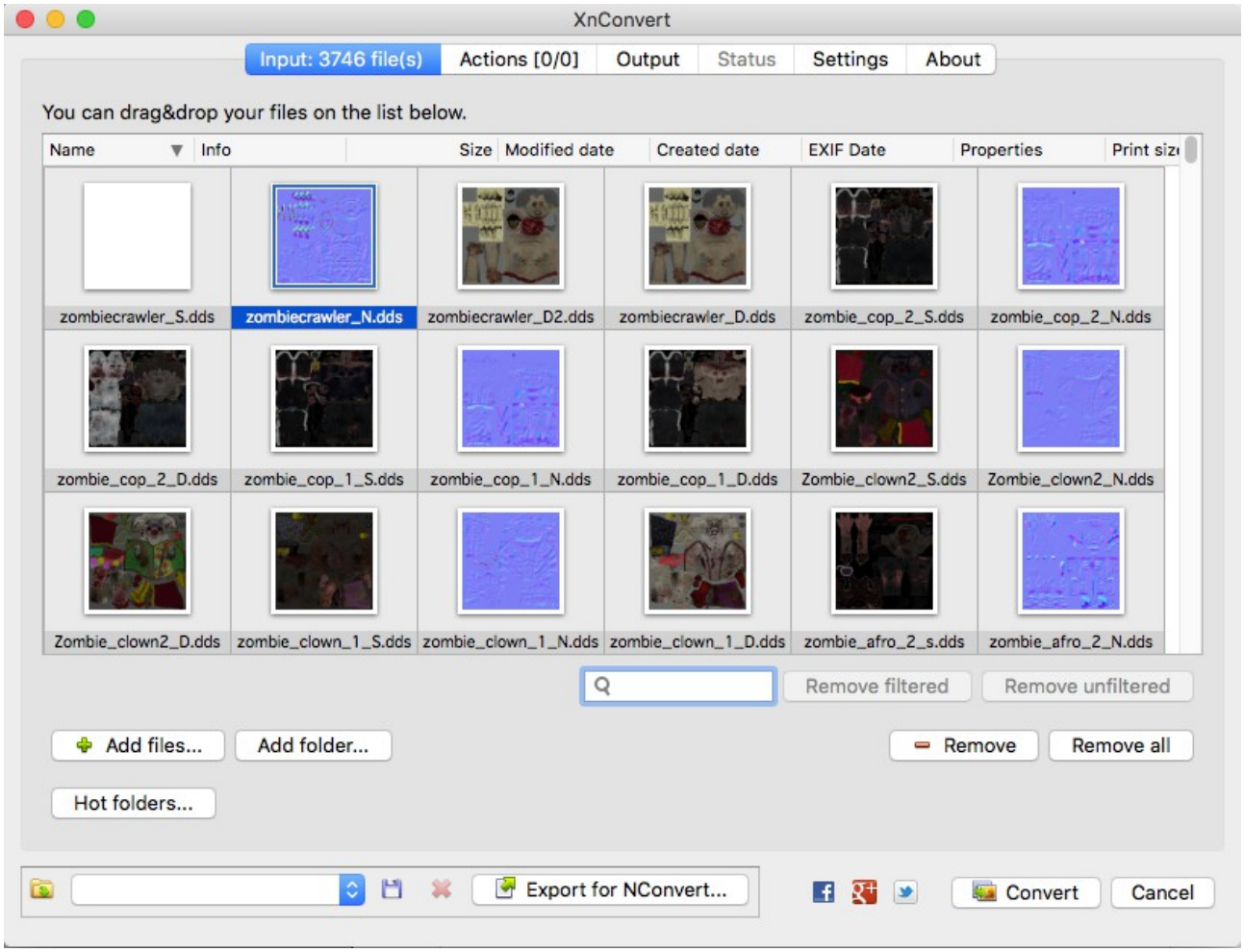
GG Loader V 1.0 (c) 2016 PLEMSOFT, Preben Eriksen.

PREPARE THE MEDIA:
In order to use GG Loader you must first prepare the GameGuru media for use in AGK.
You only need to do this the fist time you use GG Loader.

