

AppGameKit
GameGuru
LOADER >>>

USER GUIDE





Contents

Introduction	4
The Files You Get With This DLC	5
Why GameGuru Media Needs To Be Converted	8
Converting GameGuru Media Using The Converter Tool	9
Converting PBR Assets	14
Game Guru Loader – AppGameKit Source Code	15
Broadcasting Your Level	16

Introduction

Welcome to the GameGuru Loader user guide. GameGuru Loader is an easy way to convert and optimise your GameGuru Level creations into high quality, multi-platform ready, media and code that is compatible with the AppGameKit programming language, allowing for speedy development of immersive 3D environments and games.

We continue to support our amazing community with the addition of this DLC, brought to you by one of our very own community members, Preben Eriksen. We were so impressed with the tool that we wanted to make it an official add-on to our product range. It brings extended functionality to two of our flagship products, GameGuru and AppGameKit. While you'll need to own both GameGuru and AppGameKit to make use of this DLC, you'll then be free to convert your levels and media for use in your multi-platform AppGameKit creations.

The key advantage of GameGuru and AppGameKit over other development platforms is the incredibly small learning curve and the ability to broadcast your creations instantly to your platform of choice using the AppGameKit Player. Design your level in GameGuru, convert it with GameGuru Loader, and then test your creation using the generated source-code and media for your level in AppGameKit. You can also broadcast and play on your worlds to any Android and iOS devices you have.

The aim of this guide is to help you learn how to:

- Convert all your GameGuru media to a format compatible with AppGameKit
- Convert your levels to run in AppGameKit
- Load and test your levels in AppGameKit
- Tweak display settings to suite your target devices

GameGuru Loader is a media converter, and collection of AppGameKit source files and advanced shaders, that allow you to load the 3D content from GameGuru into your AppGameKit projects. This means you can design a 3D environment with GameGuru and load it in AppGameKit. It will not play an entire GameGuru game within AppGameKit, although great strides are being made in additional functionality, such as vehicle physics, simple NPC AI that allows non-player characters to follow GameGuru waypoints, lens-flare effects, proximity sensitive doors, real-time shadows, water reflections, cartoon shaders, per-object bloom effects and more.

Now, let's get started...

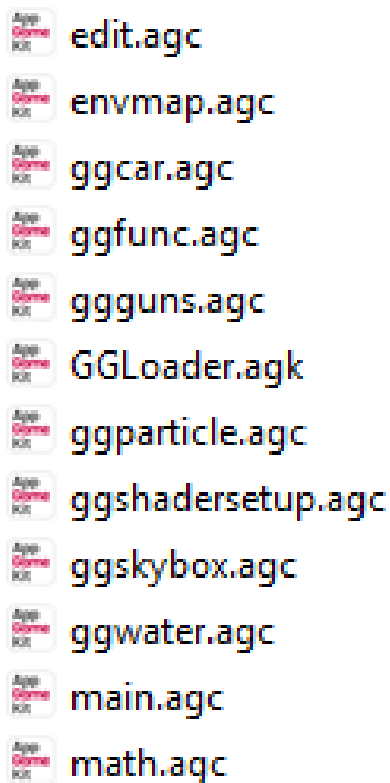
The Files You Get With This DLC

This DLC contains two archive files;

1. Ggloader142.zip

This is a package of source-code and shaders designed to load the converted 3D GameGuru levels into AppGameKit. In the package you will find the following files (both Windows and MacOSX versions are included, but for the purpose of this document, we will focus on the Windows files):

