RCT-3.ORG

THE THRILLING WEBSITE

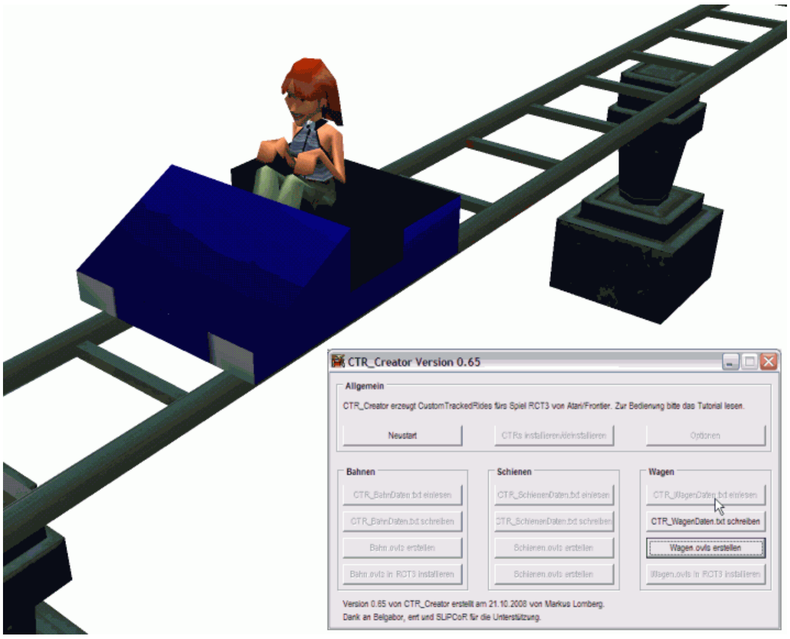
CTR_Creator use for cars (Beginner)

Tutorial for RCT3

Preliminary note

This tutorial explains the basic creation of custom cars for the game Rollercoaster Tycoon 3 (RCT3). To do this, you'll need to use the tool CTR_Creator, which you can create Custom Tracked Rides (CTR) with. This tutorial explains how beginners can create a first, simple, self-made CTR car with a little time and effort. In another tutorial for advanced users, the detailed explanations and information needed for more complex CTR cars, e.g. with animations and several models, will be explained. For pros finally there will be another tutorial that covers the use of CTR_Creator with all functions for complex cars.

This tutorial is on version 0.65 of CTR_Creator. With this version, custom coasters and cars can be made; custom tracktypes will be available in later versions.



CTR_Creator Version 0.65

Allgemein

CTR_Creator erzeugt CustomTrackedRides fürs Spiel RCT3 von Atari/Frontier. Zur Bedienung bitte das Tutorial lesen.

Version 0.65 von CTR_Creator erstellt am 21.10.2008 von Markus Lomberg.
Dank an Belgabor, ert und SLiPCoR für die Unterstützung.

Neustart
CTRs installieren/deinstallieren
Optionen
Bahnen
CTR_BahnDaten.txt einlesen
CTR_BahnDaten.txt schreiben
Bahn.ovls erstellen
Bahn.ovls in RCT3 installieren
Schienen
CTR_SchienenDaten.txt einlesen
CTR_SchienenDaten.txt schreiben
Schienen.ovls erstellen
Schienen.ovls erstellen
Wagen
CTR_WagenDaten.txt einlesen
CTR_WagenDaten.txt schreiben
Wagen.ovls erstellen
Wagen.ovls in RCT3 installieren

Introduction

It is essential to read the whole tutorial carefully and follow it exactly. Simply skimming is not enough, problems will occur then for sure. Only by following this tutorial step by step will you be rewarded with a working CTR car in the end!

In the following it's assumed that you already know CTR_Creator and the creation of CTR rides. This knowledge is the basic requirement for the creation of CTR cars explained in this tutorial. So if you don't know the tool and how to make CTR rides, please don't read any further, but go through the tutorial "CTR_Creator use for rides (Beginner)" first and create some CTR rides as an exercise. The information in there remains valid without changes.

Also you need some knowledge and talent for 3D object modeling. If you created CSOs before and used the CSO Importer, this knowledge will help you immensely – otherwise you will have to learn more (not part of this tutorial; see below for a list of links).

For CTR cars you need the latest version of the tool CTR_Creator you can download in the download area of www.RCT-3.org. The download contains some helpful supplementary files that make the use of CTR_creator easier; e.g. empty templates, example files and all data for an example car. All files contained it the download will overwrite the ones in your CTR work folder. We will return to them later.