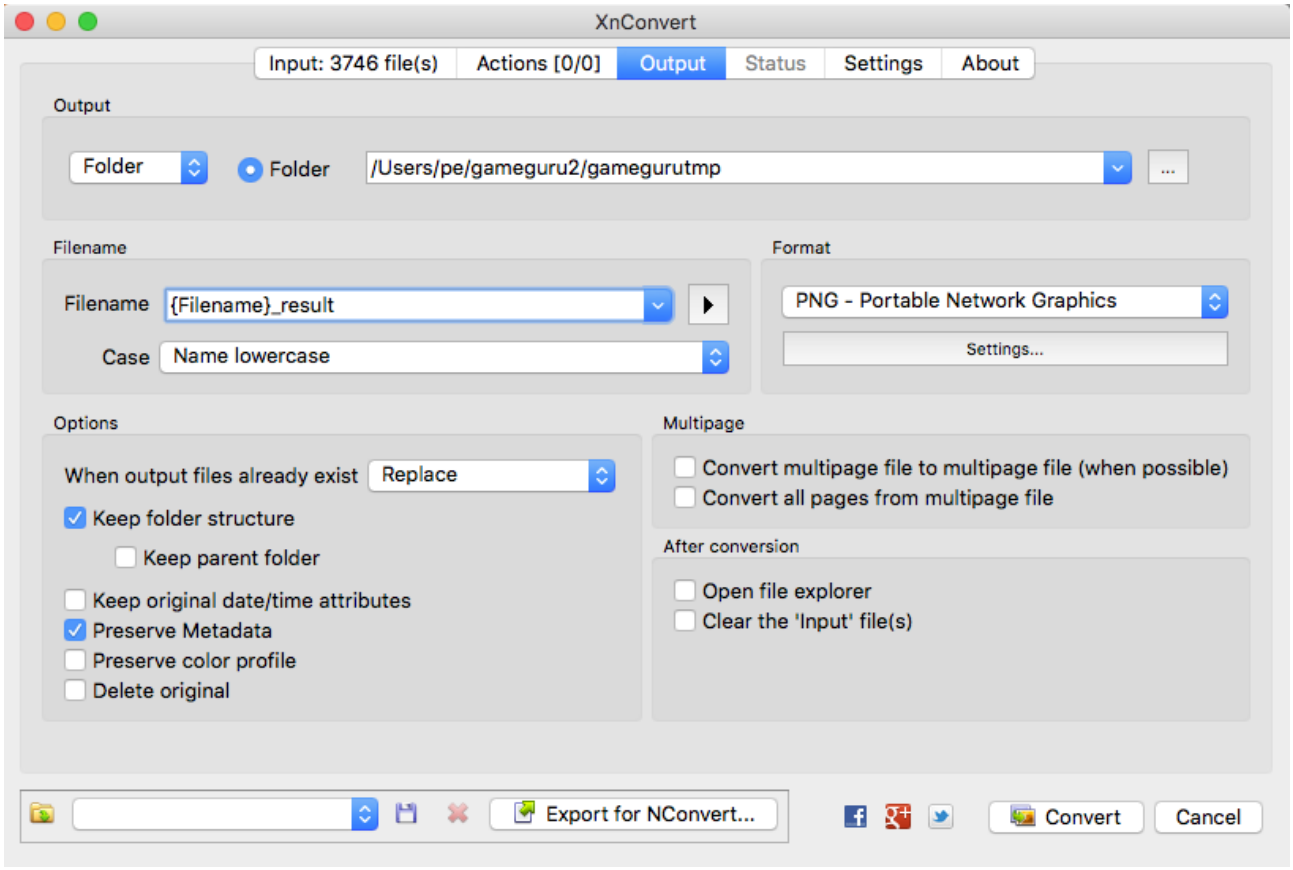
To do this you must convert ALL dds texture to png format.
you can use a program like http://www.xnview.com/en/xnconvert/ to batch convert everything. by default xnconvert add "_result" to the texture names so we use this, so when you convert "home_d.dds" it should be called "home_d_result.png". To use mobile friendly textures you also convert everything and call them "_result_mobile" in xnconvert, and remember to set xnconvert to "resize" the texture to 50%. , if you need even smaller textures for old mobiles , you could also convert everything and resize to 25%, call them "_result_mobile2", if you need to use these smaller textures change this line in "ggfunc.agc" from:
ggmediatexturem$[ml] = ReplaceString(ggmediatexture$[ml] ,"_result.png", "_result_mobile.png" , -1)
to:
ggmediatexturem$[ml] = ReplaceString(ggmediatexture$[ml] ,"_result.png", "_result_mobile2.png" , -1)

Here is some screenshots to illustrate the process:

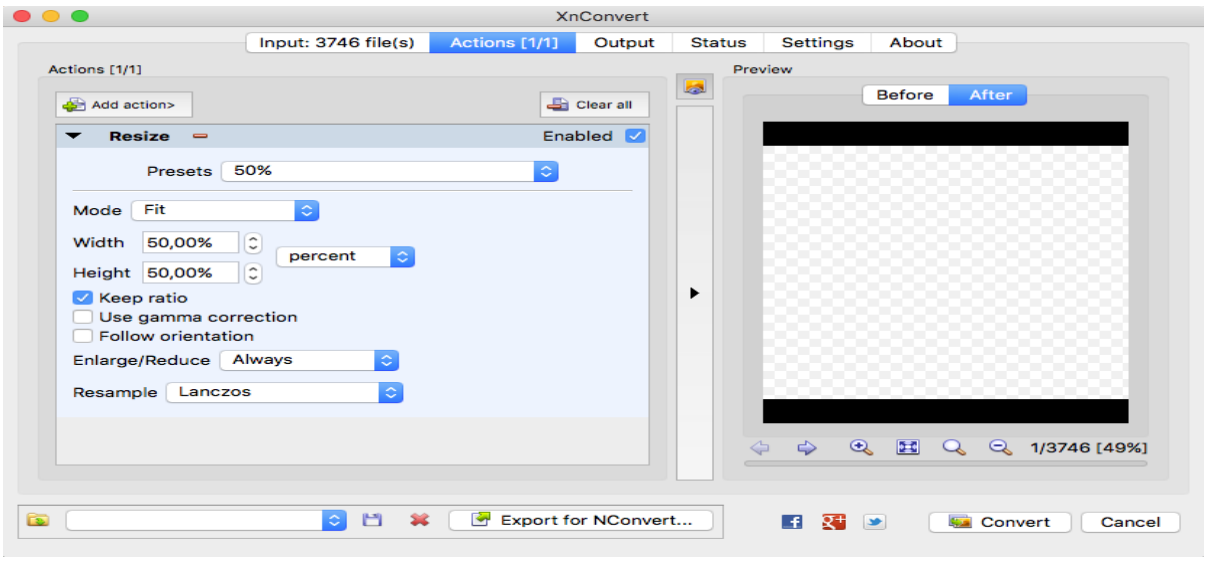
Use "Add folder ..." , Select the GameGuru root folder. Depending of how many gg packs you have installed it should display around 15.000 files. In the "Quick search" field enter ".dds" and click "Remove unfiltered", This should leave you with around 3500+ dds files:



Click the “Output” tab at the top.
Select: Output = Folder , in the “folder” field , select a new folder call it gamegurutmp.
Under “Filename” it should say: “{Filename}_result”.
Under Case: select “Name lowercase”.
Under Format: select “PNG”
When output files already exist: select “Replace”
Make a checkmark at “Keep folder structure” and “Preserve Metadata”.
This is how it should look:



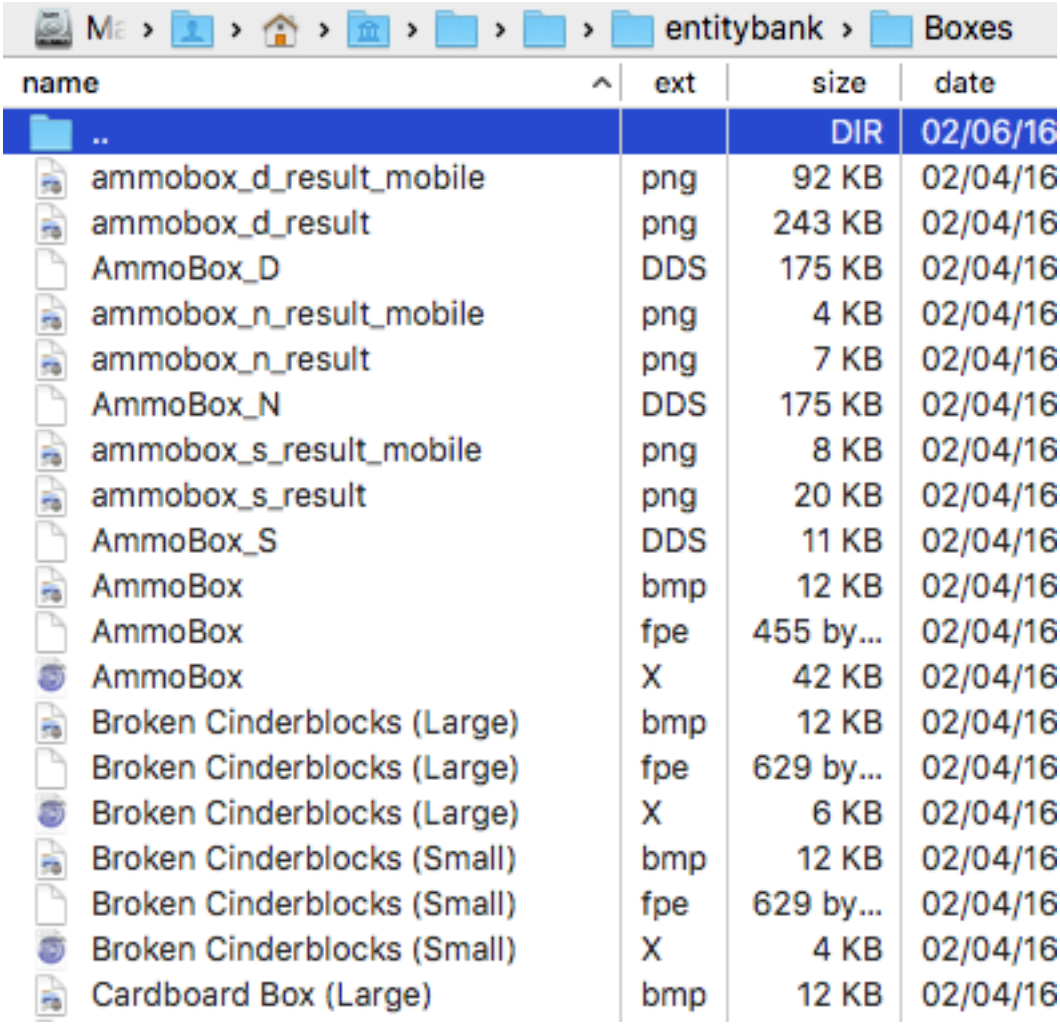
Now just press “Convert” , please be patience it will take some time to convert everything :)
When it finish DONT exit , click the “Actions (0/0)” tab at the top:
From here click “Add action”-> “Image” -> “Resize”. In the Presets select “50%” :



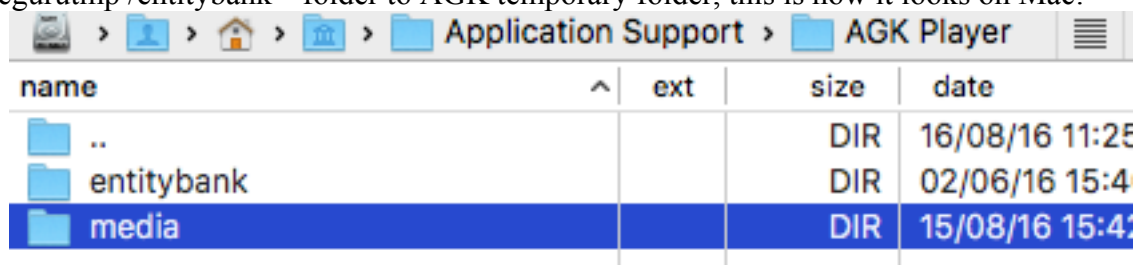
Now press the “Output” tab again at the top and change the filename field to “{Filename}_result_mobile” , click on “Convert” again and let it finish.
If you also like to generate even smaller textures, repeat the process click “Actions (1/1) set the “Presets” to “25%” , click “Output” and change the “filename” files to “{Filename}_result_mobile2” and click convert.

When everything is finish you should end up with a folder “gamegurutmp“ this folder contain all the dds textures in png format , and in different sizes.

