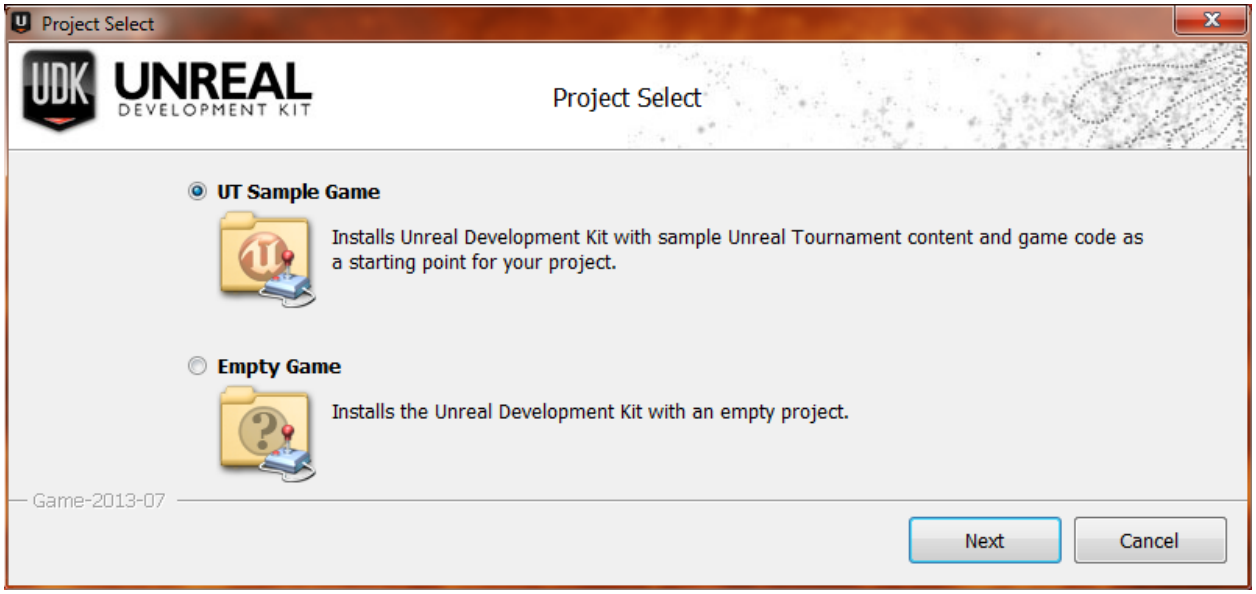
GEAS Install Steps
By Alexander B. Strout

As GEAS was built within the Unreal Development Kit, a copy of the UDK is required to build the scripts and run the project. Additionally, the Development Kit toolset itself is unfortunately available in Windows only. However, stand-alone deployables can be made for multiple platforms, including both PC and Mac (of which profiles for GEAS are included in this package) which will not require installation of the UDK.

The Unreal Development Kit is freely available at
http://www.unrealengine.com/en/udk/downloads/
GEAS was built on the July 2013 UDK Beta (http://download.udk.com/UDKInstall-2013-07.exe), which is the current latest version. However, it should work with any future versions of the UDK.

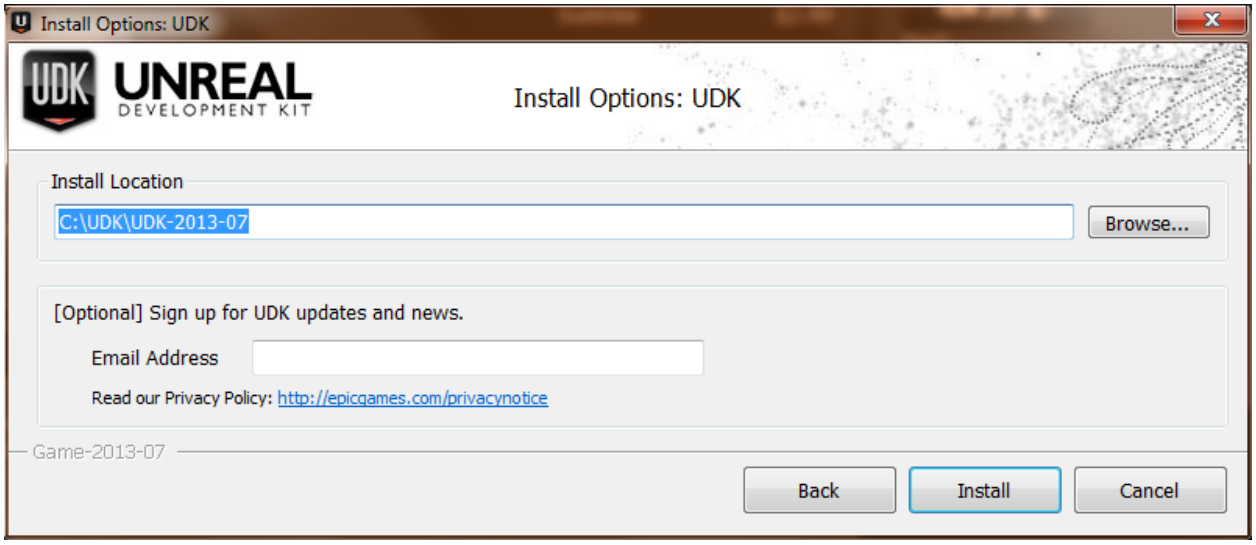
The installer for the 07-13 beta is ~1.9GB, so this may be a good time for a cup of coffee.

Upon running the installer, it will ask for a fairly standard license agreement. After, we will be presented with this screen:



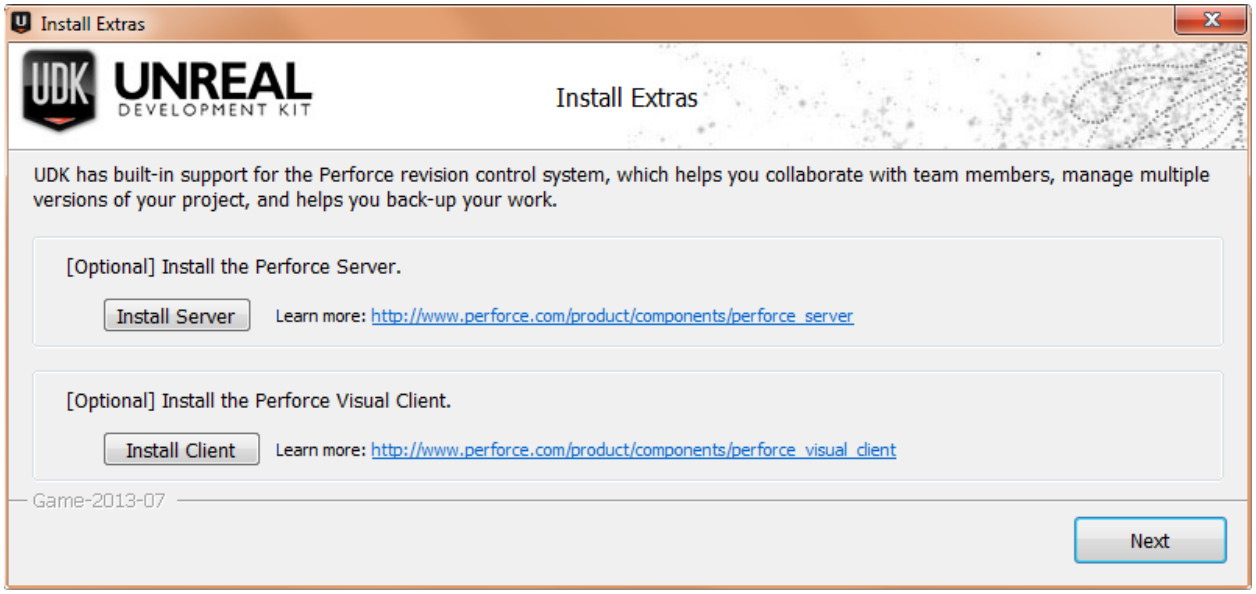
GEAS was built using a few select pieces Sample UDK Content, so be sure to pick Sample Game. When building, it will only include the assets used, which are very few compared to what it provides.