To make the creation of your first CTR car easy, this tutorial is limited to a very simple car. I.e. the car has no animation or other extras and there will be only one seat for a peep.

So that you know what steps are needed to create a simple CTR car, I listed them for you:

- First your car has to be created as textured **3D data.** You'll need a 3D modeling program for this. **Blender or SketchUp** will work.
- Next this 3D data will be converted into the RCT3 model format with the **CSO Importer** and the model saved as an **OVL file**.
- These **model OVLs** and the necessary additional data will be processed in **CTR_Creator** and put out as complete **CTR car**.

This sounds laborious and difficult, and the first few times it will be, but with practice you will become experienced. In the following we'll do these steps slow and thorough.

Note: As said in the list above, I often use the term "**3D data**" when talking about CTRs (in my tutorials and also in my posts in the forums) for the car created in the 3D modeling program. The terms "**model**" and "**model OVLs**" always indicate the car created in the CSO Importer. You should memorize this difference as it helps to avoid misunderstandings.

Creating the 3D data for the car

As said above, a CTR car doesn't start in CTR_Creator, but in a **3D modeling program, like Blender or SketchUp**. For simple CTR cars as in this tutorial it doesn't matter whether you use Blender or SketchUp. For more complex CTR cars later, I recommend using Blender. Only Blender has all the functions needed for the creation of complex CTR (and CFR, too). Also there is a new export script (for .modxml format, thanks to Belgabor) which makes the work with the CSO Importer much easier. For this reason I will explain the way to create a car using Blender in this tutorial and only refer SketchUp occasionally.

The use of Blender or SketchUp for the creation of the 3D data of your car isn't part of this tutorial. I'll only talk about the rough procedure and some advice; details of the usage won't be explained here. In principle everything works like creating CSO and there are several good and detailed tutorials on this. Here a selection for Blender:

- Blender Book by Eddi22 (German)
  http://cso.rct-3.org/eddi22/005TutorialsMenu/BlenderBuchMenu/Blender1_7.html

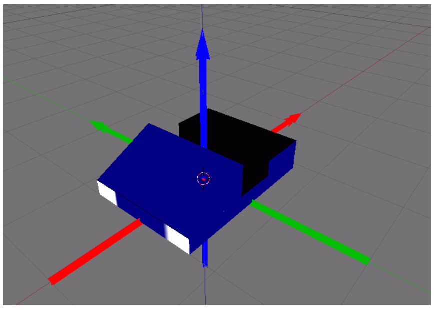
- Creating Custom Scenery Objects with Blender by YesMyDear
  http://csomodelling.blogspot.com/

- Flatride Tutorial by Belgabor
  http://sourceforge.net/project/showfiles.php?group_id=181524

- Blender 3D: Noob to Pro on Wikibooks
  http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro
  -- very detailed, so possible long download time

The first two tutorials for Blender are geared towards use with RCT3 and recommended for beginners. The Flatride Tutorial by Belgabor includes the useful knowledge about EffectBones and animation (whereas animation of CTR cars will be explained in the advanced tutorial only). The last tutorial covers Blender generally and explains the general usage (Blender 3D: Noob to Pro – Beginner Tutorial) and things that might be interesting for special 3D data.

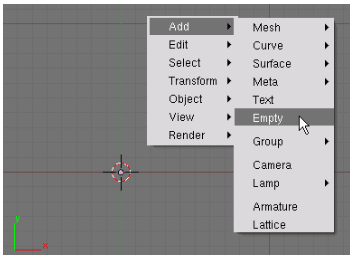
I recommend you to create a very simple geometry (e.g. a cuboid with an edge length of 0.5 Blender units) instead of a complex car with many details for your first try of a CTR car. Later more time and labor can be invested in the second or third try.



The first thing to pay attention to when designing the car in Blender is the **orientation in the coordinate system**. The direction of the car has to be along the negative X-axis (red) and the height along the positive Z-axis (blue). The adjacent image visualizes this for better understanding. If the car is not positioned correctly in the coordinate system of Blender, it will run backwards, sidewise or headlong in RCT3. [SketchUp uses an identical coordinate system (right-handed, Z-up), so the same orientation has to be used when designing the car.]

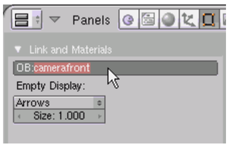
After you created the textured car correctly in the coordinate system, the work in the 3D modeling program is almost done. Only the so called "**EffectBones**" have to be added. But what are these EffectBones? Different than CSO, CTR cars have to interact with the environment in game, e.g. peeps need to board the cars, the cars should run correctly on the rails and it's nice to have cameras that show the ride. For this interaction, additional information CSO don't need is necessary. This additional information is defined through so called EffectBones. (If you ever created CS with lights, particle effects or fountains, you already know EffectPoints which are similar to EffectBones.)

