

# UT3 HEAD MOD

Dec. 2007



## BRIEF

The following deals with the addition of a custom head model to one of the existing factions in Unreal Tournament 3.

This easily applies to other body parts such as arms, torso, helmets aso. The difference lies in the fact that some of these extra bodyparts require sockets to be set on specific bones. This is covered in the tutorials linked in the UT3 Polycount Pack thread. Note that this page was made mostly as a memo hence the lack of images and links.

I will keep verbose details for side notes. If you just want to get things done you can skip the big grey blocks.

Requirements:

3d application with the corresponding ActorX plugin,

Unreal Tournament3,

Unreal Editor 3, ships with the game,

A text editing application. I use PsPad.

A 3d asset ready to be ported to the game.

#### NOTE

This tutorial is text only but should be rather easy to follow. For more information on how the editor works please refer to UDN and to the video tutorials from the bonus DVD provided with the game. The custom character process might seem huge at first but is actually very simple and fast. As a matter of fact if one sticks to the default skeletons having a custom character ported from a 3d program to the game only takes a couple minutes just like in the UT2003 days. I don't know yet if custom skeletons, custom factions and custom character shaders are possible with the 1.1 version of the game.

## EXPORTING THE MESH

I assume you have a head mesh ready with UVs and textures, aligned with one of the skeletons provided by Epic Games. These can be found on UDN in 3DSMax9 format.

If you use another 3d application or an earlier 3DSMax release you need to get these skeletons as FBX files. The Max to FBX conversion has been taken care of, the converted bones are available online.

Head textures are 2048\*2048 in Unreal Tournament 3 and the UVs needs to fit in the upper half of the UV space.

#### NOTE

The lower half of the UV space is for apparent skin bits like chest, eyes, arms aso. In fact if you have a look at torso body armor texture in the editor you will notice that the fleshy bits are not on the actual torso armor texture. My guess is that UT3 uses a multimaterial system to have apparent skin bits stay coherent with the skin type of the face. But since we are just focusing on creating a custom head we will not create a custom upper body armor piece - hence no need to worry about the lower part of the UV space for your custom asset. Better give your custom head a skin type that fits within the skin types Epic created. If you don't you'll have to create custom torso and arms.

#### NOTE

You can guess the other bodypart layouts by looking at existing UT3 character textures. Check out existing character packages to see how it works: In Ued open the Generic Browser (little blue grid icon), and rightclick > fully load some CH\_ packages. For even more example you can check the extra ones not loaded by default with the editor in C:\Program Files\Unreal Tournament 3\UTGame\CookedPC\Characters

If you work from an old head mesh of yours chances are the UVs do not conform to the UT3 norm. You can either move the texture bits into place in an image editing program and tweak your models UVs accordingly; or use Render To Texture (Max) or Transfer Maps (Maya) to automatically move texture chunks into a newly created UV layout.

To make the custom head ready import the desired reference skeleton in Max or Maya. Make sure the

smoothing groups/normals of your model are set properly. Convert to Edit Poly and reset Xforms (Max)/Delete history and freeze transformations (Maya). Skinweight your custom Head to the head bone and the neck to the neck bone with the Skin Modifier (Max) / Skin>Smooth Bind (Maya). Use the b\_ bones not the biped\_ ones if these came with the skeletons you use. There should be 70 active bones for a default character. Export the PSK to a temporary folder using ActorX. I recommend you put all your material there, ie copies of the textures aso. The ActorX plugin is located in the 'more' section of the utilities tab in 3DSMax / Maya users need to make sure it is loaded in the plugin manager, then type the required command line.

### **IMPORTING THE MESH IN UEd3**

Launch UEd3. Open the Generic Browser (little blue grid icon). Import the newly created skeletal bound psk into a new package. I used the AnimSet type, this creates an unneeded animation asset that can be deleted later. Name the package CH\_ModelName\_Female and the mesh SK\_CH\_ModelN\_Female\_Head01

#### NOTE

If you need a helmet-specific head variant (in case you have a ponytail or even just hairstrands sticking out) it seems like you need to call it SK\_CH\_ModelN\_Female\_Head01\_ForHelmet. I still have to check that in practice.

The skeletal mesh goes to a group called Mesh. The textures images and material tree will go to a group called Materials. You can import them now or later. You can decide which group an asset belongs to upon import or move it to a group later by rightclicking on the asset and using the 'rename' tool.

Double clicking the head will popup the animation browser.

#### NOTE

You might want to click the little Jesus icon on the far left to check if the bones look fine. If yes go to the 'Extra Mesh 1' dropdown and select a reference character of the same sex than your custom asset to see how it lines up. It is also a good time to scroll down through other assets to check the compatibility of your custom asset with existing shoulderpats, facemasks aso (if needed). To check against more assets that do not load by default : go back to the general browser and open > [...]Program Files\Unreal Tournament 3\UTGame\CookedPC\Characters\CH\_IronGuard\_Female.upk. Now go back to your package, and scroll down the 'Extra Mesh 1' list again, more assets will appear. You can even start building a preview of your character with up to 3 extra meshes. You can also check how it behaves with animations in the 'anim' tab.

In SkeletalMesh > Origin > change Z from 0 to -51 (in the case of a female Ironguard) In SkeletalMesh > RotOrigin > change Yaw from 0 to -90. For other sexes and factions simply follow the values used in the original Epic models.

Save your package. Location should be similar to : C:\Documents and Settings\Username\My Documents\My Games\Unreal Tournament

3\UTGame\Published\CookedPC\CustomChars\CH\_ModelName\_Female.upk

In Vista 'My Documents' might be called 'Documents'.

## SETTING UP MATERIALS

If not done already import your 2d textures into the package. Put them in a group called 'Materials'. More on this later in the texture settings chapter.

The head needs to refer to an existing 'Parent' material. You can find these materials in the CH\_ALL package to have a look at their shader tree.

However you will need an intermediate instance material since you want to use the parents structure but most likely not the same textures.

### NOTE

Numerical parameters such as emissive color RGB values, translucency strength also are also accessible but I don't think that parameters tweaked for, say, a lone head bodypart would translate in a character made out of this new asset + existing bodyparts from the game. If you created a complete set of bodyparts I think it's fine ; but if you just did one that will be mixed with other bodyparts chances are only one set of parameters will remain. The custom body asset will inherit the surface properties of the rest.

### NOTE

Ex. if you open the CH\_Necris\_Female package and go to the 'Materials' group you will find the 2d textures for the Necris female and instanced materials for each bodypart. However if you try to open these you won't find the actual material node tree, just parameters you can adjust. This parameter-only material is an instance of 'CH\_All.Materials.MI\_CH\_All\_Necris\_Base'. This one points to a parent material : 'MI\_CH\_Necris\_Global' which is the mother of all Necris shaders. This parent being different from the topmost IronGuard and TwinSouls shaders makes sense : Necris characters call for fancy cubemap based reflections while IronGuards and TwinSouls are shaded in a more simple way - hence they use the simpler MI\_CH\_All\_Base\_02 parent shader. I recommend you use that one. I still have to see if one can create his own topmost parent shader for specific fancy needs - these means creating a new faction also. For the moment let us stick to Base\_02. This base was used by Epic to shade the IronGuard and Ronin characters and covers most shaders needs. If you need reflections and more complex skin effects try using the Necris base.

### NOTE

Pointing to an existing parent material is the only way I managed to get a custom character working so far. If you create a custom material with specific effects and apply that to your custom model it will display fine in the character customisation screen, but it won't display in the game ; same even if you create a custom parent material. My understanding is that the character customization screen calls the raw assets, but the game itself uses a texture collage/atlas to render the whole character in one go. Hence the whole character needs to be pointing to a single parent shader obviously. (ex. a head with a non UT3-specific UV layout and/or a custom shader will work in that screen, but will be badly rendered in the game). I still have to check if one can

have a custom head pointing to a necris material, but part of the Ironguard faction. That would mean two shaders for one character... Most unlikely.

#### NOTE

If someone comes up with a crazy shader for a very specific visual purpose it might be possible to get a custom character to use this shader. It won't work well along with existing faction bodyparts but as a unique whole it might work. And you can also create a whole character bodypart set and tweak the numerical values of them all, even if using an existing parent. You can come up with something quite different from the defaults.

The shader hierarchy is as follows from top to bottom :

M\_CH\_All\_Base\_02 (stored in the default CH\_All.upk package installed with the game, do not modify)

MI\_CH\_ALL\_IronG\_Base (in CH\_All.upk installed with the game, do not modify)

MI\_CH\_ModelN\_FHead01\_V01

for team skins : MI\_CH\_Model\_FHead01\_VBlue / VRed

Time to create this shader. Go to your own package, rightclick the grey area and pick choose 'New Material Instance Constant'. Put it in the 'Materials' group and name it 'MI\_CH\_ModelN\_FHead\_V01'. This is the lower child material we will apply to the custom model.

The Material Instance Editor pops up. To point to a Parent you need to have it selected (ie clicked once) in the general browser, then click the little green arrow in the parent field. You will also use the green arrow later to point to textures, materials also. the little magnifying glass icon brings up the Generic Browser window - useful when you don't have the needed resource currently selected.

In our case the parent link is:

MaterialInstanceConstant'CH\_All.Materials.MI\_CH\_ALL\_IronG\_Base'

Since this material already has its own parent setup (M\_CH\_All\_Base\_02) you are all set and linked to the top.

You are now free to override the values of the topmost material with your own from inside your shader MI\_CH\_ModelN\_FHead\_V01.

You need to have your texture maps imported in the package to point to them so import them now if you didn't do it earlier. Set their settings as follows (right click on an imported texture for name and package options, doubleclick it for texture properties)

#### **Diffuse map**

T\_CH\_ModelN\_FHead\_D01\_V01\_SK1

CompressionNoAlpha set to on if you don't have an alpha stored in the texture

CompressionSettings : TC\_Default

Filter : TF\_Linear

LODBias 0

LodGroup : TEXTUREGROUP\_Character

SRGB on

Streamed DXT1

### **Normal map**

T\_CH\_ModelIN\_FHead\_N01\_V01\_SK1

CompressionNoAlpha set to on if you dont have an alpha stored in the texture

CompressionSettings : TC\_Normalmap

Filter : TF\_Linear

LODBias 2

LodGroup : TEXTUREGROUP\_CharacterNormalMap

SRGB off

Streamed DXT1

### **Specular map**

T\_CH\_ModelIN\_FHead\_S01\_V01\_SK1

CompressionNoAlpha set to on if you dont have an alpha stored in the texture

CompressionSettings : TC\_Default

Filter : TF\_Linear

LODBias 0

LodGroup : TEXTUREGROUP\_CharacterSpecular

SRGB on

Streamed DXT1

### **Emissive map**

T\_CH\_ModelIN\_FHead\_E01\_V01\_SK1

CompressionNoAlpha set to on if you dont have an alpha stored in the texture

CompressionSettings : TC\_Default

Filter : TF\_Linear

LODBias 0

LodGroup : TEXTUREGROUP\_Character

SRGB on

Streamed DXT1

Save your package.

Open your material. You have access to many parameters including which texture to point to.

Point to the desired textures. To point to a texture highlight it in the Generic Browser and click the green arrow.

For color settings (vectorParameter) clicking the little magnifying glass will bring up a color picker. See notes above if this doesn't work as expected.

When done active the desired tickboxes.

Doubleclick your custom mesh, highlight the newly created material in the Generic Browser, and back in in the AnimSet Editor go to Mesh tab > SkeletalMesh > Materials > [0] > green arrow.

Your mesh now has the correct material. Save your package. Content creation is done!

## ADDING CUSTOM BODYPARTS AND NEW CHARACTER DEFINITION TO THE GAME

You just created a custom bodypart but it needs to be referenced somewhere for the game to see it. You now need to add a 'Part' line to

```
C:\Documents and Settings\Username\My Documents\My Games\Unreal Tournament
3\UTGame\Config\UTCustomChar.ini
```

Also if you want your bodypart to appear as part of a new custom character with its own character name and characteristics you need a 'Characters' definition line. Note that I used Z as the ID for my new bodypart to be sure I was not conflicting with an existing head. It'd be even more safe to use complex names for IDs to avoid any possible conflict. Like MODELNAMEPARTNAME followed by A, B, C as a suffix if you made 3 helmet variants for instance.

### NOTE

You really need that character line if you created a custom head since there is no head swapping options in the character customization menu. If you created a helmet or a torso you don't need to create a custom character, you can simply take an existing character from the menu and customize it so that its torso piece is the one you created. You can basically create complete armor sets that way with no worry about modelling a head.

Bodypart line is :

```
Parts=
```

```
(
Part=PART_Head,
ObjectName="CH_ModelName_Female.Mesh.SK_CH_ModelName_Female_Head01",
PartID="Z",
FamilyID="IRNF"
)
```

Character line is:

```
Characters=
```

```
(
CharName="Ingame name of your character",
Description="",
CharID="Z",
bLocked="false",
Faction="Ironguard",
```

```
PreviewImageMarkup="",  
CharData=  
(  
FamilyID="IRNF",  
HeadID="Z",  
HelmetID="A",  
TorsoID="A",  
ShoPadID="B",  
bHasLeftShoPad=true,  
bHasRightShoPad=true,  
ArmsID="A",  
ThighsID="A",  
BootsID="A"  
)  
)
```

## WRAP-UP

Done! Launch the game, your character should appear in the faction you put it into. I assume you can then create custom character portraits and character bios. Now you want all the other bodyparts! Remember to follow the existing UV cutout for each kind of bodypart. Also check out tutorials dealing with bone sockets. You can make that up easily by closely looking at existing models though.

### ONE LAST NOTE

If you want ponytail movement or loincloth simulation you will notice that even though existing UT3 characters have bones for this (character cloth is not really using cloth parameters, it's just physics-driven bones), the character skeletons provided by Epic Games don't have these bones included. Still you can eyeball these in your 3d application and call them just like the ones from the game, it works. You'll find all needed naming info in the UEd animation browser. These bones behave like cloth thanks to physics assets located in the CH\_ALL package. I don't think Physics Assets can be created from within Ued. Maybe the old KAT still works. Hence we are stuck with default loincloth and ponytails for now.

Please contact me if I wrote anything wrong. Hope it will help!