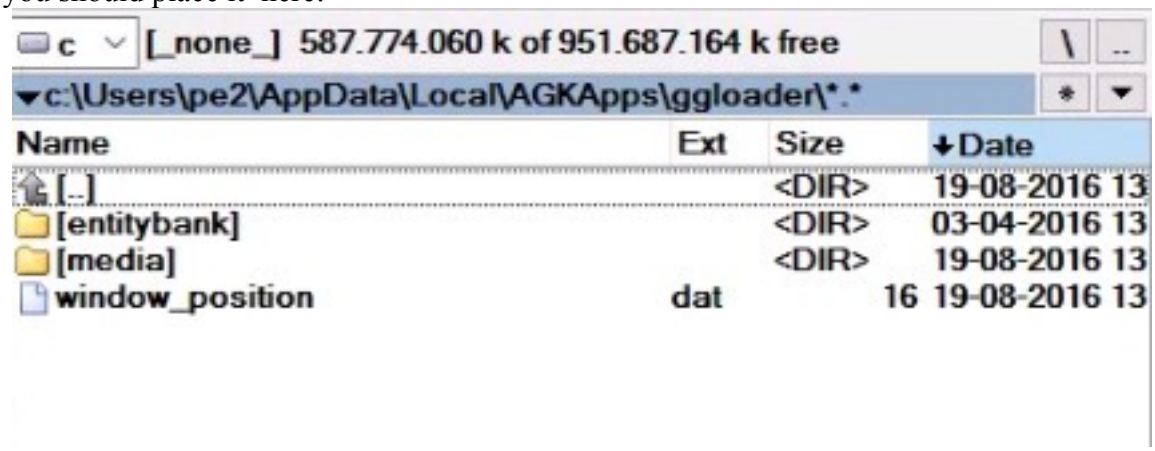
GG Loader only use the “entitybank” folder , but its still not complete, we also need ALL the objects and fpe files. So you must copy from the original GameGuru folder , EVERYTHING inside the “entitybank” folder into the new “gamegurutmp/entitybank” folder, so you end up with a “gamegurutmp/entitybank” folder that contain everything like this:



Before GG Loader can access the “entitybank” it need to be available to AGK, the default location i use for the “entitybank” folder is in the root of the AGK temporary folder , so move the new “gamegurutmp /entitybank” folder to AGK temporary folder, this is how it looks on Mac:



On windows this folder must be located inside the AGK temporary folder for the project, so in this case you should place it here:



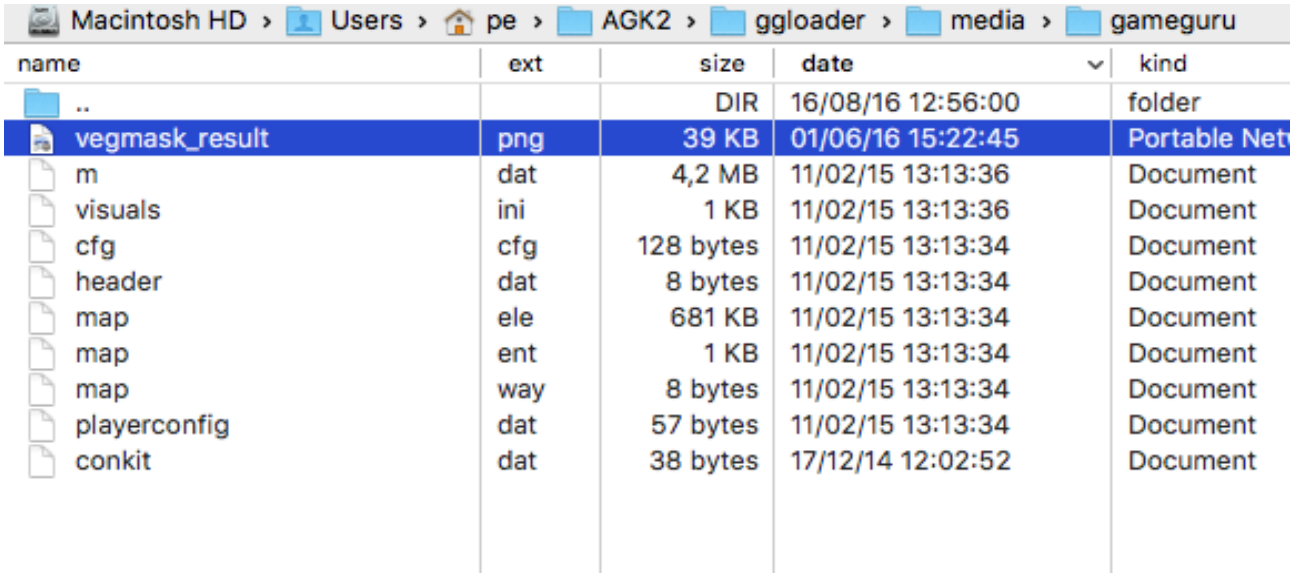
ggloader: is the AGK project name.
GG Loader will use this folder to collect all the unique media you use in you levels.

The media is now ready to be used by GG Loader , you should keep the “gamegurutmp” folder ,as you will be able to use the Terrain textures , Skybox , Guns etc.. by copying the textures you need to the AGK project media folder , you find more info about this in the source code.