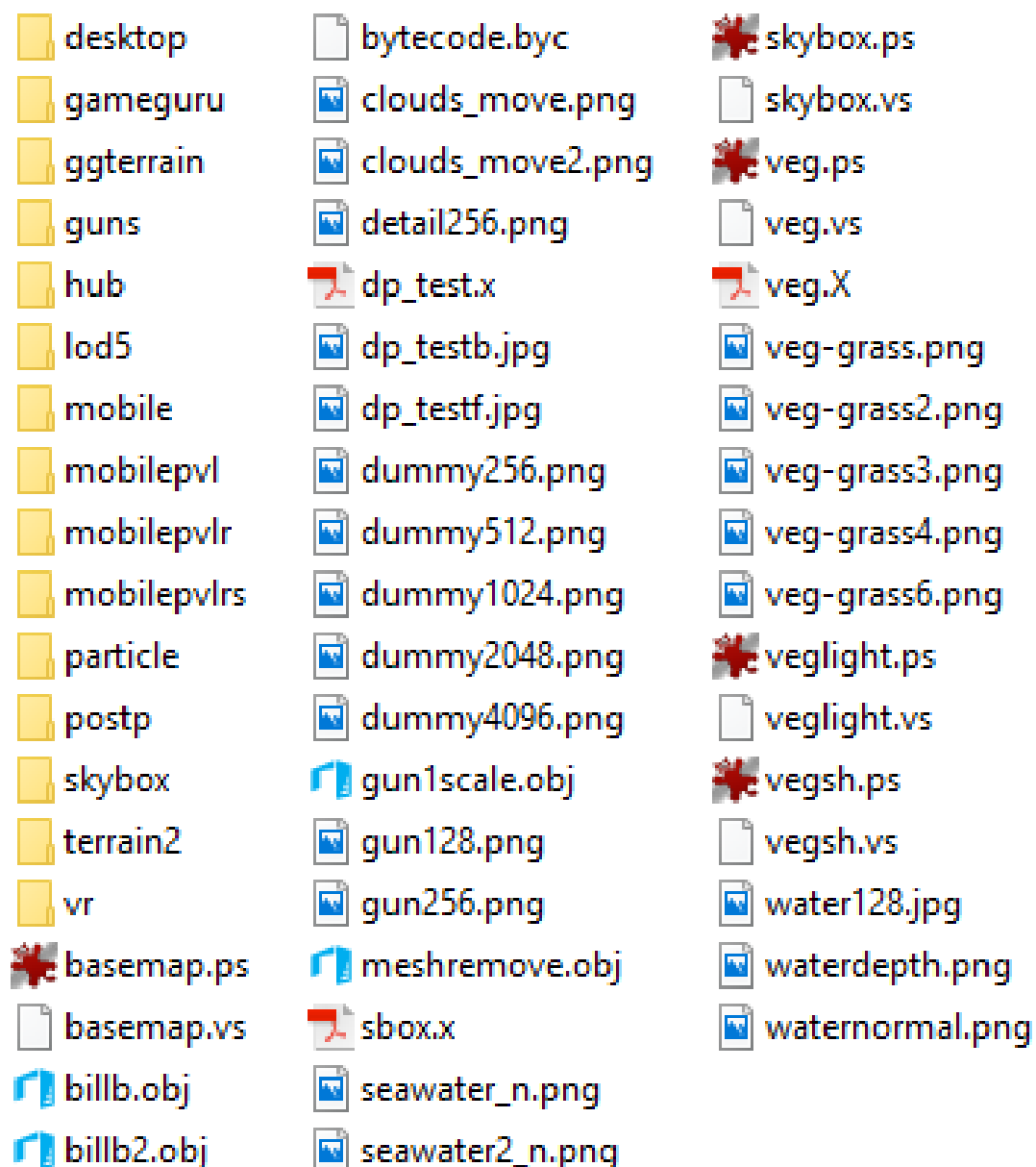
The GameGuru Loader AppGameKit project file and source code for loading your converted levels:



Documentation for converting levels without using the Graphical User Interface GGLoaderGUI:

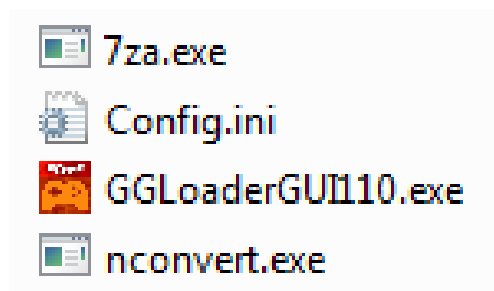


The media folder (including sub-folders), containing all the advanced shaders and default textures and objects required by GameGuru Loader:



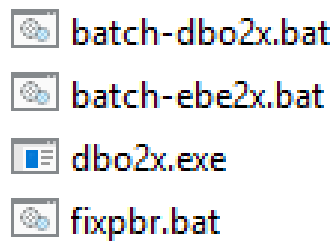
2. GGLoaderGUI110.zip

This archive file contains the third party batch file conversion, compression and helper tools to make converting your levels a breeze:



3. Fixpbr.zip

This contains a tool for converting PBR assets into a format that can be used by AppGameKit, see below for more details.

