Importing Textures, Sounds and Static Meshes in AA285 Editor

(by OBAN)

Since the import option is not present in AAO 285 Editor, we must import Textures, Sounds and Static Meshes using another way. In this case we are going to do it using commands.

Unfortunately the option to input commands in the Editor is not present. But fortunately the Editor allows others ways to execute commands.

But before using any command, we need to do some preparations.

1. Go to the “System” folder of the game and create a folder called “to_import”
2. inside the new folder create 3 folders “textures”, “sounds” and “static_meshes”

```
to_imports
          |_  textures
          |_  sounds
          |_  static_meshes
```

We will use these folders to store the objects we want to import.

Importing Textures

To start we need some texture to import. In this example I will use the following Image.

Photograph of my wall taken with my cellphone camera

The image is in .jpg format and its size is 4000x3000 pixels.

The first thing we need to do is to resize the image to a size the editor can accept.

The editor only accepts sizes that are a **power-of-two number** of pixels across, for example **64x128**, **256x256**, **512x512**, **32x512**… the maximum size you can use is **2048x2048**.

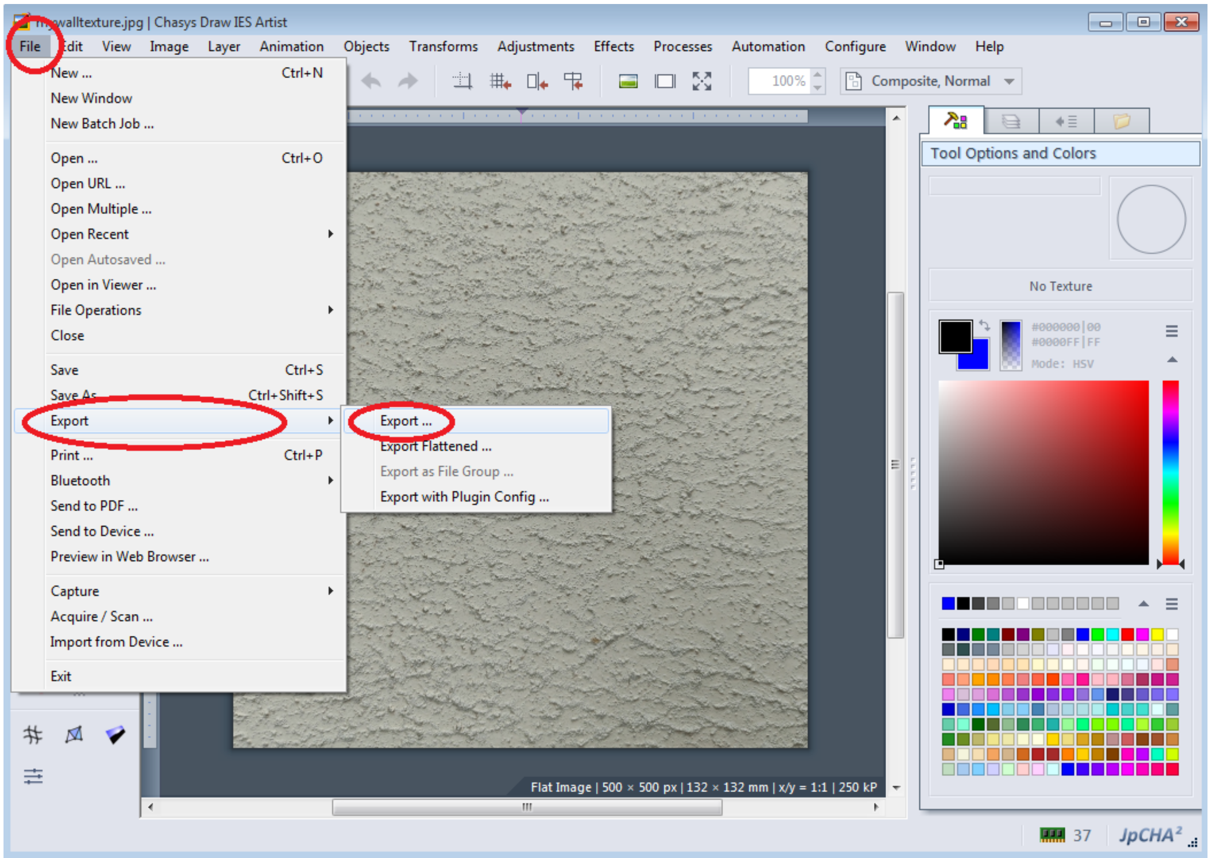
In this case I will crop the image to **3000x3000** pixels then I will resize the image to **512x512** pixels.(to do this I use a free software called fire Alpaca)