Next, we're presented with a simple install location:

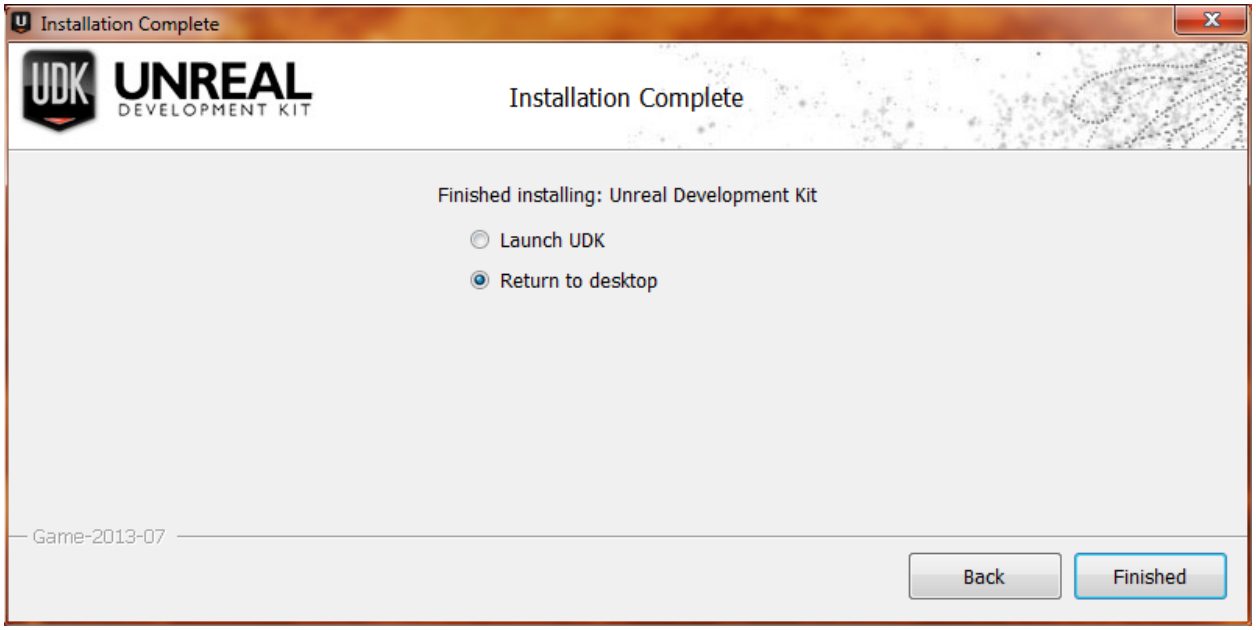


Any location is fine - just be sure to **make note** as we will need to navigate to it later.

Next, an option to install Perforce version-control will be presented. Just skip this step.