COPY YOUR GameGuru LEVEL TO AGK “GG Loader”:

In your original GameGuru root folder you find your levels under “mapbank” , when you browse this folder you see a lot of .fpm files , locate your level and copy “yourlevel.fpm” to a temp folder, rename this file to “yourlevel.zip”. In the “GG Loader” AGK project folder under media you find a folder called “gameguru” , unzip all files from “yourlevel.zip” to this folder the zip password is “mypassword”, convert “vegmask.dds” to “vegmask_result.png” don't resize it keep it at 2048x2048. Thats it your now ready to try your GameGuru level in AGK.
To reduce size you can remove these files from the folder:
/Users/pe/AGK2/ggloader/media/gameguru/vegmaskgrass.dat
/Users/pe/AGK2/ggloader/media/gameguru/watermask.dds
/Users/pe/AGK2/ggloader/media/gameguru/vegmask.dds
They are not needed anymore.

So you end up with a folder looking something like this:



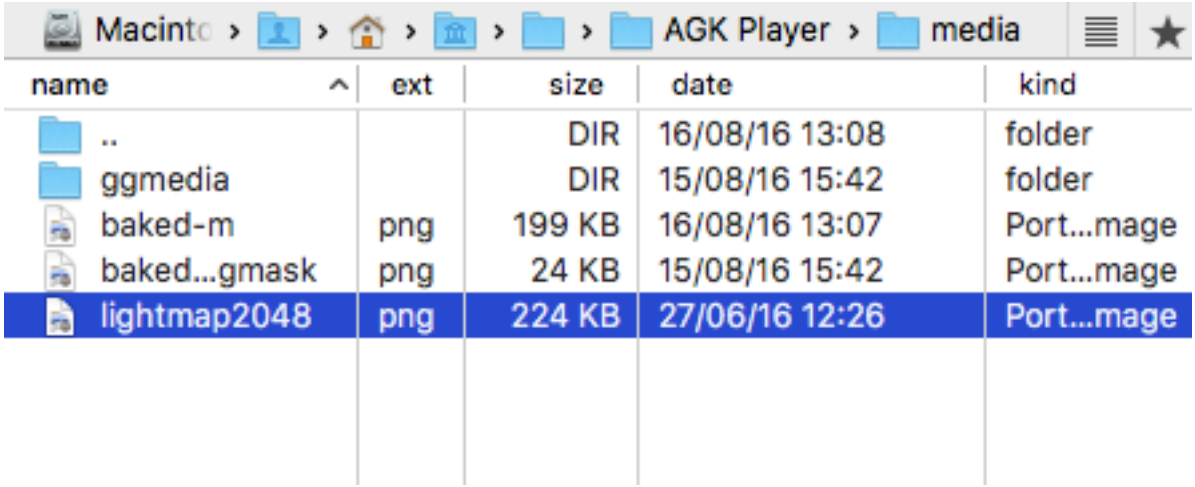
So when you make changes to your GG level this is the process you need to get it to AGK:

1. Rename your GG “yourlevel.fpe” file to “yourlevel.zip”
2. unzip all files from “yourlevel.zip” into the GG Loader folder “gameguru” , the zip password is “mypassword”.
3. inside the “gameguru” folder convert vegmask.dds to vegmask_result.png (keep it at 2048x2048).

FIRST TIMES YOU RUN YOUR LEVEL IN AGK:

The first time you run you GG level in AGK , GG Loader will collect all the unique media used and generate all the billboards / shadow light maps (This can take some minutes) that you need to run your level. It will take more time the fist time you run your level then normal.
NOTE: The billboard creation will flash your window, but this is ONLY the first time you run your level , the next time they are already generated and ready for use.

Depending of what settings you use in the GG Loader code , you will end up with some files in the AGK temporary folder under “media” , that looks something like this:



The folder “ggmedia” contain ALL media used in your level.

The files:

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

can be copied into the “/ggloader/media/gameguru/” folder to prevent GG Loader from generating these again the next time you run your level , so it will be mush faster. If you are not publishing your game , you should only copy the lightmap2048.png file, as this can take up to 10 minutes to generate , but the heightmap texture , terrain splatmap texture only take a few sec. So its fine that they are generated each time you run/test your level .

