

## Blender 2 Trackmania2

Welcome to my nightmare! 🤪

After trawling through many tutorials and forums trying to find out just how to get a car from Blender to TM2, I decided to get my problem solving head on and tackle it myself. I managed to get a fair amount of detail from the tutorial at '<http://hideou.se/trackmania/>', but even this didn't address everything and was based on 3DS Max, so some of the terminology was not clear, but it was the closest in terms of getting any results at all.

What follows is the result of a good weeks worth of bashing my head against a wall and trying all sorts of combinations to get what I wanted. By laying out this process I hope someone else is spared the frustration I felt. I will state up front, given that the process seems to change over time, this works as of Feb. 2016 with the current version of TM2 and will give you a car you can use in Valley, Canyon or Stadium, or at least, it should. 😊

I am only going over the basic version of a car here, no "damage" versions of geometry, no advanced settings of materials and such, the tutorial above helps some with that. Once you have the basics under your belt, hopefully, the process of adding more details should be easier, it may be a good idea to refer to the CanyonCar, ComboCV and Valleycar models & templates for advanced stuff.

I am also assuming you have some familiarity with Blender and are not completely new to how to use it. We will be doing basic marking of seams for unwrapping Uvs, separating geometry and setting of pivot points. I shall be doing all my texturing in Cycles, so when it comes to baking, if you are using the BI renderer, adjust the process accordingly.

Without further ado, let's dive right in!

I have provided several ZIP files on my Google Drive, one with the Blend files in, (with the model in various stages), another with the textures in GIMP and psd form and one with the resulting TM2 files I used to create a pack and the final repainted and working car model in TM2.

Blend Files: <https://drive.google.com/file/d/0B7AJLNQrsW-NeEZIMjltV1h2NWs/view?usp=sharing>

Textures: <https://drive.google.com/file/d/0B7AJLNQrsW-NZHpMckxRSH1bFk/view?usp=sharing>

Directory for TM2 used by NadeoImporter to build initial in-game ZIP file:

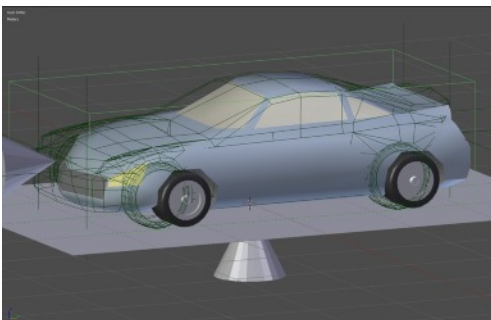
<https://drive.google.com/file/d/0B7AJLNQrsW-NOFpQNjRyTEZYdWM/view?usp=sharing>

Final repainted TM2 car which can be used in game:

<https://drive.google.com/file/d/0B7AJLNQrsW-NSGs0ekZ0TUtrVUU/view?usp=sharing>

### BLENDER PART.1.

Once you have unzipped the files somewhere, load up the blend file 'CarBSL\_tut00', this is a very basic car I threw together for testing. I have given it some basic materials to make it easier to separate later on.



Layer 1 has the camera and sun light needed for creating the "FakeShad" image. Layer2 holds the actual car model. Layer 20 (Alt+0) holds the CanyonCar collision model and bounding box, along with the FakeShad plane, nulls for the front and rear light locations and the light / shadow projectors needed by TM2.

The items on Layer20 come into play mainly later on, but when you create, or import a car model, use the box and collision models to make sure your car falls within the accepted boundaries for the game. (See Left)

## Sizing and UV Maps

You have to bear in mind that if your car is high and/or long, then even if it goes into the game okay, you will crash into walls when you appear to be nowhere near them, that is because all models use the CanyonCar bounding box to detect collision.

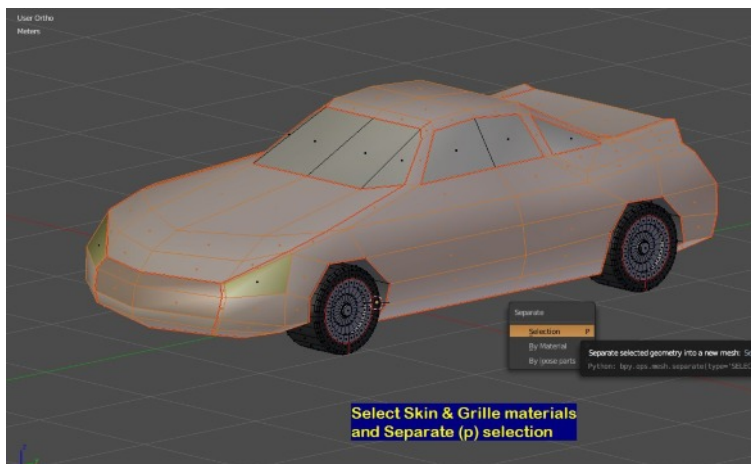
You can either take account of that whilst playing, or put some “bumpers” or outriggers on your car to more closely match the collision area.

Once you have sized and positioned your model how you want in relation to the bounding box, make sure the 3D cursor is at centre and then set your objects pivot point to be at the 3D cursor. This is key, the skin, details and glass parts of the car MUST be at 0,0,0. (The pivot points for the wheels will be at the centre of their rotation once done.)

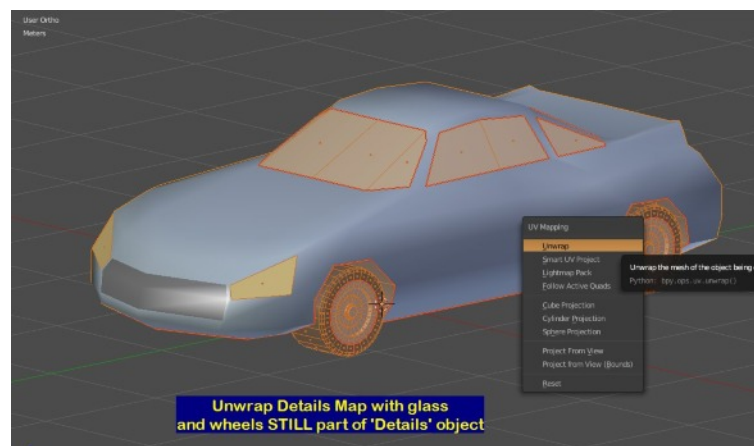
First things first, we need to create our UV Maps, (I stuck to one UVMap per object, with the wheels sharing the ‘details’ map along with the glass). I created an image in the UV/Image editor to use as a guide for unwrapping, ‘CarSL\_UV’, (1024x1024) create your own if you desire.

Select the car on Layer2, go into edit mode, make sure everything is deselected and you are in ‘face’ mode.

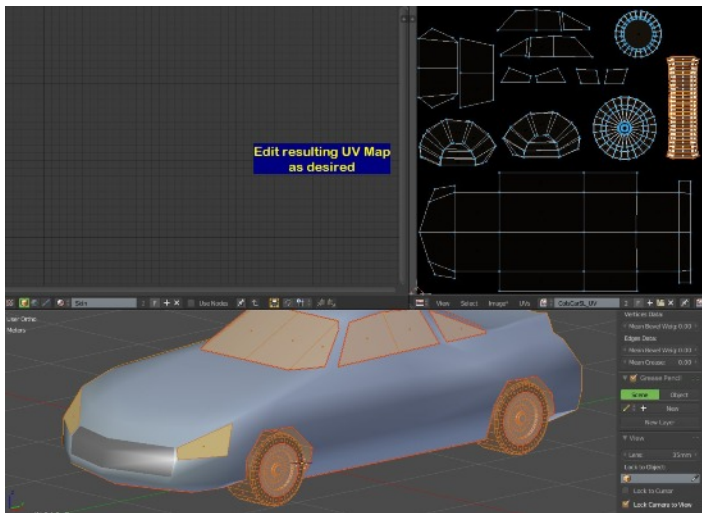
In order for the lights to work in the game, they have to be part of ‘details’ and NOT part of the skin, so we will separate the skin first. Navigate to the materials tab in the properties panel, highlight the ‘skin’ material and click the select button. Do the same for the ‘grille’ surface, then press ‘p’ to separate and do so by selection.



