HammUEr Quick-start guide

by Joe Wintergreen (@joewintergreen)

Overview

HammUEr is a plugin for Unreal Engine 4 which allows you to import levels, textures, materials and in some cases models from older game engines like Source and Quake 1 and 2, which had better level design tools than Unreal does. This lets you make maps in Hammer, Trenchbroom, or your preferred BSP editing tool, and use them in your Unreal Engine games.

This documentation might seem intimidating at first, but you'll get the hang of it all pretty quickly.

Terms

- When I use the word "**Hammer**" in this document, it's probably interchangeable with whatever BSP level editor you use. Trenchbroom, Radiant, Worldcraft, whatever.
- When I use the word "**Materials**" in this document, it refers to **UE4 materials** or **material instances**. If I'm talking about Source engine materials (.vmt files) I'll say VMT. If I'm talking about Quake 1/2 or Half-Life 1, or Source Engine textures (.vtf files), I'll say "textures".

Getting Started With HammUEr

To install HammUEr, you just place the HammUEr folder in a folder called Plugins within your Unreal project folder, for instance UnrealProjects\MyAwesomeGame\Plugins. If you don't have a Plugins folder, just create one. When you open that project in Unreal, you should have a HammUEr icon in your toolbar. If you don't, open Edit -> Plugins and make sure HammUEr is enabled (it's under the NT Entertainment category).

Click the HammUEr button on the toolbar to bring up HammUEr.

Importing Textures/VMTs

If you're importing a level from an earlier engine, first you'll need to import their textures (and/or VMTs if it's a Source Engine level).

> ***If you already have all the materials you need in Unreal, skip this step.***
> ***If you don't need your level to be textured, you can skip this step.***
> ***If you're not working from Source Engine map files, ignore anything about VMTs.***

Select the tab of HammUEr labelled "**TextUEr**". This is the part of HammUEr that deals with textures and materials. Right now we're only interested in the "Import" section.

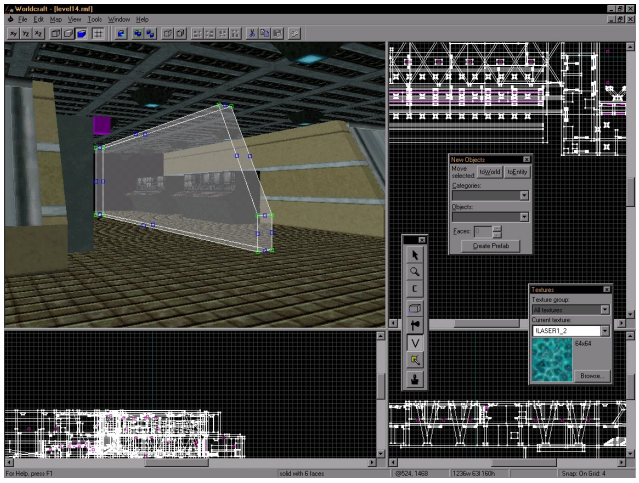
Click the "Choose source directory" button and browse to the folder where your textures or VMTs are kept. Textures can be image files or bundled up in a .WAD file as is common for Quake 1/2/Half-Life 1. In the "Import to" field, specify any folder name. This will be a subdirectory in your UE4 project's Content folder.

HammUEr will import all textures and create **material instances** for them (each texture will not get its own **material**, because that would be wasteful and unwieldy).

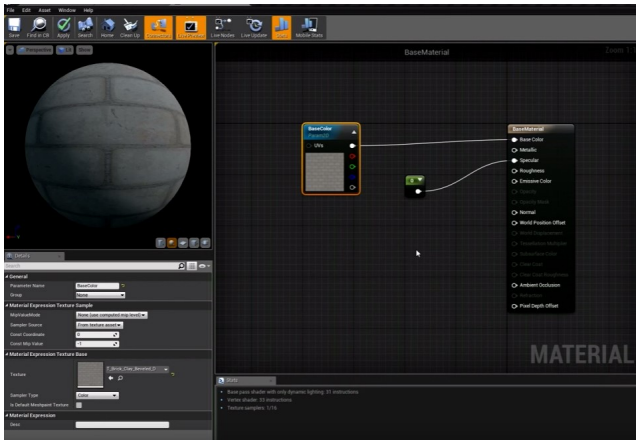
The material instances will be derived from the parent material you specify in the Source Material field. This must be a **material** with, at minimum, a texture parameter hooked up to BaseColor, which you will then specify in TextUEr as the Base Parameter. The material can be as complex as you like as long as it has that. If importing from VMTs, you can also specify texture parameter slots for the VMT-specified normal and specular maps.