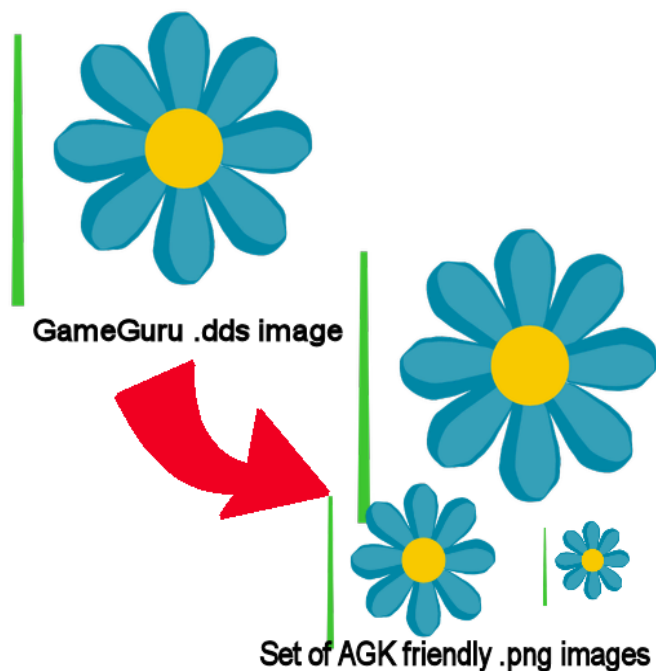


Why GameGuru Media Needs To Be Converted

GameGuru uses specific types of images and objects designed to be optimised for Windows gaming. Unfortunately these types of files are not compatible with AppGameKit, and do not offer the resource efficient file sizes required for mobile gaming. We need to convert those files into something AppGameKit can use.

Within this DLC you can find the original *doc-ggloader.pdf* guide which is a step-by-step guide to manually converting all the GameGuru media. However, we now have a helper tool called **GGLoaderGUI110.exe** to do all the conversions for us!

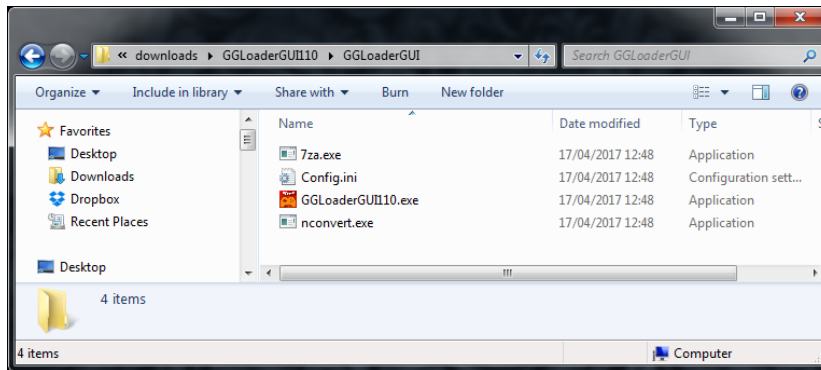
Converting the media is a very time CPU intensive task, and is rather *time consuming*. Once the conversion process is under way, be prepared for it to take several hours to complete. However, this only needs to be done once. So when all your GameGuru media is converted, it can be re-used over and over with any new level you create in GameGuru. You'll only need to convert media again if you add more media to GameGuru (i.e. if you install a new GameGuru DLC pack, items from the Asset Store, or add your own custom objects and imagery).



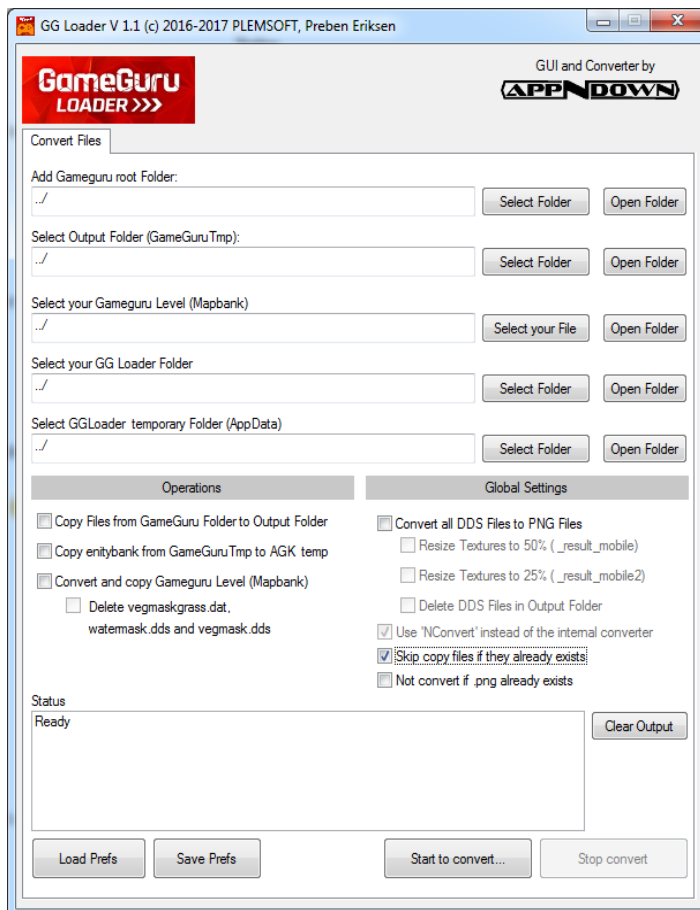
The conversion process takes images used by GameGuru for high quality desktop graphics (such as .dds images) and converts them to lower resolution .png format images.