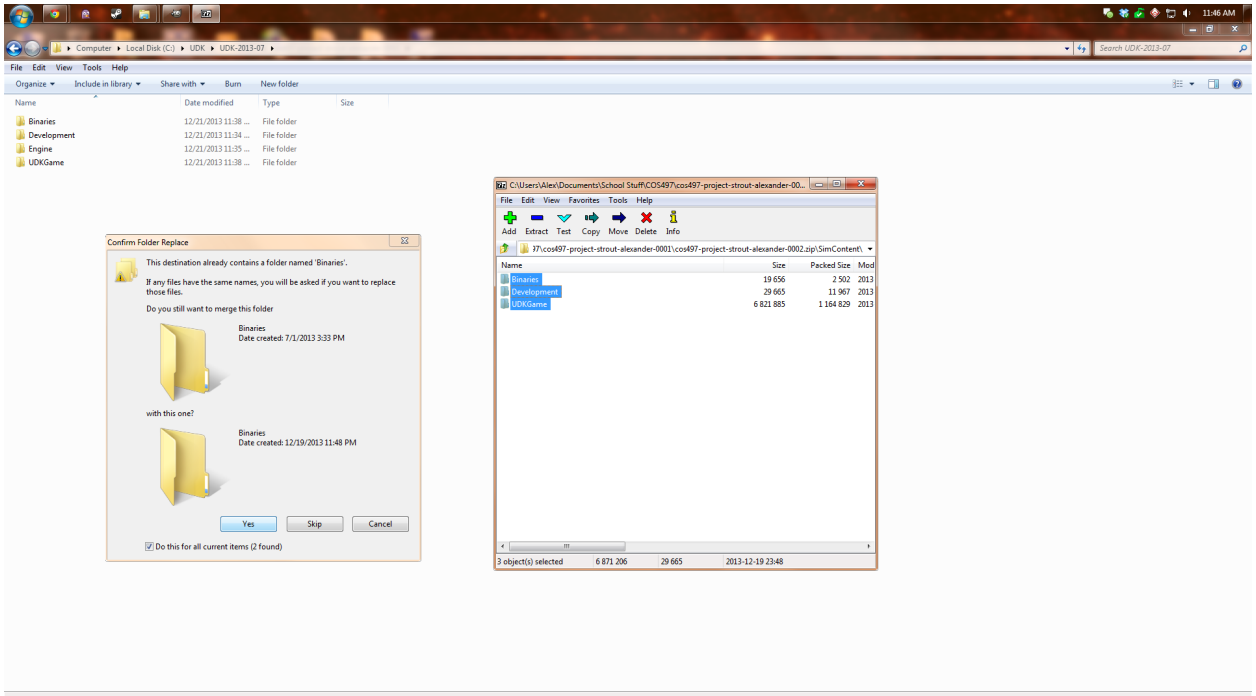


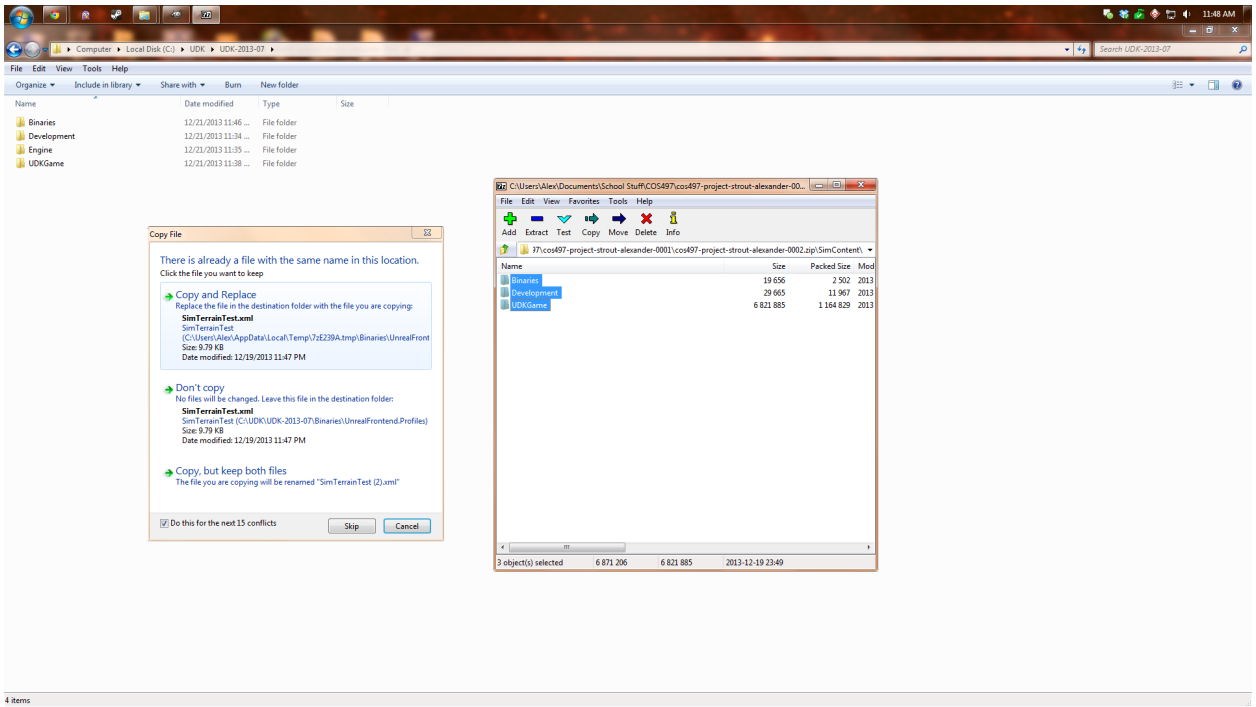
Next up, an option to run the editor will be presented:



Be sure to **not** run the editor, and instead just return to desktop - we need to install the GEAS content first so the editor knows to mount it.

Next, navigate to the UDK install directory selected earlier - it should show Binaries, Development, etc. folders as shown below. Open the GEAS zip package and extract the contents of SimContent to this folder, overwriting all files.



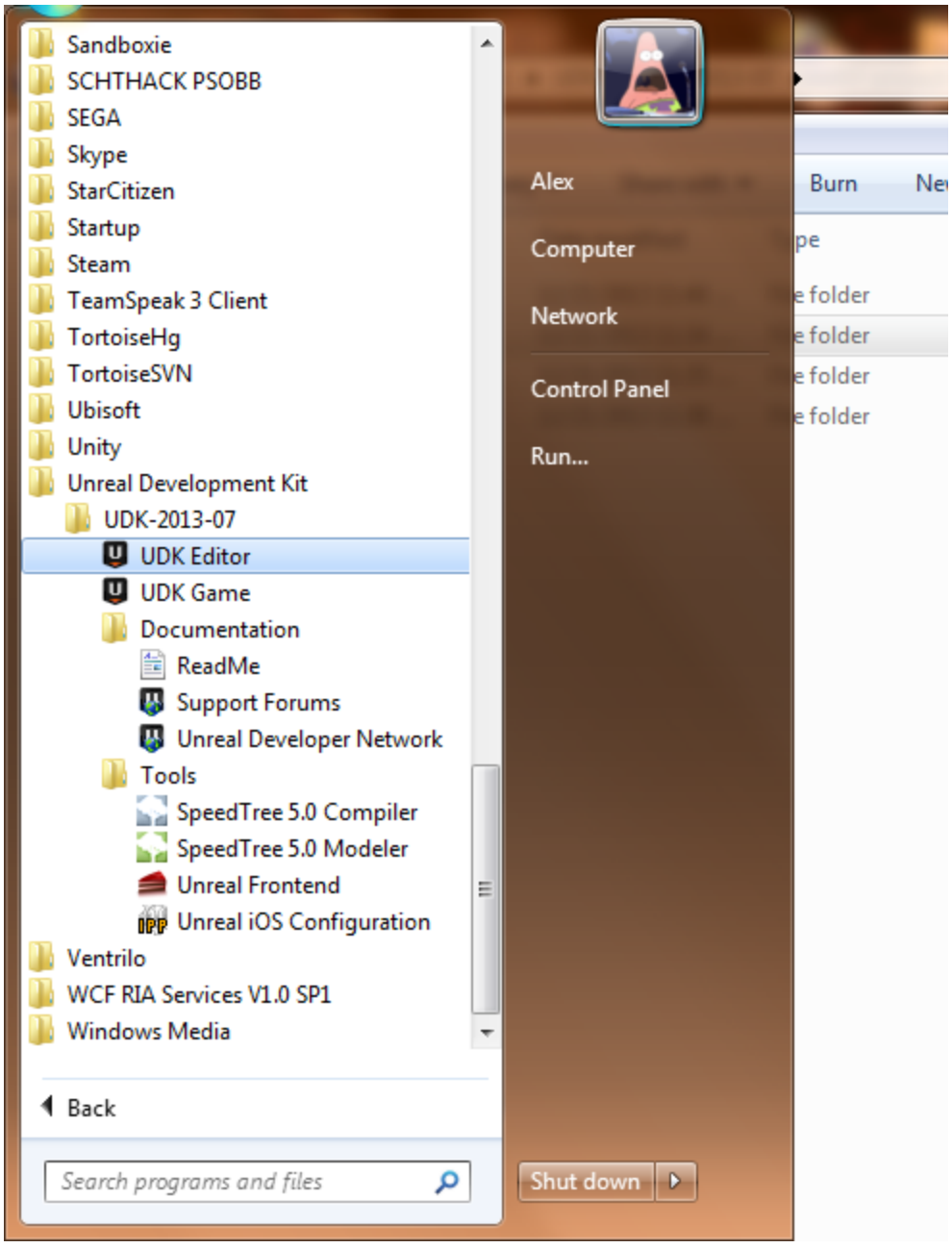


Be sure to overwrite all files when prompted.

