

2019-05-25, Folker Linstedt, Project: FPSC Classic Model-Converter for „MiniGuru“

Task:

A tiny prop placement tool and somewhat level editor for App Game Kit and other SDKs with a low usage of memory and no need of high-performance hardware, maybe working on low spec devices like mobile phones and the Raspberry Pi and or Android set top boxes

Achievable:

Models and textures have to be loaded very fast and without hustle into a simple editor, levels have to be also loaded quickly

Free models are available from TheGameCreators to create a mini 3D editor, to be able to use these models, here are the steps to convert the models:

Steps to achieve this goal:

1. Download the 3D models

<https://github.com/TheGameCreators/FPS-Creator-Classic>

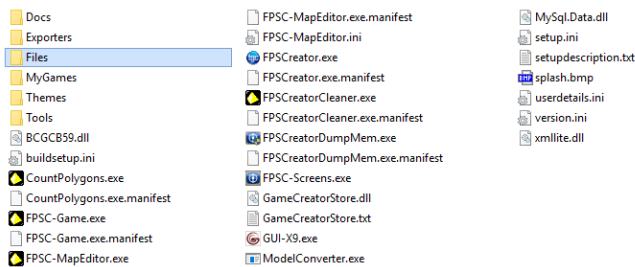
Free Model Packs

Over the years FPS Creator Classic released a lot of model packs, some created officially and some created by third parties. Those artists have generously agreed to release their packs for free as part of this open source project.

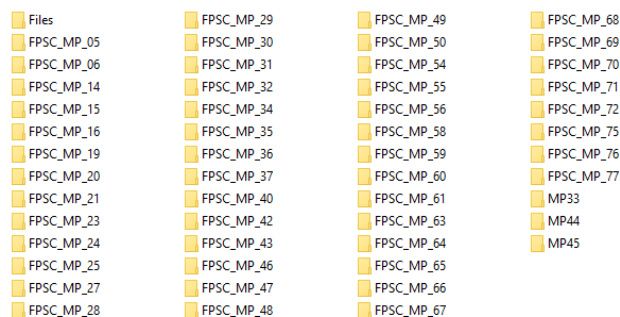
[FPSC Model Pack 01](#) [FPSC Model Pack 03](#) [FPSC Model Pack 05](#) [FPSC Model Pack 06](#) [FPSC Model Pack 07](#) [FPSC Model Pack 11](#) [FPSC Model Pack 12](#) [FPSC Model Pack 13](#) [FPSC Model Pack 14](#) [FPSC Model Pack 15](#) [FPSC Model Pack 16](#) [FPSC Model Pack 19](#) [FPSC Model Pack 20](#) [FPSC Model Pack 21](#) [FPSC Model Pack 23](#) [FPSC Model Pack 24](#) [FPSC Model Pack 25](#) [FPSC Model Pack 27](#) [FPSC Model Pack 28](#) [FPSC Model Pack 29](#) [FPSC Model Pack 30](#) [FPSC Model Pack 31](#) [FPSC Model Pack 32](#) [FPSC Model Pack 33](#) [FPSC Model Pack 34](#) [FPSC Model Pack 35](#) [FPSC Model Pack 36](#) [FPSC Model Pack 37](#) [FPSC Model Pack 40](#) [FPSC Model Pack 42](#) [FPSC Model Pack 43](#) [FPSC Model Pack 44](#) [FPSC Model Pack 45](#) [FPSC Model Pack 46](#) [FPSC Model Pack 47](#) [FPSC Model Pack 48](#) [FPSC Model Pack 49](#) [FPSC Model Pack 50](#) [FPSC Model Pack 54](#) [FPSC Model Pack 55](#) [FPSC Model Pack 56](#) [FPSC Model Pack 58](#) [FPSC Model Pack 59](#) [FPSC Model Pack 60](#) [FPSC Model Pack 61](#) [FPSC Model Pack 63](#) [FPSC Model Pack 64](#) [FPSC Model Pack 65](#) [FPSC Model Pack 66](#) [FPSC Model Pack 67](#) [FPSC Model Pack 68](#) [FPSC Model Pack 69](#) [FPSC Model Pack 70](#) [FPSC Model Pack 71](#) [FPSC Model Pack 72](#) [FPSC Model Pack 75](#) [FPSC Model Pack 76](#) [FPSC Model Pack 77](#)

Some models are also in FPS Creator Classic and Dark Basic Pro

2. Install FPS Creator Classic onto your computer, the main folder for the conversion is “Files”



3. After downloading – all assets have to be extracted, and have to be moved into the FPSC Classic files structure













A look into the FPSC_MP... folder will show, that some of them contain subfolders e.g.:

-  audiobank
-  entitybank
-  scriptbank
-  segments

these subfolders are also in the main folder "Files" from the FPSC Classic installation.