Converting GameGuru Media Using The Converter Tool

Launch the converter tool *GGLoaderGUI110.exe*



When run you will see the following screen appear:



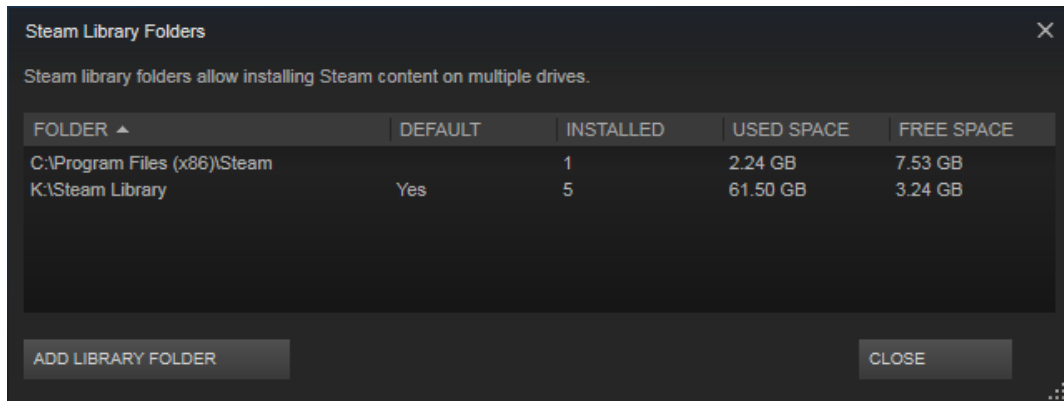
Using the 'Select Folder' buttons, you need to populate all the folder locations. It's very important you get this right for the process to work.

GameGuru root Folder:

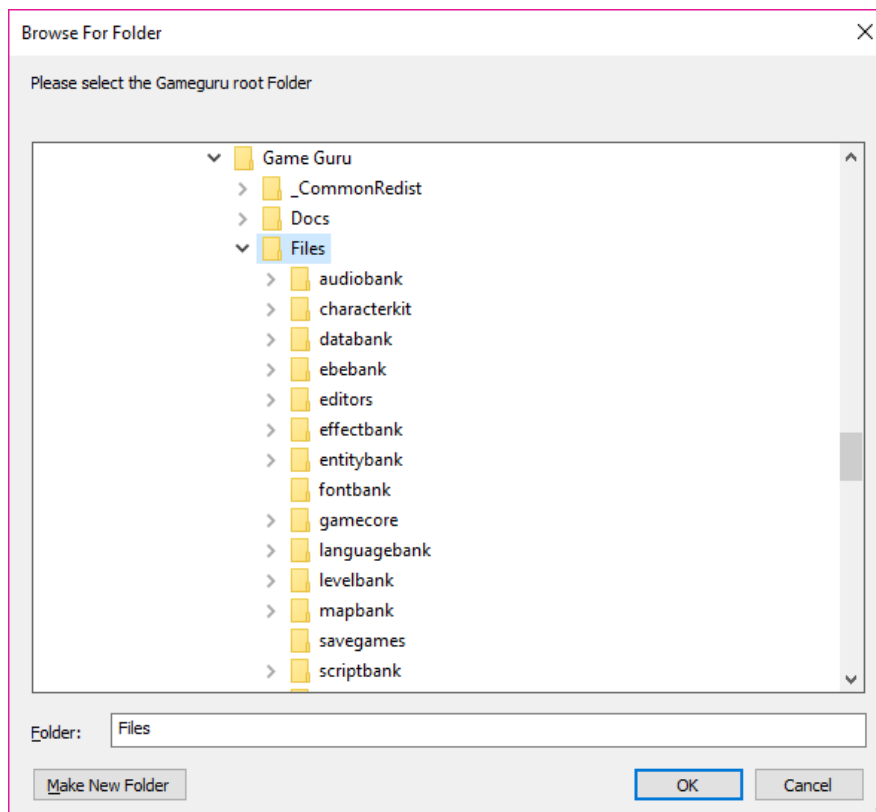
This is where you installed GameGuru on your computer. The default steam location for the GameGuru root folder on Windows is:

```
c:\Program Files\Steam\Steam Library\steamapps\common\Game Guru\
```

If you have installed GameGuru in a non-default location, you can find your Steam library folder by launching Steam, click Steam>Settings>Downloads>Steam Library Folders.



Once you know your Steam Library location, return to the GGLoaderGUI app, click the Select Folder button next to the GameGuru root folder and follow the path as above \steamapps\common\GameGuru\Files and click OK.



