer une entrée pour un essieu "dummy" dans le blueprint du bogie concerné. Vous n'avez pas à changer le modèle lui-même.

Textures

Après les bogies, c'est l'erreur la plus courante qui peut empêcher l'exportation. Si vous recevez un message d'erreur à propos des "shaders", vérifiez ce qui suit:

Avez-vous retexturé une partie, ou texturé une nouvelle partie dans votre session actuelle de Canvas ou dans une session récente? Si oui, il y a des chances que vos effets de shaders ou/et vos "dummy textures" aient été perdues (par ex. vous pourriez avoir texturé différemment une partie, avec un "shader" différent, comme une transparence). Canvas ne sauvegarde pas le nom d'usage dans les palettes.

Solution: Sélectionnez le matériel correct à partir de la palette, assurez-vous que la "dummy texture" soit en place et retapez le nom du shader dans le champ utilisateur. (n'oubliez pas le **.fx**). Maintenant, cliquez sur une partie de votre modèle qui utilise cette texture et choisissez "select by material" dans le menu. Toute les occurrences de ce "matériel" sur cet objet seront "illuminées". Allez dans le menu des "plugins" et sélectionnez "Material - Update Material". Si votre modèle est composé de plus d'un objet, vous devrez refaire la même chose avec chacun d'eux (par ex. les essieux) s'ils requièrent un shader spécifique.

Les locos ne s'attèlent pas aux wagons

Vous devez ajuster les "collision length", les pivots de la loco et/ou les pivots des bogies. Ceci à partir du blueprint ou en modifiant les fichiers .bin ou .xml.

Vous devez aussi vérifier la position des attelages - avez vous utilisé correctement le Gizmo? Encore une fois, en modifiant un fichier "manuellement", vous pouvez affiner les réglages. Est-ce que vos attelages correspondent? RS est probablement plus tolérant que MSTs en la matière, mais ça aide si, par exemple l'attelage arrière d'une loco est "à barre" et l'attelage avant du tender aussi.

Quand je clique sur une loco, les roue motrices se placent dans une position différente

Actuellement, il semble que Canvas et RailSimulator ne coopèrent pas pour exporter correctement. Il semble parfois que les animations ne se mettent pas à jour chaque fois. Dès que vous sélectionnez une loco, elle passe en mode animation, aussi si l'animation est défectueuse, cela apparaît tout de suite. Il paraît possible de solutionner le problème en réexportant l'animation, mais avec un nom différent. Changez les données de l'animation dans l'engine blueprint et cela devrait aller. C'est un problème qui devrait disparaître dans les nouvelles versions de Canvas.

Ma loco fonctionne sur mon ordinateur, mais pas sur celui de quelqu'un d'autre

Vous avez probablement utilisé un autre répertoire en dehors de celui de votre modèle (par ex. un répertoire issu des exemples), ce qui se produit facilement avec les vues extérieures de cabine. Vous devez vérifier dans votre blueprint qu'il se réfère uniquement à votre modèle, les autres utilisateurs n'ayant pas les mêmes données en place. Ainsi, si vous faites des engins avec des livrées différentes, mais avec le même modèle de base, il sera facile de leur conserver les mêmes liens avec le bogey blueprint, par exemple. A moins de réaliser tous les modèles en un seul fichier, ceci peut poser problème aux utilisateurs qui n'ont pas votre fichier original ou qui ne téléchargent pas toutes les versions.

OK. C'est tout pour le moment. Nous avons bien conscience de ne pas vous avoir donné une méthode pas à pas. Chaque fois que nous relisons, nous pensons à des choses qu'il faudrait rajouter et nous savons que la modification des fichiers .xml et .bin n'est pas facile. Cependant, nous sommes heureux d'avoir répondu à certaines questions que vous pourriez vous poser en cours de route. Et nous serons heureux d'ajouter d'autres informations et corrections dans une prochaine session. Nous essaierons. Si vous êtes capable de faire un modèle dans MSTS, avec un peu de pratique, vous pourrez le faire avec Rail Simulator, surtout si vous utilisez les fichiers de exemples et les données de base comme point de départ.

Bonne chance!

Richard Scott et Kevin martin
Juin 2008

Contacts:

richardscott@dasp.org.uk
Kngtrains@aol.com

Remerciements:

Les personnes suivantes doivent être remerciée pour nous avoir apporté des solutions, fait des fichiers d'aide ou nous avoir juste encouragés

Peter Holton

Gerrit

Ian Morgan

RSAdam and RSDerek.