

Génération de circuits pour Torcs 1.3.0

Document sous licence FDL © 20-12-2007 copyleft Michel Luc

Permission vous est accordée de distribuer des copies exactes ou modifiées de ces articles tant que ces lignes de permission et le copyright apparaissent dans vos copies et tant que ces copies restent sous licence FDL ou GPL .

Création de circuits (tracks) pour Torcs

Document Original "**Build your track for Torcs in 20 minutes**" par **Vicente Martí Centelles** :

<http://usuarios.lycos.es/f1torcs/>

Ce document n'est en aucun cas une traduction de « Build your track for Torcs in 20 minutes » mais il est évident que sans le document écrit par Vicente Martí Centelles je n'aurai probablement jamais eu envie d'écrire ce tutoriel.

Présentation

Pré-requis

Principe de création (une méthode)

Convention d'écriture

1) Création du circuit avec trackeditor

2) Modification du circuit

- 2.1 Génération des images
 - 2.1.2 Création de l'image du circuit
 - 2.1.3 Création de l'image de fond
- 2.2 Inclinaison des virages
- 2.3 Panneaux indicateurs de virage
- 2.4 Modification des surfaces
- 2.5 Création de la carte d'élévation
- 2.6 Environnement Mapping
- 2.7 Les Caméras
- 2.8 Ajout d'objets
- 2.9 Création de la carte des ombres

3) A propos du circuit du tutoriel

- 1. Contenu de l'archive
- 2. Les fichiers
- 3. Les images
- 4. Les liens internet

Présentation

Vous aurez besoin des outils suivants :

Blender (<http://www.blender.org/>)
The GIMP (<http://www.gimp.org/>)
acc (inclus dans les sources de Torcs 1.3 : <http://torcs.sourceforge.net/>)
trackgen (inclus dans les sources de Torcs 1.3)
trackeditor (inclus dans les sources de Torcs 1.3)
et un simple éditeur de texte pour modifier le fichier XML

Vous pouvez récupérer l'archive du circuit de ce tutoriel ici :
http://cern91.tuxfamily.org/pub/docs/Jeux/Torcs/tuxland_tut-track.zip

Pour créer le circuit de base, trackeditor est suffisant et très simple à prendre en main.
Pour générer le circuit trackgen remplira très bien cette tâche.
Pour ajouter des éléments (objets) au circuit de base, accc se chargera de combiner ces divers éléments (nb: accc utilise stripe).

Mais pour obtenir un circuit (piste + terrain + objets) qui corresponde à ce que vous souhaitez faire il va vous falloir 'mettre les mains dans le camboui' en modifiant le fichier XML.

Principe de création d'un circuit :

Avant de commencer la réalisation de votre circuit, voici une description, pas trop technique, du principe de création d'un circuit sous Torcs.

Le circuit sera obligatoirement créé sous \$TORCS_BASE/tracks/nom_de_la_catégorie/nom_du_circuit , c'est à dire par exemple si torcs est installé sous /usr/local/share/game et que vous construisez un nouveau circuit nommé '**le-mans**' dans la catégorie '**road**' cela va vous donner ce chemin :
`/usr/local/share/game/torcs/tracks/road/le-mans`

_ Vous construisez la base de votre circuit et le générez avec trackeditor
_ Le fichier .ac est créé par trackeditor depuis son menu File -> export --> ac3d (ou export all)

_ Puis vous pouvez lancer torcs et tester votre nouveau circuit pour vérifier/modifier la piste (track) de ce circuit. Si le terrain n'est pas construit vous pouvez le faire, dans un terminal, avec la commande : **trackgen -c nom_de_la_catégorie -n nom_du_circuit -a**
Attention cela peut prendre plusieurs minutes.

Ensuite il vous faudra :

_ ajouter un fond au décor, **background.png**
_ éventuellement modifier les textures du mapping de la piste
_ construire une carte d'élévation du terrain, **elevation-map.png**
_ construire des objets, fichiers .ac, avec Blender ou ac3d
_ ajouter ces fichiers .ac des objets au fichier .ac du circuit
_ créer une carte des objets, **objects-map.png**
_ réaliser un 'mapping' des ombres du circuit, **shadow2.rgb**
_ ajouter une image '**nom_du_circuit.png**' pour la page de présentation du circuit dans Torcs

_Après chaque modification du fichier XML, vous utiliserez trackgen :
trackgen -c nom_de_la_catégorie -n nom_du_circuit

IMPORTANT! Convention d'écriture :

Dans ce qui suit je vais utiliser des noms de variable pour désigner le nom du circuit et le nom de la catégorie, que vous remplacez par les noms de votre circuit et de sa catégorie :

\$name nom de votre nouveau circuit (ici, \$name = tuxland)
\$category nom de la catégorie où vous avez placé votre circuit (ici, \$category = road)

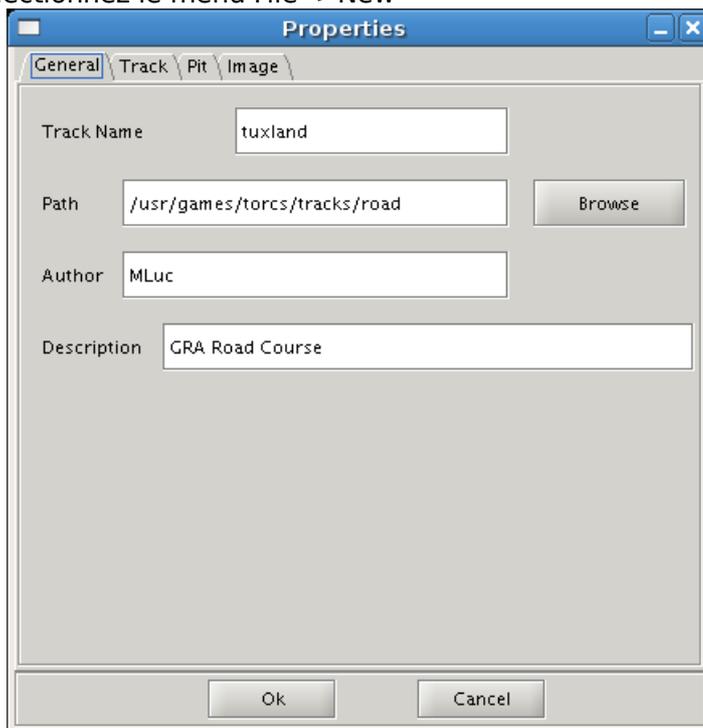
1 Création du circuit avec trackeditor :

Trackeditor est inclus dans les sources de Torcs 1.3.0, mais vous pouvez également le récupérer à cette adresse : <http://www.berniw.org/trb/download/trackeditor-0.6.2c.tar.bz2>

La documentation incluse dans trackeditor est un tutoriel avec des captures d'écrans suffisamment explicites qui, même si l'anglais n'est pas votre tasse de thé, vous permet de maîtriser cet outil en quelques minutes.

Je ne vais donc pas ici vous expliquer le fonctionnement de trackeditor, mais simplement préciser quelques points pour vous éviter de chercher.

Au lancement vous sélectionnez le menu File ->New



Une fenêtre des propriétés s'ouvre, et vous renseignez les champs suivants :

Track Name avec le nom de votre nouveau circuit

Path avec le chemin de la catégorie du circuit de torcs

puis votre nom et une courte description du type de course.

Dans les autres onglets :

Track vous indiquez la largeur, en mètres, de la piste.

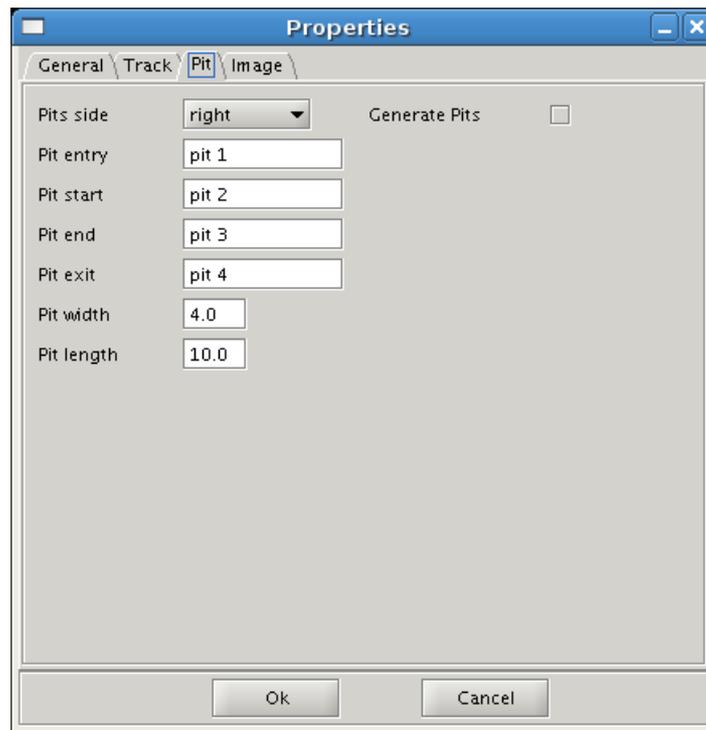
Image vous permet de sélectionner une image comme modèle pour vous aider à construire votre piste sous trackeditor (show background image), vous pouvez également indiquer l'échelle. Cette image (PNG) pourra éventuellement être utilisée comme image dans la présentation du circuit dans torcs. Pour l'échelle de l'image que vous aurez créée sous GIMP ou autre il vous suffit de mesurer le nombre pixel; Par exemple, voir tuxland.png, si vous décidez que la ligne (droite) du départ fera 800 m et que la longueur de cette ligne est de 2464 pixels, vous aurez

$$2464/800 = 3.08 \text{ comme échelle}$$

Pit les stands des 'Team' (Garage) pour les réparations :

Ils sont définis par 4 segments, la piste d'accès (Entrée) aux stands, le début des stands, la fin des stands et la piste de sortie de ces stands.

Les stands seront créés par Torcs dans le segment 2 (début) et 3 (fin) en fonction de la longueur que vous aurez attribué pour chacun de ces stands et de la longueur des segments 2 et 3.



Pits side le coté de la piste où seront positionnés les stands (ici à droite).

Pit entry le nom du segment que vous donnerez à cette piste d'accès.

Pit start le nom du segment que vous donnerez pour la début des stands.

Pit end le nom du segment que vous donnerez à la fin des stands.

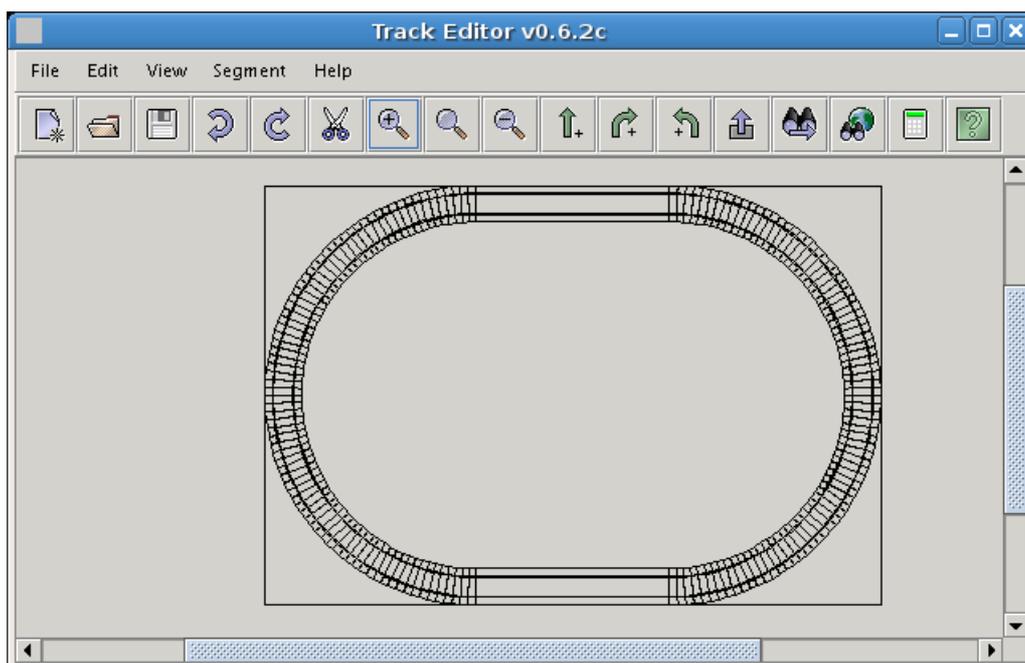
Pit exit le nom du segment que vous donnerez à cette piste de sortie des stands.

