Unreal Editor

By Jared Bishop

Unreal Editor is a tool for laying out game maps and placing actors in them. It can also be used to create scene geometry, but the editor’s modeling tools are only useful for creating orthogonal worlds. Typically, you will build and export your scene geometry from 3D studio.

The best introduction to Unreal Ed is found here:

<http://www.planetunreal.com/unrealed/index.asp?action=spec_cat&cat_id=3>

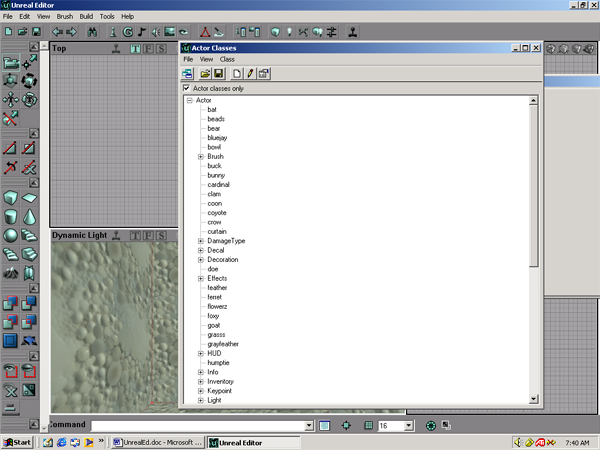
The 3 beginner tutorials are highly recommended. They will give you a basic understanding of the editor.

Geometry: Actors vs. Brushes:

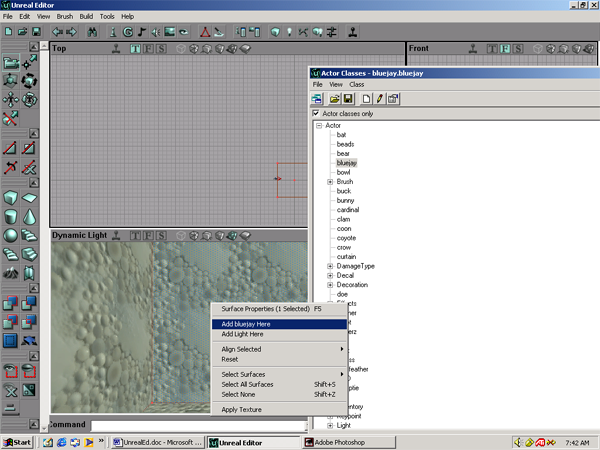
Scene geometry refers to anything you walk on, climb or run into. In other words, anything stationary with active collision detection. Scene geometry is created using “brushes” in UnrealEd—the geometry is subtracted, added, or intersected via boolean operations to the space of the map.

Animated geometry, or geometry whose properties can be set via scripting, are actors. The process for creating actors is described in the “Conversion Pipeline” portion of this document. Actors are added to the map using Unreal Ed. The process is as follows:

As long as the actor you want is listed in the “EditPackages” section of the UnrealTournament.ini file, it will show up in the actor class browser. Click on the actor in the browser.



Then, right click where you want to place the actor in the scene and click “add (some actor) here.”

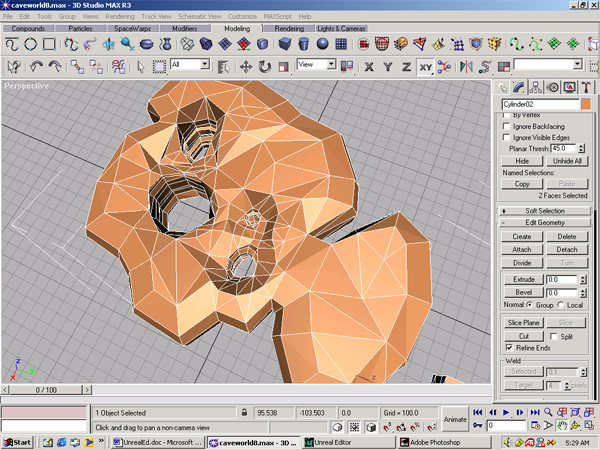


# Importing Scene Geometry into Unreal Editor from 3D Studio Max

Because UE will perform a boolean operation with it, your mesh must be seamless. It must be free of holes and overlapping polygons.

Below is a model of a cave about to be imported into Unreal Ed.

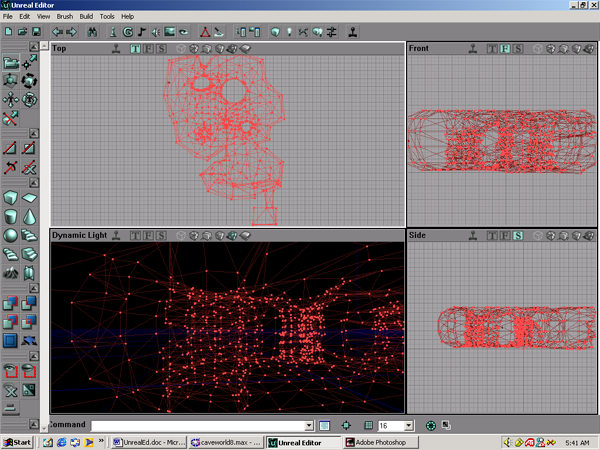
1. This brush will be *subtracted* in Unreal Ed. Therefore, the polygons in 3D studio must face the *reverse* direction they will wind up in the game. Also, Unreal Ed mirrors geometry about the Y-axis during the conversion process. This is obnoxious, but there is no way around it so you will have to mirror it in 3D studio before exporting.



2. Go to the file menu, and select Export. Choose AutoCAD .DXF file format. A “Save to Layers” dialog will open; select 1 layer.

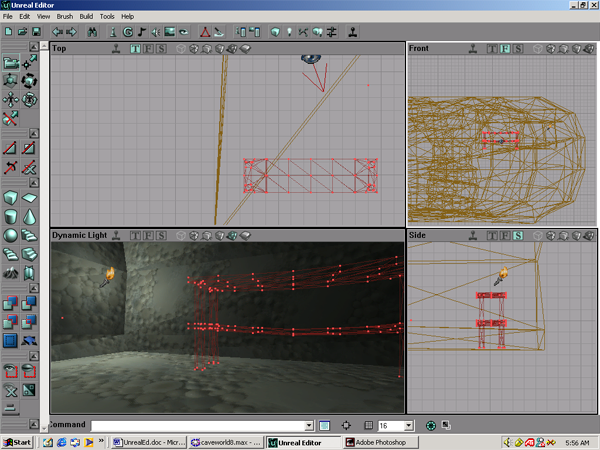
3. Open Unreal Editor. In the Brush menu, select Import and choose your .DXF file. In the import brush dialog that opens, just click OK.

4. Your .DXF file is converted to a brush, and you can move it around as you desire. Note in the screenshot how UE has mirrored the geometry.



5. Subtract the brush.

6. Now that you have carved out a space in Unreal Ed, anything inside it must be added rather than subtracted. In the screenshot below, I’m about to add a table to a room in the cave world.



In this case, the table is slightly above the floor so the brushes don’t intersect. This will ensure that the add operation goes smoothly. If the geometry of the cave and table did cross each other, the correct operation would have been deintersect.

Important things to keep in mind:

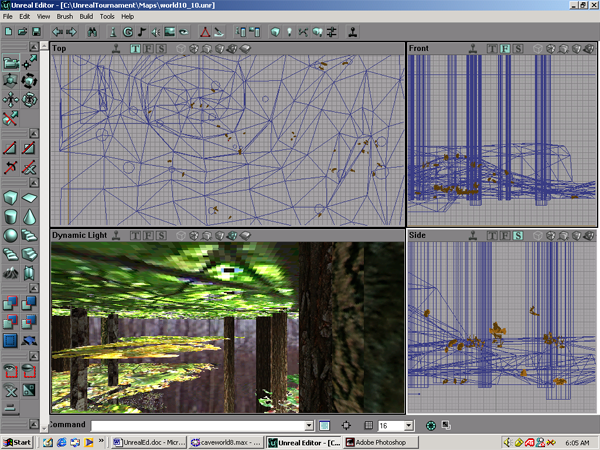
1.Keep your polygon count low. Although the UT engine can handle high polygon counts, it is easier to manage a simple mesh. There is less of a chance that you will create holes, overlapping polygons, or isolated vertices. The cave model in the example has 1643 vertices.

2. Unreal Ed will triangulate each polygon of the model. So try to model using only triangles and quads. For this reason, UE will have problems with things like the caps of cylinders, which are often single polygons consisting of 18 (or more).

3. Try not to use boolean operations when building scene geometry. This tends to create awkward geometry that the editor cannot triangulate. It also produces tiny (often invisible) holes and isolated vertices.

4. If you must boolean in 3D Studio, perform a weld (under edit geometry in vertex sub-selection mode) before trying to import into UnrealEd. Increase the weld threshold, press the “selected” button, and repeat until you get a message box that says “No vertices within weld threshold”.

5. Learn to use the CSG operations properly, rather than adding geometry ad nauseum. A good example of NOT using these operations is shown below:

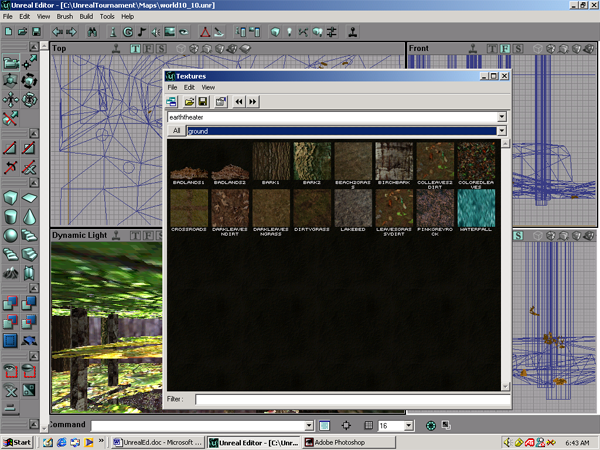


The trees were added (rather than deintersected) to this terrain such that they crossed the ground plane. Though it appeared to work for a while, this messy geometry caused Unreal Ed to blow holes in the map as it got more complicated.

6. If you’re building a 3D studio model for use as an Unreal brush, try importing and adding/subtracting it periodically to make sure it works, then save that version of the file.

# Bringing textures into Unreal

In Unreal Ed, textures are listed in the texture browser (shown below), organized into texture packages. The packages themselves are stored in the Textures folder (in the UT folder).



Button to open texture browser:



Unreal will support the following pixel sizes: 64X64, 128X128, 256X256, and 512X512. Anything other than that, like 256X255, will not import.

To import a texture:

1. You’ll probably want to make a new texture package. The best way of doing this is opening an existing package with only 1 or 2 textures in it, like castle1.utx. Then save it under a different name.
2. In the texture browser, go to file…import and select the image you want to use. Saving the package will embed the image into that package, so you will not need the original image thereafter.

Important:

1. The most common file format for Unreal textures is .PCX, in indexed color only. Unreal also supports .BMP images in indexed and RGB color.
2. If you edit a texture package, remember to save it before playing your map. Otherwise, UnrealEd will lose your edits, and any new textures you added will not show up.