TUTORIAL:

Making An Unreal Tournament Movie

by - m3ss
#UTSportCenter @ irc.gameradius.org

Introduction:

Making a nice Unreal Tournament movie is not as daunting a task as one might think. All that is required is the proper software, creativity, and a little bit of help. That's where this tutorial comes in. I will walk you through the steps of acquiring and organizing footage, rendering your footage at the desired speed/framerate, as well as tips for making enjoyable movies. Please feel free to contact me on IRC at **irc.gameradius.org #mlut** or **#UTSportsCenter** or through www.prounreal.com and my username **m3ss**. I appreciate any input and/or corrections you readers might want to give me. Thanks!

Needed/Useful Software:

- **DemoManager 3.0:** Great UT mod that can organize, autorecord, and even fast-forward your demos!
  Download Link: http://wizii.wz.cz/udemo.zip
  or **DemoManager 3.2 (with UTDC fix)** – disadvantage: you can't get to free camera
  Download Link:http://www.unrealadmin.org/forums/attachment.php?attachmentid=1152&d=1169863417
- **XviD 1.1.3 codec:** http://ftp.freenet.de/pub/filepilot/windows/multimedia/video/xvid/XviD-1.1.3-28062007.exe
- **MeGUI:** Great program used for encoding your movie to .mp4
  Download link: http://mirror01.x264.nl/x264/megui-0.3.0.0.exe
- **AviSynth**: You will need this program to encode .mp4
  Download link: http://switch.dl.sourceforge.net/sourceforge/avisynth2/Avisynth_257.exe
- **OpenGL:** Latest OpenGL driver for Unreal Tournament.
  Download link: http://cwdohnal.home.mindspring.com/utglr/utglr34.zip
- **MovieUnreal2002:** UT mod used for capturing your demo footage. A must download!
  Download Link: http://wizii.wz.cz/MovieUnreal2002.zip
- **VirtualDub 1.7.0:** Program used in aiding compression of your movie once it is rendered.
  Download Link: http://wizii.wz.cz/VirtualDub-1.7.0.zip

- **Adobe Premiere:** Editing program used for organizing and editing your UT movie and adding numerous text, transition, and sound effects. I first learned on this program, and it serves its purpose nicely. However, I prefer…
- **Sony Vegas:** Another editing program that in my opinion is much more user friendly than Premiere. It has the same functionality, but a much more straightforward user interface. NOTE: This tutorial uses Vegas.
- **Microsoft Excel:** This database and spreadsheet program comes in really handy when organizing your list of demo footage and possible movie material. Any other spreadsheet program will suffice, however.
- **Unreal Tournament:** Of course you're going to need this program if you're following this tutorial. However, the concepts presented here can be applied to almost any other game.

Step One - Finding Footage:

The first, and probably most tedious, step of making an Unreal Tournament movie is finding all of your footage. Watching demos can be very tedious, especially if you don't know exactly what you're looking for or if you will find anything worthwhile at all. However, I have some tips and procedures that can help you make this as painless as possible.

1.) The first thing you should do is download and install DemoManager. This program will make your life much easier for both finding footage, as well as capturing footage. The installation is very simple, just follow the instructions in the readme file.

2.) Once you have DM installed, you will want to bind the follow two commands to your keyboard, as they will come in very handy.
- Pause – Pauses/Unpauses the demo.
- CurTime – Displays the Current Time in seconds.

You can bind these by type "set input key Pause" in your console in UT.
Example: set input 2 Pause – This will bind pause to your 2 button on your keyboard.

3.) If you're like me, then you probably have a bunch of demos from who knows what games with who knows what type of footage on it. Well, it's time to watch demos. You might want to start up **Winamp** and listen to some possible soundtrack choices in the background while you watch your demos. Open up a demo in **DemoManager**. Conveniently, DM can inform you of any missing packages you may need in order to play a particular demo, as well as other useful information.

4.) Once the demo starts playing, you should have a **pen/pencil** handy with a **piece of scratch paper**. Write down the general demo information, such as name, date, map, etc.