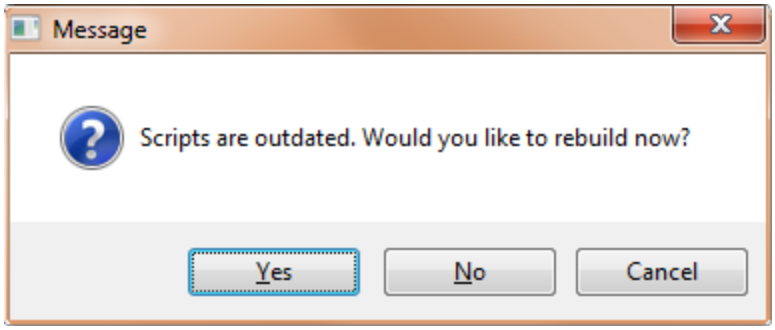
After this is completed, we're ready to build the scripts. Navigate to the Unreal Development Kit folder under the start menu. From here, there are two ways to compile:

- UDK Editor, which upon launch will prompt if you'd like to rebuild any scripts
- Unreal Frontend (under Tools), which allows to manually rebuild scripts

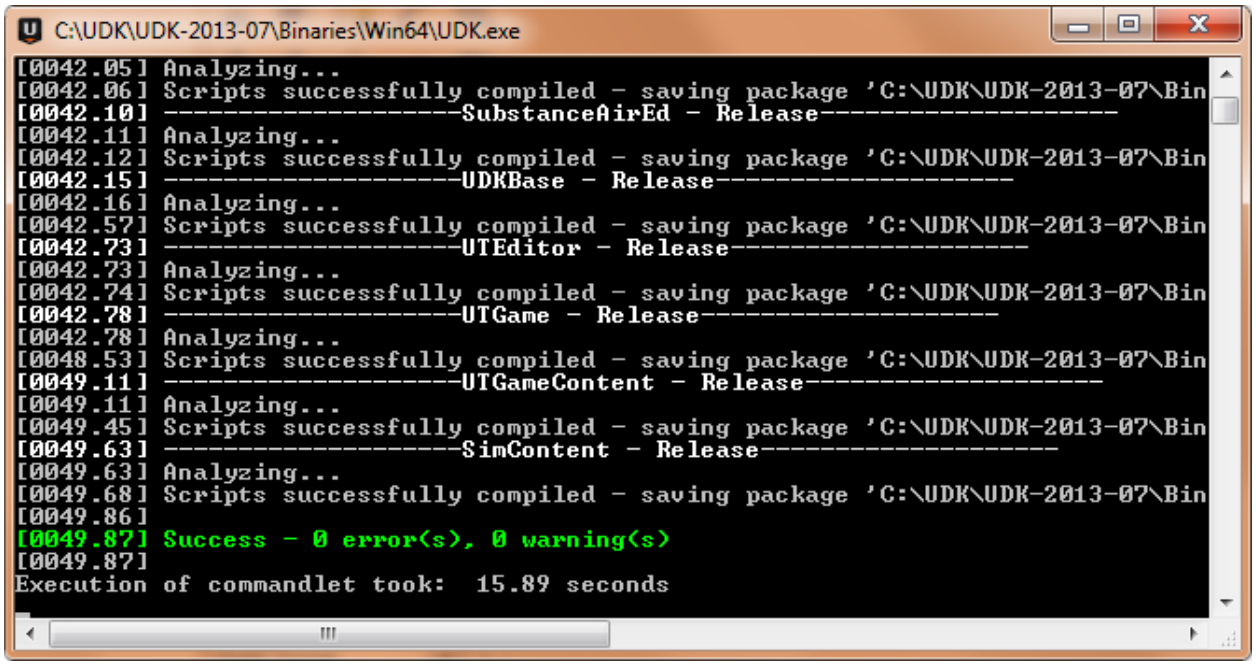
We'll touch on the Unreal Frontend later. First, let's give the UDK Editor a shot.



Upon launch, you should see this prompt:

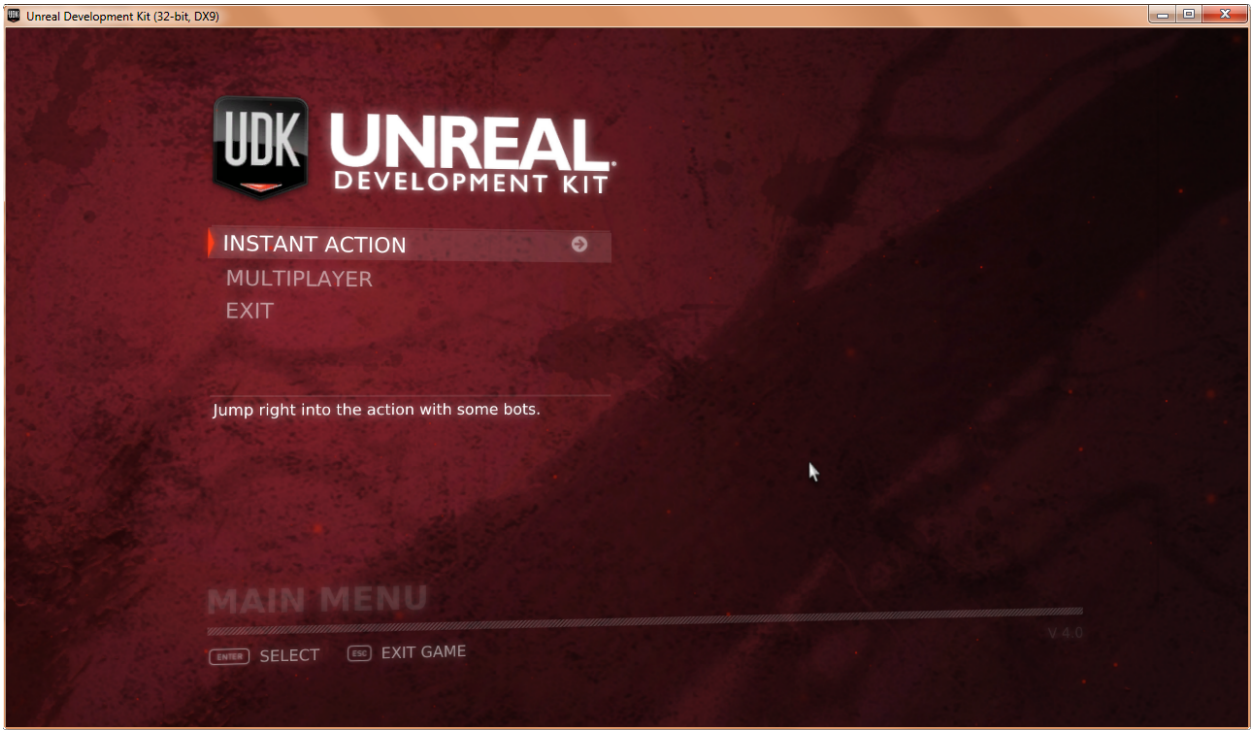


The following should show.

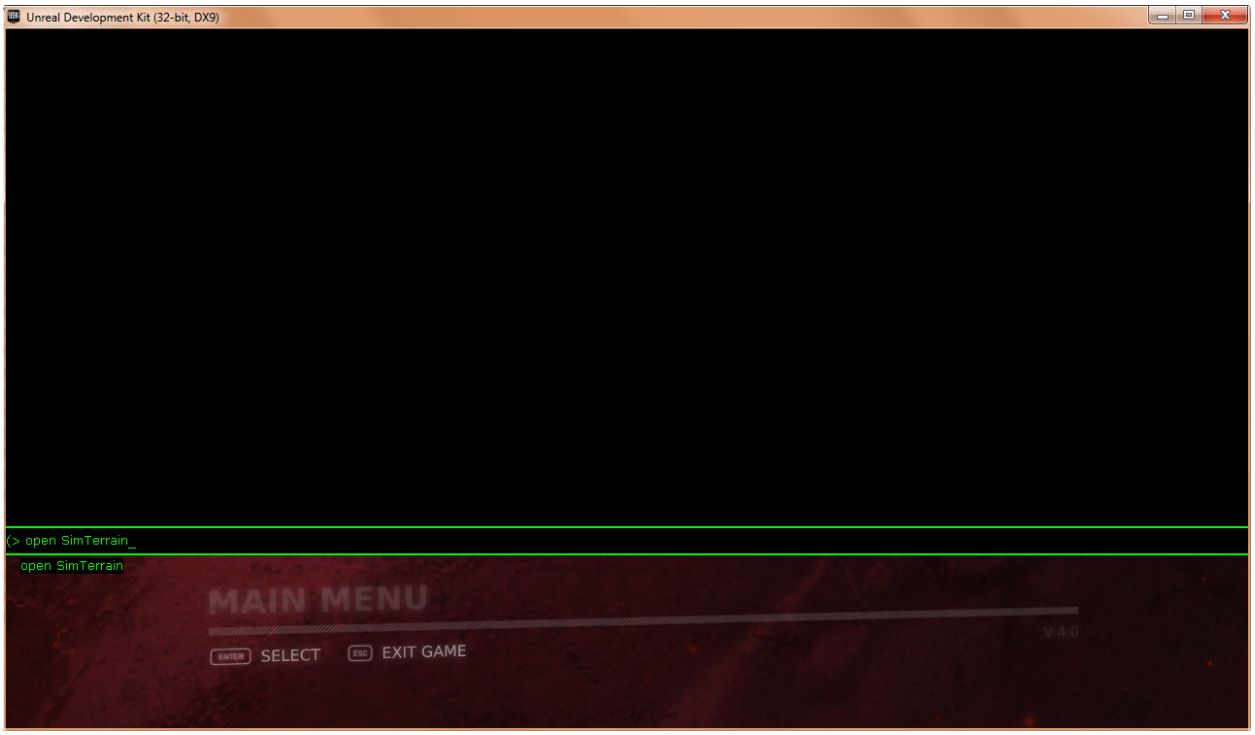


If GEAS content was installed in the folder correctly, SimContent should show as the last build step as above. Once done, go ahead and close the window. From here, launching the UDK Editor again will take you straight into the editor interface - which will get a separate document of its own. However, we're just interested in building the project, which we've now done, so let's go ahead and check it out!

Go ahead and launch UDK Game from the same Start Menu folder. Upon loading, we should be presented with this screen:



Bah! Writing a custom menu GUI was a little beyond the scope of the project - this is Computer Science after all, not New Media. Instead, open the console with the **Tilde ~** key. This should present us with this:

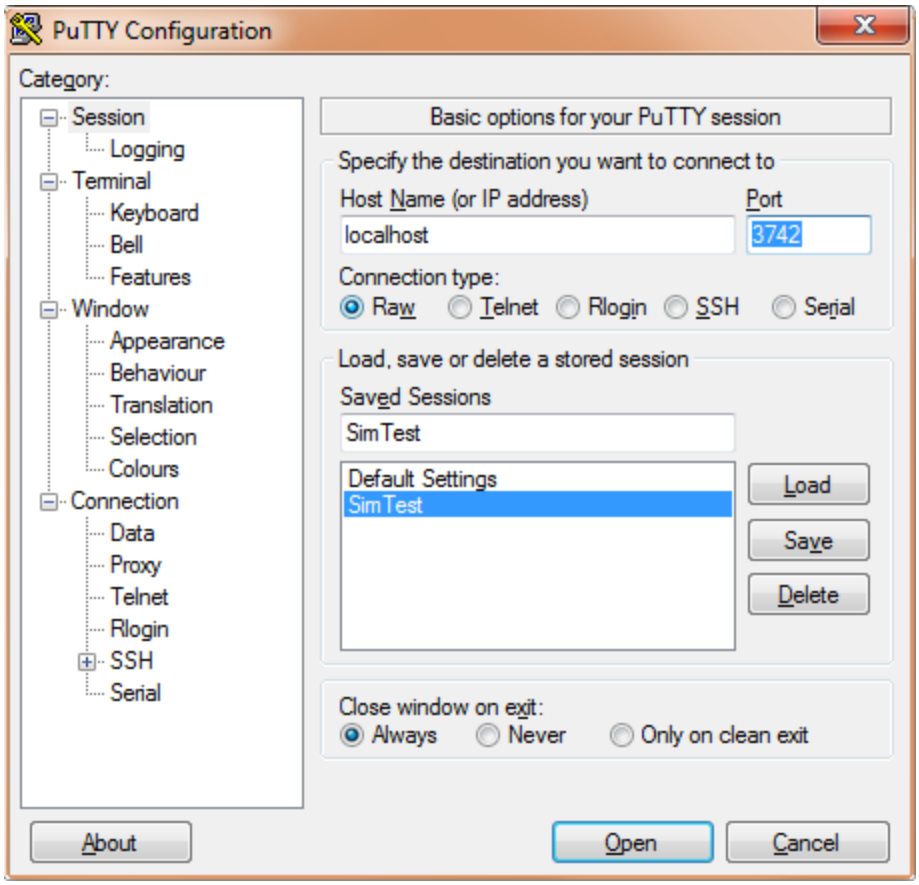


With the console open, type “open SimTerrain”. This should present some nonsense loading screen and, if everything works:

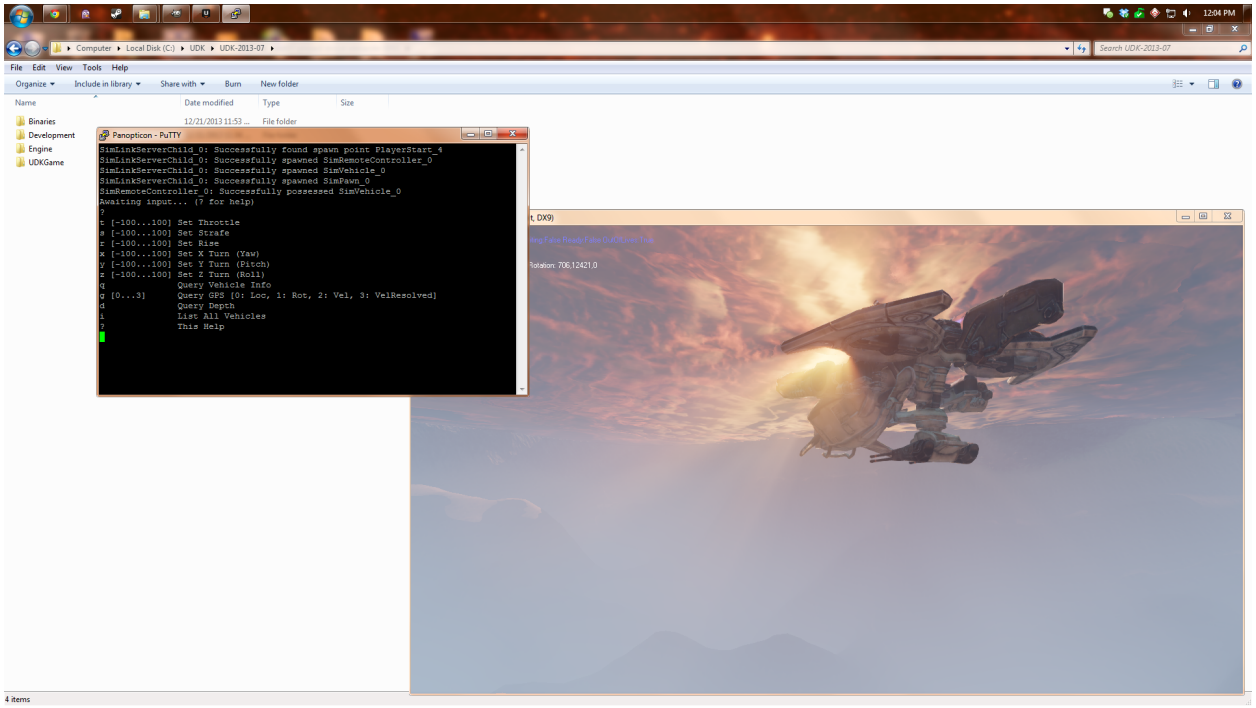
Voila!

- Camera Movement **W**, **S**, **A** and **D** keys
- Camera Rotation **Mouse**
- Cycle Active Vehicles **Left Mouse** or **Scroll Wheel**
- Return to Free View **Right Mouse**

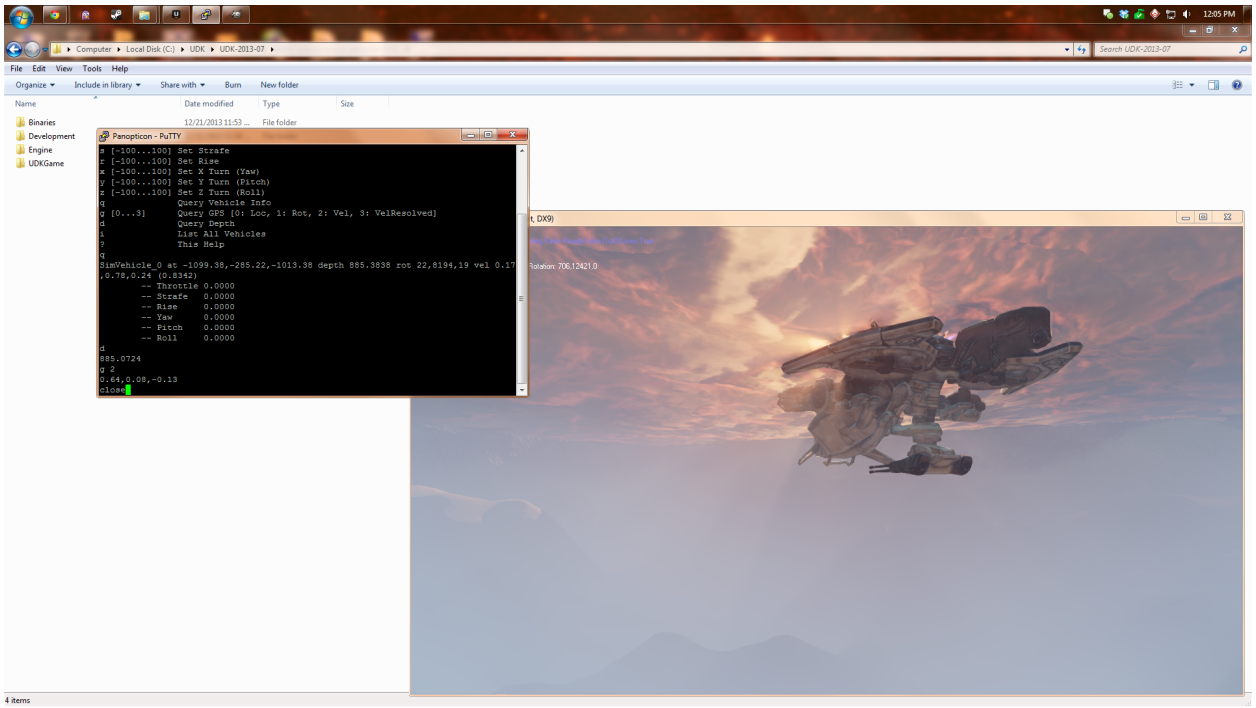
However, there are no vehicles! A socket connection needs to be opened to spawn and control a vehicle. To do so, simply open a raw connection to <address>:3742 (e.g. localhost:3742). For example, we might use PuTTY to do so:



Upon opening the connection, we should see something like so:

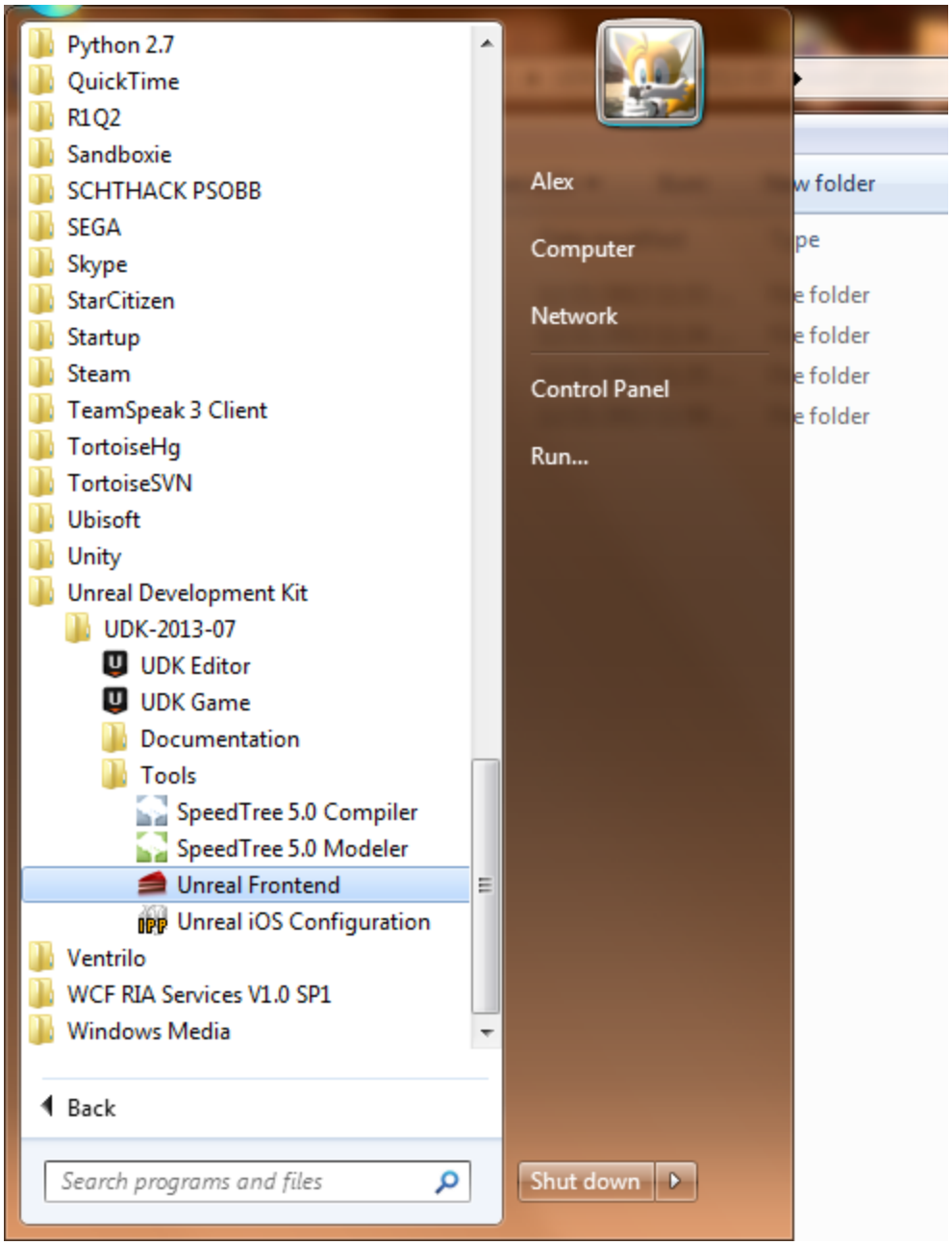


Simply input commands as desired (? for help). When finished, type “close” to close the connection (or just terminate it):

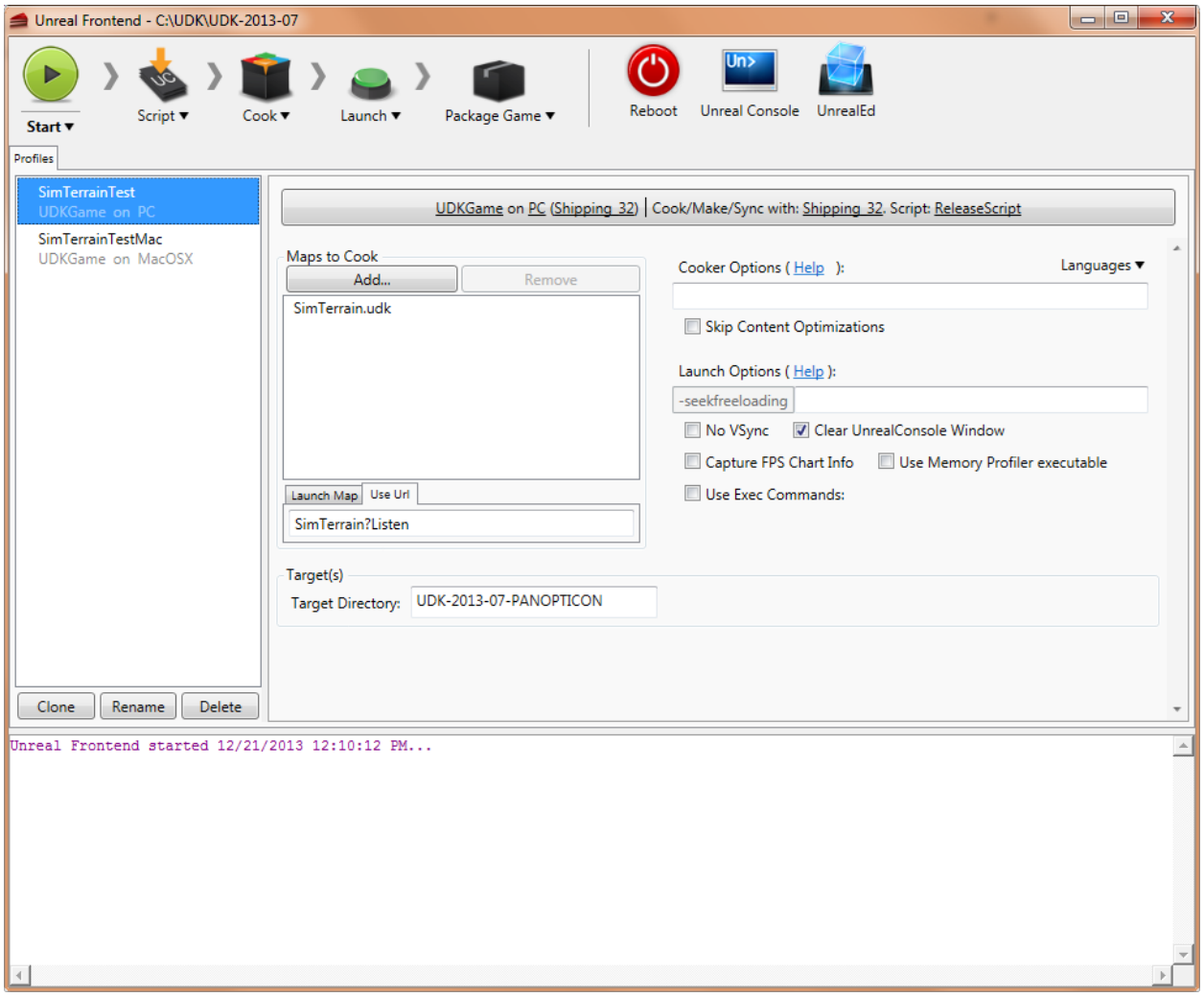


When finished with the demo, open the console again (**Tilde ~** key) and type “quit” to exit the simulation:

Next up, let's take a look at the Unreal Frontend. This allows us to not only compile the scripts (as mentioned earlier), but also do other tasks such as building deployables for other platforms. Go ahead and open this tool now:



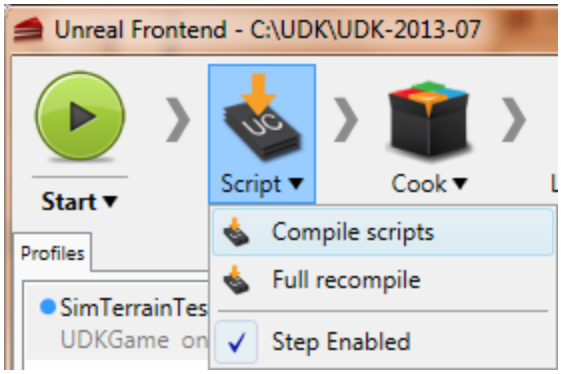
When opened, we're presented with several options. First, on the left and currently highlighted, we have the build profiles for PC and Mac stand-alone versions of the simulation demo. The options on the right allow you to customize aspects of the build process, but they should be good to go as-is.