Output Folder (GameGuru Tmp):

The output folder is where every possible item of GameGuru media you have will be exported to in its lower resolution format. So you need to create a new folder to store all the converted GameGuru media. Make sure it is not within your AppGameKit project's folder in order to prevent **all** the converted media being included in your game. An example output folder might be;



GameGuru Level (Mapbank):

Here you must select your level that you created with GameGuru. The level will be saved with a *.fpm* file extension and by default can be found within the `Files\Mapbank` folder from your GameGuru root. You can also use *.fpm* files from other locations (such as demo levels downloaded from other sources).

GG Loader Folder:

This is where you unzipped the GGLoader AppGameKit Project files.

GGLoader temporary Folder (AppData):

AppGameKit stores any files created by an AppGameKit app within the User appdata folder in Windows. By default this location would be found in:

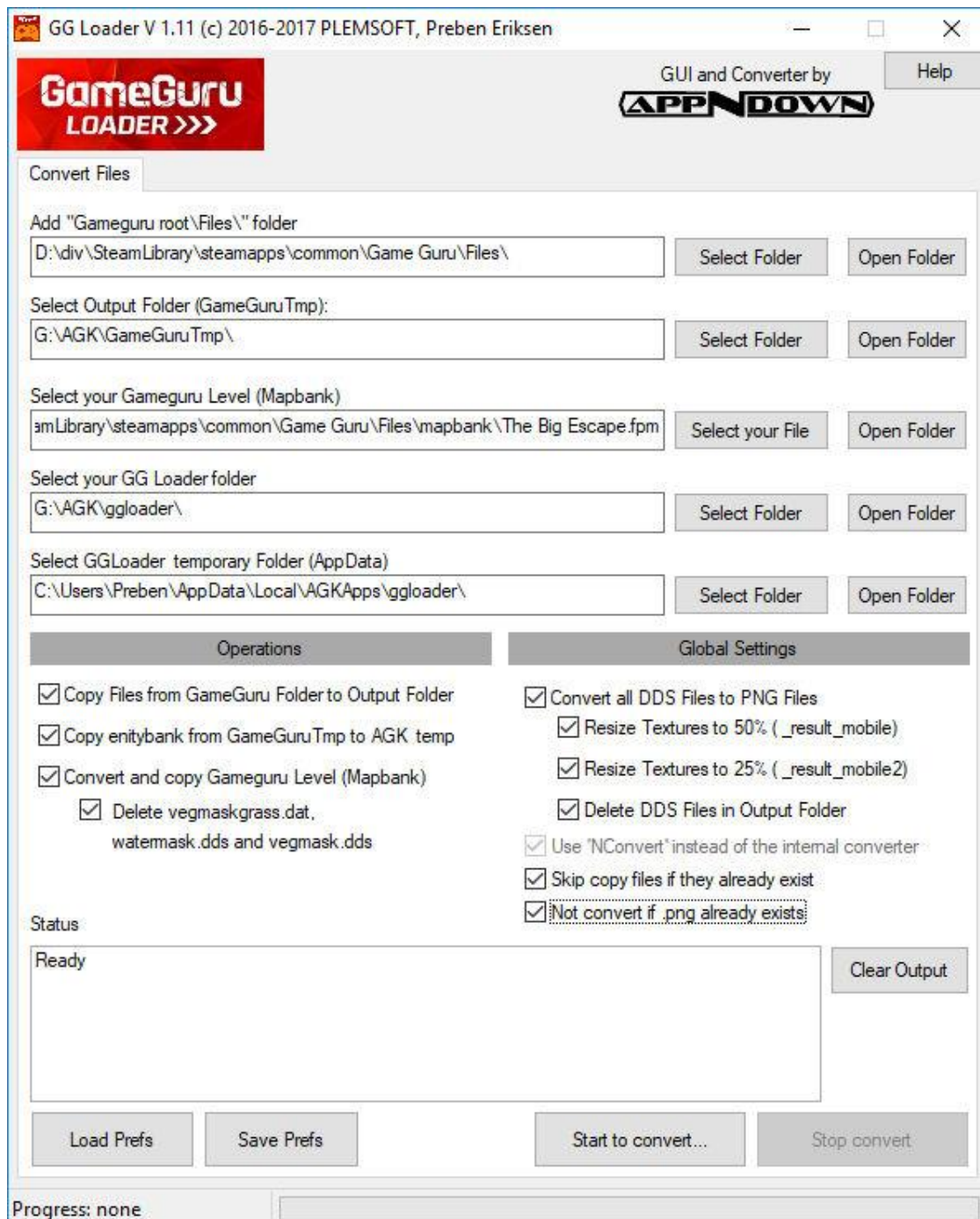
```
c:\user\<<your username>\AppData\Local\AGKApps\<<your AppGameKit project name>
```

...where `<your username>` is your Windows login, and `<your AppGameKit project name>` is the name of your AppGameKit project. For the purpose of this document, we will assume you are using the default GameGuruLoader project called `ggloader`.