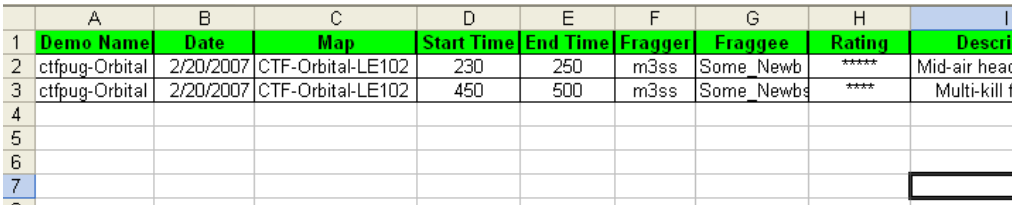
5.) While watching the demo, if you see anything worthwhile that you might want to use as movie footage, press your bind for **Pause**.

6.) The demo should pause, at which point you can press your bind for **CurTime** to get a reading of the current time in seconds that you are currently watching (paused).
7.) Subtract a buffer for the shot, about 10 seconds or so, in order to allow yourself time to find the footage later without overshooting it.
8.) Write down the **CurTime** and **shot information** on your **scratch paper**, info such as shot location/description, fragger/fraggee, etc.
9.) Press your bind for **Pause** again to unpause the demo and continue watching until you find more footage. Rinse and repeat with new demos until you have sufficient footage.

Step Two: Organizing the Footage:

At this point you really want to start to organize all of your shots into some sort of spreadsheet. Making a movie requires numerous shots, some of which may be only a few seconds long, so you can imagine how many shots you will need to fill out a multiple minute movie. As I said earlier, I like using Microsoft Excel for my organizing purposes, as it is user-friendly and I had it on my computer. So let's get started!

1.) Open up **Microsoft Excel**. First things first, you need to make yourself some **column headings**. I had information such as Demo Name, Date, Map, Start Time, Finish Time, Fragger, Fraggee, Rating, and Description. It should look something like this:



	A	B	C	D	E	F	G	H	I
1	Demo Name	Date	Map	Start Time	End Time	Fragger	Fraggee	Rating	Descri
2	ctfpug-Orbital	2/20/2007	CTF-Orbital-LE102	230	250	m3ss	Some_Newb	*****	Mid-air head
3	ctfpug-Orbital	2/20/2007	CTF-Orbital-LE102	450	500	m3ss	Some_Newbs	****	Multi-kill f
4									
5									
6									
7									

2.) **Rating** is a great category to have, as I used it as a gauge for how I felt about the quality of a certain shot. Obviously, more stars equals higher quality, but sometimes you need filler material, as finding that amazing footage isn't always easy, or fun.

Step Three: Capturing the Footage:

Now that you have a decent list of shots, you are ready to begin another somewhat tedious task in the moving making process, capturing your footage. For this step you will need to have **MovieUnreal2002** installed and working properly. Once again, this is not difficult, just follow the instructions that are included within the .rar file.

1.) It's time to use that nice organized **Shot Sheet** you just created with Excel. Either print out the shot information you intend to use, or simply jot down the info on another piece of scratch paper. The information you will need is the **Demo Name** and the **Start Time**.

2.) If you have installed MU2k2 correctly, you should find an .exe file called **MovieUnreal436.exe** in your UT/System folder. This is the program location for MovieUnreal2002. You might want to create a new shortcut on your desktop referencing this file, as you will be using it a lot. Download wIz4's latest config from http://files.filefront.com/settingszip/;9505357;/fileinfo.html and copy it to your System folder (don't forget to back up config).

3.) Upon running MU436.exe you should see an options window that includes **Output Settings, Rendering,** and **Frame Size**. F

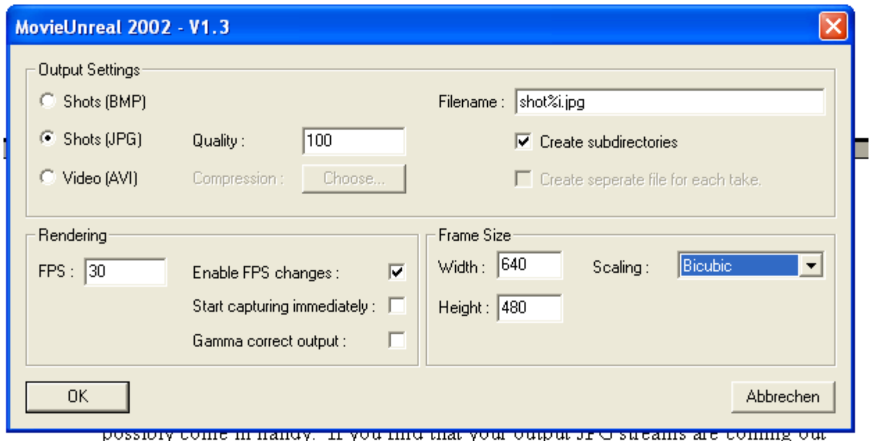
4.) Under **Output Settings**, for the best quality, select **Video (AVI)** radio button, click on **Compression** and choose **Uncompressed** or **Huffyuv** (lossless codec, **download link:** http://www.freecodecs.net/files/huffyuv_220.zip). How to install Huffyuv codec: uncompress the files to a temporary directory, right-click on the `huffyuv.inf` file and select "Install." You can delete the files in the temporary directory after the installation completes.