The main button we're concerned with is the *Start* button in the top-left - this will run all of the build steps in order, from compiling the scripts to packaging the game (it will also launch the demo, for testing.

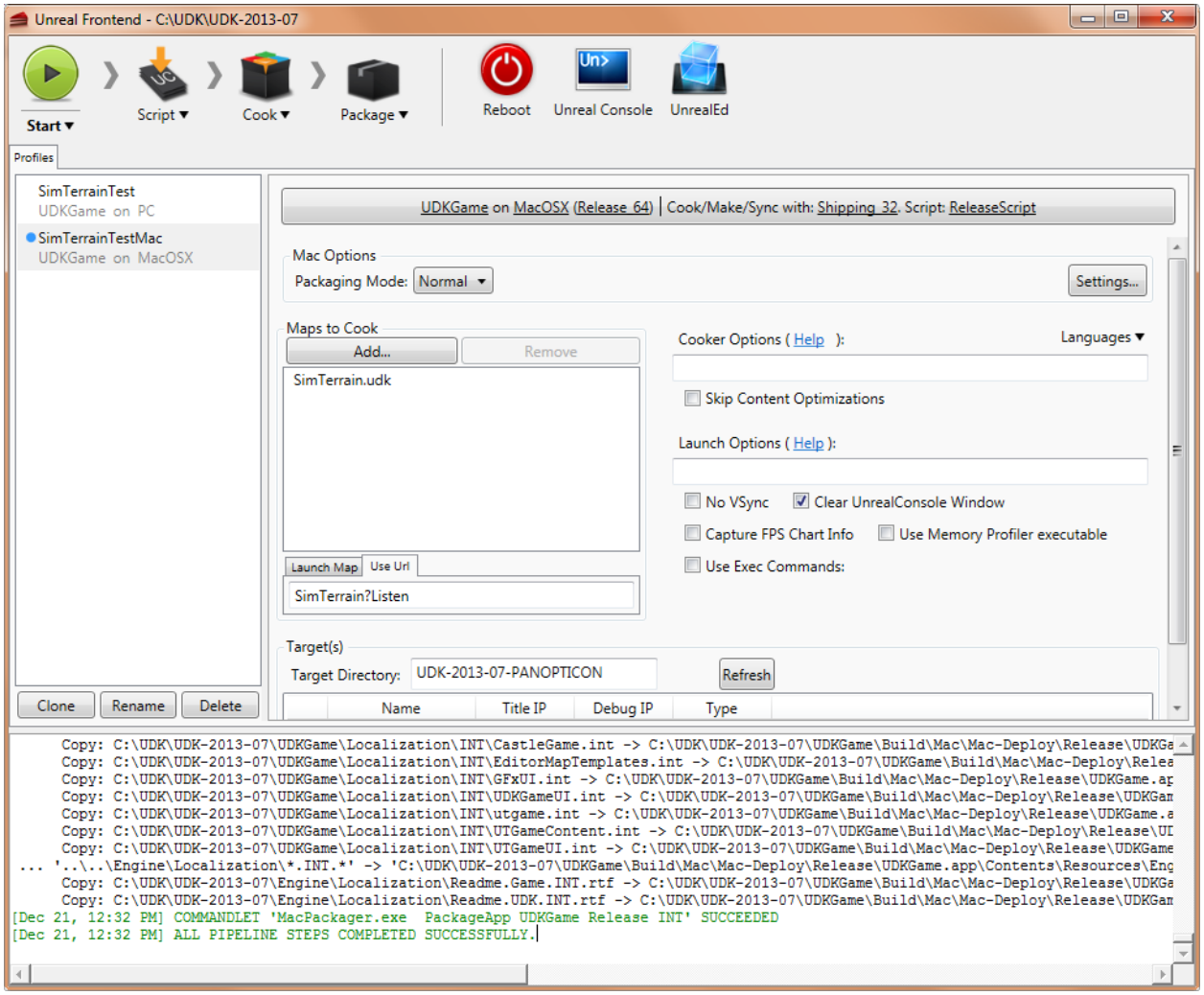
From here we can also launch the editor via the *UnrealEd* button, making this a good general-purpose interface for working with the UDK.

We can also do any of the build steps manually - for example, compiling the scripts - without running all of the other steps:

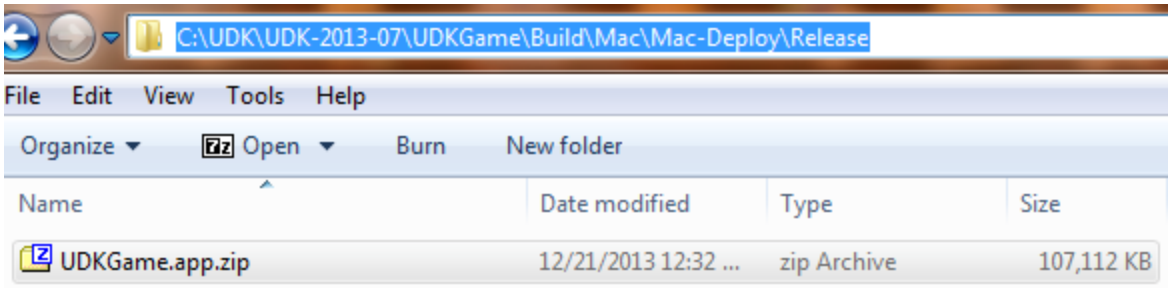


Notice the Step Enabled checkbox - we can use this to skip any of the steps for a full build.

Let's go ahead and do a full build for the stand-alone Mac version. Go ahead and select the *SimTestTerrainMac* profile and click *Start* :



Once completed (should show output in the bottom pane similar to above), it will place the built program in UDKGame\Build\Mac\Mac-Deploy\Release directory.



Copy this over to a Mac system and try it out, if desired. We're done!

Starting the demo can be done via command line:
(I forget what command line looks like on Mac, but the arguments would be the same)
UDK.exe SimTerrain - launch UDK and immediately load the demo
UDK.exe SimTerrain?Listen=1 - launch UDK and immediately load the demo as a listen server (enables other clients to connect to it using...)
UDK.exe <address> - launch UDK and connect to server at specified address
UDK.exe server SimTerrain - launch UDK as a dedicated server (no client / rendering, server only with console window)

Starting the demo can also be done via console (~ key):
open SimTerrain - open the demo
open SimTerrain?Listen=1 - open the demo as a listen server (same as above)
open <address> - connect to address as client
quit - exits the demo

To connect to the server via socket to spawn and control a vehicle, simply open a raw connection to *<address>:3742* (e.g. localhost:3742). Multiple connections can be opened and closed freely and will automatically spawn and destroy the appropriate objects within the game world.