4. ➔ move all model-pack-files into the FPSC Classic "Files"-folder! Not every model-pack contains the same folder-structure e.g:

-  audiobank
-  effectbank
-  entitybank
-  Font
-  gamecore
-  mapbank
-  meshbank
-  scriptbank
-  segments
-  texturebank

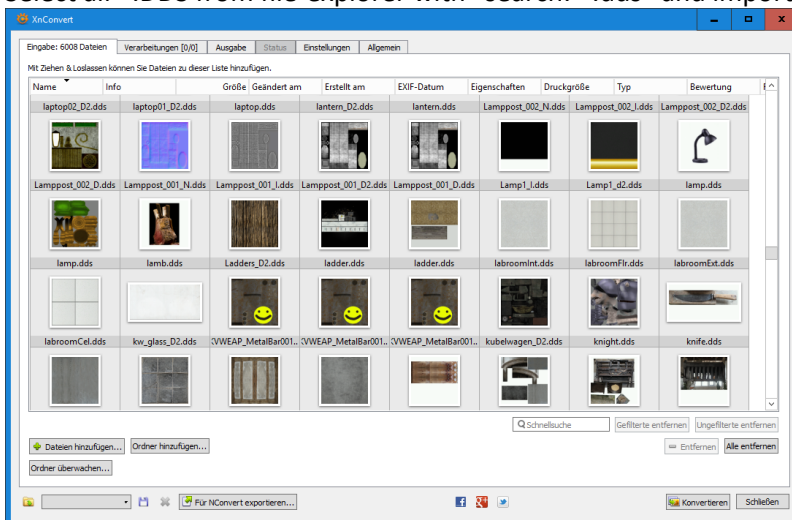
for the audiobank from the model-pack has to be "Files/audiobank"

5. Now the *.DDS texture files have to be converted into something App Game Kit can use! JPEG or PNG. For the best quality and transparency at first PNG would be the best. For conversion IrfanView or XnConvert could be used. Batch-conversion.

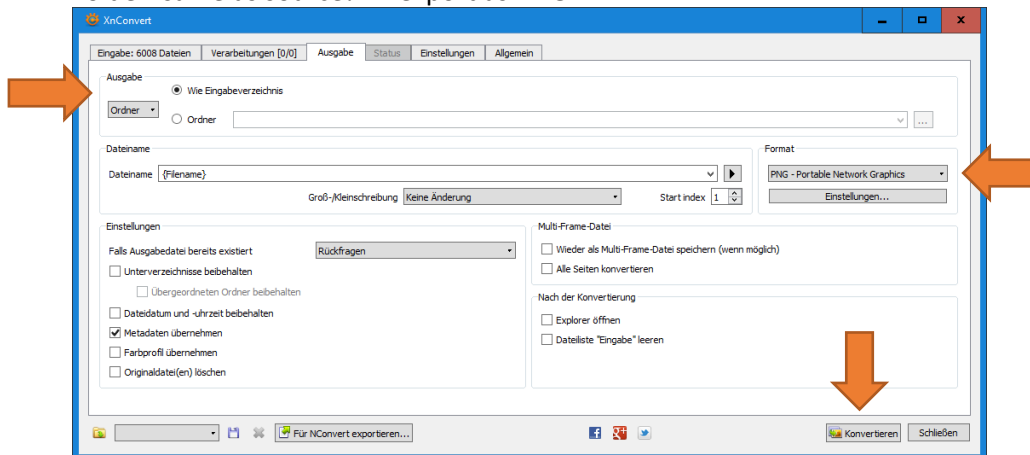
<https://www.irfanview.net/64bit.htm>

<https://www.xnview.com/de/xnconvert/>

Select all *.DDS from file-explorer with "search: *.dds" and import



Folder: same as source! ➔ export as PNG



6. Now the App Game Kit converter can find and open files from the FPSC Classic

What does the program do?