Note: By default this is a hidden folder on Windows and you may have to show hidden files to see it. You can read more information about this folder [HERE](#).

Tick ALL the options under the *Operations* and *Global Settings* headings.

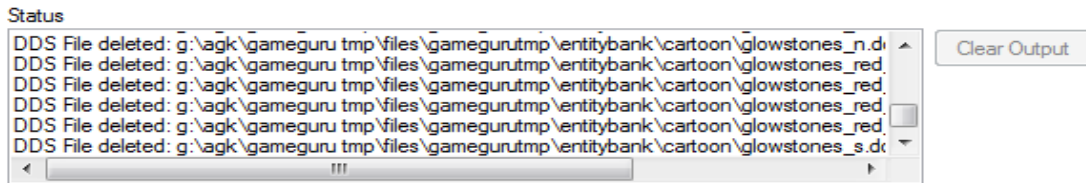
Once done, you should end up with something looking like this:



It is a good idea to click 'Save Prefs' at this point so you don't need to re-enter all those folder locations again in the future.

Now you can click 'Start to convert', sit back and relax for the next few hours whilst your entire GameGuru media library is converted. N.B. This process *really will* take hours to complete so be patient and don't turn off your PC or interrupt the process.

During the conversion process you will see progress updates in the status window. This give you some feedback to know the process is happening and files are being loaded, converted and saved in the new mobile format.



Once again, just in case you missed it, the initial conversion process **WILL TAKE SEVERAL HOURS TO COMPLETE.**

Using the folder settings you set within the application, the converter will build a new and complete media library (in the 'Output folder (GameGuru tmp)' location), and will copy all the required media and most of the converted level data into your AppGameKit project media folder and AppData media folder (some level data is created when you run the ggloader AppGameKit project). All GameGuru entities will be placed into an entitybank folder within AppData so that your AppGameKit project can find all the files it needs to build your selected level.

When the initial conversion is complete, you can safely untick the following options:

- Copy files from Game Guru to Output folder
- Convert all dds files to png
- Copy entity bank from GameGuruTmp to AGK temp

...as those processes are only required once. If you need to convert more levels, you are able to do so with the current set of options selected unless you add additional media to your Game Guru media library, in which case you will need to select all options again and click 'Start to convert' to add the new media to your converted set.

Converting PBR Assets

To use the new Physically Based Rendering (PBR) assets from the latest public release version of GameGuru you need to convert it into a format that AppGameKit is able to read. Supplied with GameGuru Loader is a new tool called **fixpbr.zip**

How to convert PBR assets

- Unzip the file **fixpbr.zip** into any folder
- Edit **fixpbr.bat** using notepad and make sure that the path is set to where you have GameGuru located. By default it is set to drive c: like this:
c:\Program Files (x86)\Steam\steamapps\common\GameGuru\Files\entitybank\...
- If you have GameGuru located in another directory change all the entries to match your setup.
- Now run **fixpbr.bat** and all the PBR media will be converted.

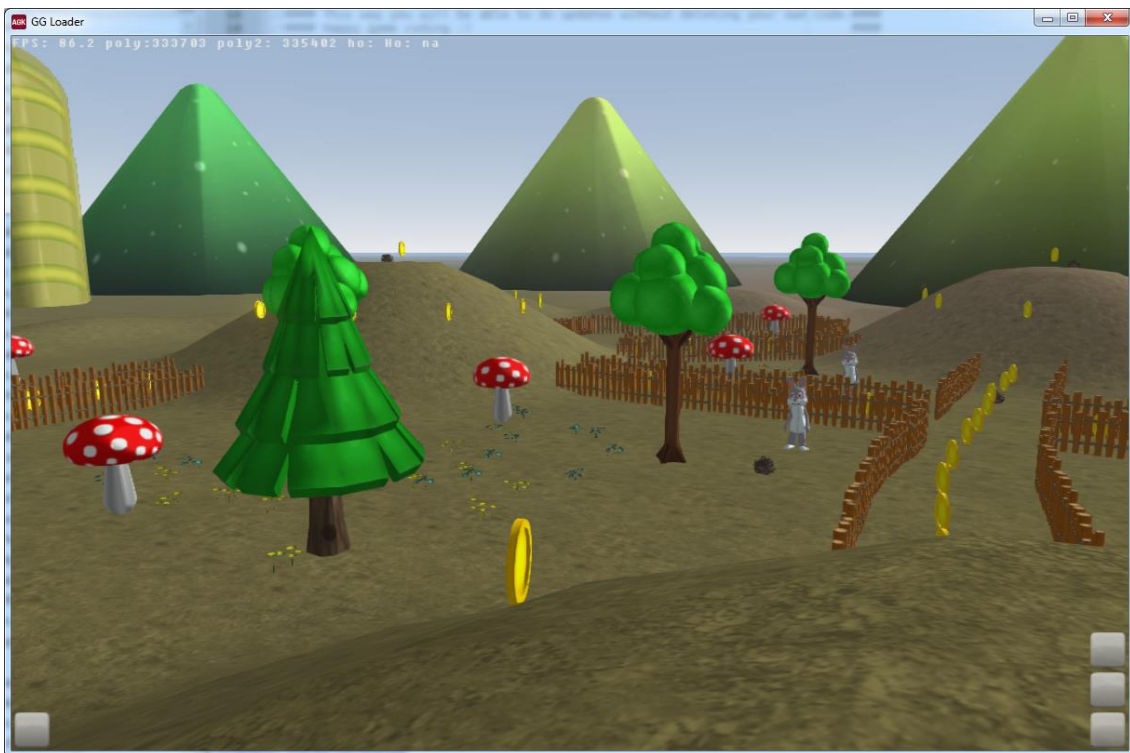
The program will convert all .dbo objects that do not have a .x version so that they can be used by AppGameKit, it will also allow you to convert EBE (Easy Building Editor) structures into .x for use in AppGameKit and / or AppGameKit - GameGuru Loader.