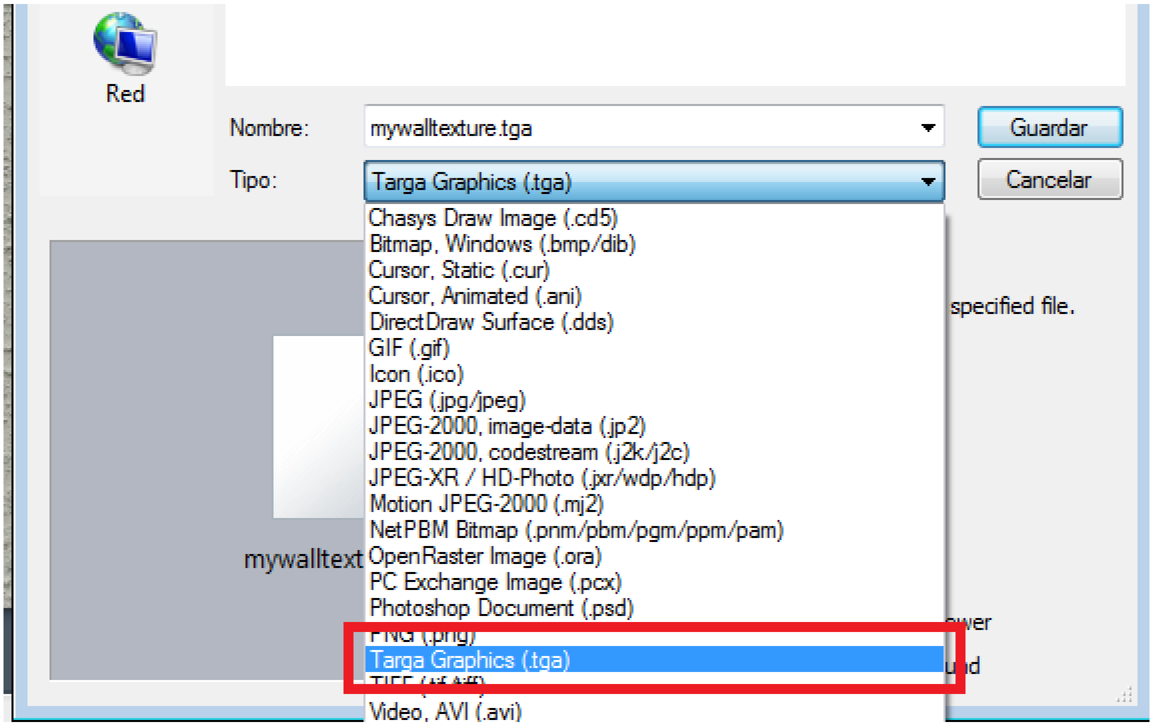
Image resized to 512x512 pixels in .jpg format

Now I have the image in a size the editor can accept, the next step is to convert the image to a format the editor accepts.

the editor accept the following formats: **.pcx**, **.bmp**, **targa**, **DXT** (this formats must be 8-bit indexed color)

I will convert the image to a .tga format(Targa). and for that I will use a software called “**Chasys Draw IES**”. I will open the image in this program and then use the export options then I will choose the .tga format and save the image as “mywalltexture.tga” as you can see in the following images.