It searches from the “main-folder” → “Files” all *.FPE files or *.FPS files. FPE are prefabs, FPS are segments. They are nearly the same, but segments can contain prefabs and are mostly more than one object and more like a combination of models

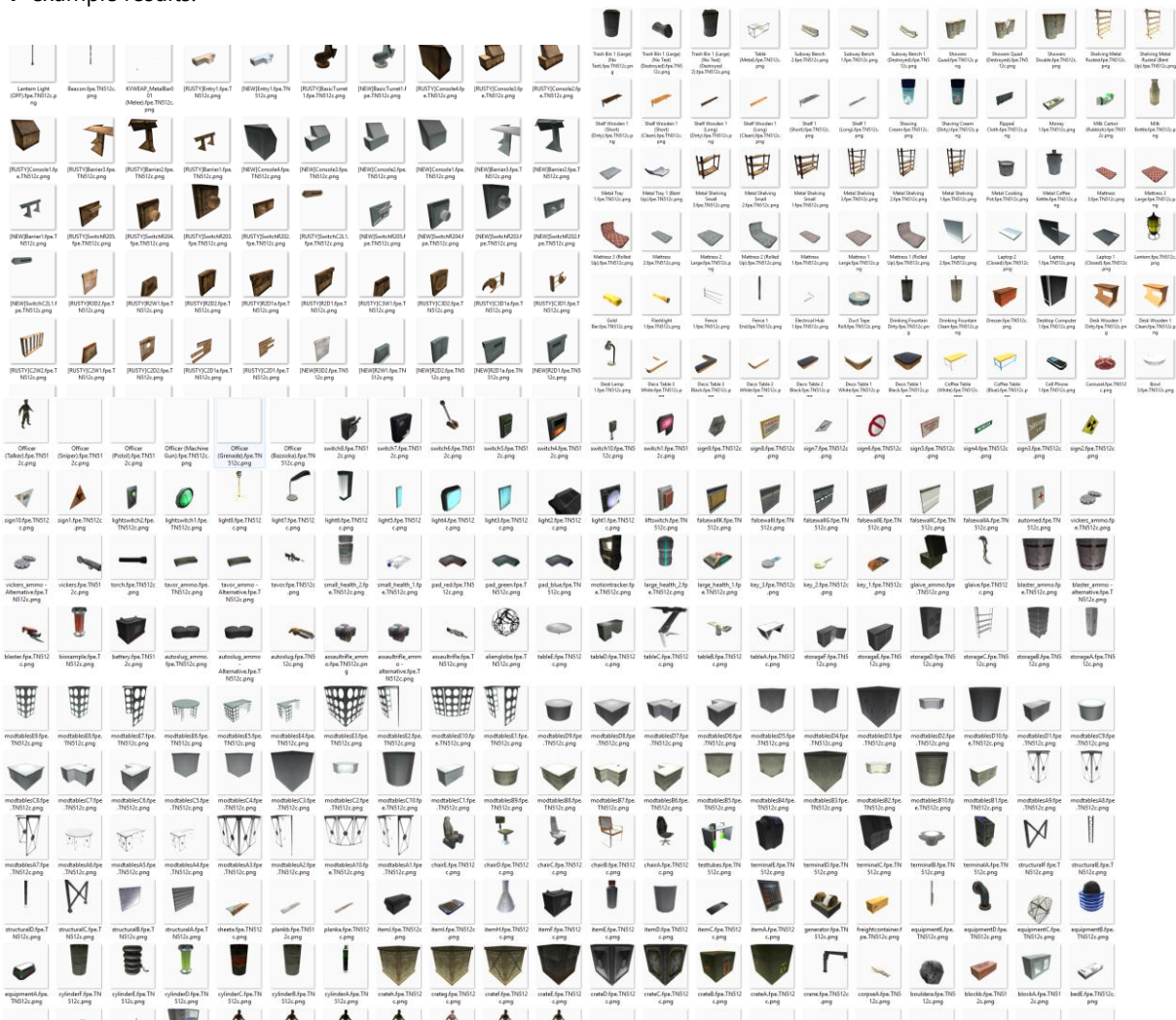
The program (the App Game Kit converter tool) creates a list of all *.FPE or *.FPS files and opens one after the other from the created list. The program could be changed to read files from another list or from a combined list and so on.

Every item of this list is opened and read. The model *.X-files are loaded, the *.DDS textures are not loaded, but if a *.PNG file in the same folder exists where the *.DDS file should have been than the texture image could be loaded and will be. After this the texture is combined with the model, the model is rotated, and the camera is positioned.

An image will be captured with a new file-name-ending in the same folder as the *.FPE or *.FPS
It will be also a text-file created with the needed folders for model, texture and FPE/FPS file.