5.) Set your filename.

6.) Under **Rendering**, set the **FPS** to **30** as a default (in few weeks/months, after you become pr0, feel free to experiment with higher/lower values). This will record your footage at standard timing. If you want slow-mo, increase the **FPS** to **60** for half-speed (or use the slomo command in UT). For extreme slow motion try to combine 60 fps and slomo 0.5 in UT. Obviously, lowering the FPS results in fast-forward recording, but I don't actually recommend this. If you set desired speed in UT, your movie will look better, because of stable FPS. If you set for example FPS 15 in order to speed up the avi, it will really go faster, but your avi will consist of only 15 frames instead of 30, which won't look that smooth. That's why I recommend speeding up and down directly in UT. Now leave all of the checkboxes unchecked in case you are recording on **OpenGL**. If you are using **D3D, check Gamma correct output**. FPS Changes usually do not work, and you usually do not want to start capturing immediately.

7.) Under **Frame Size**, set the scaling to Super Sampling (this will become handy only if you are actually scaling down your avi. For example, let's say that your resolution in UT is 1280x1024, but the resolution in MU is just 1024x768. In this case, MU will scale avi down and that's when Super Sampling is useful. But in my opinion, all that scaling is useless and you will probably lose some details on textures. All of you certainly have good computers, so setting the max in your gfx card preferences (**AA, AF,....**) **AND** the same resolution in **UT AND MU** (so it won't be scaled down) will give you nice results. Also if you set same resolution even in Windows, MU and UT, capturing your footage will be a bit faster than on different resolutions). Now just set your desired resolution (I use 1280x800 in UT and MU).

My window looks like this:



MovieUnreal 2002 - V1.3

Output Settings

○ Shots (BMP)

● Shots (JPG) Quality : 100

○ Video (AVI) Compression :

Filename : shot%i.jpg

☑ Create subdirectories

□ Create seperate file for each take.

Rendering

FPS : 30

Enable FPS changes : ☑

Start capturing immediately : □

Gamma correct output : □

Frame Size

Width : 640 Scaling : Bicubic

Height : 480

possibly come in handy. If you find that your output JPG streams are coming out

8.) Once you have your settings correct, press OK and Unreal Tournament will start normally.

9.) Open up **DemoManager** again and select the demo that contains the footage you wish to capture.

10.) Start the demo. Hopefully you have your **Shot Sheet** handy with the time information. Here is where DM works it's magic. **Pause** the demo, then open the console. In the console, type **seekto time** where the time is your **CurTime** find for the shot you wish to capture.
Example: seekto 120 – This fast-forwards to two minutes into the demo.

10.) DemoManager should then fast-forward to within a certain percentage of the time you asked for. Be patient as sometimes this process can take a minute or so, especially if you are seeking a great distance into the demo.

11.) Press **F12**, this starts MovieUnreal capturing footage. Then **Unpause** your demo.

12.) Allow MU2k4 to work its magic until you have the scene/shot you wanted completely captured. Once again you should allow yourself a buffer zone both before and after the shot in question, as you can always delete the excess frames later.

13.) Press **F12** again to stop capturing.

14.) Once you are finished capturing footage from you current demo, close MU2k2. Rinse and repeat with new shots/demos. Open up your UT/System folder and you should see a new .avi.

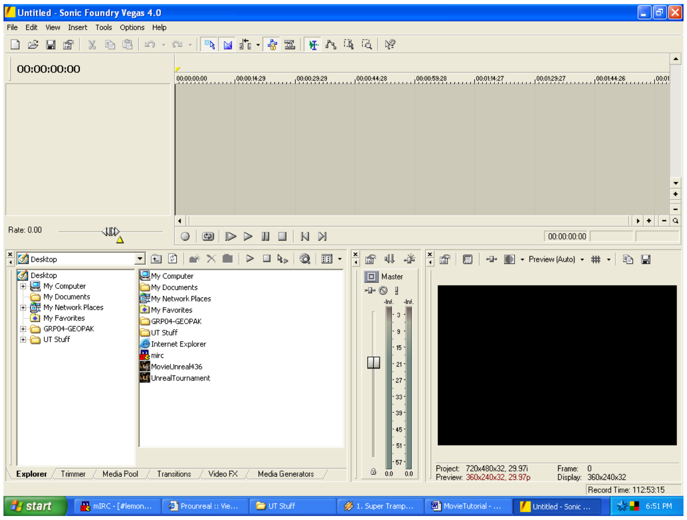
15.) Now you should rename your avi file and remove it somewhere to your project folder, otherwise there would be m3ss in your System folder.

16.) Rinse and repeat the process with new footage/demos until you think you have enough to start assembling and editing your movie.

Step Four: Making the Movie:

So now that you have captured enough footage/shots to start the movie, you get to proceed with easily the most enjoyable part of the movie making experience: the editing. This process is where you get to let your creative juices flow as you can pick what shots to use, what soundtrack to use, as well as a multitude of other advanced editing techniques. Here I will be going over some of the basics of using **Sony Vegas** for assembling and rendering your movie.

1.) Open Sony Vegas. You should see a window that looks like this:

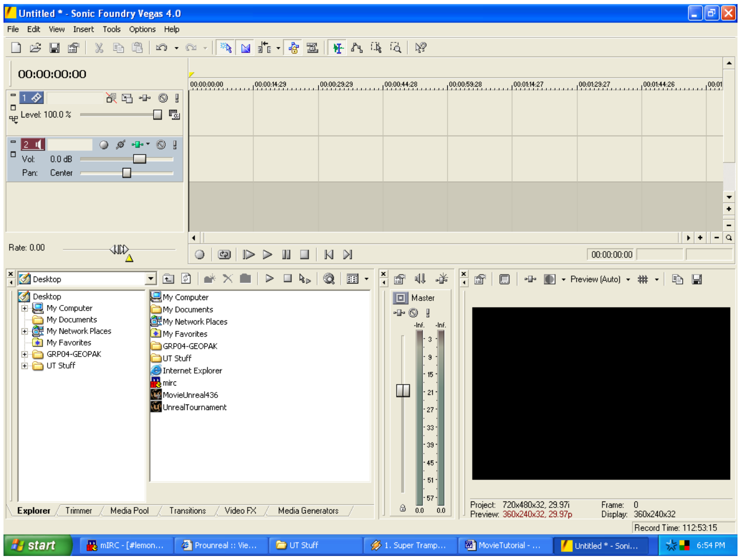


Project: 720x480x32, 29.97i
Preview: 360x240x32, 29.97p
Frame: 0
Display: 360x240x32
Record Time: 112:53:15

Now, before you start working on your movie, press **Alt + Enter** and you will get to **Project Preferences**. Now, set:

Width, Height and framerate exactly same as in MU
Field order: None
Pixel aspect ratio: 1
Full resolution rendering quality: Full
Click OK, and save your project.

2.) In the left-hand window, below the 00:00:00:00 time, right click and select **Insert Video Track**. Right click in the same window again and select Insert **Audio Track.** These are the two basic tracks you will need to make a simple UT movie. You should see a window that looks like this:



Level: 100.0 %

Vol: 0.0 dB

Pan: Center

Rate: 0.00

Project: 720x480x32, 29.97i

Preview: 360x240x32, 29.97p

Frame: 0

Display: 360x240x32

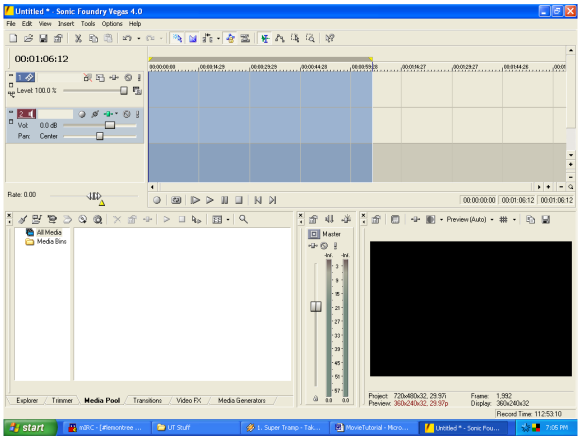
Record Time: 112:53:15

3.) Now in the bottom left-hand corner you should see various tabs starting with the Explorer tab. There are also Trimmer, Media Pool, Transitions, Video Effects, etc. Click on **Media Pool**.
Click on the **Import Media** button in the new window and select either an AVI file (or JPEG sequence) or an audio track/mp3 that you wish to include in your movie. If you are selecting a JPG sequence, make sure to check the **Open still image sequence** box in the lower left-hand corner. Select the first JPG in the sequence and this feature will automatically select the rest of the JPG's in the sequence based upon their numbering. (This is why we separated the footage into folders earlier.)

5.) Next you are basically making your movie. Import all of the footage and songs you wish to use and start adding them to their appropriate tracks. To add footage, drag a JPG sequence (AVI file) from the media pool to the appropriate track. You should see it appear in the track. To add a song, do the same thing, drag the track into the audio track window on the right and it will appear in the track.

6.) Get creative with your movie and try out some of the features of your relative editing tool, be it Premiere or Vegas. Above all else, have fun with it and make the movie to please yourself first and foremost. There are multitudes of advanced techniques that can be used, many of which I am unfamiliar with. I will likely make another Tutorial for advanced editing techniques. However, if you can't wait you can ask me questions in #UTSportsCenter.

7.) Once you have your movie set the way you like it with all of your transitions, song, and footage, it's time to render it. Select all of the footage you wish to render by dragging your mouse over the footage in the timer window. You can select the entire movie, or just a portion if you wish to edit and view just one small section of your movie. It should look like this:



Project: 720x480x32, 29.97i
Preview: 360x240x32, 29.97p
Frame: 1,992
Display: 360x240x32
Record Time: 112:53:10

8.) Now click File > Render As > **Change the Template to Default Template (uncompressed)** we will be compressing it later. Click **Custom > Video** and set:
Frame size: Use project settings
Frame rate – same as you set in MU and Project preferences
Field order: None
Pixel aspect ratio: 1
Video format: Uncompressed (yes, you will need 165106540165 GiBs – to be exact, there is an equation to figure out, how much disc space you will need: (resX * resY * 3 * FPS * lengthInSeconds) / 1024 / 1024 = filesize in MiBs

Click on Audio and **uncheck** Include audio.
Click Save and prepare to wait. Depending on the length of the clip you are rendering, this could take quite some time. Once the rendering completes, click File > Render As again and this time, **select Wave (Microsoft) .wav** instead of Avi, 'cause I bet that you want to have some nice music in your movie too:P. Let it render, close Vegas and now you should see the uncompressed .avi and .wav file wherever you saved it.

9.) Now we need to install and open up **VirtualDub**. Again, the process is simple, follow the instructions. You also should take this time to install the **XviD 1.1.3** files as well, because you will be using these for compression purposes.

10.) Open VirtualDub and click File > Open video file... Find your uncompressed .avi file that you just rendered with Sony Vegas and open it.

11.) Click on Video > Compression. Select XviD MPEG-4 Codec from the window and click OK.
You also might be interested in wIz4's XviD settings (download them at http://files.filefront.com/settingszip/;9505357;/fileinfo.html). Before you compress your movie, I have to tell you, that there is nothing like unique XviD settings. For each movie you have to create new settings, or at least change them a bit (depends on resolution, fps, scenes, textures, details, effects, ...). These settings I give you should give you decent results while encoding HD movies (1280x720 to be exact).

12.) Next click on Audio > Wav Audio and load your uncompressed .wav. Then click on Full Processing Mode. Then click Audio > Compression.
Select MPEG-Layer 3 and a reasonable kbit value and click OK.

13.) Now pay your attention. In XviD dialog, select Encoding type: Twopass - 1st pass, click on More, check Discard first pass, uncheck Full quality first pass, click OK, once again, File > Save as AVI > OK. XviD will do the first pass (basically it will gather information about your video file, so it can distribute bitrate better in the second pass later). Once this is done, click Video > Compression > XviD > Encoding type: Twopass - 2nd pass and select desired bitrate).Cclick OK, File > Save as AVI > OK.

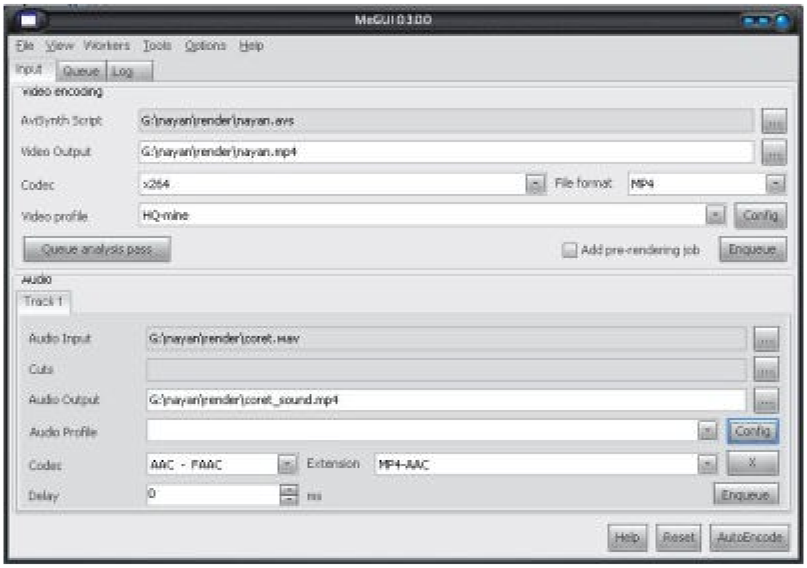
VirtualDub will begin compressing your movie. Once the process is complete, the finished product is your compressed movie. You can delete the very large uncompressed .avi (after you check out your final movie for errors etc!) file from Sony Vegas now and view your work of art. This compression will produce a reasonable quality video built for small file size. However, there are advanced compression techniques that can produce higher quality videos at reasonable file sizes. Once again you can contact me in IRC about this or wait for a newer tutorial.

Step Five: Creating .mp4 file

1.) Download and install **MeGUI** and **AviSynth** (mentioned at the beginning of this tutorial).
2.) Create a file called **output.avs** for example and put there following lines:

```
AVISource("nameOfYourFile.avi")
Crop(0,0,-0,-0)
ConvertToYV12()
```

3.) Open MeGUI and press Ctrl + U for updates. Once you have latest versions of all files, load your AviSynth script (that output.avs you created before). A preview window will pop up, but we don't need to work with it, so feel free to close it.



AviSynth Script: G:\nayan\render\nayan.avs
Video Output: G:\nayan\render\nayan.mp4
Codec: x264
File format: MP4
Video profile: HQ-mine
☐ Add pre-rendering job

Audio Input: G:\nayan\render\coret.wav
Cuts:
Audio Output: G:\nayan\render\coret_sound.mp4
Audio Profile:
Codec: AAC - FAAC
Extension: MP4-AAC
Delay: 0 ms

In Video Encoding part, set
Codec: x264
File Format: MP4
Video Profile: If you are not familiar with all those settings, just download wIz4's profile (http://files.filefront.com/settingszip/;9505357;/fileinfo.html), which will give you constant filesize 27 MiBs/min and great quality (kalias – the movie).
Click Enqueue

4.) Audio:
Audio Input: load your uncompressed .wav file
Audio Output: just some filename
Codec: AAC – FAAC
Extension: MP4-AAC
Delay: 0
Config > Set some reasonable bitrate
Now click Enqueue
5.) Click Workers > Create new worker
6.) Click Queue > Start. MeGUI will now encode your .avi and .wav to separated .mp4 files (video .mp4 file will have same filename as your .avs file). By the way, if you downloaded that HQ profile from wIz4, don't be surprised by **_VERY_ low** encoding speed. Encoding x264 is CPU-killer and those settings are very detailed, made to give very nice results. Anyway, once the encoding of both files is done, click on Tools > Muxer > MP4 Muxer, load your encoded video .mp4 file (Video input) and audio .mp4 file (Audio input), change the final filename (Muxed output), click Queue and Start. Muxing should take few seconds
7.) Go watch your movie in great quality.

I hope you have found this tutorial on making an Unreal Tournament movie helpful. If you have any questions or comments, feel free to contact **m3ss** on **irc.gameradius.org** in **#mlut** or **#UTSportsCenter**.