Nombre: mywalltexture.tga

Tipo: Targa Graphics (.tga)

Saving the image as mywalltexture.tga

Now that I have my image in the correct format and size the next step is to copy the image to the folder "textures" we create at the beginning. (System/to_import/textures)

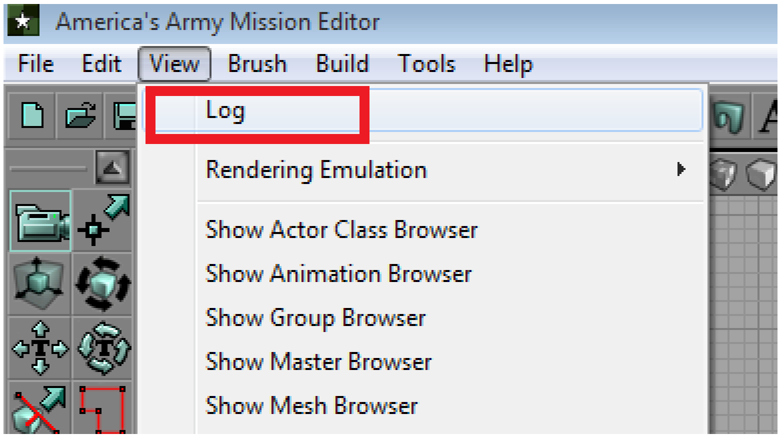
Now we need to go to the editor an execute the following command:

TEXTURE IMPORT NAME=NameForTextureToImportAs FILE=PATH\TO\SomeTexture.pcx GROUP=SomeGroup MIPS=OFF FLAGS=2 PALETTE=SomeTexPalette LODSET=2

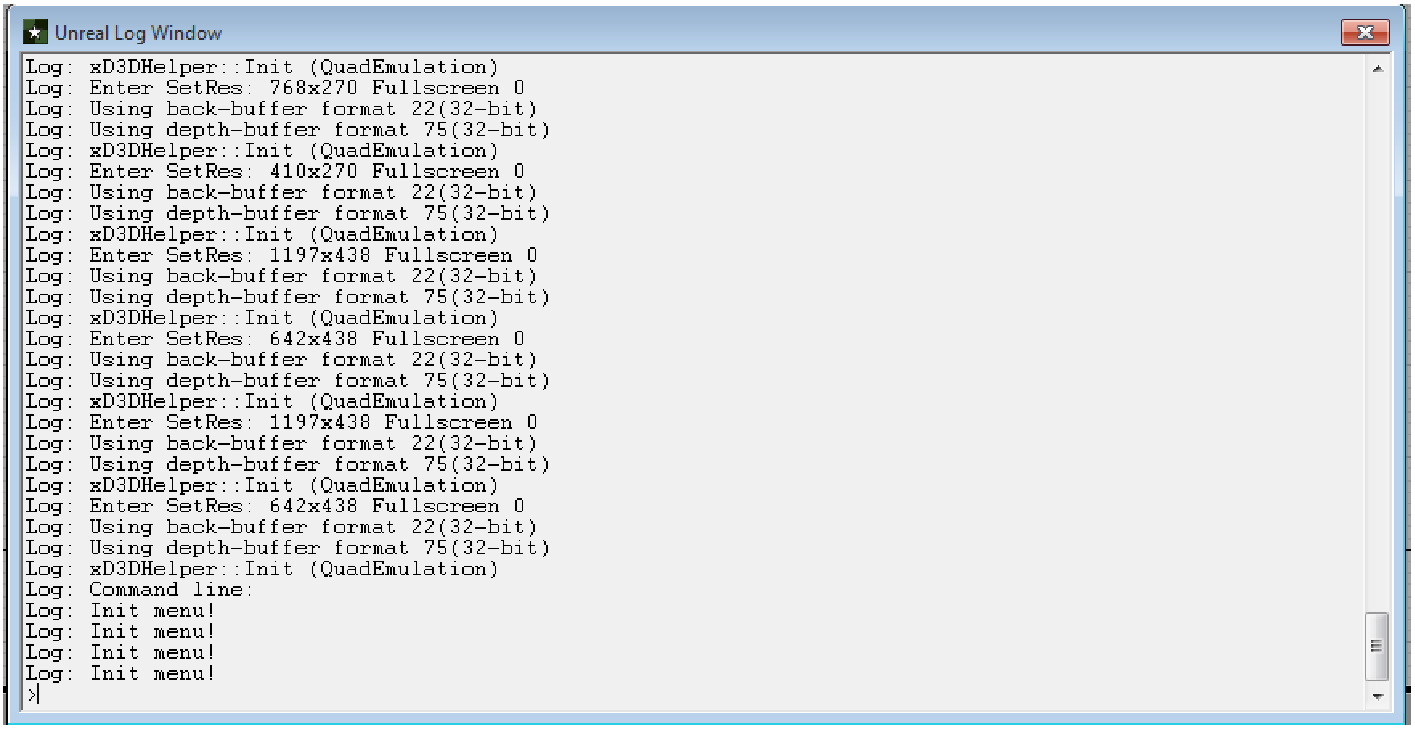
but first we need to modify some of the arguments of the command as follow:

TEXTURE IMPORT NAME=my_wall_texture FILE=to_import\textures\mywalltexture.tga PACKAGE=myLevel GROUP=Wall MIPS=OFF FLAGS=2 LODSET=2

Now in the Editor go to the **View** Menu and click on the "log" option.

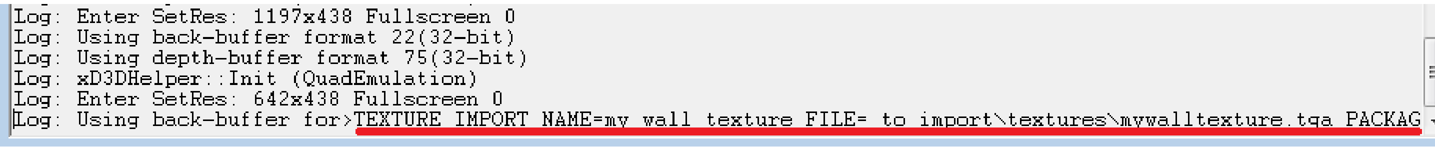


The “**Unreal log window**” will appear. Is through this window that we will be able to execute the command.



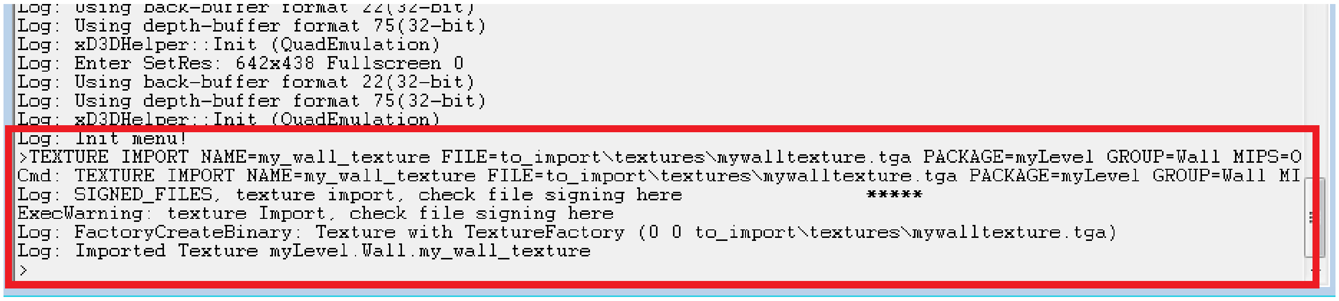
```
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 768x270 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 410x270 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 1197x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 642x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 1197x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 642x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Command line:
Log: Init menu!
Log: Init menu!
Log: Init menu!
Log: Init menu!
>
```

Now copy the command we edited earlier and paste in the log window. as you can see in the next image.