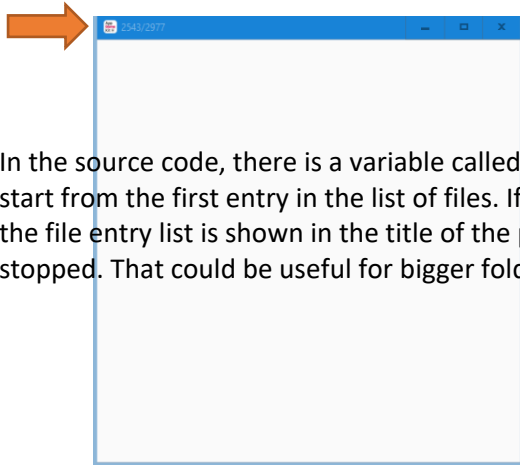
With the search from the file explorer it is possible to only show the generated images:

Only shows the images, but not the generated *.txt files, size: < 1kb: *.tn512c.png **größe: > 1kb**
→ example results:



When starting the program, it looks like that:

Program looks like:



In the source code, there is a variable called "k", this is for a starting point. If k=0 the program will start from the first entry in the list of files. If the program crashes or has to be shutdown, the index of the file entry list is shown in the title of the program, so it is possible, to start where the program has stopped. That could be useful for bigger folders, or if only a part of the files has to be re-created.

WARNING!

It will overwrite existing files. If the program runs it don't check, if *.PNG-files or *.TXT files already exist.

```
fenRenderAndSaveObject(files[k].entry$, ".TN512c.png")
```

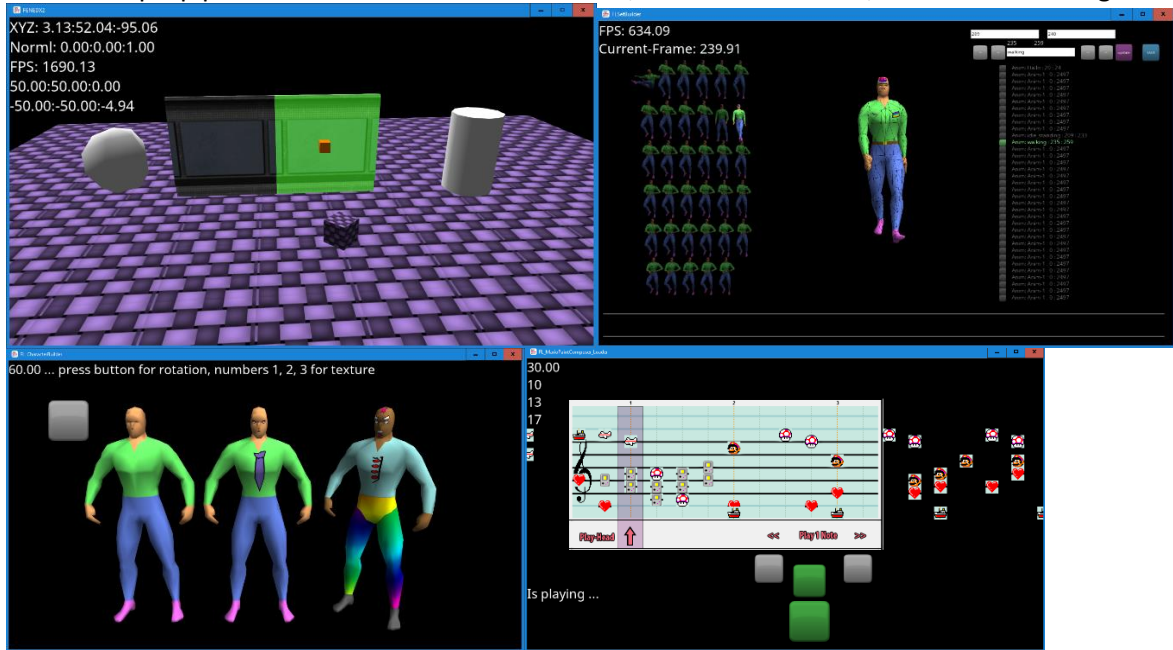
this will create the preview-files. Changing the name here could prevent thumbnails to be overwritten.

THE PROGRAM is NOT OPTIMISED. It is very slow. For testing please use only ~ 200 files or less. Also FPS files need much more time to "render", because there is no optimization and FPS files contain up to 16 files or more, and textures and models are loaded as much as needed.

Doppelganger files are not sorted out in this step! The preview could be the same, if the model file and the texture file are the same.

ToDo!

Create the prop placement editor. Character builder, animation "editor", music creator integration



And have all textures without doppelgangers on combined textures for faster loading.