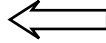
That leaves us with just the details to UV Unwrap. At this point you would normally need to decide what seams you wanted to create and do so, to save time, I have marked the seams on this model, so just select everything, then hit ‘u’ and unwrap.



Now just edit your UV map as you see fit, you may just want to pack the UV islands, or you may choose to overlay things such as wheels and tyres as I have done. (I use vertex snap in UVs to help with alignment).



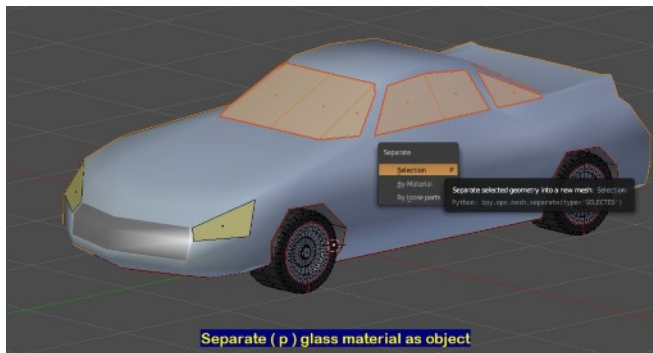
My UV Map



(As you can see, I mainly overlay the wheels, tyres & Wheel arches).

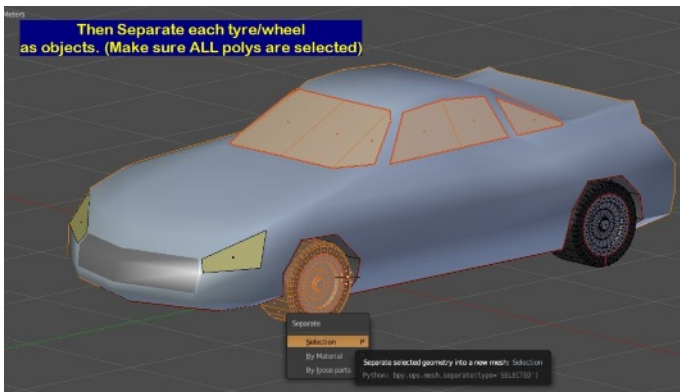
At this point, I exported my UV Map Layouts for use as a guide when building my textures. You need to do it at this point so the glass & wheels are taken account of within the 'Details' UVMaP.

When generating your UVs, allow for bake margins, so there is no overlap caused by the map being too "tight".



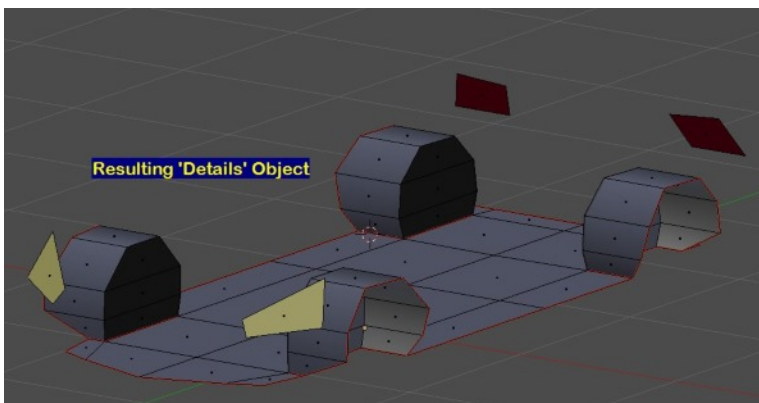
Once your UV map has been done, we can now separate the glass and wheels into individual objects whilst retaining the UVs. (There may be more efficient ways of doing this, but this worked so I stuck with it).

Make sure each wheel is a unique object as they need to rotate about their individual origins. (An easy method for the glass is to select a single poly, Shift+G and select by material).



When separating the wheels, do not just rely on 'L' or 'Ctrl+L' to select all faces, as there are UV seams which stop this. So either box select in wire mode, or use 'L' but then press 'CTRL Numpad+' to grow the selection until everything you need is highlighted.

Once again, use separate by selection. (Do not use select by material as that would grab all wheels and you need to do them one at a time).

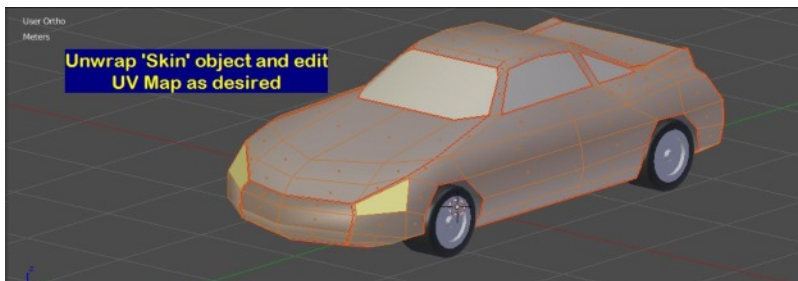


This is what your final 'details' object should look like if you go into local view mode using the '/' key on the numpad, (press again to toggle out of local view mode).

Having done this, we can exit edit mode.

Now select the 'skin' object, go back into edit mode for this. Create an Image for this, call it BSL\_SkinUV, so we have separate images, you \*could\* re-use the prior image and rename as you saved it, but there is too much of a chance of accidentally overwriting the wrong file.





Once you have created your new UV image, in the 3D view, select everything and 'u' to unwrap. Once that is done, you can edit the Uvs if you so desire. Export this UV Layout as well for use when texturing. *(Again, I have pre-marked the seams, but feel free to alter them to your own preference).*

## Object Naming

Having unwrapped everything, go back into Object mode and we can start to name objects so they have the correct names for in game use.



Name your skin object, (main body panels), to 'sBody', (note, the case IS important). Name your glass 'gBody', the details are 'dBody' and your wheels are named based on their position. FL - Front Left, RR - Rear Right. So looking from the top and going clockwise from front left, we have dFLWheel, dFRWheel, dRRWheel, dRLWheel.

At this point, it is a good idea to set the origin for the wheels to be at the centre of rotation. Setting "Origin to Geometry" with Shift+Ctrl+Alt+C should do it in Object mode. Also ensure that your gBody, sBody and dBody objects have their origin at 0,0,0 so set your cursor to centre and the origin to 3d cursor. If you get any of this wrong, you get very odd results in game, with wheels going off in weird rotations and parts not lining up.

(I use 3<sup>rd</sup> party pie menus which allow me to select geometry and 'set origin to selected', within edit mode, amongst other things, very useful).

<http://www.blenderartists.org/forum/showthread.php?348496-Addon-Wazou-s-Pie-Menus>

RMB Pie menus:

<http://www.blenderartists.org/forum/showthread.php?349758-Addon-Wazou-s-RMB-Pie-Menu>

If you wanted to, you could clear out 'redundant' materials from each object at this point, so the wheels only have rim & tyre, the skin only has skin & grille and so forth.

Now is a good point to save your work, when doing this, it pays to have several versions as you work along, just in case you need to back-track.

In order to get my colour maps for the details and skin, I set up some basic materials for baking onto the relevant UV Mappings. It is also handy for generating the initial 'Icon' render needed for the game, (though this will be overridden if you do any painting in the game itself). At this point, it is up to you to decide if you will do all your texture work outside of TM2 or if you intend to use the painter within the game.