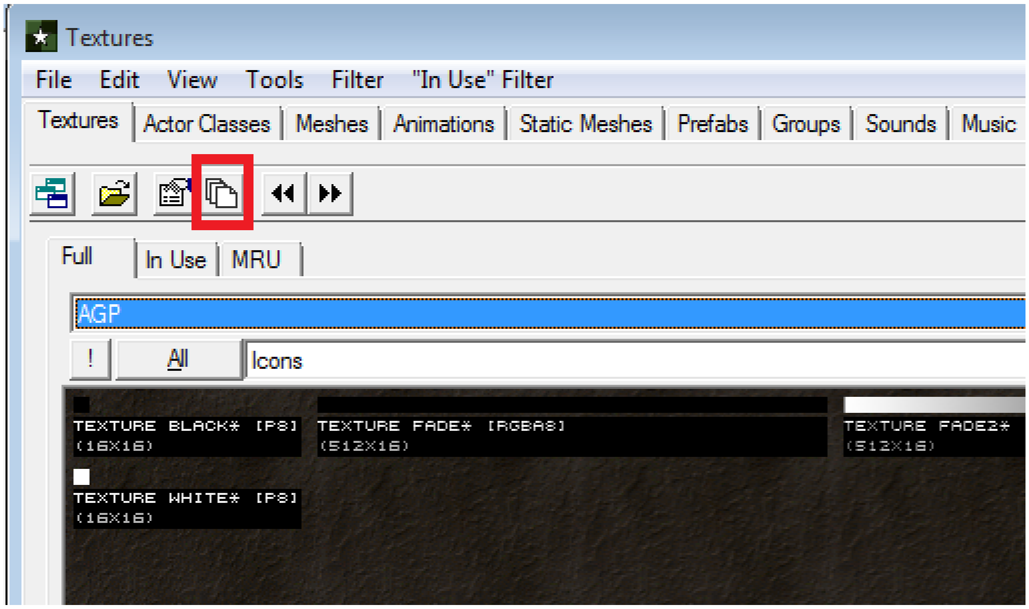
```
Log: Enter SetRes: 1197x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 642x438 Fullscreen 0
Log: Using back-buffer for>TEXTURE IMPORT NAME=my wall texture FILE= to import\textures\mywalltexture.tga PACKAG
```

the press the Enter Key to execute the command and if all went well, you should see the following messages in the log window

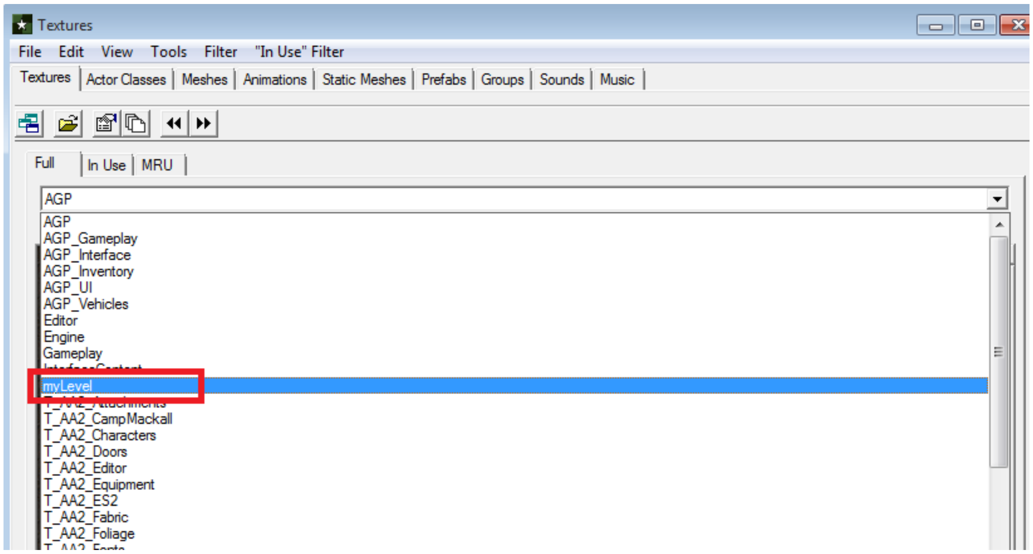


```
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Enter SetRes: 642x438 Fullscreen 0
Log: Using back-buffer format 22(32-bit)
Log: Using depth-buffer format 75(32-bit)
Log: xD3DHelper::Init (QuadEmulation)
Log: Init menu!
>TEXTURE IMPORT NAME=my_wall_texture FILE=to_import\textures\mywalltexture.tga PACKAGE=myLevel GROUP=Wall MIPS=O
Cmd: TEXTURE IMPORT NAME=my_wall_texture FILE=to_import\textures\mywalltexture.tga PACKAGE=myLevel GROUP=Wall MI
Log: SIGNED_FILES, texture import, check file signing here               *****
ExecWarning: texture Import, check file signing here
Log: FactoryCreateBinary: Texture with TextureFactory (0 0 to_import\textures\mywalltexture.tga)
Log: Imported Texture myLevel.Wall.my_wall_texture
>
```