Once you've specified your source material, parameters and source directory, click Import. Once the process finished you should have a lot of textures and material instances in subfolders under the one you specified.

Any textures/VMTs imported here will be remembered and automatically found by HammUEr when you import your levels.



Worldcraft circa 1996

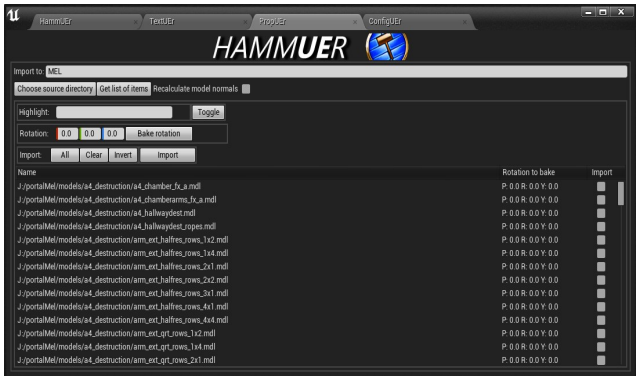


Bare minimum source material.

Material Instances for a bunch of Half-Life textures imported from halflife.wad.

Importing Props/MDLs

HammUEr also supports importing models from various Source versions via the PropUEr tab. Choose a source directory and click "Get list of items". This will create a multi-selectable list of all the models in the directory and subdirectories you've chosen, as shown. You have some options to finetune and bake in rotations for selected models, but in most cases, you will just want to click "All" and then "Import" to create Unreal versions of all the models it has found.



PropUEr model list.

Importing a level from Hammer

When you save a map from a Source Engine version of Hammer, it'll save as a **.vmf** file. If you're saving from a Quake editor you'll probably get a **.map**, and if you're saving from a Half-Life 1 engine it'll be an **.rmf**. To be able to use these, reopen the .rmf with a Source version of HammUEr and resave them to .vmf.

HammUEr also supports Doom3 **.map** and **.proc** (compiled meshes) files, for which the rest of this documentation will remain valid and the same.

In the HammUEr tab, click "Choose File". Browse to, and select, your map file, then click "Open File". HammUEr will search the file for references to textures/VMTs that it recognises (probably because you imported them earlier through TextUEr). You should see a list of all the textures used in your map, the width and height of each, a column of checkboxes called "ND", and a column of UE4 Material references.

Basically, for each row, the **texture** shown at left is going to be replaced with the **UE4 Material** specified in the dropdown on the right. The width and height must be correct, or your map won't be UV'd correctly once you import it. If HammUEr hasn't automatically found the appropriate materials to use, you can specify them here and the Width and Height will update accordingly. HammUEr will remember the materials you specify here for next time.

The ND column is important: it stands for NoDraw. Any texture with the ND box ticked will be skipped on import. You usually want to enable this for skyboxes, clip brushes, and some other things.

HammUEr material list.

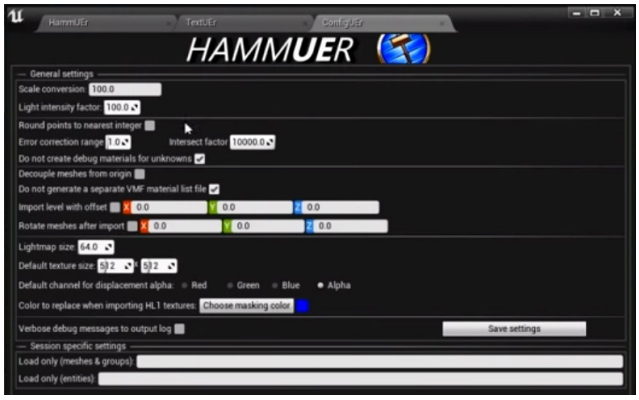
Once this list looks correct, and you've specified an import directory, hit "Go". All level geometry should be converted to meshes, imported to UE4, and placed in the level you have open.Lights will also be imported.

HammUEr imports each brush from the level as its own mesh, except for groups and brush entities. Each group and brush entity is imported as a single mesh. It's good to keep this in mind when building your maps. Elaborate structures like arches should always be grouped or made into entities so you don't end up with a single arch in Unreal being made out of a large number of meshes.

ConfigUEr

The ConfigUEr tab contains a bunch of important functions which are all fairly self-explanatory and have detailed tooltips - hover over them to see what they do. The most important is "Scale Conversion", which determines how large the level you import is going to be - 100cm = x Hammer units - and the best value is going to depend on the level you're importing and what game it was for. If it's a Half-Life level, around 37 is a good number.

For a more in-depth explanation of all the various options and possible problems you might run into, read the full documentation in the other file.



ConfigUEr