Once you are happy with your materials, you can bake the colours. The glossiness of the surface is determined by the alpha channel of the diffuse map, so there is little point in doing that unless you intend to bake out the glossy map to use as an alpha channel.

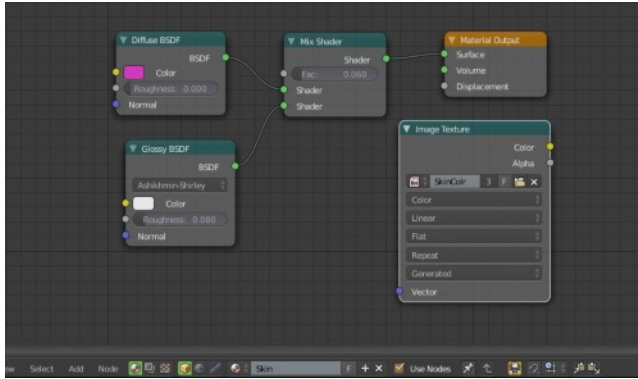
I again went with simple colours and made my changes in GIMP, as Photoshop is outside my budget. I then did some added work using the game painter and use this to generate an up to date icon, (more on that later).

At this point, your car should roughly match what is in the 'CarBSL\_tut1' blend file version, with obviously any differences in UV map layout and material colours.

## Baking UV Maps

To bake your textures in Cycles, you need to assign an image texture for the bake within each material, so firstly, within the UV/Image editor, create two images for your UV Maps and name them accordingly, (e.g. DetsColr, SkinColr), make them either 1024x1024 or 2048x2048 if you want more details).

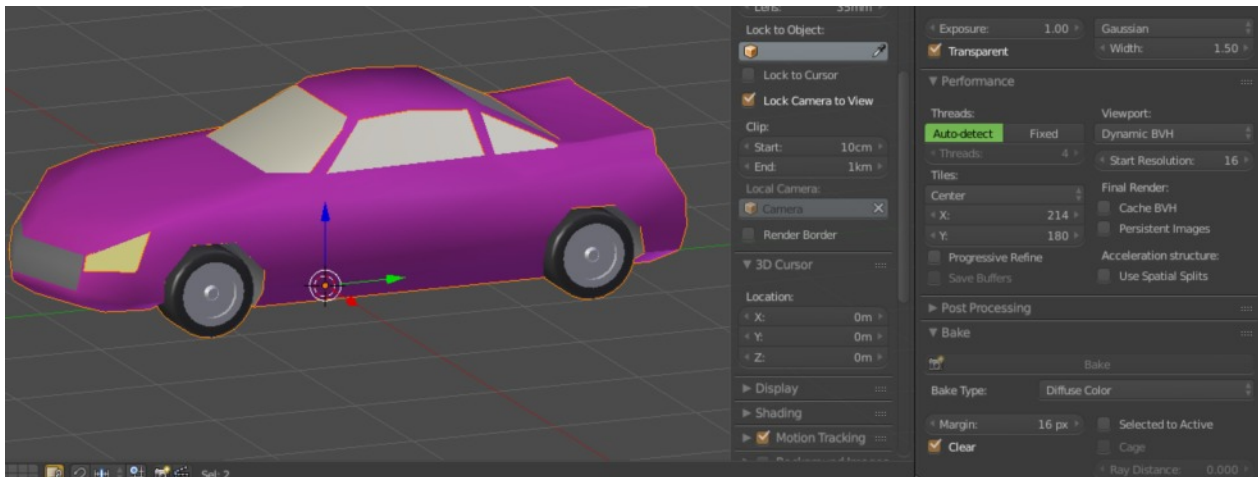
Within Each material, add a new node of "Image Texture", do NOT connect this to your material node tree, select the appropriate image for the surface. So 'glass' and 'tyre' etc would use DetsColr for example, whilst 'Skin' and 'Grille' would use 'SkinColr'



Make sure this image node is selected in the node editor, otherwise it will not be baked.

Repeat for each material, making sure the image is the only thing highlighted in each.

In the 3D View window, make sure all objects are selected, switch to the render tab of the properties panel, go down to bake, select 'Diffuse Color' and click bake. Once completed, save the two baked images using 'F3' from the UV/Image editor window and name them accordingly.

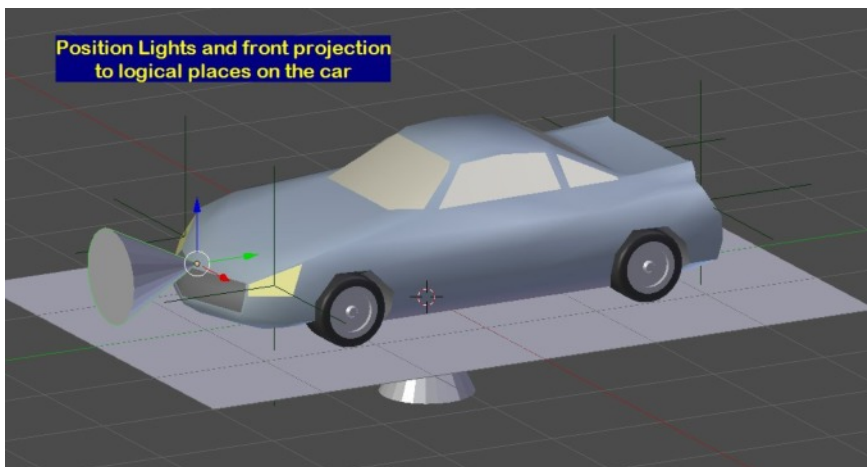


## Lights and FakeShad

Once you have baked your materials for mapping, the next step is to generate the 'FakeShad' texture and position the light markers on your car. To do this, we move things from Layer20 over to Layer2.

Go to layer 20, select and delete the collision car object and the bounding 'hitbox' object, then select everything remaining and press 'm', move it to layer 2.

Go back to layer 2, move the Emptys 'LightFL1' and 'LightFR1' to the front headlamps on the left and right side respectively, position them where you expect the light centre to be. Likewise move 'LightRL' and 'LightRR' to the rear lights.



**Note:** you can have an extra 2 sets of lights at the front, but NOT at the rear, (so LightFL2 & LightFL3 are valid).

Move the object 'LightFProj' to the front of the car, roughly central to the lights as this is used to cast the headlight beam, whereas 'LightF??' is used to light up the geometry.

Do not alter the origin for any of these objects, they stay as they are.

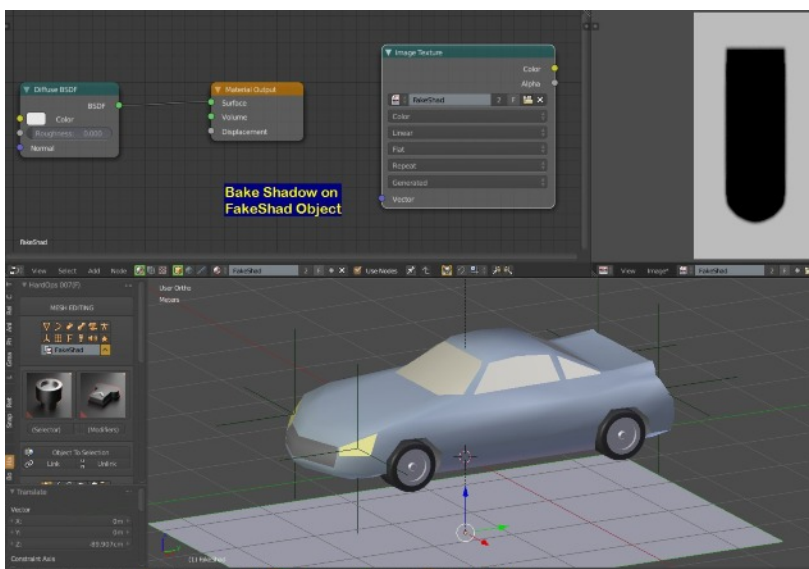
At this point, your model should be similar to the setup of the 'CarBSL\_Tut02' blend file.