In this tutorial for beginners, we'll only cover the seven basic EffectBones that **must** be specified for a working CTR car. In most cases you can use standard positions for EffectBones for simple cars, making their creation a cakewalk (that of the bones, not of the whole car ;-)).

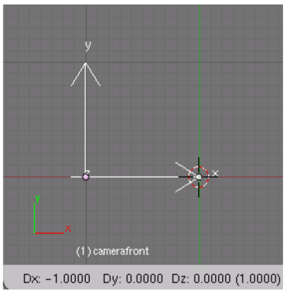


To create an EffectBone, switch to **top view (view from above on the X-Y-plain)**. You can do this by pressing "Num7". Now open the menu pressing "Space" and choose "**Add >> Empty**".

In Button window (mostly in the lower left) rename the “Empty“ to "camerafront".



OB:camerafront
Empty Display:
Arrows
Size: 1.000



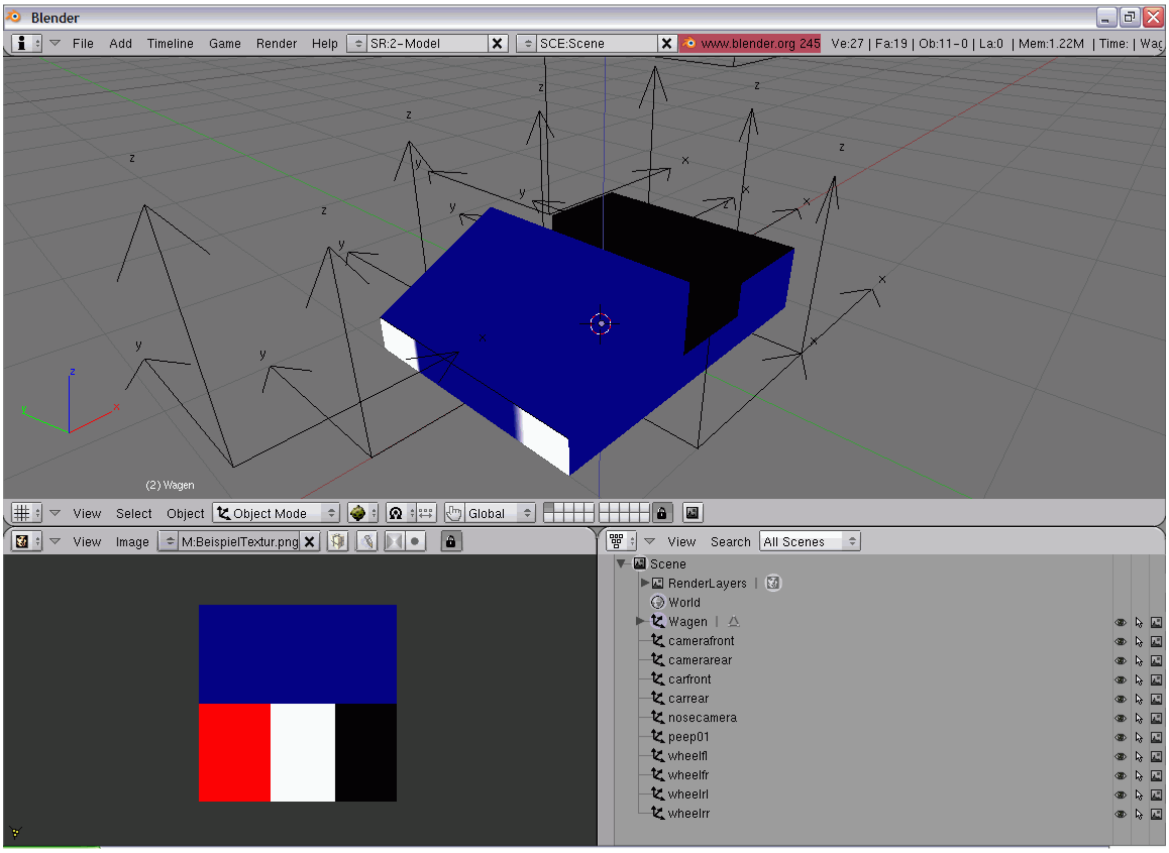
(1) camerafront
Dx: -1.0000 Dy: 0.0000 Dz: 0.0000 (1.0000)

Use key "G" (for grab) to move the EffectBone **to the desired position** – in this case for “camerafront“ about position x = –1, y = 0, z = 0. Now the first EffectBone is done.