Pit width La largeur des stands.

Pit length La longueur de chaque stand :

Par exemple vous voulez 20 stands (pits) de 10 mètres de long, vous créez un segment (pit start) nommé 'pit 2' et un segment (pit end) nommé 'pit 3' respectivement de 100 m chacun ou 150 m et 50 m, ou 80 m et 120 m... etc.

Pour finir vous cliquez sur le bouton OK.



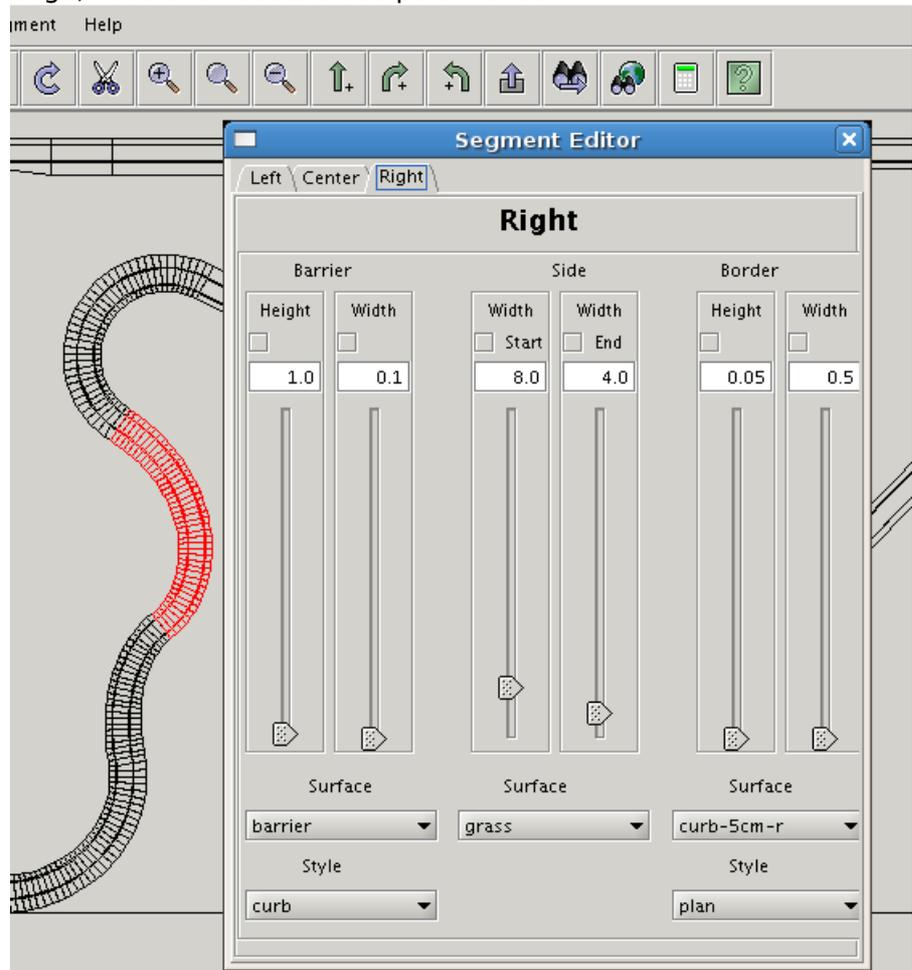
Une piste par défaut est créée que vous allez pouvoir modifier en respectant toutefois certains paramètres :

Votre **ligne de départ**, ici en haut et à gauche de la ligne droite, doit obligatoirement être placée sur une **ligne droite horizontale**.

Vos segments doivent être 'raccordés' c'est à dire la hauteur/largeur du segment doit être identique, au début avec le segment précédent et à la fin avec le segment suivant. Cela paraît évident mais... Vous faites un segment (une cote) qui part de 0m et se termine à 6m et le segment suivant (descente) vous le commencez à 4m et finissez à 0m, vous avez ainsi un dénivelé de 2m entre les 2 segments, dans Torcs ces 2 segments (centre/cotés/barrières) seront joints, et ce sera très laid visuellement et les voitures vont effectuer un saut de 2m de haut à ce passage.

En sélectionnant un segment (rouge) vous avez accès à ces paramètres :

c'est ici que vous allez donner un nom au segment (onglet Center) et modifier la pente, la largeur des cotés, l'angle pour les virages, le type des différentes surfaces (centre, bords, cotés et barrières).



Vous avez terminé votre circuit :

_ vous sauvegardez, cela va créer un fichier nom_du_circuit.prj.xml

_ Vous exportez pour obtenir un fichier XML Torcs

_ Vous pouvez utiliser le menu File -> Export -> ac3d (c'est une bonne idée) pour construire d'une part le fichier XML de Torcs et d'autre part le fichier 3D utilisé par Torcs (nom_du_circuit.ac).

Vous pouvez quitter trackeditor, c'est fini.

Le répertoire correspondant à votre circuit, dans l'arborescence de torcs,

\$TORCS_BASE/tracks/\$category/\$name

doit contenir maintenant :

\$name.ac

\$name.prj.xml

\$name.xml

\$name-msh.ac

\$name-trk.ac

Si c'est votre premier circuit, le mieux maintenant est de lancer torcs pour tester votre circuit et voir si vous n'avez pas fait d'erreur de construction avant de passer à la modification de la piste.

2 Modification de la piste :

Trackeditor a construit un fichier `$name.xml` (ici `tuxland.xml`) qui commence par :

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
file           : tuxland.xml
auto generated  : by Track Editor
version        : v 0.6.0 24/4/2005
copyright      : (C) 2005 by Charalampos Alexopoulos
email         :
-->
<!--
This program is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
-->
  <!DOCTYPE params SYSTEM "../././src/libs/tgf/params.dtd" [
  <!-- general definitions for tracks -->
  <!ENTITY default-surfaces SYSTEM "../././data/tracks/surfaces.xml">
  <!ENTITY default-objects SYSTEM "../././data/tracks/objects.xml">
  ]>

  <params name="tuxland" type="param" mode="mw">

    <section name="Header">
      <attstr name="name" val="Tux in mountain" />
      <attstr name="category" val="road" />
      <attnum name="version" val="4" />
      <attstr name="author" val="MLuc" />
      <attstr name="description" val="GRA Road Course" />
    </section>
```

Pour modifier votre circuit, inclinaison, surfaces, bords, cotés et barrières, c'est dans ce fichier que vous allez changer ces paramètres.

Après chaque modification de ce fichier XML, avant de tester dans torcs , utilisez la commande :

```
_pour reconstruire la piste      trackgen -n $name -c $category
_pour construire piste et terrain trackgen -a -n $name -c $category
```

Dans la console vous avez certainement des 'WARNING', Torcs râle car il ne trouve pas les images `shadow2.rgb` et `$name.png` : Vous allez commencer par ça :-)

Les images, excepté 3 ou 4 au format PNG, sont toutes par défaut au format SGI (*.rgb) mais l'utilisation du format PNG est possible. La condition importante pour que vos images soient prises en compte, est qu'il faut que leur dimension soit un multiple de 32 (64 128 256 512), 512x512 et 1024x1024 seront un maximum, sauf pour le fond, la carte d'élévation et la carte des ombres.

Si une image ne correspond pas (multiple de 32) vous aurez un avertissement dans la console. Si une image est absente ou incorrecte cela se traduit dans torcs par une coloration rouge de l'objet sensé recevoir l'image (UV Mapping).

2.1 Génération des images :

trackgen est capable de créer des images **PNG** correspondant à l'élévation de votre circuit en utilisant le noir comme niveau le plus bas et le blanc pour le plus élevé.

Nous utiliserons plus loin une de ces images pour construire la carte d'élévation **elevation-map.png**. Vous pouvez donc générer toutes ces images et choisir celle que vous utiliserez, avec la commande :

```
trackgen -a -n $name -c $category -E 0
```

Le type d'élévation généré dépend de **-E** :

- 0** : tous les fichiers d'élévation
- 1** : élévation du terrain + piste
- 2** : élévation du terrain + piste en blanc
- 3** : élévation de la piste seule
- 4** : élévation de la piste en 8 étapes

Pour créer la carte des ombres il est nécessaire que tous les objets soient construits avant de générer cette image, donc pour le moment pour éviter le message d'alerte dans la console vous pouvez charger **\$name-elv.png** dans GIMP, ajouter un canal Alpha et la redimensionner en **1024x512** et l'enregistrer sous **shadow2.rgb** (sans compression RLE).

L'image choisie n'a pas d'importance elle ne sera pas utilisée pour le moment, c'est juste pour calmer torcs :-)

2.1.2 Création de l'image du circuit :

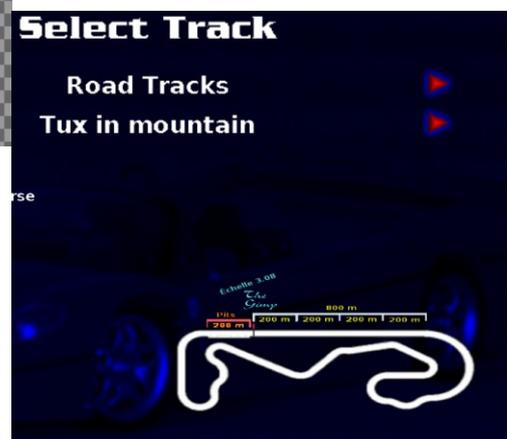
Vous avez remarqué dans Torcs lorsque vous sélectionnez un circuit, dans 'Configure Race', une image représentant la piste est affichée sur la droite de l'écran, c'est cette image (**\$name.png**) que réclame Torcs :

Vous pouvez choisir **\$name-elv4.png**, **elv3** ou **elv2** la charger dans GIMP, ajouter un calque transparent comme fond et dans le premier calque ne garder que la piste (en blanc), re-dimensionner en **512x512** et enregistrer sous **\$name.png**

Vous pouvez y ajouter du texte et la ligne de départ, par exemple :



<-- tuxland.png



dans Torcs -->

2.1.3 Création de l'image de fond :



Par défaut l'image de fond (background.png) de torcs peut ne pas vous convenir, vous avez la possibilité d'utiliser votre image de fond. Le format peut être de 512x512 mais cela n'est pas des plus esthétique, le format maximum sera de 2048x512.

Un bon compromis entre qualité et poids serait une image de 1024x512.

Vous placez évidemment cette image dans le répertoire de votre circuit et vous modifiez la section Graphic du fichier XML, comme suit :

```
<section name="Graphic">
```

```
<attstr name="3d description" val="tuxland.ac" />
```

```
<attstr name="background image" val="background.png"/>
```

background.png est le nom de votre image de fond, vous n'êtes obligé de la nommer ainsi.

Vous pouvez également régler l'ambiance (couleur/luminosité) et définir la position du soleil, en ajoutant à la suite par exemple ceci :

```
<attnum name="background type" val="4"/>
<attnum name="background color R" val="0.5"/>
<attnum name="background color G" val="0.5"/>
<attnum name="background color B" val="0.55"/>
<attnum name="ambient color R" val="0.2"/>
<attnum name="ambient color G" val="0.2"/>
<attnum name="ambient color B" val="0.2"/>
<attnum name="diffuse color R" val="1"/>
<attnum name="diffuse color G" val="1"/>
<attnum name="diffuse color B" val="1"/>
<attnum name="specular color R" val="0.1"/>
<attnum name="specular color G" val="0.1"/>
<attnum name="specular color B" val="0.1"/>
<attnum name="light position x" val="-10000"/>
<attnum name="light position y" val="10000"/>
<attnum name="light position z" val="7000"/>
```

background color, ici la combinaison des valeurs de RGB donne une coloration gris bleu, les autres paramètres définissent la luminosité et les réflexions ici une lumière blanche avec une ambiance plutôt grise.

light position XYZ définit la position de la source lumineuse, le soleil.

Faites varier ces paramètres pour voir leur incidence sur l'environnement lumineux de votre circuit, par exemple si vous positionnez R=0 G=0 B=1 de diffuse color vous allez avoir une lumière bleu et les objets/piste/terrain vont prendre une coloration bleu. Un soleil qui éclaire 'bleu' ce n'est pas commun mais cela peut être utilisé si vous voulez réaliser des circuits de nuit sans pour autant que tout soit 'Noir'. Si vous voulez faire des essais, essayez ceci, une lumière (du jour) bleuté (Montagne) et une ambiance jaune (plage) :

ambient color -> R=0,8 G=0,8 B=0,6

diffuse color -> R=0 G=0 B=0,6

Vous voyez mieux ce qui est en pleine lumière et ce qui est moins éclairé, pour avoir un résultat cohérent (pas comme celui là) il va vous falloir accorder un peu de temps à ces réglages.

2.2 Inclinaison des virages (banking)



Pour un virage à droite la valeur (degré) sera positive et pour un virage à gauche elle sera négative, par exemple pour une inclinaison de 15° :

Virage à droite (rgt)

```
<attnum name="banking start" unit="deg" val="0"/>
<attnum name="banking end" unit="deg" val="15"/>
```

Virage à gauche (lft)

```
<attnum name="banking start" unit="deg" val="0"/>
<attnum name="banking end" unit="deg" val="-15"/>
```

En plus du degré d'inclinaison il faut indiquer la hauteur de l'élévation du coté opposé au virage, pour un virage à droite cela donne :

```
<attnum name="z end left" unit="m" val="3"/>
<attnum name="z end right" unit="m" val="0"/>
<attnum name="banking start" unit="deg" val="0"/>
<attnum name="banking end" unit="deg" val="15"/>
```

ce virage aura une inclinaison de 15° et le coté gauche sera 3 mètres plus haut que le coté droit.

Pour que cela fonctionne il faut que votre virage comporte au minimum 2 segments, le premier pour élever le virage et le dernier pour l'abaisser au niveau du segment suivant. Il peut y avoir autant de segments intermédiaires que vous voulez, c'est à dire un virage peut avoir 3 ou 4 segments ou plus.

Dans l'exemple d'un virage à droite de 180° composé de 3 segments à 60°, le fichier XML (sans banking) ressemble à ceci :

```
<!--*****-->
<!-- Segment nx1 -->
<!--*****-->
<section name="curve 1">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z start" unit="m" val="0.0" />
```

```

    <attnum name="z end" unit="m" val="0.0" />
    <attstr name="surface" val="asphalt2-lines" />

<!--*****-->
<!-- Segment nx2 -->
<!--*****-->
<section name="curve 2">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z start" unit="m" val="0.0" />
  <attnum name="z end" unit="m" val="0.0" />
  <attstr name="surface" val="asphalt2-lines" />
<!--*****-->
<!-- Segment nx3 -->
<!--*****-->
<section name="curve 3">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z start" unit="m" val="0.0" />
  <attnum name="z end" unit="m" val="0.0" />
  <attstr name="surface" val="asphalt2-lines" />

```

Dans chaque segment vous devez enlever les lignes 'z start' et 'z end' en faisant attention à la hauteur qui y est indiquée, pour ajouter la valeur de 'z end' à la valeur du z end de la nouvelle ligne du banking :

```

<!--*****-->
<!-- Segment nx1 -->
<!--*****-->
<section name="curve 1">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z end left" unit="m" val="3.0" />
  <attnum name="z end right" unit="m" val="0.0" />
  <attnum name="banking start" unit="deg" val="0"/>
  <attnum name="banking end" unit="deg" val="15"/>
  <attstr name="surface" val="asphalt2-lines" />
<!--*****-->
<!-- Segment nx2 -->
<!--*****-->
<section name="curve 2">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z end left" unit="m" val="3.0" />
  <attnum name="z end right" unit="m" val="0.0" />
  <attnum name="banking start" unit="deg" val="15"/>
  <attnum name="banking end" unit="deg" val="15"/>
  <attstr name="surface" val="asphalt2-lines" />
<!--*****-->
<!-- Segment nx3 -->
<!--*****-->
<section name="curve 3">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="59.99999999999999" />
  <attnum name="radius" unit="m" val="40.0" />
  <attnum name="z end left" unit="m" val="0.0" />

```

```

<attnum name="z end right" unit="m" val="0.0" />
<attnum name="banking start" unit="deg" val="15"/>
<attnum name="banking end" unit="deg" val="0"/>
<attstr name="surface" val="asphalt2-lines" />

```

2.3 Panneaux de signalisation des virages :



Vous pouvez ajouter une signalisation pour les virages, le positionnement de ces panneaux s'effectue simplement en ajoutant ces lignes dans la section 'Terrain Generation' :

```

<section name="Terrain Generation">

<attnum name="track step" unit="m" val="20" />
<attnum name="border margin" unit="m" val="50" />
<attnum name="border step" unit="m" val="30" />
<attnum name="border height" unit="m" val="15" />
<attstr name="orientation" val="clockwise" />
<attstr name="surface" val="asphalt2-lines"/>

<section name="Turn Marks">
  <attnum name="width" unit="m" val="1.0"/>
  <attnum name="height" unit="m" val="1.0"/>
  <attnum name="vertical space" unit="m" val="1.0"/>
  <attnum name="horizontal space" unit="m" val="2.0"/>
</section>

```

Les 2 premières lignes, de la section 'Turn Marks', définissent les dimensions (largeur, hauteur) du panneau et les 2 dernières la distance verticale et horizontale du bord de la piste.

Ensuite il ne vous reste plus qu'à définir, dans le segment concerné, le nombre de panneaux et son signalement avec

```
<attstr name="marks" val="100;150;200"/>
```

pour avoir 3 panneaux, 200, 150 et 100

ou

```
<attstr name="marks" val="50;100"/>
```

pour 2 panneaux de 100 et 50

dans l'exemple du tutoriel (tuxland.xml) pour le segment 9 vous trouverez ceci :

```

<section name="curve 4">
  <attstr name="type" val="rgt" />
  <attnum name="arc" unit="deg" val="179.0" />

```

```

<attnum name="radius" unit="m" val="79.0" />
<attnum name="z start" unit="m" val="0.0" />
<attnum name="z end" unit="m" val="0.0" />
<attstr name="marks" val="150;200;300"/>
<attstr name="surface" val="asphalt-sw3" />

```

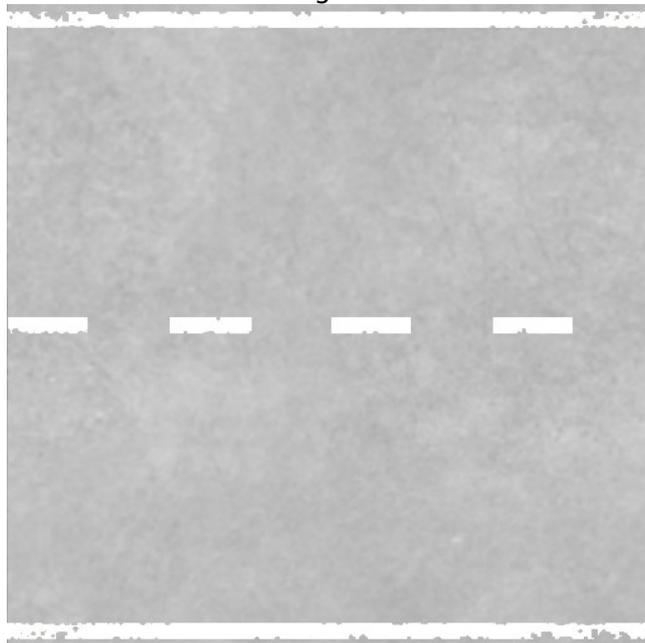
Vous pouvez modifier ou créer vos panneaux, il suffit de les nommer turn200L.rgb, turn200R.rgb, turn150L.rgb, turn150R.rgb... etc. Et de les placer dans le répertoire de votre circuit.

-Voir .../torcs/data/textures

2.4 Modification des surfaces

C'est ici que vous allez pouvoir modifier l'aspect de la piste, des cotés, des bords et des barrières. Pour cela vous devez créer une image de la texture (UV Mapping) qui sera appliquée à la surface que vous aurez choisi, par exemple vous voulez modifier la surface du centre de la piste (asphalte) :

_Vous créez une image, texture, que vous nommez tr-road-as1.rgb



Cette image peut être de
128x128, 256x256 ou 512x512

1024x1024 étant un maximum

Pour remplacer la texture 'asphalt2-lines' par défaut du fichier XML, vous allez commencer par définir les paramètres de la nouvelle surface, dans la section 'Surfaces', votre fichier XML doit commencer et comporter au moins ces lignes :

```

<!DOCTYPE params SYSTEM "../././src/libs/tgf/params.dtd" [
<!-- general definitions for tracks -->
<!ENTITY default-surfaces SYSTEM "../././data/tracks/surfaces.xml">
]>

```

ensuite, au début ou à la fin du fichier vous trouverez la section Surfaces :

```

<!--Surfaces part-->
<section name="Surfaces">
    &default-surfaces;
</section>

```

que vous allez modifier ainsi :

```

<section name="Surfaces">
    &default-surfaces;

```

```

<section name="asphalt-sw3">
  <attnum name="color R1" val="0.1"/>
  <attnum name="color G1" val="0.1"/>
  <attnum name="color B1" val="0.1"/>
  <attnum name="color R2" val="0.2"/>
  <attnum name="color G2" val="0.2"/>
  <attnum name="color B2" val="0.2"/>
  <attstr name="texture name" val="tr-road-as1.rgb"/>
  <attstr name="texture type" in="discrete, continuous" val="discrete"/>
  <attnum name="texture size" val="15.0001" unit="m"/>
  <attnum name="texture mipmap" val="0.0" unit="m"/>
  <attnum name="friction" val="1.15"/>
  <attnum name="rolling resistance" val="0.02"/>
  <attnum name="roughness" val="0.01"/>
  <attnum name="roughness wavelength" val="1.0"/>
</section>

</section>

```

_ Dans la première ligne (section name="asphalt-sw3") vous indiquez le nom (asphalt-sw3) que vous utiliserez pour cette surface, nous y reviendront.

Je passe sur la coloration, je vous laisse faire des essais.

_ Ensuite, 'texture name', là vous indiquez le nom que vous avez donné à votre image et que vous avez bien sûr placé dans le répertoire de votre circuit.

La ligne suivante, discrete/continuous, indique seulement la façon dont va se plaquer/raccorder la texture sur l'objet (surface).

_ 'texture size' c'est là qu'il faut définir la largeur de votre surface, si vous avez une piste de 10m et que vous mettez ici une valeur de 5 votre texture sera répétée 2 fois.

_ 'friction' ce paramètre important indique en fait l'adhérence des pneumatiques sur l'asphalte, ici la valeur 1.15 correspond à une bonne adhérence. Si vous montez au-dessus de 1.5 ça devient de la 'super glue' et si vous descendez en-dessous de 1 ça commence à 'glisser' sérieusement. Par exemple pour les cotés enneigés j'ai utilisé une valeur de 0.6 qui est juste un peu plus glissante que de l'herbe.

Les autres valeurs sont elles aussi en relation avec l'adhérence, l'ondulation ou déformation, la rugosité et la profondeur de la déformation (creux des bosses :-). Pour cette dernière valeur 'roughness wavelength' (en anglais la profondeur de la vague ou de l'onde) sur une piste de GP formule 1 vous pouvez descendre en dessous de 1, mais ici nous sommes sur une route de montagne, pour les bas cotés (sable, herbe, terre...) vous pouvez utiliser des valeurs bien supérieures comme 3, 4 ou 5.

Ne mettez pas non plus des valeurs trop 'parfaite' ce serait peu réaliste.

Bon vous venez de définir cette surface, pour l'utiliser vous remplacer l'ancien nom (asphalt2-lines par exemple) par celui, **asphalt-sw3**, que vous avez attribué à votre nouvelle surface (texture) correspondant donc à l'image **tr-road-as1.rgb**. C'est à dire dans le(s) segment(s) où vous voulez cette surface, vous changez

```

en
  <attstr name="surface" val="asphalt2-lines" />
  <attstr name="surface" val="asphalt-sw3" />

```

C'est le même principe pour TOUTES les surfaces de votre circuit y compris celle du terrain.

Regardez le fichier XML du circuit (Tuxland) de ce tutoriel et tester le dans torcs pour avoir une meilleur idée.

2.5 Création de la carte de l'élévation du terrain :

Dans un circuit sur route avec peu de dénivellation c'est une opération très simple à réaliser... Mais j'ai choisi un paysage de montagne enneigé qui est certainement loin d'être le plus facile à faire.

Pour commencer, si ce n'est déjà fait, charger le circuit du tutoriel dans torcs et regardez le découpage de la montagne. Les défauts ne passent pas inaperçus, vous avez des sommets plats, des pentes abruptes mal placées : C'est la carte d'élévation qui va dessiner ce 'dénivelé'.

Je vais vous indiquer comment créer cette carte (image) et à vous d'essayer d'améliorer le rendu.

```
trackgen -a -E 1 -n $name -c $category
```

génère un fichier `$name-elv.png`

```
trackgen -a -E 2 -n $name -c $category
```

génère un fichier `$name-elv2.png`

J'ai utilisé le premier `$name-elv.png`, à vous de voir :

Maintenant, dans Gimp vous charger `$name-elv.png`, vous ajouter un calque sur le lequel vous allez travailler en noir et blanc (dégradé de gris), vous ajoutez des zones claires et des zones foncées pour modifier la hauteur du terrain, le blanc étant le plus haut et le noir le plus bas.

N'en faites pas trop pour commencer enregistrez le fichier sous **elevation-map.png**

Voici l'exemple du tutoriel (Tuxland)



carte `$name-elv.png`

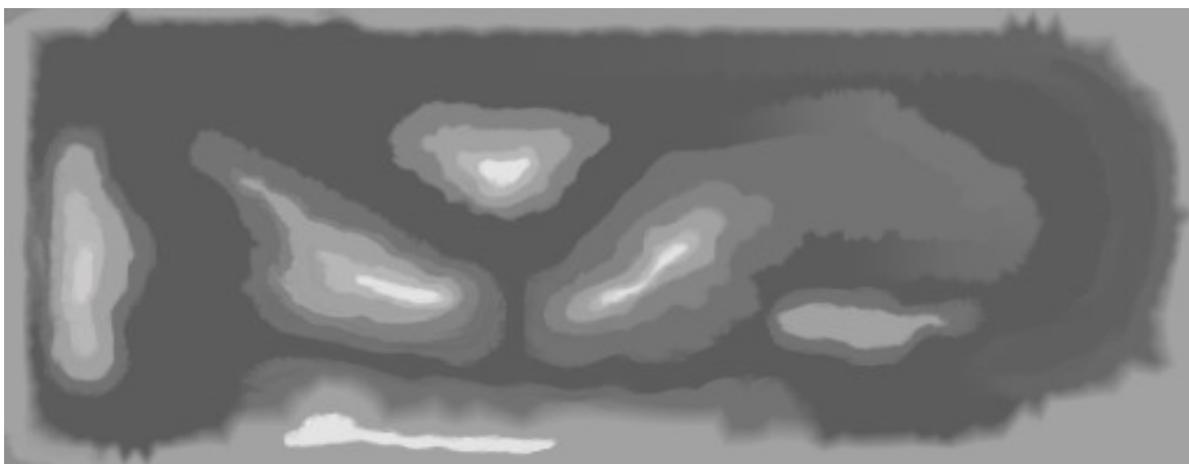
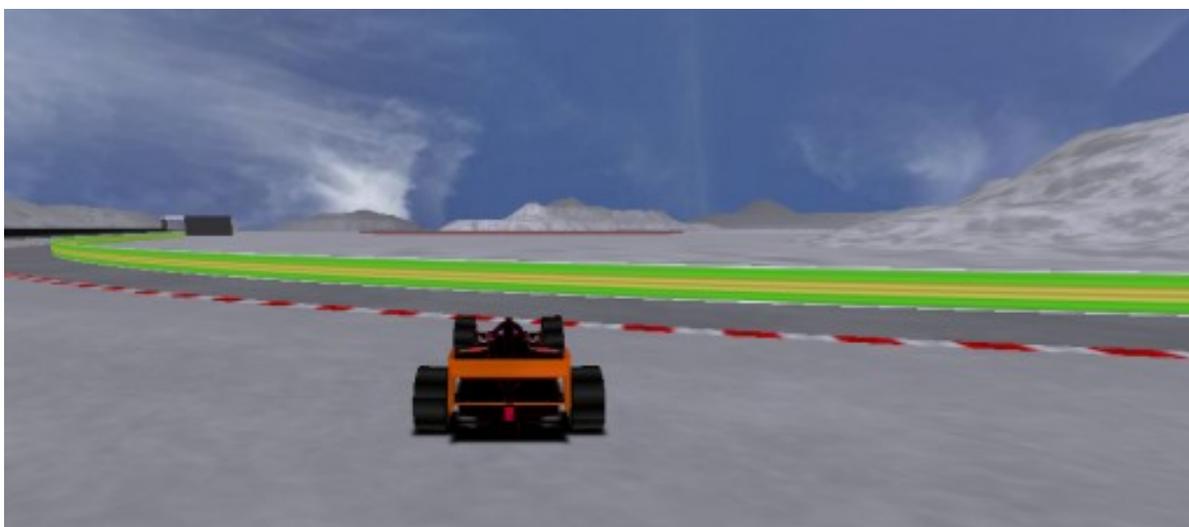


image modifiée dans Gimp

Il vous reste à modifier le fichier `$name.xml` pour y inclure l'élévation du terrain, dans la section « Terrain Generation » ajouter les lignes en rouge :

```
<section name="Terrain Generation">  
  
  <attnum name="track step" unit="m" val="20" />  
  <attnum name="border margin" unit="m" val="50" />  
  <attnum name="border step" unit="m" val="30" />  
  <attnum name="border height" unit="m" val="15" />  
  <attstr name="orientation" val="clockwise" />  
  <attstr name="surface" val="asphalt-sw3"/>  
  
  <attstr name="elevation map" val="elevation-map.png"/>  
  <attnum name="minimum altitude" val="-20"/>  
  <attnum name="maximum altitude" val="35"/>  
  <attnum name="group size" val="150"/>
```

minimum altitude et maximum altitude désigne la valeur en mètres minimum et maximum de la hauteur du terrain correspondant respectivement au Noir et Blanc de votre image.



résultat dans torcs

tester le résultat dans torcs après avoir créé le fichier `.ac` correspondant :
trackgen -c \$category -n \$name -a

2.6 Environnement Mapping

Votre circuit commence à ressembler à ce que vous vouliez en faire... Mais il lui manque encore une petite chose pour ajouter au réalisme, c'est le reflet de l'environnement sur les objets et plus précisément sur les voitures qui se déplacent à l'intérieur de votre circuit.



Pour que cet effet soit réussi il faut utiliser une image peu contrastée et en teinte de gris 'délavé', évitez la couleur sauf peut être si votre circuit est entouré uniquement d'arbres et encore.

Vous pouvez lui donner une teinte générale dans les ocres, bleus ou gris (orientez vous vers des tons pastel plutôt que fluo :-). Cela dépend de l'environnement de votre circuit et si c'est en plein soleil, au crépuscule... etc. A vous d'en juger.

Vous trouverez une image (env.png) sous ../torcs/data/textures



fig.1 env.png



fig.2 env.png

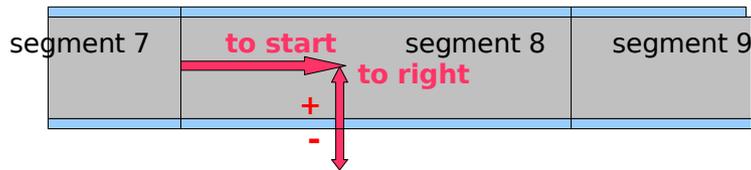
La fig.1 est l'image par défaut de torcs, la fig.2 est celle qui a servi à faire la capture au-dessus: Je pense que c'est assez parlant.

2.7 Les caméras :

Cela n'est vraiment pas indispensable en mode multi-joueurs, mais ça devient intéressant si vous faites des courses de 'robot' ou pour faire une vidéo.

Donc vous pouvez ajouter des caméras dans les segments là où vous voulez un angle de prise de vue différent.

Dans le principe il suffit de définir les coordonnées de la position de la caméra par rapport au bord droit du départ du segment en indiquant la distance latérale depuis ce bord (**to right**) et la distance depuis le début de ce segment (**to start**) ainsi que la hauteur (**height**), altitude .



Mais prenons un exemple :

Pour placer des caméras (une par segment) dans l'enchaînement de virages des segments 20, 21 et 22 du circuit du tutoriel, nous allons placer la première un peu avant le virage (20) c'est à dire dans la ligne droite (19) à 300m de son début, à 5m du bord et à 8m de haut

```
<attstr name="segment" val="straight 4"/>
<attnum name="to right" val="5"/>
<attnum name="to start" val="300"/>
<attnum name="height" val="8"/>
```

maintenant il faut lui indiquer la zone active c'est à dire les segments de début et de fin, ici :

```
<attstr name="fov start" val="straight 4"/> <!-- segment 19 -->
<attstr name="fov end" val="curve 11"/> <!-- segment 21 -->
```

les segments 19 et 20 (straight 4 et curve 10) seront inclus mais pas le segment segment 21

Voici un exemple de jeu de caméras pour ces virages, que vous placerez à la fin du fichier XML :

```
<section name="Cameras">

  <section name="cam 1">
    <attstr name="segment" val="straight 4"/>
    <attnum name="to right" val="5.0"/>
    <attnum name="to start" val="200"/>
    <attnum name="height" val="8"/>
    <attstr name="fov start" val="straight 4"/>
    <attstr name="fov end" val="curve 10"/>
  </section>

  <section name="cam 2">
    <attstr name="segment" val="curve 10"/>
    <attnum name="to right" val="-3.0"/>
    <attnum name="to start" val="5"/>
    <attnum name="height" val="4"/>
    <attstr name="fov start" val="curve 10"/>
    <attstr name="fov end" val="curve 11"/>
  </section>

  <section name="cam 3">
    <attstr name="segment" val="curve 11"/>
    <attnum name="to right" val="0.0"/>
    <attnum name="to start" val="5"/>
    <attnum name="height" val="4"/>
    <attstr name="fov start" val="curve 11"/>
    <attstr name="fov end" val="curve 12"/>
  </section>

  <section name="cam 4">
    <attstr name="segment" val="curve 12"/>
    <attnum name="to right" val="2.0"/>
    <attnum name="to start" val="20"/>
    <attnum name="height" val="10"/>
    <attstr name="fov start" val="curve 12"/>
    <attstr name="fov end" val="straight 5"/>
  </section>

</section>
```

Attention à la hauteur et au décalage droit, en montage la caméra risque de se retrouver sous le niveau du terrain ou derrière une élévation.

Une caméra peut, être active sur plusieurs segments, ne pas être dans l'axe de la piste si le terrain (visibilité) le permet. Et il peut y avoir plusieurs caméras sur un même segment.

Si vous n'avez rien modifié dans votre fichier XML, l'ajout des sections 'camera' ne nécessite pas la reconstruction du circuit avec trackgen, sinon :

trackgen -n \$name -c \$category -a

chargez le circuit dans Torcs, sélectionnez 'Practice' puis choisir un robot, et vous pouvez regarder le résultat en mode '**TV Director View**' touche F11.

2.8 Les Objets :

Note : Dans le cas où l'ajout d'objets, si vous avez utilisé Blender ou ac3d, pose un problème lorsque vous lancez Torcs qui se plaint parce que votre fichier .ac comporte des 'crease 30' ou 'crease 45', il suffit de supprimer ces lignes du fichier .ac.

Si vous êtes sous Linux utilisez cette commande pour effectuer la suppression des lignes 'crease' :

```
sed '/crease/d' $name.ac > $name.ac2 && cp $name.ac2 $name.ac
```

Placer des objets comme des arbres par exemple est quelque chose de très simple à faire, mais pas évident à visualiser, vous allez comprendre.

Utiliser des objets définis dans Torcs :

Vous allez, pour ajouter des objets à votre circuit, réaliser une image (carte) contenant la position de ces objets. Pour que cela fonctionne il faut vérifier la présence de lignes de paramètres, sinon ajoutez ces lignes, au début de votre fichier XML :

```
<!DOCTYPE params SYSTEM ".././../src/libs/tgf/params.dtd" [  
<!-- general definitions for tracks -->  
<!ENTITY default-surfaces SYSTEM ".././../data/tracks/surfaces.xml">  
<!ENTITY default-surfaces SYSTEM ".././../data/tracks/objects.xml">  
>
```

puis vous ajoutez une section 'Objects', après la section Surfaces (texture) est une bonne idée, mais cela n'a pas d'importance :

```
<section name="Objects">  
    &default-objects;  
</section>
```

c'est tout, pour le moment! :-)

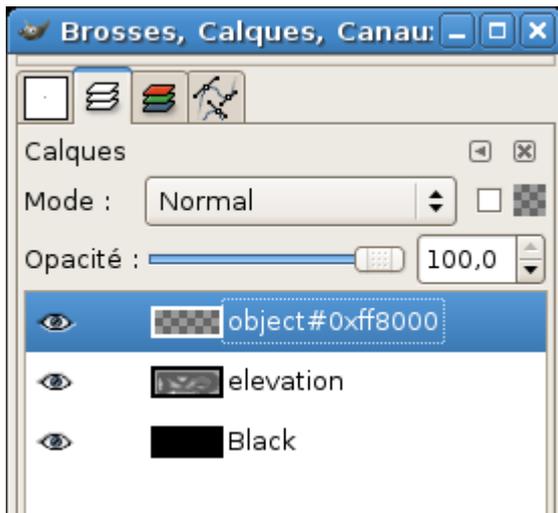
Le principe est enfantin, vous créez une image de la taille de votre carte (1024x398 pour tuxland) dans laquelle vous allez placer des points de 1 pixel de couleurs différentes, chaque couleur correspond à un objet défini dans le fichier torcs/data/tracks/objects.xml.

Donc pour connaître par exemple la couleur à utiliser pour un arbre qui se nomme 'tree15', vous éditez ce fichier objects.xml et à la ligne 100 vous trouvez ceci :

```
<section name="tree15">  
    <attstr name="object" val="tree15.ac"/>  
    <attnum name="color" val="0xFF8000"/>  
    <attstr name="orientation type" val="random"/>  
</section>
```

la valeur ff8000 correspond à une teinte orange, et alors....

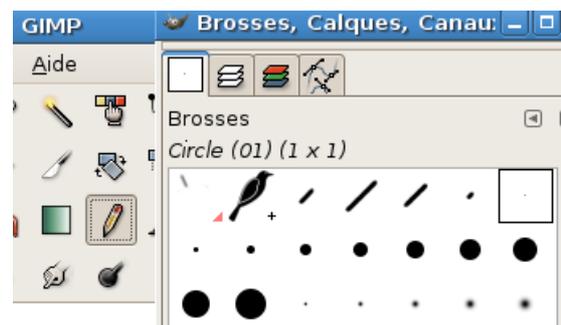
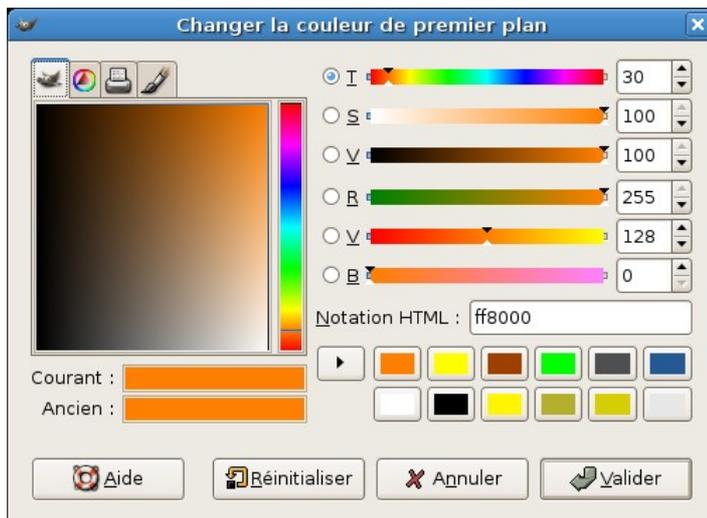
Pour chaque pixel orange (ff8000) que vous placez sur votre carte, trackgen va construire un arbre 'tree15' qui ressemble à cela -->



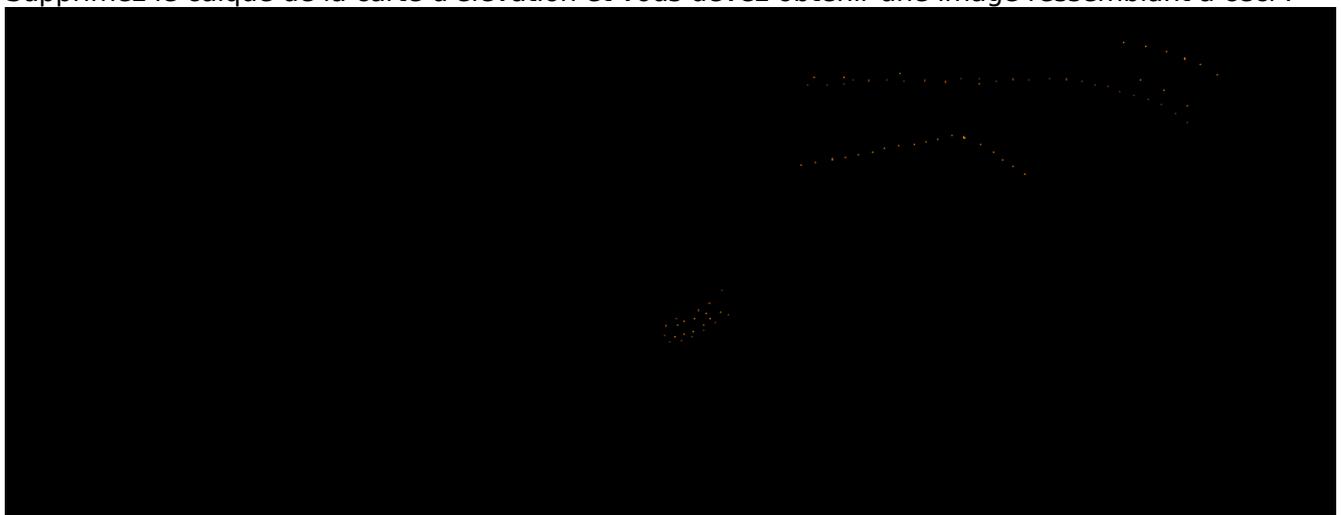
Mais comme ce n'est pas évident de les placer au bon endroit, vous allez vous aider d'une carte d'élévation, \$name-elv2.png par exemple, qui servira de guide pour positionner vos arbres ou autres objets.

Vous chargez donc votre carte d'élévation dans Gimp, vous ajoutez un calque transparent en première position (en haut) dans lequel vous placerez vos pixels de couleurs, puis vous ajoutez en dernière position (en bas) un calque transparent que vous remplissez entièrement de couleur noire.

En vous basant sur la description du fichier Objects.xml vous allez changer la couleur du premier plan pour l'utilisation du crayon (peindre) avec une brosse ronde pleine de 1 pixel (la plus petite) et peindre vos pixel là où vous désirez avoir des arbres ou autres objets correspondant à la couleur que vous avez défini.



Supprimez le calque de la carte d'élévation et vous devez obtenir une image ressemblant à ceci :



et vous enregistrez l'image sous **objects-map.png**

Pour que trackgen utilise cette carte il faut lui indiquer le nom en ajoutant, dans votre fichier XML, une section 'Object Maps' :

```
<section name="Terrain Generation">

  <attnum name="track step" unit="m" val="20" />
  <attnum name="border margin" unit="m" val="50" />
  <attnum name="border step" unit="m" val="30" />
  <attnum name="border height" unit="m" val="15" />
  <attstr name="orientation" val="clockwise" />

  <section name="Object Maps">
    <section name="map 1">
      <attstr name="object map" val="objects-map.png"/>
    </section>
  </section>
</section>
```

Pour finir on construit le circuit avec la commande :

```
trackgen -c $category -n $name -a
```

Et le résultat dans Torcs :



Ça c'est la méthode simple et rapide pour ajouter des objets, il y a d'autres façons de faire. Mais avant de voir 2 autres possibilités de placement d'objets, vous avez sans aucun doute remarqué, dans l'image ci-dessus, le peu de crédibilité de ces arbres dans un paysage de montagne sous la neige.

Je n'ai donc ni utilisé ces arbres ni cette méthode pour ce circuit de démo, mais j'ai modélisé les objets pour y plaquer des textures de sapins, pins... etc. blanchis par la neige !

Ce type d'objet est simple à réaliser, ce sont 2 plans à 90° dont l'intersection est le centre (tronc) de l'arbre. Vous trouverez dans l'archive du circuit du tutoriel tous les fichiers (et sources) qui ont servis à sa réalisation.

Bon continuons avec ce principe d'insertion d'objets, donc les objets disponibles ne vous conviennent pas et vous préférez modéliser les vôtres... Aucun problème :-)

Supposons que vous avez réalisé 3 type d'arbre et un bâtiment, vous placez les fichiers .ac dans le répertoire de votre circuit et vous éditez un fois de plus votre fichier XML.

Il suffit, comme dans le fichier Objects.xml, de définir les paramètres pour ces objets dans la section 'Objects' :

```
<section name="Objects">
    &default-objects;

    <section name="arbre1">
        <attstr name="object" val="arbre_01.ac"/>
        <attnum name="color" val="0x00FFFC"/>
        <attstr name="orientation type" val="random"/>
    </section>

    <section name="arbre2">
        <attstr name="object" val="arbre_02.ac"/>
        <attnum name="color" val="0xBAFF00"/>
        <attstr name="orientation type" val="random"/>
    </section>

    <section name="arbre3">
        <attstr name="object" val="arbre_03.ac"/>
        <attnum name="color" val="0xE14038"/>
        <attstr name="orientation type" val="random"/>
    </section>

    <section name="stand_nord">
        <attstr name="object" val="tribune_01.ac"/>
        <attnum name="color" val="0xE138D7"/>
        <attstr name="orientation type" val="standard"/>
        <attnum name="orientation" val="180" unit="deg"/>
    </section>

    <section name="stand_sud">
        <attstr name="object" val="tribune_01.ac"/>
        <attnum name="color" val="0xE100D4"/>
        <attstr name="orientation type" val="standard"/>
        <attnum name="orientation" val="0" unit="deg"/>
    </section>

    <section name="stand_est">
        <attstr name="object" val="tribune_01.ac"/>
        <attnum name="color" val="0xE168DA"/>
        <attstr name="orientation type" val="standard"/>
        <attnum name="orientation" val="90" unit="deg"/>
    </section>

    <section name="stand_ouest">
        <attstr name="object" val="tribune_01.ac"/>
        <attnum name="color" val="0xE194DC"/>
        <attstr name="orientation type" val="standard"/>
        <attnum name="orientation" val="270" unit="deg"/>
    </section>

</section>
```

Attention si vous utilisez une couleur déjà définie dans Objects.xml, cela ne fonctionnera pas!

Pour les arbres ça reste encore simple, leur orientation étant aléatoire, vous devez juste indiquer le nom du fichier ac utilisé et la couleur pour le pixel dans votre carte.
Pour le bâtiment, ici une tribune (Stadium), ça se complique. La tribune a une orientation, à positionner, du fait qu'elle est dirigée vers le circuit et comporte au moins 4 cotés à positionner sur les axes x,y et z.

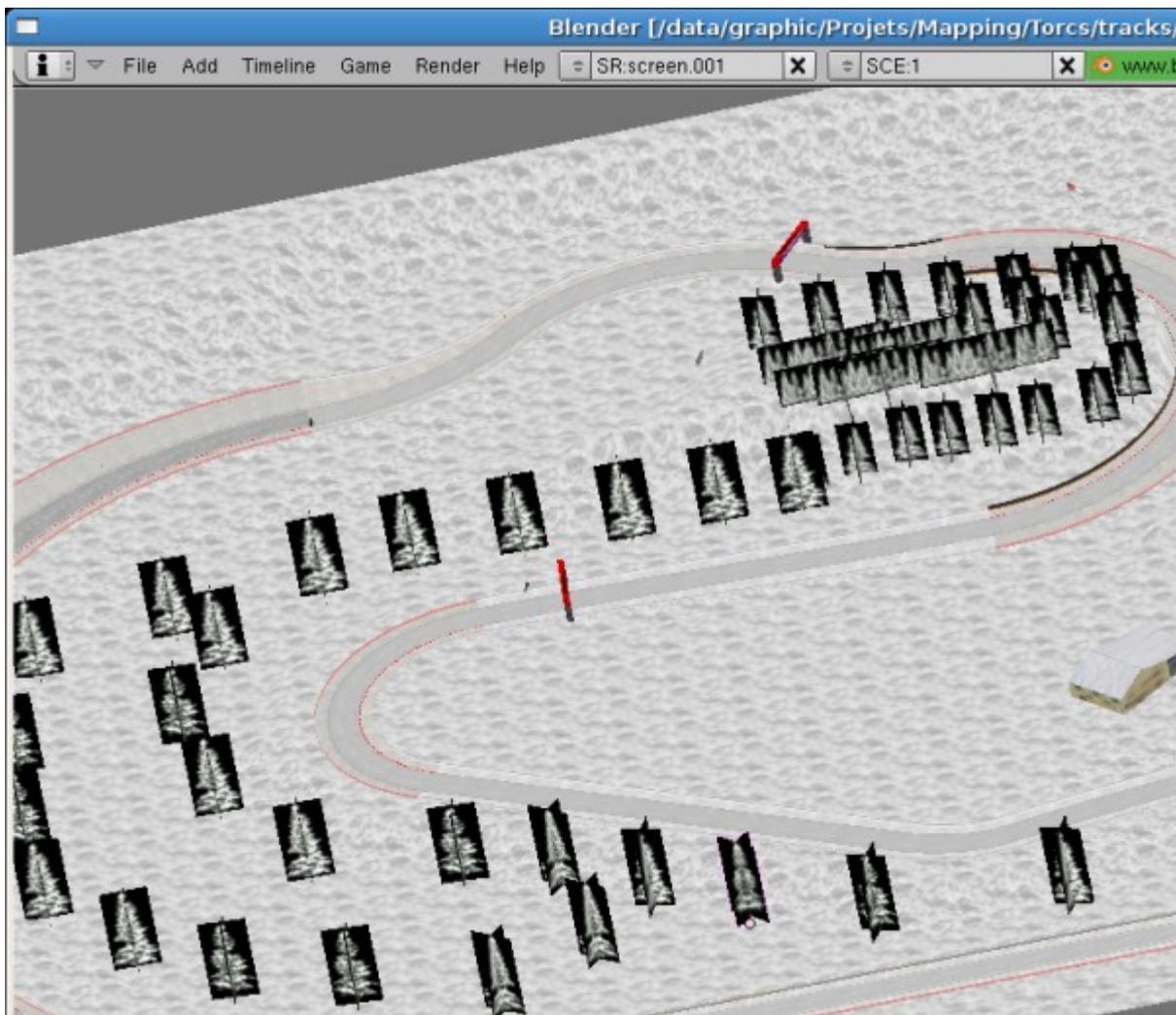
Utiliser vos objets modélisés :

Cela reste en fin de compte certainement la méthode la plus utilisée et la plus évidente. Personnellement j'utilise Blender, mais étant donné la quantité de modeleurs 3D disponibles, vous utilisez le votre.

Donc vous construisez vos objets dans Blender, vous les positionnez, vous leur appliquez une texture (UV mapping) et vous exportez votre circuit au format ac3d (.ac).

Vous n'avez plus besoin, dans le fichier XML, de la section 'Objects' ni des images 'objects-map.png', surtout à partir de ce moment vous faites une sauvegarde de votre fichier .ac ainsi exporter et vous ne modifiez plus votre circuit avec trackgen et le fichier XML.

L'initiation à la modélisation n'étant pas le but de ce tutoriel, je vous laisse choisir votre méthode d'ajout d'objets dans votre circuit.



2.9 Création de la carte des ombres :

La méthode 'coué' ne fonctionnant pas trop ici, je vous en propose 2 autres :-)

A partir du fichier de description 3D (*.ac) :

Pour créer l'image qui va servir de base, dans un terminal vous tapez la commande :

```
trackgen -a -E 1 -n $name -c $category
```

cela va générer un fichier **\$name-elv.png**, ou avec cette commande :

```
trackgen -a -E 2 -n $name -c $category
```

pour générer un fichier **\$name-elv2.png**, vous choisissez le fichier 'd'élévation' qui vous convient le mieux.

Ensuite vous chargez ce fichier d'élévation dans **Gimp** et vous ajoutez un calque blanc que vous descendez en dernière position.

Vous sélectionnez le premier calque (\$name-elv.png) et pour obtenir l'image des ombres vous allez appliquer un filtre **Filtre -> Distorsion -> Repoussage**

Pour diminuer un peu le contraste obtenu vous allez à nouveau appliquer un filtre pour obtenir un flou cinétique avec un angle à 360° **Filtre-> Flou -> Flou cinétique**

Maintenant il faut mettre ces ombres sur un fond blanc. Vous sélectionnez, dans le premier calque, avec l'outil de sélection de couleur, la couleur dominante de votre calque qui doit être un ton de gris et vous la supprimez Edition -> Couper.

Vous re-dimensionnez l'image en 1024x512, image -> taille et échelle de l'image.

Vous enregistrez votre image sous shadow2.rgb sans compression RLE.

A l'aide de Blender à partir du fichier .ac :

Le but ici n'est pas d'expliquer le maniement de **Blender**, peut être pour un prochain tutoriel. Je suppose donc que vous en connaissez les manipulations de base.

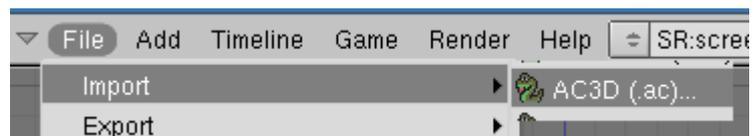
Avec Blender il est nécessaire de placer un éclairage pour simuler le soleil, cela va demander quelques manipulations supplémentaires par rapport à la méthode précédente. Il est donc préférable d'avoir placé tous les objets, bâtiments, arbres, véhicules, ponts... etc. pour ne pas avoir à refaire cette construction. Je vous rappelle, pour éviter les mauvaises surprises, qu'il ne faut pas oublier lorsque vous construisez des objets, dans Blender, de les transformer en triangles avec la combinaison de touches **CTRL + T**.

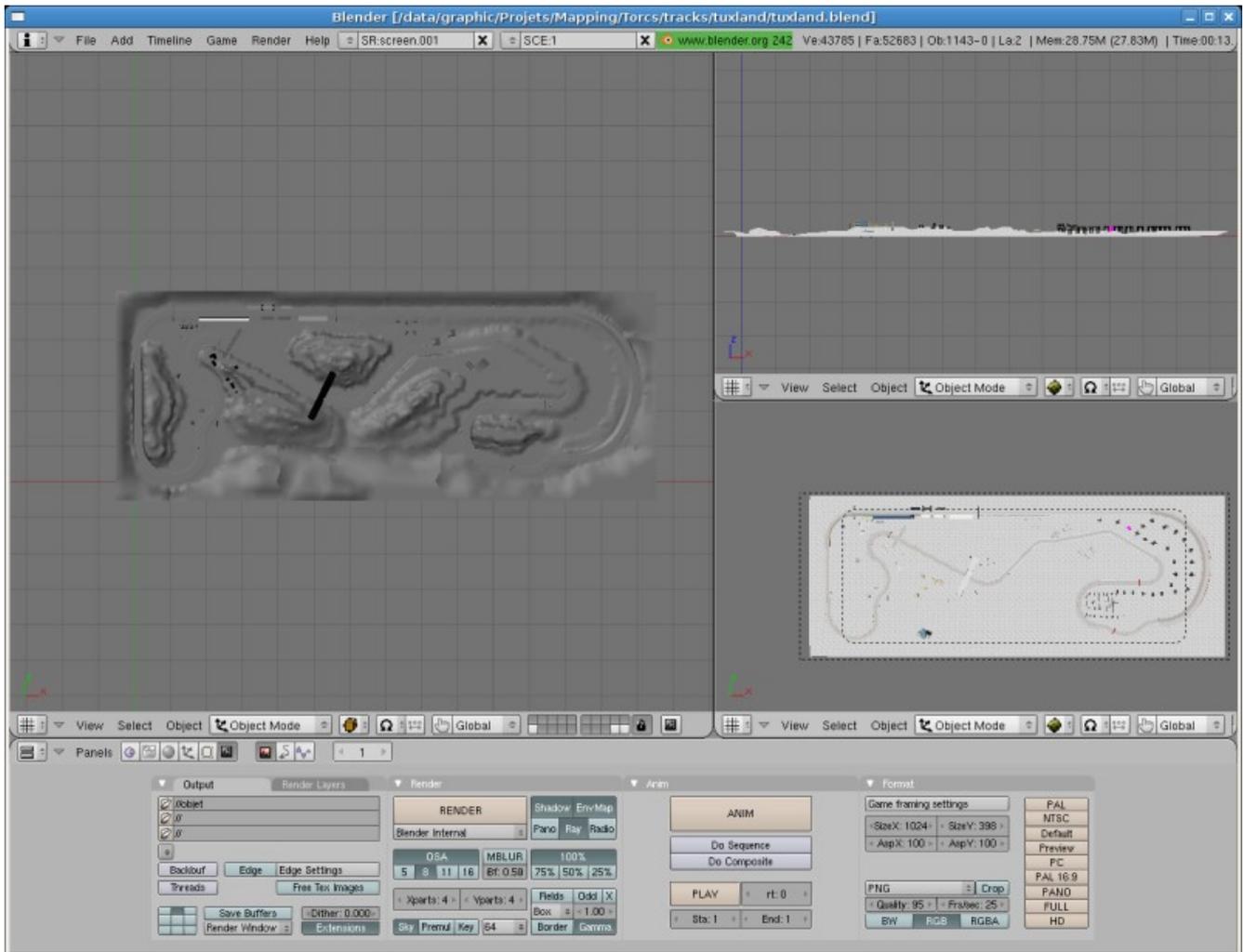
Comme les ombres dans votre circuit sont créées par la position du soleil que vous avez défini dans le fichier xml, commencez par relever ces données:

```
<section name="Graphic">
  <attstr name="3d description" val="tuxland.ac" />
  <attstr name="background image" val="skyclouded.png"/>
  <...
  ....>
  <attnum name="light position x" val="-10000"/>
  <attnum name="light position y" val="10000"/>
  <attnum name="light position z" val="7000"/>
```

Chargez votre fichier .blend ou importez le fichier .ac de votre circuit dans Blender. Pour Importer ou Exporter un fichier ac (ac3d) :

Menu File -> Import/Export -> AC3D (ac)





Placer la lumière :

Ajoutez une lampe (qui produit des ombres) du type 'Spot' ou 'Sun'.

Sélectionnez votre lampe et ouvrez le menu des coordonnées (touche **n**)

Vous entrez les valeurs relevées dans votre fichier xml, c'est à dire ici :

$x = -10000$, $y = 10000$ et $z = 7000$

Ces valeurs sont importantes car elles correspondent à la position du soleil qui est tout de même très éloigné de votre circuit :-)

Vous éditez les paramètres de la lampe pour lui donner une portée (Dist:) supérieure à 2000 et vous augmentez la puissance (Energy:) de façon à ne pas obtenir une image trop sombre.

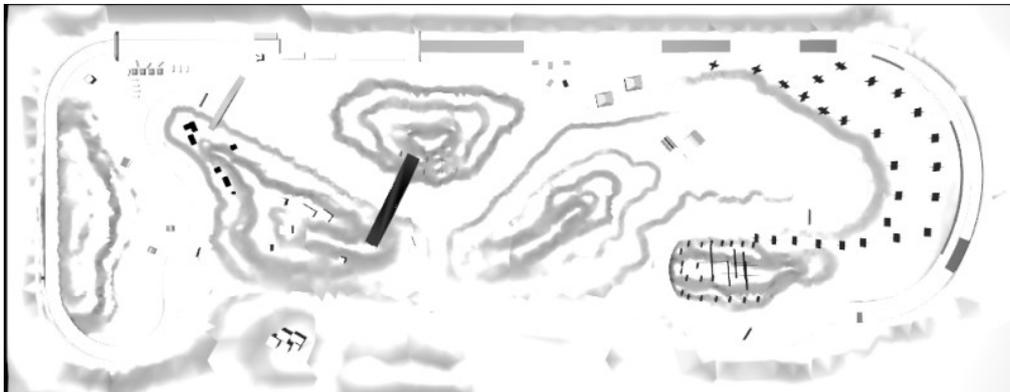
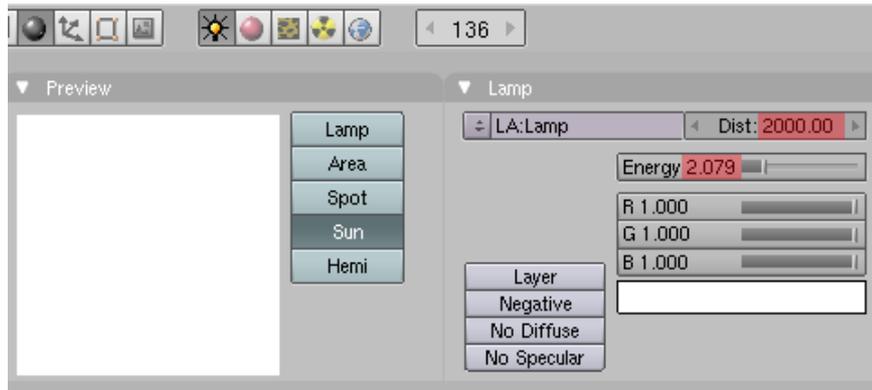


fig.4 rendu sous blender

La figure (fig.4) a été obtenue avec une lampe de type 'Sun' et avec les réglages figure 5.



fig,5 Éclairage avec Sun

Mais avant de pouvoir faire ce rendu il faut positionner la caméra.

Placer la caméra :

Comme pour la lampe, vous pouvez entrer les valeurs définissant la position dans la fenêtre des coordonnées de la caméra, et pour reprendre l'exemple de **Vicente Martí Centelles** (fig.6), vous pouvez aussi calculer la position de votre caméra (valeurs de x et y).

Sinon la méthode simple consiste, en vous basant sur ce que voyez dans la vue camera, à déplacer (G) la caméra jusqu'à obtention d'une position correcte affichant toute la carte du circuit à l'intérieur du cadre de la caméra.

Pour le calcul des valeurs x et y correspondant au centre de votre circuit, relevez les coordonnées du coin gauche en bas et du coin droit en haut puis appliquer cette règle, $X = (x1+x2)/2$ et $Y = (y1+y2)/2$ le centre étant situé à la moitié de la largeur (y) et de la longueur (x).

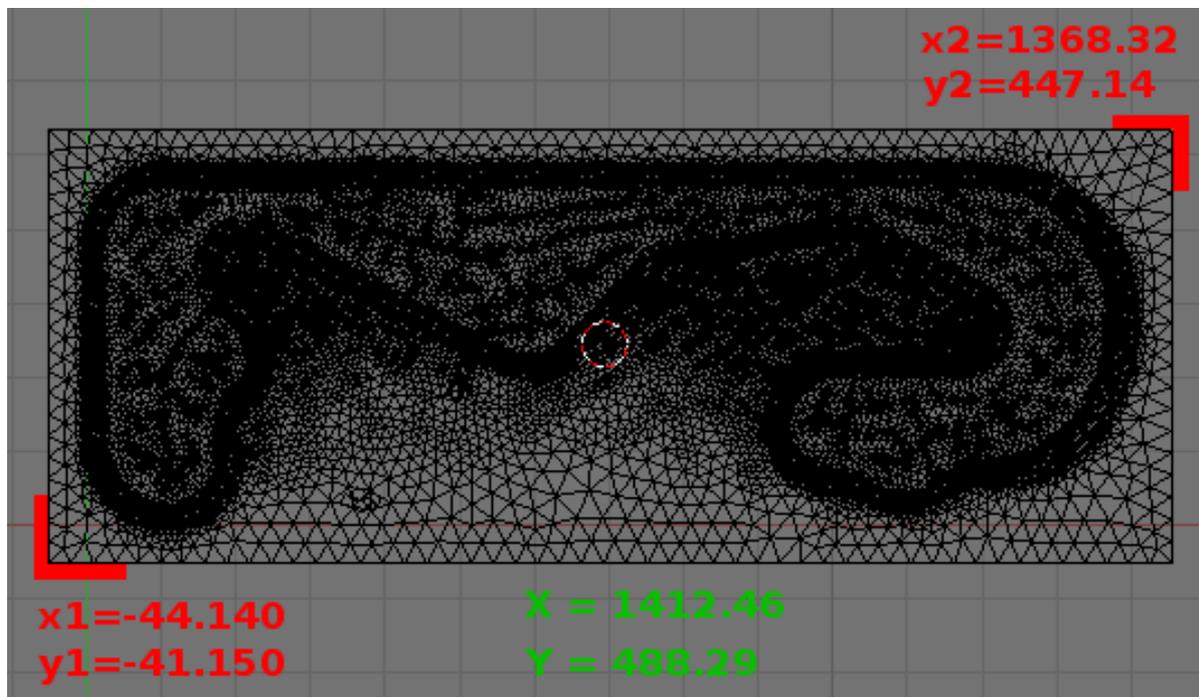
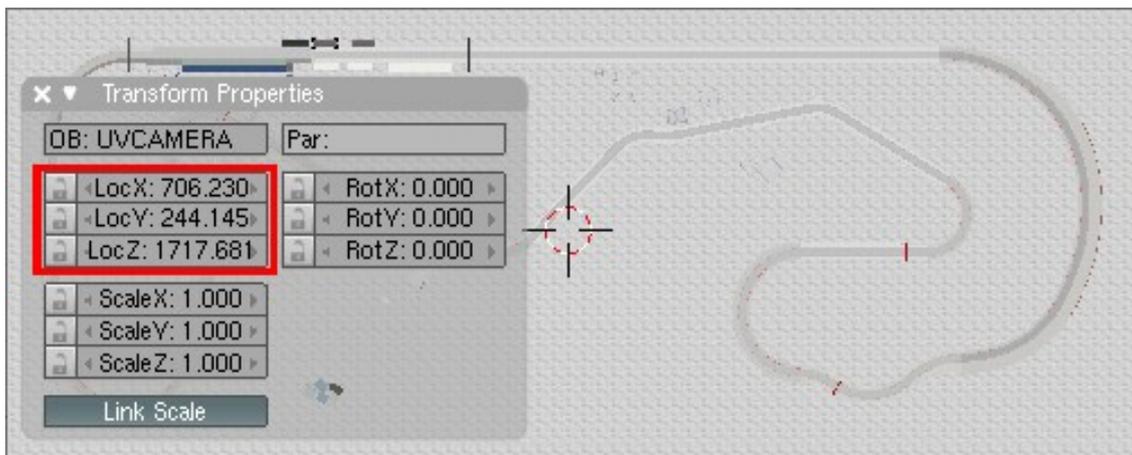


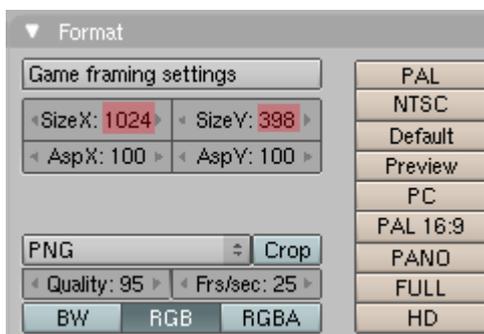
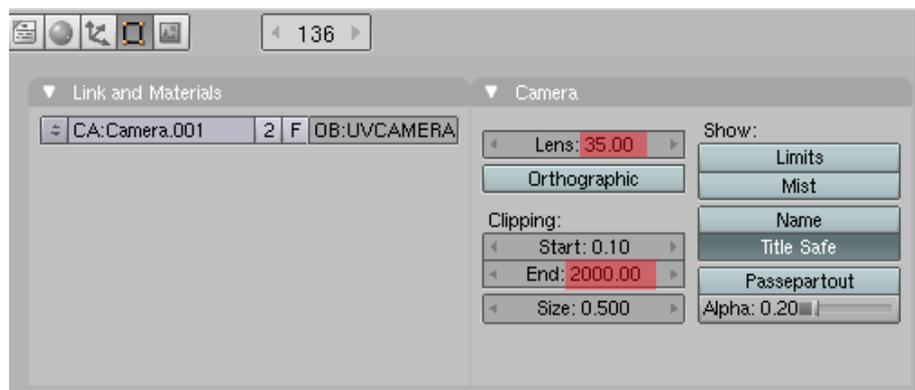
fig.6 coordonnées X Y dans Blender

Ensuite sélectionnez votre caméra et ouvrez sa fenêtre des coordonnées :



la hauteur, valeur de Z, devant être suffisante (entre 1000 et 2000) pour voir entièrement votre circuit dans la vue caméra, cela dépend évidemment de la taille du circuit.

Vous aurez certainement besoin de modifier d'autres valeurs des paramètres de la caméra, comme la focale et la distance maximum de visibilité.



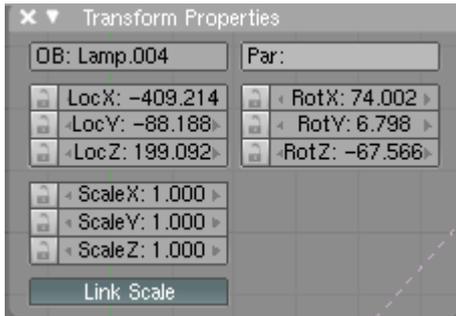
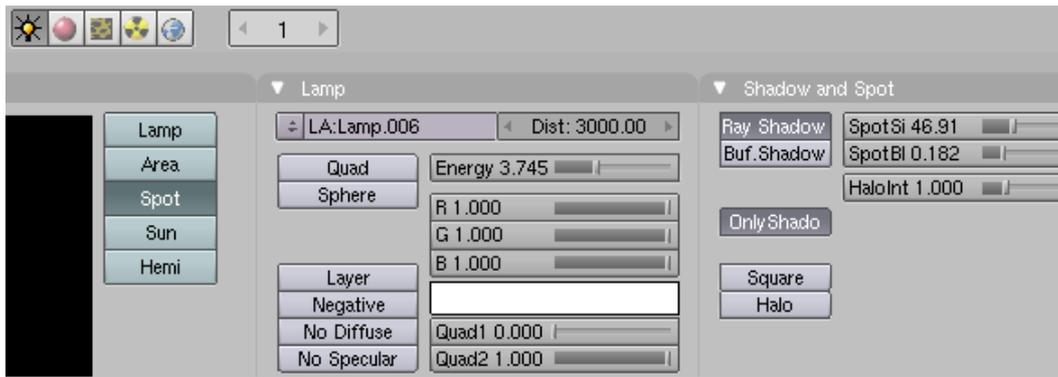
Avant de faire un rendu de votre circuit et de l'enregistrer au format PNG, vous devez indiquer la taille de l'image qui doit correspondre, par exemple, à celle de \$name-elv.png soit 1024x398 dans ce tutoriel.

Vous cliquez sur le bouton 'Render' et vous enregistrez l'image sous shadow2.png

Bon ce n'est pas toujours très satisfaisant comme résultat du fait que l'éclairage est très haut et très à gauche, mais il ne faut pas oublier que pour ce circuit de démo nous sommes en montage et descendre la position du soleil risque de produire des ombres trop importantes; Mais il est toujours possible de tricher 'un peu' :-)

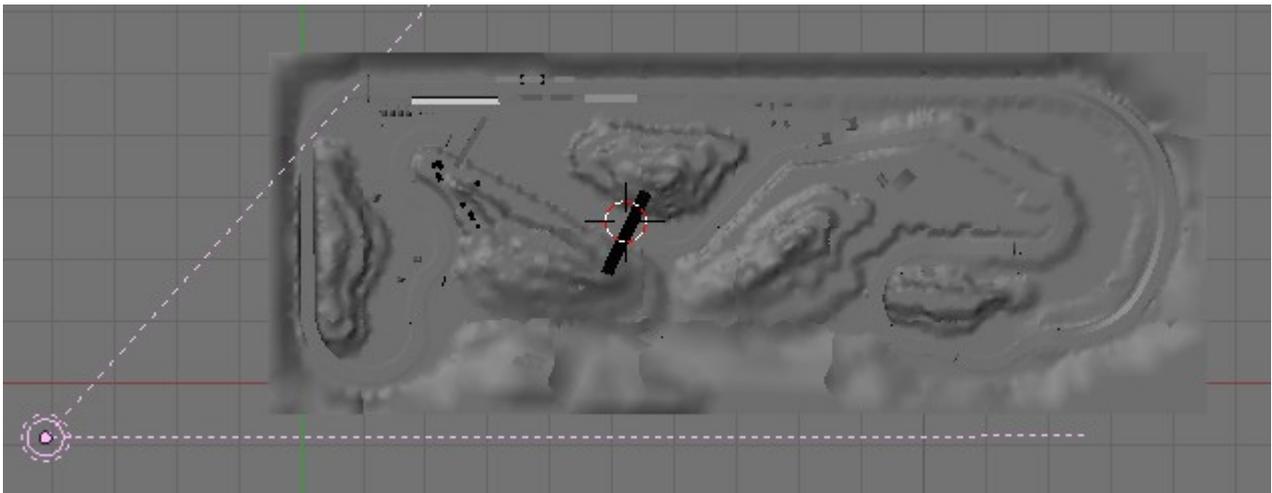
Alors voici ce que j'ai fait pour obtenir des ombres plus marquées pour ce circuit.

- _ Vous baissez la puissance de votre lampe 'Sun', entre 0,7 et 0,5
- _ Vous ajoutez une lampe 'Spot' pour produire, uniquement, des ombres
- _ Vous réglez les coordonnées de ce 'Spot', images ci-dessous
- _ Et vous faites un rendu qui devrait ressembler à la figure 7 (fig.7).



pour la position
 $x = -409,214$
 $y = -88,188$
 $z = 199,092$

pour l'angle (rotation)
 $x = 74,002$
 $y = 6,798$
 $z = -67,566$



C'est loin de correspondre à la position du soleil, cela n'a pour unique but que d'allonger les ombres.



fig.7 rendu dans blender

Avec cette image ainsi obtenu il vous faut l'éditer dans Gimp et faire les modifications décrites dans la première méthode pour obtenir un fichier shadow2.rgb convenable.

Le résultat après les manipulations dans Gimp est celui de la figure 8.



fig.8 shadow2.rgb

Génération des ombres :

Cette image **shadow2.png** va maintenant servir à construire un fichier .acc, à l'aide de **acc**, qui correspondra à votre fichier .ac plus les ombres :

Dans un terminal, se placer dans le répertoire de votre circuit