( CarBSL\_tut03 is similar with textures set up to use baked colour maps, just so I could tweak them before exporting, in terms of model setup, only the materials are any different ).

The 'FakeShad' object already has a UVMap, now we need to create the fake shadow texture. If your object is close in size to the CanyonCar, you could probably get away with simply copying the CanyonCar FakeShad.dds into your build. To build one of your own, simple go into the UV/Image editor, create a new image of 512x1024 (width/height), call it FakeShad, if you go into edit mode and select the face of the FakeShad object, it should already be UV mapped correctly.

If not, in the 3D view, go into top view ortho, and unwrap with 'Project from View (bounds)', this should ensure the image is correctly mapped.

Back in object mode, ensure the FakeShad object is selected, we will bake the shadow texture to the new image. The object should already have a 'FakeShad' material defined for it. Go into the node editor, add an image texture node to one side, select the FakeShad image you just created, ensure the image texture node remains highlighted. (Just as per setup for baking the details & skin colours).



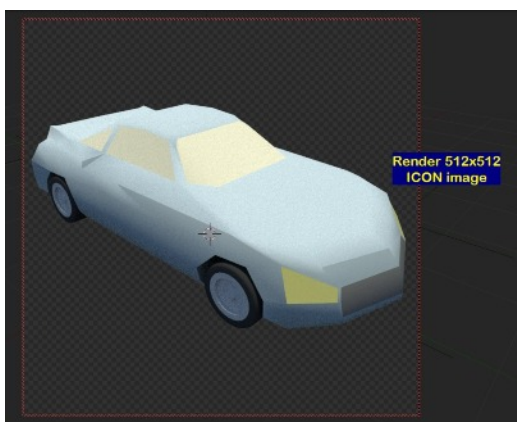
Make sure layer 1 is selected in the layers list to include the sun lamp in the bake.

Go to the render tab, change the bake mode to "Shadow" and hit Bake.

You may want to move the FakeShad object down a couple of units to reduce the darkness of the shadow, and/or play with the sun settings. (I did my tweaking in Gimp). If you do move the 'FakeShad' object, , make sure to remember to move it back to 0,0,0 after you have baked the texture.

Once the texture is baked, save the FakeShad image.

As the 'hideou.se' tutorials states, if the edge of this image is NOT pure white, (which it rarely is), you end up with a square shadow under your car, so I edit the image in Gimp, block fill the edges with pure white, tweak the colour levels and add a bit of blur to the result.



If you haven't already done so, you can now create an Icon image for use in game, make sure this is 512x512. Don't spend too much time on this, if you save in the game painter, it always creates an icon based on a 3/4 view so if you like, you could even get away with a very basic render such as on the left. (Remember to make sure the light/shad projection and fakeshadow objects are set to hidden from render).

I find the best way to work is to create my textures, then when I am happy, go into the painter in-game and click 'save' to update my car icon. Then I copy the icon.dds & SkinDiffuse.dds from the final zip file back to my source textures directory. (I also pull them back into my GIMP master file as a layer).

If you haven't already done so, it would be a good time to save your file, once more under a new iteration / filename. Okay, we are on the last stretch within Blender itself.

## Exporting Collada

We are now going to make one simple change to the scene so that when it exports, it is orientated correctly within the game. If you do not do this, when you go to paint your car, it will be pointing into the ground.

Now, in my tests, I found that prior to export, just a basic diffuse shader on each object was all that matters, any shader settings are, apparently, totally ignored, so no point in cluttering your final output with shader trees that serve no purpose. Though I did ensure that the wheels, glass and details object all shared the one material, with the 'Skin' having another. Again though, so long as there are separate UVMaps, I don't think the importer or game care about that, but as the example templates have separate materials, I thought I would "play safe".

Double check at this point that all your origins are correct and your car object is sat "on the ground".

Before I export, I go to Layer1 and delete the camera and sun light, so my export only has the data the game needs. Once done, return to layer2.

**Key Point** - Make sure your pivot center for rotation/scaling is set to 3D Cursor and your cursor is at 0,0,0. If you do not do this and end up rotating around a median point, you may find when you go into the game, your car, though not pointing downwards, is partly buried in the ground.

With your pivot point set as above, select everything, then rotate on the X axis by -90 degrees. So, assuming nothing is selected, press 'a' then 'rx-90', then press RETURN to confirm this, your car should now be pointing with its nose in the air. We now need to "lock" this, so CTRL+A and apply both rotation AND scale. *(Do NOT apply location, so likewise, do NOT select "transforms", otherwise this will throw your pivot point of your wheels out).*

This done, simply choose Export Collada and save your car under a suitable name, this will be copied to the respective filenames needed with the game importer. Righty, that is your work in Blender done and dusted.

**Note:** You can create damaged versions of your geometry which the game will use but I have avoided this as I did not wish to complicate matters. Simply put, a damaged model is a copy of say the 'sBody' object, named with a prefix of "\_", so "\_sBody", push & pull the vertices around to create your damage.

Be aware, do NOT add or remove geometry as the damaged model MUST have the same number of vertices as the undamaged one. Once you are familiar with the export of models from Blender into TM2, you can tackle this if you so wish, as yet though, I have not done so, therefore cannot pass on any help in that in terms of "gotchas" etc.

## Naming your model

Create a directory which will act as your model template for the Nadeo Importer, so for example 'CarBSL', copy your 'dae' file into this directory folder. Now, rename it to MainBody.dae, copy this file, paste it and name that copy as 'MainBodyHigh.dae, paste it again and name the third file as MainBodyVeryHigh.dae

This is the basis of your model, with a "proper" model, these obviously represent the varying levels of detail in the game. For this example, being a low poly model, I simply copy across to all 3 files.

If you do a car from scratch, the key thing to bear in mind is the detail level limits.

MainBody - This is the 'Low Poly' version and MUST be below 4,250 vertices

MainBodyHigh - This is the main model used and can be up to 20,000 vertices

MainBodyVeryHigh - This is your high resolution model and can be 60,000 vertices

Most of the time, I would imagine you will be copying your MainBodyHigh.dae to MainBodyVeryHigh.dae

To create your low poly, you would do it by decimation and removal of edge loops, the key thing is to ensure you do not upset your UVMaps as all models use the same UVMapped textures. It is either that, or create a specific low poly version then shoe-horn the map to fit the texture.

If you do create a specific low poly version, remember to save this in its own blend file. You can also delete any damage version as low poly models do not use this apparently.

## Setting up Textures

Textures for use in game MUST be in the DirectDraw 'dds' format, most are in DXT5 form with interpolated alpha and use mipmaps. How you create them depends upon what software you use for texture painting.

If you are using Photoshop, you can get the official DDS plugin from Nvidia. If you are using GIMP, there is a dds plugin available.

Paint.Net has built in dds support and also, you can convert most images to DXT using the DXTBmp stand alone program. *(I have provided links for these on page 13).*

Personally, I use GIMP with the DDS plugin as well as Serif Photoplus with DXTBmp depending on my file type and what it is I want to do.

The key thing to remember with the details & skin 'Diffuse' textures is, the alpha channel does NOT represent the transparent part of the image, but rather either controls the glossyness, or, in the case of DetailsIllum, the illumination level, of the texture.

Only in the damage / dirt textures does the alpha directly control where the dirt/damage is visible.

*(The Icon.dds does have a 'basic' transparency).*

SkinDiffuse.dds - DXT5 with Alpha & Mipmaps - Alpha indicates glossyness - black = matt, White = Shiny.

DetailsDiffuse.dds - As above

WheelsDiffuse.dds - As Above

DetailsIllum.dds - DXT5 with Alpha & MipMaps - Alpha indicates light function  
1 - Dashboard and night lights and general glow  
0.5 → 0.67 - Headlamps,  
0 → 0.05 - Brake Lights, (average intensity, stronger when braking).

SkinDirty.dds - DXT5 with Alpha and mipmaps - Alpha controls dirt opacity

DetailsDirty.dds - As Above

WheelsDirty.dds - As Above

SkinDamage.dds - DXT5 with Alpha and mipmaps - Alpha controls damage opacity

DetailsDamage.dds - As Above

FakeShad.dds - DXT5 with MipMaps, NO ALPHA

Icon.dds - DXT5 NO mipmaps, Interpolated Alpha, (alpha indicates transparency)

ProjWheelShad.dds - DXT1 No Alpha, (similar to FakeShad, but unless you don't want a wheel shadow, you can pretty much take this from any car download / template on ManiaPark).

**NOTE:** If you do not include the dirt / damage / illumination textures, the game will automatically use the ones from CanyonCar, which could produce very odd results.

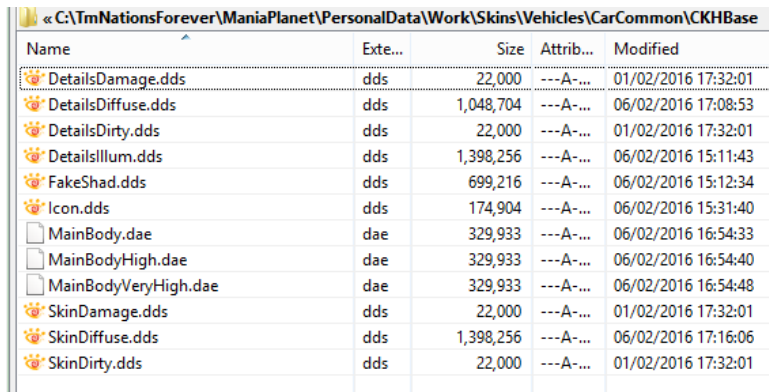
If you do not have dirt or damage textures, simply include a basic 128x128 texture, with a BLACK alpha channel saved as DXT5 with mipmaps & alpha for these to prevent any problems during import or game play.

I am not going to guide you through the process of using layer masks or editing alpha channels as, not only is that a full blown tutorial on it's own, it depends on what software you are using as to the procedure. If you look at the example files I have provided, you should get a rough idea of what I did for my textures. I am however, very much a "noob" when it comes to working with Gimp.



Once you have created all your 'dds' textures, making sure the names are correct, (*including being case sensitive*), move or copy these to the same folder as your 3 'dae' models.

You should end up with something like this:



Name	Ext...	Size	Attrib...	Modified
DetailsDamage.dds	dds	22,000	---A-...	01/02/2016 17:32:01
DetailsDiffuse.dds	dds	1,048,704	---A-...	06/02/2016 17:08:53
DetailsDirty.dds	dds	22,000	---A-...	01/02/2016 17:32:01
DetailsIllum.dds	dds	1,398,256	---A-...	06/02/2016 15:11:43
FakeShad.dds	dds	699,216	---A-...	06/02/2016 15:12:34
Icon.dds	dds	174,904	---A-...	06/02/2016 15:31:40
MainBody.dae	dae	329,933	---A-...	06/02/2016 16:54:33
MainBodyHigh.dae	dae	329,933	---A-...	06/02/2016 16:54:40
MainBodyVeryHigh.dae	dae	329,933	---A-...	06/02/2016 16:54:48
SkinDamage.dds	dds	22,000	---A-...	01/02/2016 17:32:01
SkinDiffuse.dds	dds	1,398,256	---A-...	06/02/2016 17:16:06
SkinDirty.dds	dds	22,000	---A-...	01/02/2016 17:32:01

## Using NadeoImporter

Once you have the NadeoImporter from the link on p.13 of this document, unpack it into the same folder as your ManiaPlanet executables, in my case, that is C:\TmNationsForever\Maniaplanet.

The folder for use by the importer must be off your personal data folder, by default, that will likely be in the 'users' path, such as 'C:\Users\Peter\Documents\ManiaPlanet'. In my case, I prefer everything to be off the game program folder, so it is 'C:\TmNationsForever\Maniaplanet\PersonalData'.

From this directory, make sure the directory 'Work\Skins\Vehicles\CarCommon' exists as well as a 'Skins\Models\CarCommon' directory. You also need to make sure a 'Temp' directory exists off the personal data folder.

(These MUST be so named, I tried using 'Work\Skins\Models\CarCommon' instead of then above and the importer simply wouldn't work).

Okay, now make a copy of the whole folder containing your dae files and textures, copy this folder into the 'Work\Skins\Vehicles\CarCommon' directory.

Go to the program directory, (e.g. 'C:\TmNationsForever\Maniaplanet'), and open a command window, you need to run the importer in command window, if you double click the exe in explorer, it simply opens and closes. You could, I suppose, create a batch file and give it the required arguments, then run it from the start→run menu option. (I have included such a batchfile, place it in the same directory as the programs and importer, all you need to do is pass your directory name as it's single argument when you run it).



**Batch File Link:** <https://drive.google.com/file/d/0B7AJLNQrsW-NRjRqRy1mMWQ2Unc/view?usp=sharing>

If you have the command window open and are not using the provided batch file, then type in a command as follows:

```
nadeoimporter CarSkin Skins/Vehicles/CarCommon/CarBSL /MaxSize=4096 /HqDds /LogMeshStats
```

Where 'CarBSL' is the name of your folder, which should be the name you want your resulting car package to have, (in the case below 'ColsCarSL').

```
C:\TmNationsForever\ManiaPlanet>nadeoimporter CarSkin Skins/Vehicles/CarCommon/ColsCarSL /MaxSize=4096 /HqDds /LogMeshStats
NadeoImporter 2014-10-22_12_37
Created :user:\Skins\Vehicles\CarCommon\ColsCarSL.zip
```

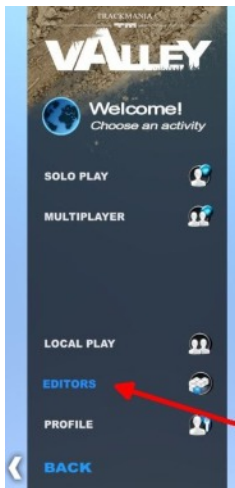
If all went well, as in the above image, you should get a message:

“Created :user:\Skins\Vehicles\CarCommon\CarBSL.zip”

Go to the game ‘Skins\Vehicles\CarCommon’ directory, (note, NOT the ‘work’ subdir, that houses ONLY the unpacked source files). Cut the ZIP file from here and paste it into the ‘Skins\Models\CarCommon’ directory, this is where the game expects any 3<sup>rd</sup> party models to be.

### **Finally, on to the Game import, Yay!**

Run ‘ManiaPlanet’, This first part is just to ensure that the model is behaving itself, at this point, you CANNOT select the model to race with in the game. However, before we commit to building a model pack and getting it ready, it is wisest to check that the model looks and behaves as expected so if need be, we can edit and re-export from Blender and re-import prior to finalising the model in-game.



Select your game, (or select from Valley / Canyon / Stadium if you have more than one). Go into Editors, then go to ‘Paint A Car’. You should be able to select your car here, do NOT go into the “Paint!” option, we just want to test the model out.

Your model should be visible here and you can paint it, you just can’t actually use it in game yet. Now, let’s check things are as they should be.

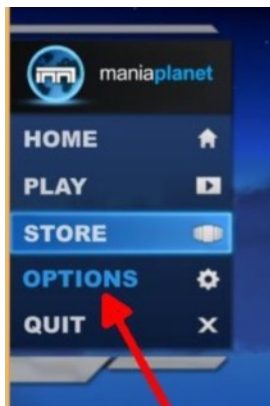
Press ‘h’ to bring up the helper keys menu. I turn off the turntable animation with Ctrl+A. Click and hold the RMB to rotate the car view so you can see the rear lights.

Press CTRL+B (toggle) to check the brakes. Use CTRL+L to check lights, (rotate so you can see the headlights and use CTRL+L to check them).

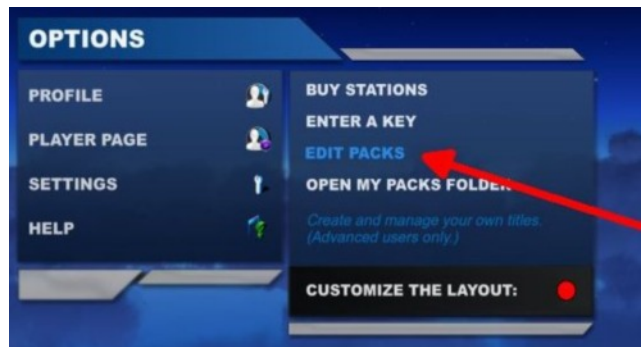
Toggle Ctrl+T to make sure your wheels are rotating correctly and are not offset or doing some odd behaviours. If everything is ok, we can move on. If not, go back and check your origins etc in Blender and re-export.

Once you are happy that wheels etc behave as they should, it is time to build the model pack.

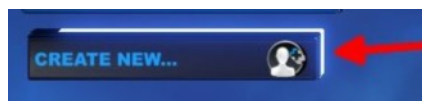
Back all the way out of the game until you are at the main menu.



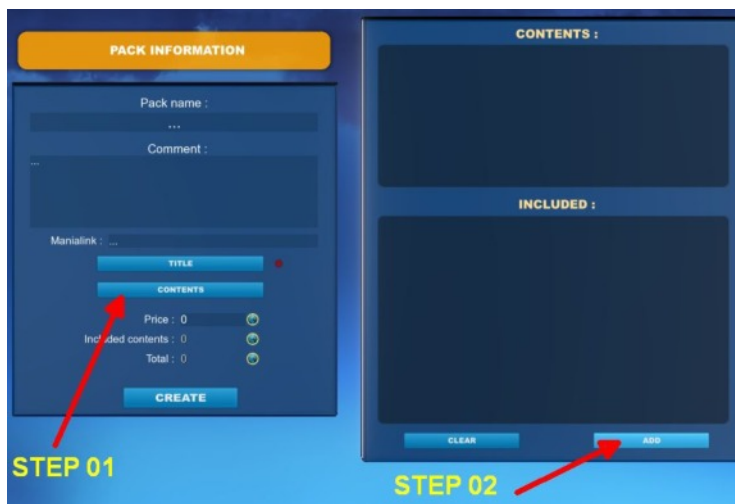
Select options.  
Then select ‘Edit Packs’  
from there. →



From the pack listings, choose ‘Create New’:



The main Pack menu will then load up.



Step 01: Select 'Contents', when this window pops up, choose 'Add' (Step02).

Navigate to the Skins\Models\CarCommon folder and select your new zip file, select 'Open'.

The pack name will automatically take the name of your Zip file.

Click 'Create' in the Pack Information panel and say 'ok' when prompted to "Create A Pack" and on the "pack successfully registered" pop up. (Say NO if asks to select skin for profile as we are not yet done).

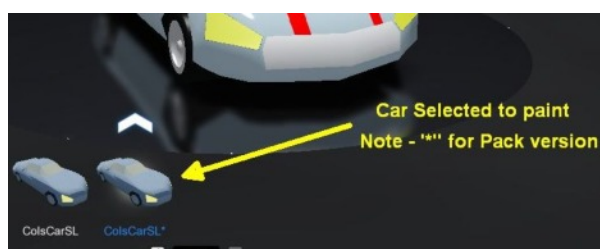
Now go back to the main menu, select a game title and enter to bring up the game options.

Again, at this point, you CANNOT yet load a car to race, we need a couple of more steps yet.

From the menu, as before, select 'Editors', then "Paint A Car".

Now, the crucial bit, it may seem counter intuitive, but this is the "hidden" secret.

Browse the cars, you will see TWO instances of your pack name, both will show the car, however, one will have an asterisk (\*) after it, (*usually further along the list*). THIS is the model you should choose to paint. It refers to the "pack" model and not your original car object.



Select this model, then select the 'Paint!' Option on the left hand side...



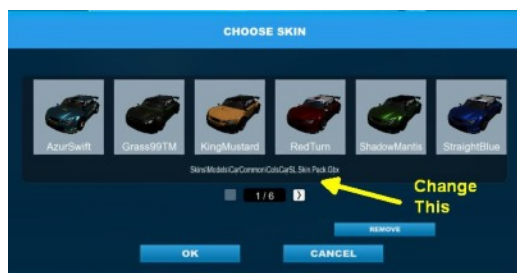
Scary, your skin map has vanished! you can see why I didn't choose this initially! However, this IS what we want.

Now select the 'Save As' Icon.



When the option pops up, it asks you to choose a skin, (*it will initially display the pack name*).

Click in the name box, it should read CarBSL.Skin.Pack.Gbx, change this to something like CarBSLSkin with no further text. Then press OK. (*I avoid underscores as I am not sure they are 'valid'*).



Once that is done, if you look in Windows Explorer, you should see your original zip file, a file named 'CarBSL.Skin.Pack.Gbx' and a file named 'CarBSL\_Skin.zip'. (In the Skins\Models\CarCommon folder).

We need to copy our texture from our original zip file to the new 'CarBSL\_Skin.zip' file.

Before we do that however, let's just make sure we can indeed see the car for selecting in-game.

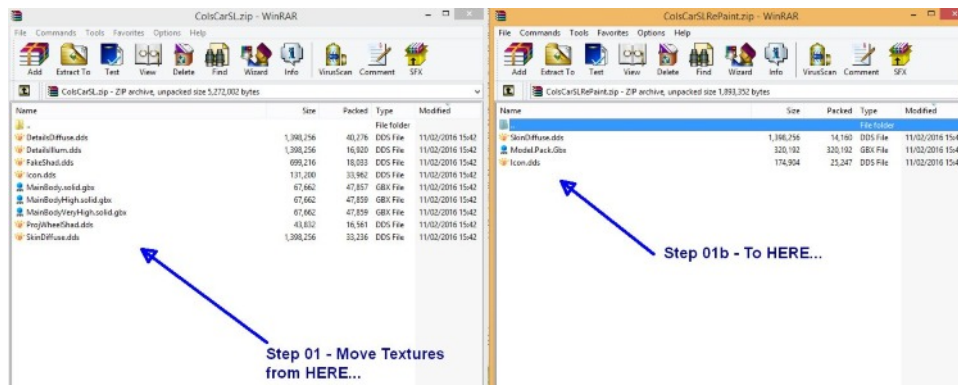
If you come out of the paint editor, go into “Profile” → “Models”, you should be able to select this car to race with, it’s skin texture is a flat colour, but it shows the model should be able to be selected. *(You may have to go to the end of your models list to see it for selecting).*

Back out to the main game menu, so you are not in “Profile” or “Paint a car”.

In windows explorer, open up both the new model Zip file and your original, *(or alternatively open up the New zip file but make sure you can see your original source folder for your textures).*

Right, either copy your files from your source folder into the new ‘CarBSLSkin.zip’ file, replacing the SkinDiffuse.dds, or pull it from your original zip file.