[In SketchUp use the text tool to name and position the EffectBone.]

Create the remaining EffectBones that are listed in the following table in the same way.

EffectBone name	**description of the values**
camerafront	Coordinates for the front camera of the car. Beginners can use the standard position x = –1, y = 0, z = 0 for their first car.
camerarear	Coordinates for the rear camera of the car. Beginners can use the standard position x = +1, y = 0, z = 0 for their first car.
carfront	Defines where the front of the CTR car is. (The carfront EffectBone is necessary and sufficient to define the front-to-rear extent of the car. The actual 3D-geometry is not taken into account.) Beginners can use the standard position x = –2, y = 0, z = 0 or the foremost point of their car.
carrear	Defines where the rear of the CTR car is. (The carrear EffectBone is necessary and sufficient to define the front-to-rear extent of the car. The actual 3D-geometry is not taken into account.) Beginners can use the standard position x = +2, y = 0, z = 0 or the rearmost point of their car.
peep01	Position of the peep in the car. Beginners can use the standard position x = 0, y = 0, z = 0 or the position of the peep.
wheelfl	Coordinates of the front left wheel. (Note: It's not necessary to have the EffectBone matching the actual wheel exactly.) Beginners can use the standard position x = 0, y = –1, z = 0 for their first car.
wheelfr	Coordinates of the front right wheel. (Note: It's not necessary to have the EffectBone matching the actual wheel exactly.) Beginners can use the standard position x = 0, y = +1, z = 0 for their first car.



Finally the 3D data of you car will be prepared for the next step in CSO Importer by running the export script "Model XML" (.modxml format). When saving and exporting the car, you may only use letters and digits, **no special characters** (i.e. no spaces, umlauts, underscores, minus...). Also you should avoid starting the name with "CTR...", as this can lead to problems later.

The export script "Model XML" is part of the package "Blender Scripts" by Belgabor you can download at http://sourceforge.net/project/showfiles.php?group_id=181524 [SketchUp-Users save as ".ASE" like for CSO.]

Converting the 3D data into RCT3 model format with the CSO Importer

Using the **CSO Importer**, the saved 3D data of your car will be converted into the RCT3 model format. It's essential to use Importer version 18 (or higher) so CTR_Creator can load the models correctly! The procedure is similar to the creation of CSO, too, so this tutorial won't cover the use of the importer in detail, because there are enough good tutorials on this. Here are some to mention:

- Tutorial for Importer V17 by Eddi22 (containing useful files) (German)
  http://cso.rct-3.org/eddi22/005TutorialsMenu/Importer17Menu/TutoImporter.html
  -- on the lower right side (a bit difficult to find)

- Beginner tutorial for CSO Importer by Belgabor
  http://rct3.sourceforge.net/tutorials/beginner_tutorial.html

For CTR cars, the following procedure, which is easier than the complete procedure of creating CSO, is necessary. Important things that differ from CSO will be shown in detail.

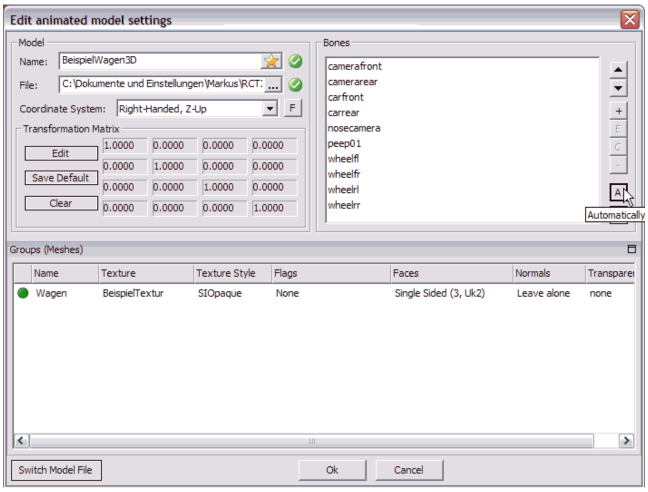
- In the main menu of the Importer click **"Create OVL" button**.

- At first – like with CSO – specify the **textures used**.

- It's essential to **switch to "Animated Models"** in the lower left to load your model there. [Note: Static Models can't be used for CTR cars (at the moment); in RCT3 Static Models are only used sporadically for the original cars as this has no advantages.] By the way: Your "Animated Models" don't have to be animated really; CTR cars work without animation which is the easiest way for beginners (but without animated restraints or similar the cars don't look that good of course).

- In the window "Edit animated model settings" you now have to **enter the created 3D data of your car** (in .modxml format if you followed my advice and use Blender). The settings necessary are identical to those needed for CSO.

- Next the **EffectBones** you created in your 3D modeling program have to be loaded (in the upper right in the "Edit animated model settings" window). To do this, press button "A" (for Automatically Add).


Edit animated model settings

Model

Name: BeispielWagen3D

File: C:\Dokumente und Einstellungen\Markus\RCT:

Coordinate System: Right-Handed, Z-Up

Transformation Matrix

1.0000	0.0000	0.0000	0.0000
0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	1.0000

Bones

camerafront
camerarear
carfront
carrear
nosecamera
peep01
wheelfl
wheelfr
wheelrl
wheelrr

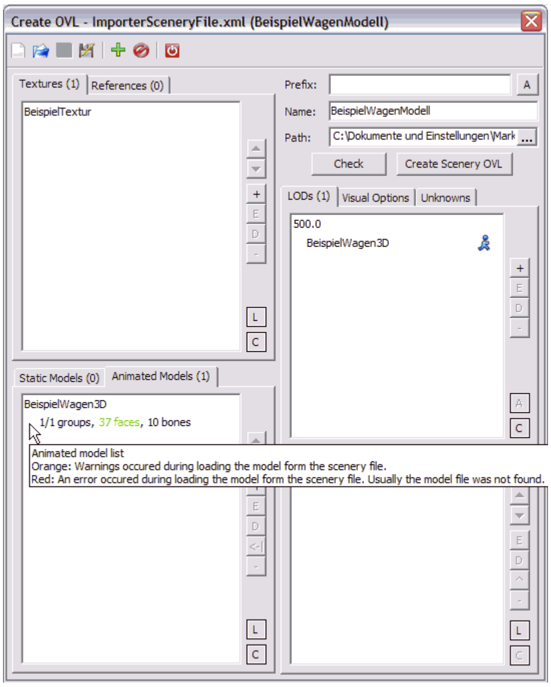
Groups (Meshes)

	Name	Texture	Texture Style	Flags	Faces	Normals	Transpare
●	Wagen	BeispielTextur	SIOpaque	None	Single Sided (3, Uk2)	Leave alone	none

- When the EffectBones are done, go back to the "Create OVL" window, where you specify the **LODs and eventually other settings** (identical to creating CSO).

- Finally the **file name of you car model** has to be entered in the upper right of the "Create OVL" window. For this name, only use letters and digits, but **no special characters** (i.e. no spaces, umlauts, underscores, minus...), too. Also you should avoid starting the name with "CTR...", as this can lead to problems later. The field "Prefix" has to stay empty.

- After selecting a path to save to, click **"Create Scenery OVL" button**. A warning that there are no animations in the "Animated Mode" can be ignored. Two files named *file name*.common.ovl and *file name*. unique.ovl are created that contain the model of the car in the RCT3 model format. Copy these two files into the working directory of the CTR_Creator; strictly speaking copy them to the place where you will save the WagenDaten.txt file later (if you use subdirectories).



BeispielTextur

Prefix:

Name: BeispielWagenModell

Path: C:\Dokumente und Einstellungen\Mark

500.0
BeispielWagen3D

BeispielWagen3D
1/1 groups, 37 faces, 10 bones

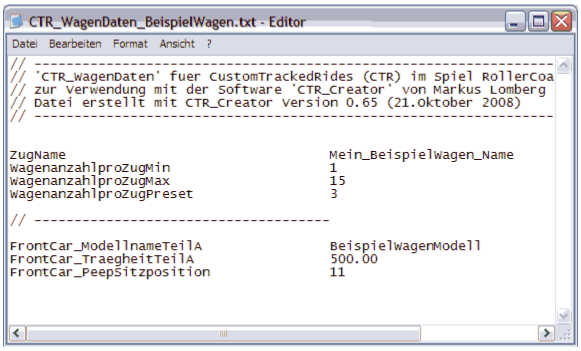
Animated model list
Orange: Warnings occured during loading the model form the scenery file.
Red: An error occured during loading the model form the scenery file. Usually the model file was not found.

In the next step, we'll create the final CTR car.ovls for RCT3 out of these car model OVLs and additional values for the car using CTR_Creator.