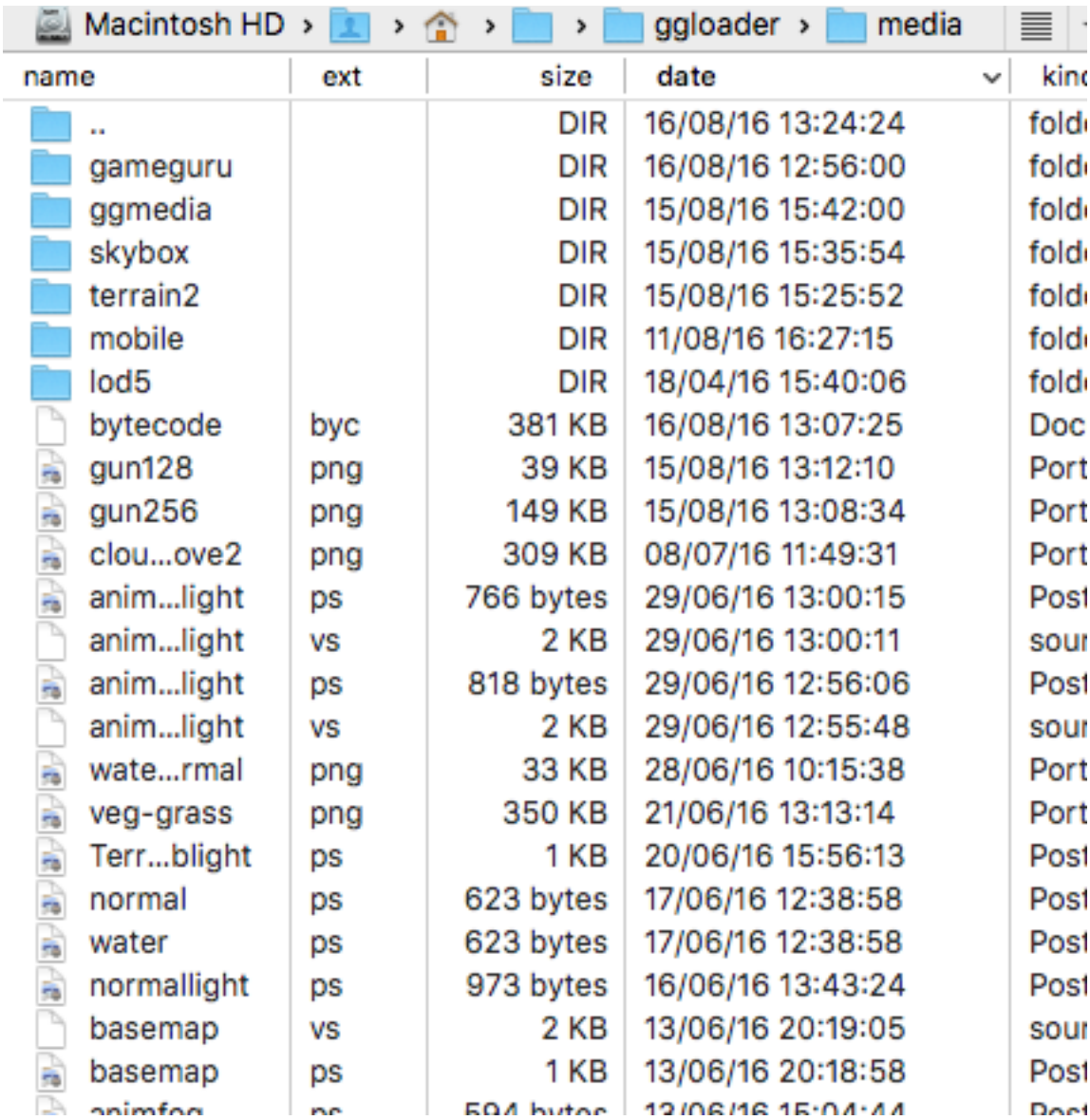
READY TO BROADCAST, PUBLISH YOUR GAME:

To test your level using AGK “broadcast” or before you publish anything all the media must be in your GG Loader AGK project media folder, so copy:

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

To:
“/ggloader/media/gameguru/”

And copy the “ggmedia” folder to “/ggloader/media/ggmedia” , so you end up with:



You have now collected all the media needed, and your ready to broadcast/publish your game.

When your coding your game and just testing you don't need to do this, ONLY when you want to broadcast (and made changes to the media), or are ready to publish.