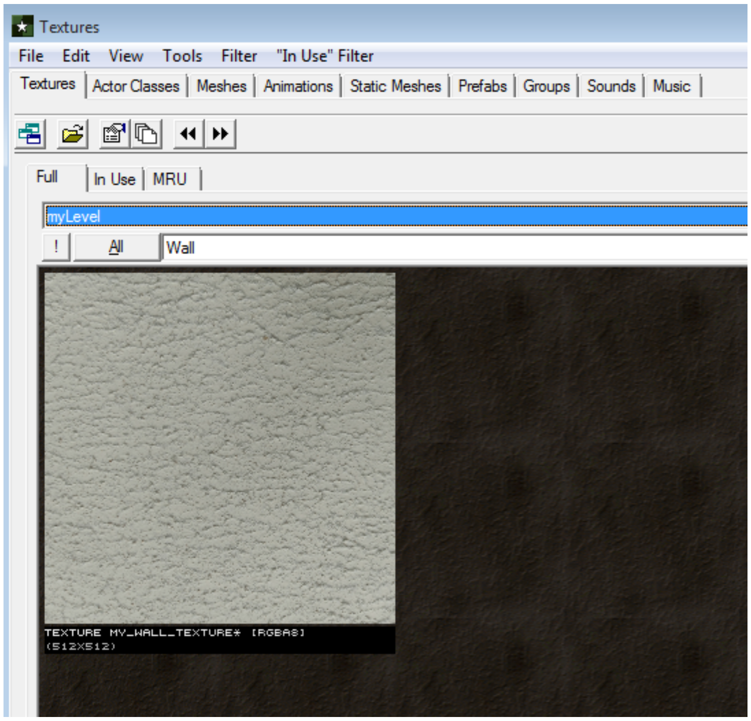
Now you can go to the texture browser a press the button **“Load Entire Package”** to refresh the list of packages, as you can see in the following image



then click on the combobox to see the list of packages, and select **“myLevel”** as you can see in the image



Now you will see the texture loaded in the editor ready to be used. as you can see in the following image.


Textures
File Edit View Tools Filter "In Use" Filter
Textures | Actor Classes | Meshes | Animations | Static Meshes | Prefabs | Groups | Sounds | Music
Full | In Use | MRU
myLevel
! All Wall

TEXTURE MY_WALL_TEXTURE* [RGBA8]
(512X512)

And this is all the procedure to import textures. If you want to import other textures you only need to follow the same procedure and edit the command arguments.

Importing Sounds

To import sounds, the first thing is to have a sound file that complies with the format required by the Unreal Engine.

The engine only accepts **uncompressed wave files**. No ADPCM! No MP3 wave! Only plain old PCM wave audio files.

The engine only accepts mono, so no stereo are allowed or other multi-channel format

The engine wants a standard sampling rate at 8 or 16 bit per sample. For UT you should stick to 11 or 22 kHz, newer games will definitely also handle 8, 16, 32, 44 and 48 kHz, but here you should find the lowest sampling rate that still sounds good. In most cases you will want to use 22 or 44 kHz at 16 bits/sample.

Once you have the audio file that meets the engine requirements, you just need to execute the following command to be able to import sounds to the editor.

```
AUDIO IMPORT NAME=NameForSoundToImportAs FILE=PATH\TO\Soundfile.wav
PACKAGE=SomePackage GROUP=SomeGroup
```

for example if you had the audio file **mysound.wav** that you want to import into the editor. first you have to copy the file to the sounds folder we created at the beginning, then you should edit the command as follows:

```
AUDIO IMPORT NAME=my_sound FILE=to_import\sounds\mysound.wav
PACKAGE=myLevel GROUP=mysound
```

and finally you must execute the command using the editor's log window as we saw when importing textures.

Importing Static Meshes

Before importing static meshes, you need to know that a static mesh is made up of the 3D model and the textures or texture of the model. This means that to import a static mesh you need to import textures and the 3D model. We have already seen how to import textures so that is not a problem, now we will see how to import the 3D model.