Adding additional data and creating the car in CTR_Creator

Now your car models with some EffectBones exist in RCT3 data format with *file name*.common.ovl and *file name*.unique.ovl. In this step, some additional data that describe the behavior of the car has to be added. This is done with the WagenDaten.txt file (similar to the BahnDaten.txt file for CTR rides). Using CTR_Creator the car model OVLs and the additional data will be merged into a complete working CTR car. The basic procedure in CTR_Creator is identical to the procedure needed for CTR rides.

In this tutorial for beginners we'll only cover single cars or trains with identical cars. This one car is always defined as **FrontCar**, i.e. the keywords and values described following have to be applied for the FrontCar in the WagenDaten.txt file; otherwise your CTR car won't behave like you intended.



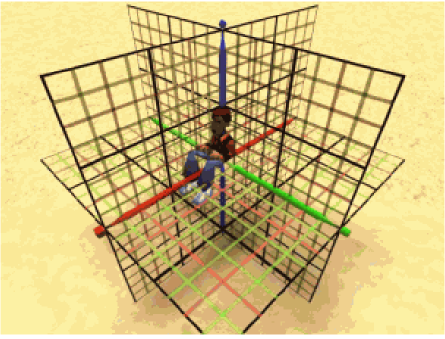
```
// ------------------------------------------------------------------
// 'CTR_WagenDaten' fuer CustomTrackedRides (CTR) im Spiel RollerCoa
// zur Verwendung mit der Software 'CTR_Creator' von Markus Lomberg
// Datei erstellt mit CTR_Creator Version 0.65 (21.Oktober 2008)
// ------------------------------------------------------------------


ZugName                              Mein_BeispielWagen_Name
WagenanzahlproZugMin                 1
WagenanzahlproZugMax                 15
WagenanzahlproZugPreset              3

// ------------------------------------

FrontCar_ModellnameTeilA             BeispielWagenModell
FrontCar_TraegheitTeilA              500.00
FrontCar_PeepSitzposition            11
```

At first, the desired attributes of the car will be written into the **WagenDaten.txt file**. Use the empty template for this or change the example file. The adjacent image shows such a text file for the additional data (in this image, the data is already filled in).

On the left you can see the so-called keywords. Behind these, you have to write the keyword values. If you're not sure, you can leave them empty. CTR_Creator will then add a default value that matches most use cases automatically.

keyword	**description of the value**
TrainName	Name of the car or train shown in the game. This name will be displayed in RCT3 when looking in the car selection; strictly speaking it is the name of the train but it is also used for single cars. Space characters and special characters are not allowed here. *Example: CarForSuperScreamer*
CarCountPerTrainMin	Minimum number of cars per train you can set in the game. *Example: 1*
CarCountPerTrainMax	Maximum number of cars per train you can set in the game. *Example: 15*
CarCountPerTrainPreset	Preset number of cars per train. *Example: 3*
FrontCar_ModelNamePartA	Name of the 3D car model from the CSO Importer. Enter the *filename* of the OVLs created by the CSO Importer here; i.e. the name without the ending common.ovl or unique.ovl. *Example: BeispielWagenModell*
FrontCar_InertiaPartA	Inertia of the car. This value defines the inertia of the car that has an influence on the performance of the car. Simplified you can imagine some kind of mass/weight. You should enter a value between about 200 and 600 here. *Example: 400*
FrontCar_PeepSeatingPosition	Style of the seating position of the peeps in the car. This value defines which position the peeps will have in the car: 0=Stand, 1=Sit, 2=Sledge, 3=Lay, 4=Bike, 5=Car, 6=Horse, 7=Row1Oar, 8=Row2Oar, 9=SitVReel, 10=SitHarness, 11=SitBar, 12=Observation, 13=Sit_Harness_ Feet_Dangle, 14=Horse_Steeple, 15=Pedalo, 16=Pedal_Bike, 17=Canoe,

	18=Dinghy, 19=Lay_Front, 20=Row_L_Oar, 21=Row_R_Oar, 22=Horse_Merry_Go_Round, 23=Horse_ SteepleChase, 24=Jump, 25=Bum_Bounce, 26=Dance. There are additional values for RCT3 with addons Soaked and Wild for the seating position: 27=Surf_Board, 28=Wind_Surf, 29=Fishing, 30=Slide_Body_Back, 31=Slide_Ring, 32=Inflatable, 33=Sit_ Stadium, 34=Swim Thanks to Belgabor for discovering this list! *Example: 11*