```
cd ../torcs/tracks/$category/$name
```

Les paranos peuvent faire une sauvegarde :

```
cp $name.ac $name-bak.ac
```

acc va utiliser le fichier **\$name.ac** pour construire les ombres à partir de l'image shadow2.rgb, et le résultat sera enregistré dans le fichier **\$name-shade.ac**

```
acc +shad $name.ac $name-shade.ac
```

il reste à combiner les ombres avec le circuit dans un nouveau fichier **\$name.acc**:

```
acc -g $name.acc -l0 $name.ac -l1 $name-shade.ac -d3 1000 -d2 500 -d1 300 -S 300 -es
```

si le fichier shadow2.rgb n'est pas dans votre répertoire **\$name** vos textures vont prendre une teinte rouge, vous devez utiliser également les options -d1 -d2 -d3 et -S, vous pouvez omettre ces options au risque de voir disparaître votre terrain.

Pour plus d'info. sur les options de **acc** : **acc -help**

Un script **generate.sh** fera ce travail, vous le trouverez dans l'archive du circuit de ce tutoriel (tuxland_tut-track.zip), disponible à cette adresse :

<http://cern91.tuxfamily.org/pub/docs/Jeux/Torcs/>

3 A propos du circuit de ce tutoriel :

Pourquoi ne pas avoir utilisé un circuit existant dans Torcs ?

D'une part il n'est pas facile d'expliquer comment et pourquoi avoir procédé d'une certaine façon lorsque cela a été réalisé par une autre personne, d'autre part je voulais que ce circuit n'utilise que des textures qui ont été créées pour ce circuit. Le but étant de montrer que cela représente un grand nombre d'images et que tout, ou presque, est possible sous Torcs.

Ce que je n'ai pas eu le temps de faire est un plan d'eau et une traversée de piste (pont). Mais Eric Espie (merci :-) a réalisé ce type de circuit, pour une voie en traversant une autre par un pont vous pouvez voir ça dans le circuit 'wheel-2', pour la surface d'eau je n'ai pas le nom en mémoire mais cela existe et ne présente pas de difficultés particulière, une dénivellation (trou) avec un plan horizontal sur lequel on applique une texture pour simuler la surface de l'eau.

Un oubli, j'ai commencé ce circuit en 2007 et c'est en écrivant ce texte, en 2008 que je me suis aperçu avoir complètement oublié, de faire et d'en parler, des inclinaisons de la piste (cotes).

Bon ceci est possible en modifiant les valeurs de 'z' de début et fin de chaque segment

```
<attnum name="z start" unit="m" val="0.0" />
```

```
<attnum name="z end" unit="m" val="2.0" />
```

mais Torcs offre une autre possibilité beaucoup plus intéressante qui est d'ajouter et de modifier le pourcentage de la pente de la piste et son profile ("profil tangent") à l'aide de la valeur attribuée à "grade" qui sera positive pour monter et négative pour descendre.

on monte :

```
<attnum name="grade" unit="%" val="2" />
```

```
<attnum name="profil end tangent" unit="%" val="1" />
```

on descend :

```
<attnum name="grade" unit="%" val="-4" />
```

```
<attnum name="profil end tangent" unit="%" val="-4" />
```

Vous avez un excellent exemple d'utilisation, en montage, avec alpine-2 de Eric Espie.

Éditez son fichier XML pour avoir une meilleur idée de la façon de faire, et/ou vous pouvez aussi récupérer, dans l'archive des fichiers sources de pistes, le circuit « sevennes » que j'utilise pour le réglage de mes véhicules.

Ce circuit n'est absolument pas abouti, mais il est jouable et vous pouvez vous y amuser, mais son but est de vous permettre de refaire les manipulations décrites dans ce tutoriel.

Ensuite, pourquoi pas, essayer d'en faire un 'Vrai' circuit :-)

3.1 Contenu de l'archive :

Le nom du répertoire est donc '**tuxland**' à placer soustorcs/tracks/**road**/

ce répertoire contient un sous-répertoire '**src**' dans lequel vous trouverez les fichiers sources qui ont servis à créer les images, textures, carte d'élévation, carte Objects Map... etc.

Vous trouverez également 2 scripts, **generate.sh** pour créer votre circuit + terrain à partir du fichier XML et **cleanblendac.sh** pour générer votre circuit + terrain à partir du fichier ac exporté par Blender.

note: Les fichiers dans le sous-répertoire src avec l'extension **xcf** sont les textures, images+calques, réalisées dans **The GIMP**.

3.2 Les fichiers et les images :

Le fichier XML (tuxland.xml) utilise le fichier ac (tuxland.ac) et non le fichier acc, pourquoi ?

Simplement parce que j'ai réalisé très rapidement ce circuit sous Blender en important des objets d'autres fichiers blender et que l'origine (centre des axes) de ces objets est différente de l'origine du circuit et du terrain, plusieurs se trouvent même à l'extérieur de la carte, chose que accc semble ne pas accepter pour la construction des ombres (tuxland-shade.ac) il râle et refuse de construire le fichier acc.

Mais vous avez un fichier tuxland_blender.acc (avec les ombres) qui a été construit avec l'option '-nts' que vous pouvez tout de même essayer, sinon le fichier acc (tuxland_ac3d.acc) est l'original généré par accc à partir du fichier ac (tuxland.ac) créé avec trackgen depuis le fichier XML (tuxland.xml), ce fichier a donc bien les ombres mais pas les objets ajoutés sous Blender. Le fichier ac d'origine est tuxland_ac3d.ac, le fichier modifié sous blender est tuxland_blender-clean.ac.

Vous avez donc :

_un fichier tuxland.ac avec les objets mais sans les ombres.

_un fichier tuxland.acc avec les ombres mais avec des erreurs de construction.

_un fichier tuxland.xml utilisant tuxland.ac et non pas tuxland.acc

Vous trouverez également un fichier tuxland_bender-export.ac qui le circuit avec les objets exporté au format ac3d depuis Blender.

Si vous désirez utiliser un fichier acc, n'oubliez pas de modifier le fichier XML en conséquence.

Personnellement j'utilise le format PNG pour les textures et autres images, mais pour ce circuit les images sont au format RGB (SGI) pour vous éviter, si vous êtes sous Linux, quelques désagréments lors de la création des cartes d'élévation.

Le format SGI (.rgb) utilisé pour les images est un rgb sans compression. Mais avec la version 1.3 de Torcs les images rgb compressées fonctionnent très bien.

Les objets et textures ont été réalisés pour ce tutoriel avec GIMP et BLENDER, excepté un pont (vous le reconnaitrez) qui est celui de Eric Espie emprunté pour ce tutoriel au circuit g-track-3.

3.3 Liens Internet :

Licence Art Libre : <http://artlibre.org/>

Licence GPL/FDL : <http://www.gnu.org/licenses/licenses.fr.html>

Site officiel de Torcs : <http://torcs.sourceforge.net/>

Le site Bernhard Wymann (Berniw) : <http://www.berniw.org/>

La page Torcs de Vincente Marti : <http://usuarios.lycos.es/f1torcs/>

Ce tutoriel **Torcs_track-Tut-FR.pdf** et le circuit **tuxland_tut-track.zip** sont disponibles ici :

<http://download.tuxfamily.org/debux/Torcs/>

vous y trouverez également un autre tutoriel en français, **Torcs_car-Tut-FR.pdf.zip**, sur la réalisation/conversion de voitures pour Torcs et les archives des fichiers sources de mes voitures (et robots) ainsi qu'une dizaine de circuits de test (offroad, dtm et f1) pour ces voitures,

Torcs_LnX-cars.zip et **Torcs_LnX-tracks.zip**

4 Note de l'auteur :

Problèmes rencontrés :

Sous Linux, en utilisant des images au format PNG pour les textures, lors de la création du terrain et des cartes d'élévation trackgen réclame des images SGI. La solution est de convertir les textures PNG en RGB, sans supprimer les images PNG ni modifier le fichier XML, trackgen utilisera alors les images au format RGB que vous pourrez supprimer après la création des cartes d'élévation.

Lors de la création du fichier acc (Shading) avec un fichier .ac exporté depuis Blender, acc se plaint de ne pas connaître le nom des maillages qu'utilise Blender, comme 'Mesh.0001' par exemple, et affiche des messages d'erreur avec ' Unknown verb Mesh.0001 '.

Vous pourrez utiliser le script 'cleanblendac.sh' pour nettoyer (supprimer les lignes Mesh.) le fichier ac et construire votre fichier acc.

Attention l'utilisation de ce script rend inutilisable le fichier ac par Torcs, pour pouvoir à nouveau utiliser un fichier ac, à la place du fichier acc, il vous faut le reconstruire à partir du fichier ac exporté par blender avec ce script cleanblendac.sh mais en ne supprimant pas les lignes contenant Mesh.

si vous avez une explication et/ou une solution 'propre', merci de me contacter.