In order to import a 3D model, you must have a file containing the model that is in **.lwo(Lightwave)** or **.ase** format, which are the only two formats that the editor accepts.

then you have to use the following command to import the model:

```
NEW STATICMESH NAME=NameForStaticMesh FILE=PATH\TO\StaticMesh.ase
PACKAGE=SomePackage GROUP=TeamWeapons
```

let's see an example:

we have a static mesh made up of the following 3 files:

- truckmodel.ase (the 3D model)
- trucktexture.tga (texture of the model)
- truckwindows.tga (texture of the model)

The first thing is to copy these files to the folders we create at the beginning of this tutorial. truckmodel.ase to “static_meshes” folder, trucktexture.tga and truckwindows.tga to “textures” folder.

From experience I advise you to first import the textures. so you have to edit the command to import textures as follow

```
TEXTURE IMPORT NAME=truck_texture FILE=to_import\textures\trucktexture.tga
PACKAGE=myLevel GROUP=vehicles MIPS=OFF FLAGS=2 LODSET=2
```

```
TEXTURE IMPORT NAME=truck_windows FILE=to_import\textures\truckwindows.tga
PACKAGE=myLevel GROUP=vehicles MIPS=OFF FLAGS=2 LODSET=2
```

To import the 3D model you also have to edit the arguments of the command as follows:

```
NEW STATICMESH NAME=truck_model FILE=to_import\static_meshes\truckmodel.ase
PACKAGE=myPackage GROUP=vehicles
```

after importing all the necessary files you should be able to see the 3d model in the Static meshes explorer.

where to get static meshes to import into the game?

there are 2 options you can create them yourself in a 3D modeling program like 3d max or maya, or you can get them from other games.

If you are interested in getting static meshes from other games I recommend you use the “ue viewer” program also known as umodel. https://www.gildor.org/en/projects/umodel

Command to import textures in detail

Syntax:

```
TEXTURE IMPORT NAME=NameForTextureToImportAs FILE=PATH\TO\SomeTexture.pcx
GROUP=SomeGroup MIPS=OFF FLAGS=2 PALETTE=SomeTexPalette LODSET=2
```

NAME
(optional) Name the texture is imported as. Defaults to the file name without extension if omitted.
FILE
Path to the texture being imported. Paths are local to the root of the package directory.
GROUP
(optional) Name of the group the texture is classified under (used in, for example, the UEd Texture browser.)
MIPS
(optional) OFF or ON. Generate MipMaps.
FLAGS

(optional) Numerical value. Assigns the texture rendering style. Valid styles are:

- 1 Normal
- 2: 2-sided – isn't that masked? →Wormbo →Kirin Yes Wormbo it is → SabbathCat: In most UT code FLAGS=2 is commented as just "two sided".
- 3: Translucent 2-sided
- 4: Masked 2-sided
- 5: Modulated 2-sided

PALETTE
(optional) Color palette to use if not default.

LODSET
(optional) Numerical value. Level of detail. Homologous to the LODSet Texture property in UEd. Affects texture clarity as dictated by the user's settings.

- 0 no detail settings applied.
- 1: LODSET_World – affected by user's World Texture Detail setting.
- 2: LODSET_Skin (or LODSET_PlayerSkin for (UT2003)) – affected by user's Skin Detail setting.
- 3: (UT2003) LODSET_WeaponSkin – used for weapon textures
- 4: (UT2003) LODSET_Terrain – used for Terrain textures
- 5: (UT2003) LODSET_Interface – used for menu items like HUD and Player Portraits
- 6: (UT2003) LODSET_RenderMap – used for shaders
- 7: (UT2003) LODSET_LightMap – used for light maps
- 8: (SWAT4) LODSET_NormalMap

ALPHA
(UT2003) (optional) Binary value. Defines whether a texture is imported as a Texture with an Alpha Channel
UCLAMPMODE
(UT2003) (optional) can be CLAMP or WRAP, WRAP is default, so you don't have to add it then. I'm not totaly sure what is does, but it's needed for projector textures, and for team symbols.

VCLAMPMODE
(UT2003) (optional) see UCLAMPMODE.

DXT
(UT2003) (optional) One of: 1/3/5. DXT format to use when compressing the imported texture. Note that it's possible but usually not practical to include a PACKAGE=... parameter. There may also be other parameters, but the ones listed above should be all you ever need.