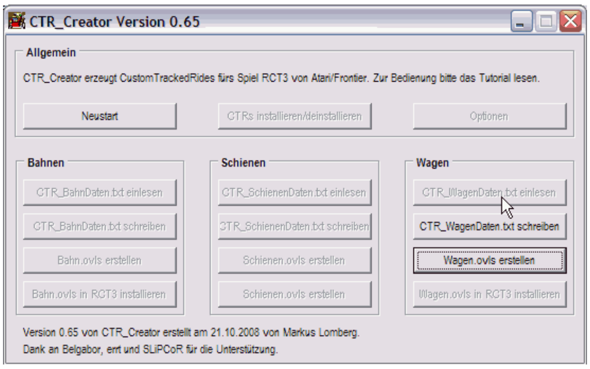


By the way: SliPCoR made **screenshots of all peep seating positions** of the adults, teens and children (see example on the right). You can download these at:

http://s345.photobucket.com/albums/p367/slipcor/peeps/

On the one hand this list gives you a better understanding of the postures and the coordinates of the positions mentioned above. On the other hand the images are very helpful to design the geometry of the car, the seats and restraints correctly in the 3D program in the first try, when overlying the model with the screenshots of the three views as semitransparent background images in the 3D program. Thanks to SLiPCoR for this helpful list of images!

After entering all desired values, save the text file with a new name. As mentioned above, this WagenDaten.txt file has to be saved to the same directory your car model OVLs created using the CSO Importer (*file name*.common.ovl and *file name*.unique.ovl) are in necessarily! The file name of the text file has to start with "CTR_WagenDaten_", followed by the **name of the directory** your car will be put to in the RCT3 directories, so e.g. "CTR_WagenDaten_SuperScreamerCar.txt". The name is used internally by RCT3 and is never shown in the game. This is the name you will have to use for the CTR ride with the keyword "CarTypeName". This will be explained again below.


CTR_Creator Version 0.65
Allgemein
Neustart
CTRs installieren/deinstallieren
Optionen
Bahnen
CTR_BahnDaten.txt einlesen
CTR_BahnDaten.txt schreiben
Bahn.ovls erstellen
Bahn.ovls in RCT3 installieren
Schienen
CTR_SchienenDaten.txt einlesen
CTR_SchienenDaten.txt schreiben
Schienen.ovls erstellen
Schienen.ovls erstellen
Wagen
CTR_WagenDaten.txt einlesen
CTR_WagenDaten.txt schreiben
Wagen.ovls erstellen
Wagen.ovls in RCT3 installieren

CTR_Creator erzeugt CustomTrackedRides fürs Spiel RCT3 von Atari/Frontier. Zur Bedienung bitte das Tutorial lesen.

Version 0.65 von CTR_Creator erstellt am 21.10.2008 von Markus Lomberg.
Dank an Belgabor, ernt und SLiPCoR für die Unterstützung.

In CTR_Creator you now can **load this CTR_WagenDaten text file**. If everything was right (i.e. you didn't do any bigger mistakes with the values behind the keywords in the text file), two new buttons will appear.

The **button "Write CTR_WagenDaten.txt"** will overwrite your text file and CTR_Creator will automatically format everything correctly, enter the default values behind keywords without value and add more keywords that are interesting for advanced users and pros only.

If there are problems with the CTR car in the game, it's possible to see whether the data was interpreted correctly by the CTR_Creator with this CTR_WagenDaten text file. If errors occur with the interpretation, you can correct these in the text file and (after clicking the "Restart" button) load the corrected CTR_WagenDaten again.

Note: Pressing the “Write CTR_WagenDaten.txt“-button is not necessary for importing your car, you only need to do that to get the reformatted txt-file as described above.

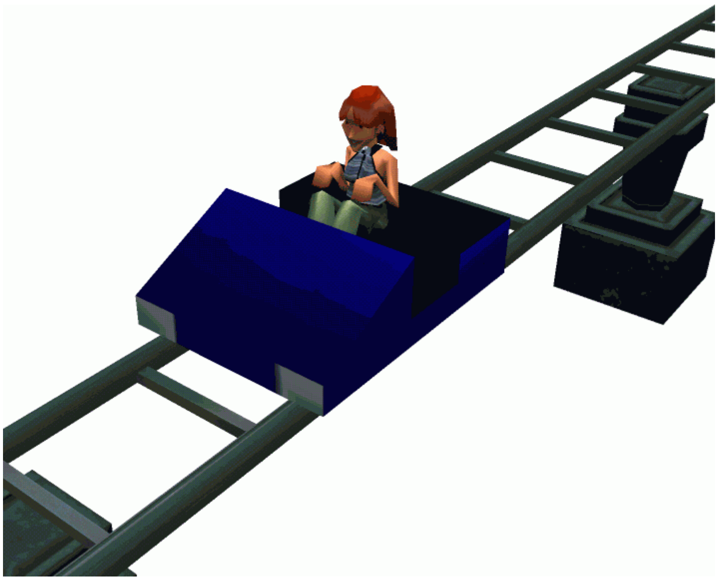
The most important **button "Create Car.ovls"** will create the car.ovl files usable for RCT3 from the CTR_WagenDaten read in and the model OVLs created in the CSO Importer before. It will create a directory containing the two car.ovls in your working directory of CTR_Creator.

You can give this directory containing the CTR car to other users. You can also install the car.ovls into RCT3 by hand by copying this folder into "\Cars\TrackedRideCars" in your RCT3 directory.

The easier way is the automated installation using the **button "Install Car.ovls in RCT3"** that appears after the car.ovls are created in your working directory. Of course, you don't have to click this button if you prefer installing the car.ovls by hand.

Now the creation of your first CTR car is done! To use your CTR car in the game, you need a **CTR ride using your CTR car** of course. You already know how to create a ride using CTR_Creator. When creating this CTR ride, the name of the directory of the car, e.g. "CTR_SuperScreamerCar" (not the name you entered with the keyword "TrainName"!) has to be entered with the keyword "CarTypeName" in the WagenDaten.txt.

Finally you can play with the self-made CTR car in RCT3 using the matching CTR ride. The way to this point was laborious and long-winded certainly, but the view of the custom car is an amazing regard!



Additional information

The **lists behind keywords** can be separated by spaces, tabulators or commas.

To **uninstall a CTR car** go to the directory „\Cars\TrackedRideCars“ in your RCT3 directory where all created CTR car directories are. To uninstall simply delete the directory of the CTR car undesired. Usually you should not forget to delete the CTR ride using this car, too, as in most cases the ride will be useless without the car.

The tool doesn't have an effective **error handling** or logging. In case of an error you should close CTR_Creator and restart it to avoid an unexpected state and problems.

CTR_Creator doesn't **check for senseless inputs** (e.g. negative inertia). So don't complain if your senseless inputs produce curios effects in RCT3.

There can also be **unexpected results** if you create cars with special features and use them with a CTR ride not adapted for it. The custom rides and cars give so many different possibilities of settings the RCT3 programmers couldn't foresee, so unexpected effects or problems can occur.

For beginners that want to do a step forward, I'll explain some additional **EffectBones for several peeps and for the rear wheels** now. The EffectBones peep01, peep02, peep03, etc. give the ability to place several peeps in your CTR car. The EffectBones for the rear wheels help to keep the car on the track better, as the front points alone can cause the rear of the car to pull out. Also the four EffectBones of all wheels are needed for the correct function of advanced CTR cars, so you should get accustomed to always specify the four wheel coordinates.

EffectBone name	**description of the value**
peep##	Position of more than one peep in a car. For every peep in a car coordinates have to be specified. ## is a consecutive number. I.e. for a car with four peeps you need to specify different values for peep01, peep02, peep03 and peep04 (otherwise they will all seat in the same place which looks strange).
wheelrl	Coordinates of the rear left wheel. (Note: It's not necessary to have the EffectBone matching the actual wheel exactly.)
wheelrr	Coordinates of the rear right wheel. (Note: It's not necessary to have the EffectBone matching the actual wheel exactly.)

Closing words

If you read the tutorial carefully and still have problems with your CTR cars, you can get help in the forums of www.RCT-3.org or www.ataricommunity.com. Problems already known will be discussed, new questions will be answered by experienced users. I won't answer mails or PMs!

If you have gained enough experience already so the CTR car settings for beginners become boring, I suggest the tutorial for advanced users. The settings for CTR cars with many more and more complex options (e.g. animations, more than one model) will be explained there.

Special thanks to Belgabor, without his new Importer18 CTR cars would not have been possible. Also Belgabor helped me a lot with his tireless effort for RCT3, his scripts for Blender and many tips. I also want to thank errt and SLiPCoR for their extensive support. Also thanks to all the beta testers that helped me to find (hopefully) all of the bugs. Thanks to Lonestar for providing the forums on www.RCT-3.org.

Written Markus Lomberg on 12-12-2008 (version 1.3) exclusive for www.rct-3.org
Translated by errt on 28-12-2008 (version 1.3e) based on the German version.