CODE SETTINGS:

Here are some of the main settings your able to use (from main.agc):

global gameguruscale# = 17 // Scale everything inside gameguru level.
global useveg = 0 // 1 = include vegatation on level. not really mobile friendly takes around 20 sec on old mobile devices to generate.
global maxveg = 10000 // max number of vegatation objects to use.
global vegtexture$ = "veg-grass.png" // texture to use for vegatation objects.
global vegspreading = 8 // 10=normal: set spreading of vegatation , lower mean more vegatation.
global mobileversion = 0 // 1=terrain textures / heightmap / skybox use lower size textures.
global usemobiletextures = 1 // 1=use mobile lower size textures on all objects.
global usebillboards = 0 // use billboards.
global usebillboardsections = 1 // use billboard sections.

```
global usebillboardfade = 1 // fade billboard to normal object
global usealphamask = 0 // use alpha mask on all transparent objects , workaround.
global loadcharaters = 0 // load in gameguru charaters and animate.
global maxobjects = 3000 // 3000: max gameguru objects.
global maxuniquemedia = 130 // 130: max gameguru .fpe object to prepare.
global runonmobile = 1 // collect and use media on mobile, for broadcast. MUST be 0 until all ggmedia is collected. 1 also print load progress.
global uselodstages = 0 // use lod stages inside gameguru objects.
global useSetObjectScreenCulling = 1
global includemovingsky = 1 // include a moving sky
global includeskybox = 1 // include a  skybox
global skybox$ = "default" // default
global skySpeed# = 15.0 // how fast should the moving sky move.
global SkyboxRotateToFitSun# = -48.0 // rotate skybox so sun fit the shadows and light position
global WaterSpeed# = 0.00010 // how fast should the water move.
global distclip# = 120 // set later.
global billboarddist# = 100 // set later
global billboardfadedist# = 100 // set later
global billboardsections# = 100 // set later
global ggpathe$ = "/entitybank/" // path to gameguru media , objects and textures.
global ggpathe2$ = "/entitybank" // copy .fpe files from.
global ggmdiapath$ = "/media/ggmedia/"  // media objects,textures destination folder, to collect media for use on mobile.
global ggmdiapath2$ = "/media/ggmedia"  // used for .fpe files destination folder.
global ggdefaultground$ = "default" // terrain texture combinations, HOW terrain splatmap is combined, default should do in most cases.
global basemaponly = 0 // 1 = generate a basemap and use the basemap terrain shader, mush faster on mobiles.
global basemapsize = 1024 // texture size of basemap.
global lightmapsize = 2048 // combined light and shadow texture size.
global generatelightmap = 1 // generate and use shadow lightmaps yes=1
global excludeterrainshadows = 1 //dont include terrain in shadow mapping
global uselightmaponstatic = 0 // use lightmap on static objects. if 0 only terrain get the lightmap
global terrainshader = 1 // default terrain shader.
global includenormalmap = 1 // include normal mapping and specular mapping
global normalshaderpath$ = "mobile/" // path to mobile friendly normal,specular mapping shaders.
global terrainonlyusenormalspec = 0 // 1= only terrain gets normal and spec.
global terraindistancedetails = 1 // improve terrain distance quality.
global disablerallglass = 0 // 1=disable non alpha mask transparent objects like glass, there can be some depth problems.
global usebuildinshader = 0 // 1 = use agk built in shaders. ONLY on gg objects , not water,terrain.
global screensizex = 1280 // use screen width
global screensizey = 800 // use screen height.
global vegdist# = 0 // set later.
global fogR# = 100 // fog color R ( 103 fit gg sblue skybox )
global fogG# = 100 // fog color G ( 93 fit gg sblue skybox )
global fogB# = 101 // fog color B ( 72 fit gg sblue skybox )
global fogDist# = 0 // set later.
global fogDists# = 0 // set later.
fogDist# = 25000/gameguruscale# // fog distance.
fogDists# = fogDist# + (2560/gameguruscale#) // fog distance and add a bit so we dont hide skybox.
```

distclip# = fogDists# // disabled all objects at this distance.
vegdist# = 3000/gameguruscale# // when should we disable vegatation.
global waterlevel#
waterlevel# = 500.0/gameguruscale# // set where you want the water level.
global playerheight#
playerheight# = 38/gameguruscale# // Define the player height , will ALSO change collision checks on player.

They settings are self explained in the comments, there are other minor setting in the other source files.

These setting can be combined in any way you like (well i think i have tested most combinations) , to define how your level is going to be displayed, here is the source files used: