

## Lab #3 – UDK Mutators

For this lab you will be creating a mutator for UDK that modifies a weapon as well as its projectile (so this excludes modifying non-projectile weapons like the Enforcer) and puts it in the players' default inventory

All mutators should be inside of a package named after your last name (for example, "jhala.u").

### Submission

The assignment should be submitted on moodle in the locker as one .zip file with only ".zip" as extension. You should include a readme file detailing what each mutator does, at what resolution your configuration menus look the best, and anything else you think I should know. You should also include all necessary localization and configuration files. Your submission should be zipped and follow the folder structure below. I should be able to unzip it into my UTGame directory and have the files go into the appropriate folders.

- /Development/Src
  - /jhala
  - Readme.txt
  - /Classes
    - ArnavsMutator.uc
    - CustomWeapon.uc
    - Whateverelse.uc
  -

**Setting up WOTgreal for UDK** (based on modifications from Mike Dominguez from NCSU and tutorial from <http://itu.dk/people/peder/> )

1. Export the script files from the Unreal Editor

1. Open the Unreal Editor

2. Open the generic browser.

3. Select the Actor classes tab.

4. From the file menu, select "Export all scripts". This will export the scripts to /My Documents/My Games/UDK/UTGame/ExportedScript/

2. Install WOTgreal

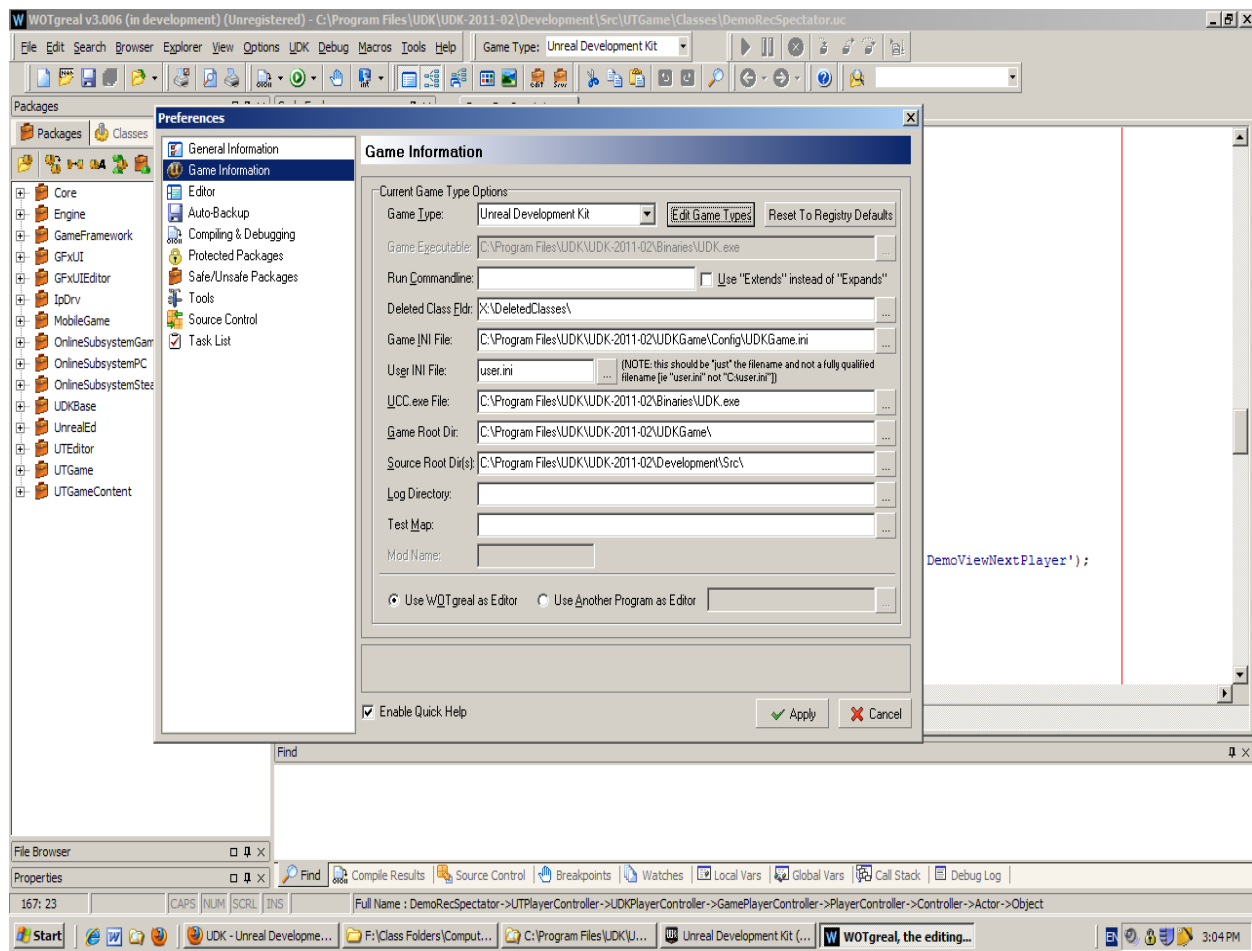
1. Download and run the newest WOTgreal installer (3.006 development build) from [www.wotgreal.com](http://www.wotgreal.com).

2. *To run WOTgreal with UDK, you are going to need version 3.06, so you must download the development build, and extract the files into the WOTgreal installation directory, overwriting the old files.*

3. WOTgreal 3.06 has a setup wizard that automatically detects installed unreal games, except from Unreal Engine 3 games, so just skip through the wizard.

3. Configure WOTgreal for the UT3 game type

1. Go to Options -> Preferences, and then choose "game information" from the menu.



2. Hit the "Edit Game Types" and, press "Add". If Unreal Development Kit is already on the list, select it and press the edit button.
3. Enter the following settings:
  1. Display Name (This is the name your game type configuration will be stored under) = **UDK**
  2. Game Exe Name (This is the same for all game types and mods) = **UDK**
  3. Default UCC Name (The compiler) = **UDK.exe**
  4. Menu Name = Same as Display Name
  5. Select UE3 under game architecture
4. Go back to the "Game Information" window, and select the new/modified UDK game type.
5. In the field, "UDK.exe file", browse to the location of the UDK executable :  
C:\Games\UDK\Binaries\UDK.exe
6. In the field, "Source Root Dir(s)", enter the directories where you have stored the UnrealScript files for your project, followed by a ";" and then the folder that contains the UDK script files (Example:  
"C:\Users\Arnav\Documents\My Games\UDK\UTGame\Src;  
C:\Users\Arnav\Documents\My Games\UDK\UTGame\ExportedScript"). This will build the class tree once you close the

property window.

#### 4. Other notes

1. After creating or adding source files, you need to go to UDK-> Refresh Package/Class Tree for them to appear in the sidebar.

## Mutator Development Steps

Here is a quick overview of the steps you need to follow to create a mutator (excluding the configuration screen). For more detailed instructions see Ambershee's tutorials in the **Helpful Links** section.

- Follow the instructions for setting up WOTgreal.
- Create **jhalas** folder in **Documents\My Games\UDK\UTGame\Src**.
- Create **Classes** folder inside that. All UnrealScript files will go in here.
- We need to make sure the game knows where to find our new code.
  - Open **UTEditor.ini** in **Documents\My Games\UDK\UTGame\Config**
  - Go to the **[ModPackages]** heading
  - Add the bottom of this section add **ModPackages=jhalas**
- In WOTgreal, write (for example) **DefectiveShockRifle.uc** and **FirstMutator.uc** and save them in the **Src\jhalas\Classes**.
  - To compile, copy your UT3 Editor shortcut, then in its target replace **editor** with **make**. The compiled code will be here: **My Documents\My Games\UDK\UTGame\Unpublished\CookedPC\Script\jhalas.u**
- Compiling created the **UTjhalas.ini** file in **Documents\My Games\Unreal Tournament 3\UTGame\Config**. Edit the entries for **FriendlyName** and **Description**.
  - Go to **Documents\My Games\UDK\UTGame\Unpublished\CookedPC\Localization** (need to create that folder if it doesn't exist) and create **jhalas.int**. Add the lines for the **DefectiveShockRifle**, like this:
    - **[DefectiveShockRifle]**
    - **ItemName="Defective Shock Rifle"**
    - **PickupMessage="Defective Shock Rifle"**
- Need to add **-useunpublished** to the target of your UT3 shortcut for the mutator to appear in game.

## Development Tip

Use **Unreal FrontEnd** (located in **UDK\Binaries**) to speed up development. It will allow you to jump right into a game with your mutator enabled. In the SP tab, browse to find the map you want to play. Then, in the Extra Options field, enter the following, for example:

- **?Mutator=jhalas.FirstMutator?numplay=0 -useunpublished -nomoviestartup**

I would also suggest checking the "Show Log" button – this will bring up a window with the game log, which is helpful for debugging. Then, click on the Launch button, and your game will automatically launch into the level with your mutator enabled. Numplay specifies the number of bots. Since this bypasses all menus, when working on your configuration menus a shortcut like this is helpful:

- **"C:\Program Files\Unreal Tournament 3\Binaries\UDK3.exe" -useunpublished -windowed -**

**resx=1280 -resy=960 -nomoviestartup -log**

## Example Mutators

The mutators that are included with Unreal Tournament 3 can be found in the UTGame package of the exported script. Look for class names that begin with “UTMutator”.

## Helpful Links:

<http://udn.epicgames.com/Three/UnrealScriptReference.html> - Overview of UnrealScript

<http://www.codekisk.com/unreal/ut3/scriptref/> - UT3 UnrealScript Reference Documentation

<http://gearsforums.epicgames.com/forumdisplay.php?f=350> – Epic’s UnrealScript Forums

<http://www.moddb.com/members/ambershee/tutorials> - Ambershee’s UnrealScript Tutorials (most relevant are #1, #2, #3, and #7)

<http://www.wotgreal.com/> - WOTgreal

If you find links that are helpful as you are working on the project, please post them to the class’s blog.

## Corrections to Ambershee’s Configurable Mutator Tutorial

If you want to go through his tutorial to create your menu, here are some errors that you’ll come across.

**ButtonBar.AppendButton("", OnButtonBar\_Back);**

Should be:

**ButtonBar.AppendButton("<Strings:UTGameUI.ButtonCallouts.Back>",  
OnButtonBar\_Back);**

**“Once you’ve done that, we’re going to need three more sliders -  
sliJumpHeight, sliHeath and sliMaxHealth.”**

Should be:

**“Once you’ve done that, we’re going to need three more sliders -  
sliJumpHeight, sliStartHeath and sliMaxHealth.”**

**UIConfigScene=UI\_Scenes\_PlayerTweak.PlayerTweakConfigMenu**

Should be:

**UIConfigScene=UTUI\_Scenes\_PlayerTweak.PlayerTweakConfigMenu**

For the tutorial to work. However, realize that following naming conventions it should be

UI\_Scenes\_PlayerTweak.

Also, comment out all “SliderValue.CurrentValue = 100.f;” lines, and add this at the bottom of the if statement:

**PlayerSpeed.SliderValue.CurrentValue =  
class'PlayerTweak.PlayerTweakMutator'.default.Speed;  
PlayerJumpHeight.SliderValue.CurrentValue =  
class'PlayerTweak.PlayerTweakMutator'.default.JumpHeight;  
PlayerStartHealth.SliderValue.CurrentValue =  
class'PlayerTweak.PlayerTweakMutator'.default.StartHealth;  
PlayerMaxHealth.SliderValue.CurrentValue =  
class'PlayerTweak.PlayerTweakMutator'.default.MaxHealth;  
PlayerSpeed.UpdateCaption();  
PlayerJumpHeight.UpdateCaption();  
PlayerStartHealth.UpdateCaption();  
PlayerMaxHealth.UpdateCaption();**