Copy **ONLY** the skin diffuse texture across from your original zip file / folder into the new zip file. If you copy anything more than this, you can no longer select the car from your Profile→Models option. *(I presume this is something to do with the FakeShad & Details textures now being packed into the “Model.Pack.Gbx” file so having them in the final Zip file ‘confuses’ things. This tripped me up for a while as a car that was once selectable suddenly vanished from my model list).*



Go back into “paint a car”, select the new skin name, go into “paint!”, You should now see your car with the correct paint job. Use the left-hand “Save” icon to update your model zip file Icon image. Exit painter, go into “Profile” → “Models” and you should be able to select your model with its correct icon. If you select it, you can now go racing with this model.

With that done, you can either move the original zip file and pack file elsewhere for backup, or delete them. If you look at the above image, the key point here is the new zip file has a ‘Model.pack.gbx’ file whereas the original has the 3 files are per our Collada exports. It is this which is key to making a model “visible” to choose in game. Saving the ‘pack’ model, (the one with the asterisk suffix), as a new name, generates this file and only ever provides the icon and SkinDiffuse textures.

*(Side Note: As I initially tried to follow the ‘hideou.se’ tutorial, which only had details and no ‘skin’, I could not reach this stage so went round in circles for a bit!)*

At this point, you can actually rename your zip file within windows explorer and the game reflects this, there is nothing within the model pack file which stores its original name as far as I am aware. You have to exit the game and reload it otherwise it if you do this though to ensure the new name is “read” as the game loads.

Should you modify and re-export your model from Blender, you need to go through the whole import procedure again, but this time updating your pack with the Options→Edit Packs and then Contents→Add, it will then prompt your to “Update” rather than “Create”. Once this is done, you need to go through the steps outlines above to generate a renewed ‘Model.Pack’Gbx’ object. *(This car has no interior so seems “see through to the game, a good exercise to try would be to add a basic interior into the “dBody” object).*

Phew!

Sounds a lot doesn’t it, but truth is, once you’ve done it once or twice, the thing that takes the longest is painting your textures and setting them up. If you simply output a flat colour from Blender and paint in the game, I bet you’re only looking at a couple of minutes to get from Blender to a fully working / paintable car in game.



## **Final Notes**

As I have stated previously, this is a super-basic version of getting from Blender to TM2, you can add damage, add dirt maps, add other geometry and maps to make the wheels more detailed and we didn't even touch on suspension. You can separate the skin & details so bits fly off them, (you'll need to be aware of pivot points if you do this). Likewise the glass object can be broken down.

Simply put, the game looks for a prefix, 'd' for detail, 's' for skin, 'g' for glass, it then behaves accordingly.

As a test of your own, homework if you will, try separating the glass object 'gBody' into Front & rear windshields and side glass. 'gFWShield', 'gRWShield', 'gBody'. Export this as a new model and see if it behaves any differently in the game when you crash.

## **RESOURCE LINKS - All links valid as of Feb. 2016**

ManiaPark Car Downloads, (get both download AND Template)

CanyonCar: <http://www.maniapark.com/ressource.php?id=1>

ComboCV: <http://www.maniapark.com/ressource.php?id=2235>

Gimp: <https://www.gimp.org/>

Gimp dds Plugin: <http://registry.gimp.org/node/70>

DXTBmp: <http://www.mwgfx.co.uk/programs/dxtbmp.htm>

Paint.Net: <http://www.getpaint.net/index.html>

Nvidia Photoshop

DDS Plugin: <https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>

NadeoImporter: <http://doc.maniaplanet.com/creation/importer>

A little Xtra: <https://drive.google.com/file/d/0B7AJLNQrsW-NbDBYRkxaNVVKb2c/view?usp=sharing>

(Try exporting this to TM2, it CAN be done, and it's fun to race with!)

## **Authors Note:**

Hopefully this tutorial has helped you get your own car into TM2, if it has, job done! If you feel there are areas which are not quite clear, or there is a step you think is missing, please let me know and I shall look into them. I have not covered setting up your model to make it available online for other players as, naturally, so far, I wasn't even able to get a model into the game so never dealt with that side of things.

It probably took me many times longer to put this together than it does to actually go through the process, but it was worth it. Please bear in mind the zip files I have provided are subject to Google Drive imposing any daily bandwidth limits, so if you get a message saying it cannot be downloaded at present, give it a day and try again. Though I doubt there will be a mad rush by thousands for this info. Hehe...

*If I can find the time and any interest is shown, I may create a video tutorial of this on YouTube, going from taking a 3<sup>rd</sup> party car from say Blendswap through to getting it running in TM2. I can't promise anything, but it is tempting. :-)*

If you would like to show your gratitude, you may do so by making a donation, however small, to the Gentleshaw Wildlife Centre where I volunteer, ( <http://www.gentleshawwildlife.co.uk/> ), there is a paypal button on the front page, but do not feel obligated to do so.

Happy Blending ...and Racing!



P.S:

I have added a few pages here which I found useful in my working out. I took the DAE models of the ComboCV car and the CanyonCarHigh and produced a list of parts / materials / uvmaps within Blender. You may find them useful, or not.



=====

ComboDAE\_ListObjDets.txt (**ComboCV\_DAE**)

04/02/2016

G:\Blender\_Resources\Colkai\_Blends>ListObjDets.txt

OBJECT (Mesh) - UVLayers - LayerName (TextName)

-----

dbody	- 1 UVMaps: sBody_005-mesh-map-0 [MATERIALSSkinDetails]
dFLGuard	-UVMap:dFLGuard-mesh-map-0[MATERIALS: Details]
dFLWheel	-UVMap:dFLWheel-mesh-map-0[MATERIALS: Details]
dFRGuard	-UVMap:dFRGuard-mesh-map-0[MATERIALS: Details]
dFRWheel	-UVMap:dFRWheel-mesh-map-0[MATERIALS: Details]
dHood	-UVMap:dHood-mesh-map-0[MATERIALS: Details]
dLDoor	-UVMap:dLDoor-mesh-map-0[MATERIALS: Details]
dRDoor	-UVMap:dRDoor-mesh-map-0[MATERIALS: Details]
dRLGuard	-UVMap:dRLGuard-mesh-map-0[MATERIALS: Details]
dRLWheel	-UVMap:dRLWheel-mesh-map-0[MATERIALS: Details]
dRRGuard	-UVMap:dRRGuard-mesh-map-0[MATERIALS: Details]
dRRWheel	-UVMap:dRRWheel-mesh-map-0[MATERIALS: Details]
dTrunk	-UVMap:dTrunk-mesh-map-0[MATERIALS:SkinDetails]
FakeShad	-UVMap:FakeShad-mesh-map-0[MATERIALS:FakeShad]
gBody	-UVMap:gBody-mesh-map-0[MATERIALS: Glass]
gFWShield	-UVMap:gFWShield-mesh-map-0[MATERIALS: Glass]
gLDoor	-UVMap:gLDoor-mesh-map-0[MATERIALS: Glass]
gRDoor	-UVMap:gRDoor-mesh-map-0[MATERIALS: Glass]
gTrunk	- 1 UVMaps: sBody_004-mesh-map-0 [MATERIALS: Glass]
LightFProj	-NoUVMap[MATERIALS: Details]
sBody	-UVMap:sBody-mesh-map-0[MATERIALS: Skin]
sFLWheel	-UVMap:sFLWheel-mesh-map-0[MATERIALS: Skin]
sFRWheel	-UVMap:sFRWheel-mesh-map-0[MATERIALS: Skin]
sHood	- 1 UVMaps: sBody_007-mesh-map-0 [MATERIALSSkin]
Sizer_Cube	- 0 UVMaps: NONE
sLDoor	- 1 UVMaps: sBody_008-mesh-map-0 [MATERIALS: Skin]
sRDoor	- 1 UVMaps: sBody_009-mesh-map-0 [MATERIALS: Skin]
sRLWheel	-UVMap:sRLWheel-mesh-map-0[MATERIALS: Skin]
sRRWheel	-UVMap:sRRWheel-mesh-map-0[MATERIALS: Skin]
sTrunk	- 1 UVMaps: sBody_002-mesh-map-0 [MATERIALS: Skin]
wFLWheel	-UVMap:wFLWheel-mesh-map-0[MATERIALS: Wheels]
wFRWheel	-UVMap:wFRWheel-mesh-map-0[MATERIALS: Wheels]
wRLWheel	-UVMap:wRLWheel-mesh-map-0[MATERIALS: Wheels]
wRRWheel	-UVMap:wRRWheel-mesh-map-0[MATERIALS: Wheels]

## USED MATERIALS

-----  
Skin  
Details  
FakeShad  
Glass  
Wheels

=====

G:\Blender\_Resources\TrackMania2>ListObjDets.txt (*CanyonCarDets*)

### OBJECT (Mesh) - UVLayers - LayerName (TextName)

_dBody	- UVMa: Geometry-_dBody-CanyonCardetails [MATERIALS: CanyonCarDetails]
_dHood	- UVMa: Geometry-_dHood-CanyonCardetails [MATERIALS: CanyonCarDetails]
_dLDoor	- UVMa: Geometry-_dLDoor-CanyonCardetails [MATERIALS: CanyonCarDetails]
_dRDoor	- UVMa: Geometry-_dRDoor-CanyonCardetails [MATERIALS: CanyonCarDetails]
_dTrunk	- UVMa: Geometry-_dTrunk-CanyonCardetails [MATERIALS: CanyonCarDetails]
_gBody	- UVMa: Geometry-_gBody-CanyonCardetails [MATERIALS: CanyonCarGlass]
_gFWShield	- UVMa: Geometry-_gFWShield-CanyonCardetails [MATERIALS: CanyonCarGlass]
_gLDoor	- UVMa: Geometry-_gLDoor-CanyonCardetails [MATERIALS: CanyonCarGlass]
_gRDoor	- UVMa: Geometry-_gRDoor-CanyonCardetails [MATERIALS: CanyonCarGlass]
_gTrunk	- UVMa: Geometry-_gTrunk-CanyonCardetails [MATERIALS: CanyonCarGlass]
_sBody	- UVMa: Geometry-_sBody-CanyonCarSkin [MATERIALS: CanyonCarSkin]
_sHood	- UVMa: Geometry-_sHood-CanyonCarSkin [MATERIALS: CanyonCarSkin]
_sLDoor	- UVMa: Geometry-_sLDoor-CanyonCarSkin [MATERIALS: CanyonCarSkin]
_sRDoor	- UVMa: Geometry-_sRDoor-CanyonCarSkin [MATERIALS: CanyonCarSkin]
_sTrunk	- UVMa: Geometry-_sTrunk-CanyonCarSkin [MATERIALS: CanyonCarSkin]
dBody	- UVMa: Geometry-dBody-CanyonCardetails [MATERIALS: CanyonCarDetails]
dFLGuard	- UVMa: Geometry-dFLGuard-CanyonCardetails [MATERIALS: CanyonCarDetails]
dFLWheel	- 2 UVMa: (0) - Geometry-dFLWheel-CanyonCarSkin (1) - Geometry-dFLWheel-CanyonCardetails [MATERIALS: CanyonCarDetails]
dFRGuard	- UVMa: Geometry-dFRGuard-CanyonCardetails [MATERIALS: CanyonCarDetails]
dFRWheel	- 2 UVMa: (0) - Geometry-dFRWheel-CanyonCarSkin (1) - Geometry-dFRWheel-CanyonCardetails [MATERIALS: CanyonCarDetails]
dHood	- UVMa: Geometry-dHood-CanyonCardetails [MATERIALS: CanyonCarDetails]
dLDoor	- UVMa: Geometry-dLDoor-CanyonCardetails [MATERIALS: CanyonCarDetails]
dRDoor	- UVMa: Geometry-dRDoor-CanyonCardetails [MATERIALS: CanyonCarDetails]
dRLGuard	- UVMa: Geometry-dRLGuard-CanyonCardetails [MATERIALS: CanyonCarDetails]
dRLWheel	- 2 UVMa: (0) - Geometry-dRLWheel-CanyonCarSkin (1) - Geometry-dRLWheel-CanyonCardetails [MATERIALS: CanyonCarDetails]
dRRGuard	- UVMa: Geometry-dRRGuard-CanyonCardetails [MATERIALS: CanyonCarDetails]
dRRWheel	- 2 UVMa: (0) - Geometry-dRRWheel-CanyonCarSkin (1) - Geometry-dRRWheel-CanyonCardetails [MATERIALS: CanyonCarDetails]
dTrunk	- UVMa: Geometry-dTrunk-CanyonCardetails [MATERIALS: CanyonCarDetails]
FakeShad	- UVMa: Geometry-FakeShad-FakeShad [MATERIALS: FakeShadMat]
FLLight	- NoUVMa [MATERIALS: FLLightMat]
FRLight	- NoUVMa [MATERIALS: FRLightMat]

gBody	- UVMap: Geometry-gBody-CanyonCardetails [MATERIALS: CanyonCarGlass]
gFWShield	- UVMap: Geometry-gFWShield-CanyonCardetails [MATERIALS: CanyonCarGlass]
gLDoor	- UVMap: Geometry-gLDoor-CanyonCardetails [MATERIALS: CanyonCarGlass]
gRDoor	- UVMap: Geometry-gRDoor-CanyonCardetails [MATERIALS: CanyonCarGlass]
gTrunk	- UVMap: Geometry-gTrunk-CanyonCardetails [MATERIALS: CanyonCarGlass]
LightFProj	- NoUVMap [MATERIALS: CanyonCarDetails]
RLLight	- NoUVMap [MATERIALS: RLLightMat]
RRLight	- NoUVMap [MATERIALS: RRLightMat]
sBody	- UVMap: Geometry-sBody-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sFLWheel	- UVMap: Geometry-sFLWheel-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sFRWheel	- UVMap: Geometry-sFRWheel-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sHood	- UVMap: Geometry-sHood-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sLDoor	- UVMap: Geometry-sLDoor-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sRDoor	- UVMap: Geometry-sRDoor-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sRLWheel	- UVMap: Geometry-sRLWheel-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sRRWheel	- UVMap: Geometry-sRRWheel-CanyonCarSkin [MATERIALS: CanyonCarSkin]
sTrunk	- UVMap: Geometry-sTrunk-CanyonCarSkin [MATERIALS: CanyonCarSkin]
wFLWheel	- UVMap: Geometry-wFLWheel-CanyonCarWheels [MATERIALS: CanyonCarWheels]
wFRWheel	- UVMap: Geometry-wFRWheel-CanyonCarWheels [MATERIALS: CanyonCarWheels]
wRLWheel	- UVMap: Geometry-wRLWheel-CanyonCarWheels [MATERIALS: CanyonCarWheels]
wRRWheel	- UVMap: Geometry-wRRWheel-CanyonCarWheels [MATERIALS: CanyonCarWheels]

## USED MATERIALS

-----  
 CanyonCarDetails  
 CanyonCarGlass  
 CanyonCarSkin  
 FakeShadMat  
 FLLightMat  
 FRLightMat  
 RLLightMat  
 RRLightMat  
 CanyonCarWheels