Note that at the time of writing (Jan 5th 2017) only the latest Public Beta version of GameGuru features these new PBR assets. So if you're running the release version then you will not need this yet.

Game Guru Loader – AppGameKit Source Code

Once the conversion process is complete, you are ready to load and test your level in AppGameKit! Open the AppGameKit IDE and load the GGLoader project that came with this DLC. Compile and run. The first run can take some time to start, and you may notice the project window flicker for a while, whilst all the required media from the freshly created entitybank folder is extracted and copied, and whilst terrain maps are built etc. This is all handled within the AppGameKit program files. After a short delay, your level should appear, ready to play, with a default set of buttons and WASD controls enabled. Wander around your creation in awe, and look up at the sun to see the lovely lens-flare in action. :)

First run of the Cartoon Antics level:



Broadcasting Your Level

To broadcast your creation to mobile, or to be able to export your creation to an AppGameKit project, you need to tweak a little bit of code within the *main.agc* file, and make sure all the required media for your level is included in your project media folder.

So far, AppGameKit has been loading the required media from the AppData temporary location. So, using Windows Explorer, copy the following files from your GGLoader AppData folder:

```
lightmap2048.png  
baked-m.png  
baked-vegmask.png
```

to:

```
ggloader/media/gameguru (where ggloader is your GGLoader project)
```

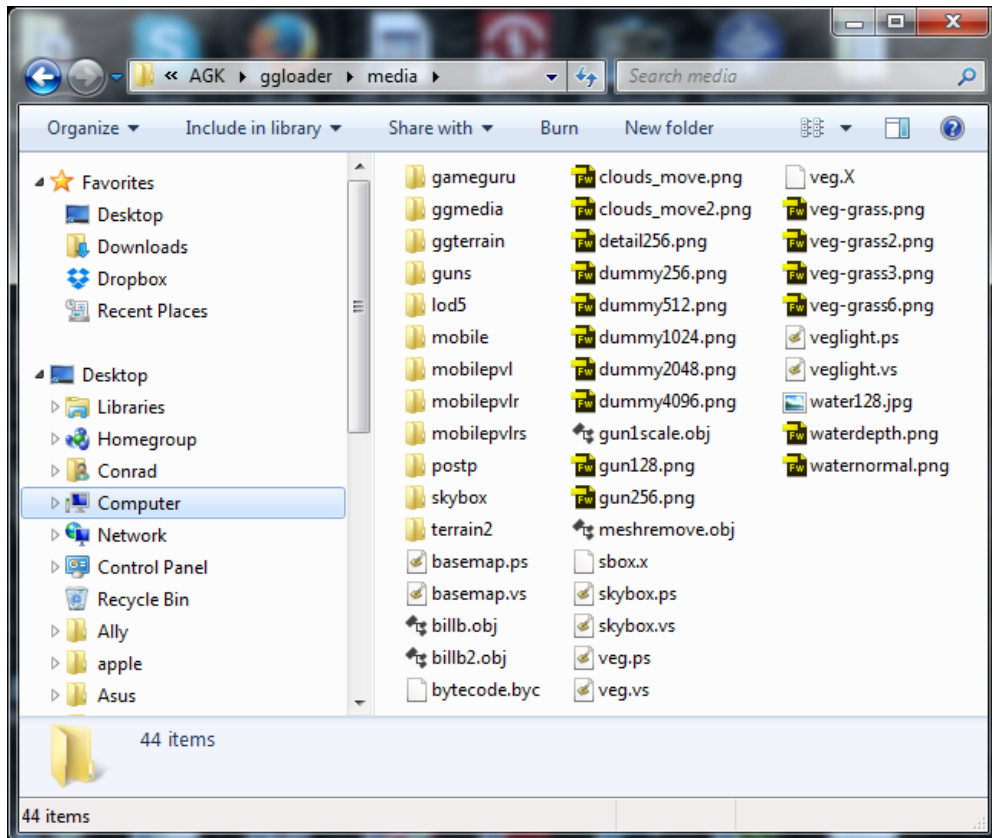
Notes:

- *You only need lightmap2048.png if your using the "generatelightmap = 1" option in GameGuru Loader*
- *When you use mobileversion = 1, baked-m.png will be called "baked-m_mobile.png"*

Next copy the entire ggmedia folder from your GGLoader AppData folder to:

```
/ggloader/media
```


Your ggloder project media folder should now look like this:

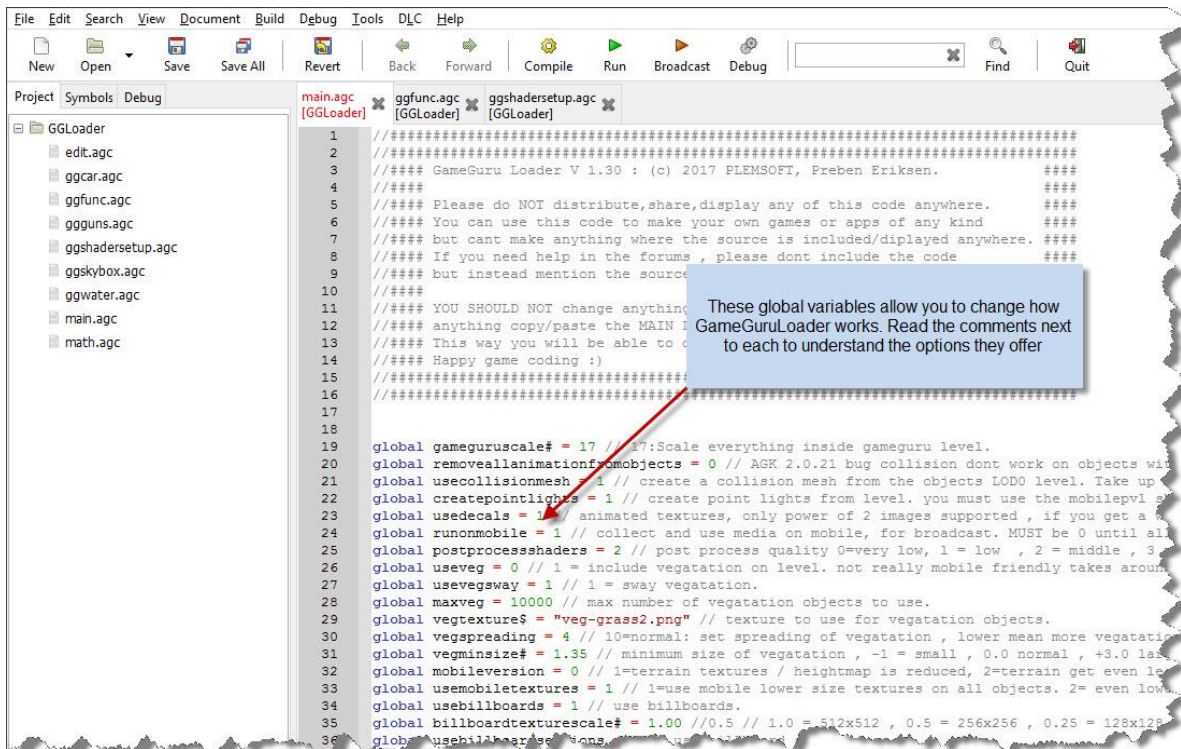


Within the ggloder main.agc file, change line 24 from:

```
global runonmobile = 0
```

to

```
global runonmobile = 1
```



This indicates to the program that you have successfully completed copying the required media over to your project and are prepared to test on a mobile device. You're then ready to hit broadcast!

Fire up the AppGameKit Player on the mobile device that you want to test with. Depending on the size of your level, it may take a while to transfer the media over, and then load the media, but you should end up with your GameGuru level running on your mobile device within a few seconds.

Here's an actual screenshot of the Cartoon Antics level running on an Android mobile phone via Broadcast:



There are many advanced features and options to tweak within the ggloader source-code, however this is enough to get you started for now. Please visit the AppGameKit forums for help and support on the more advanced options. Here are some other sources of help and advice you might want to check out;

- [The GameGuru Loader Forum Thread](#)
- [The AppGameKit Chat Board](#)
- [Preben's YouTube video showing the whole process](#)

Happy level building!