**Setting up NIUI**

**Fore note:**

This is a generic guide for setting up NIUI with different versions of UDK. NIUI was built in the December 2010 version of UDK, so I strongly recommend that you use that if you are not comfortable with following the steps below.

In the **Quick Start** folder of this download, there is a quick start guide written specifically for the December 2010 version. This will get the sample project going with very little work on your part. You can try the steps on other versions of UDK but I can’t guarantee that it won’t ka-splode UDK.

**Step 1:**

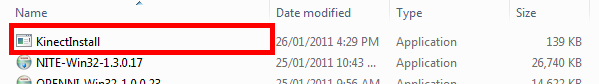
Firstly, I strongly recommend that you uninstall **\*ALL\*** of the current Kinect drivers, the OpenNI binaries and NITE binaries before you begin. I have found that it is best to start from a clean slate for the setup process.

**Step 2:**

*IMPORTANT: Make sure that your Kinect is unplugged from your computer before you begin this step.*

Open the folder called “**Installers**” found in the download. This will have a subfolder titled “**KinectInstall**” which contains a several files. The one that we are interested in is called “**KinectInstall.exe**”, which is a small program that automates the setup process for OpenNI / NITE / Kinect. Credit for this goes to:

<http://babaandthepigman.wordpress.com/2011/01/26/openni-kinect-getting-set-up-on-windows/>

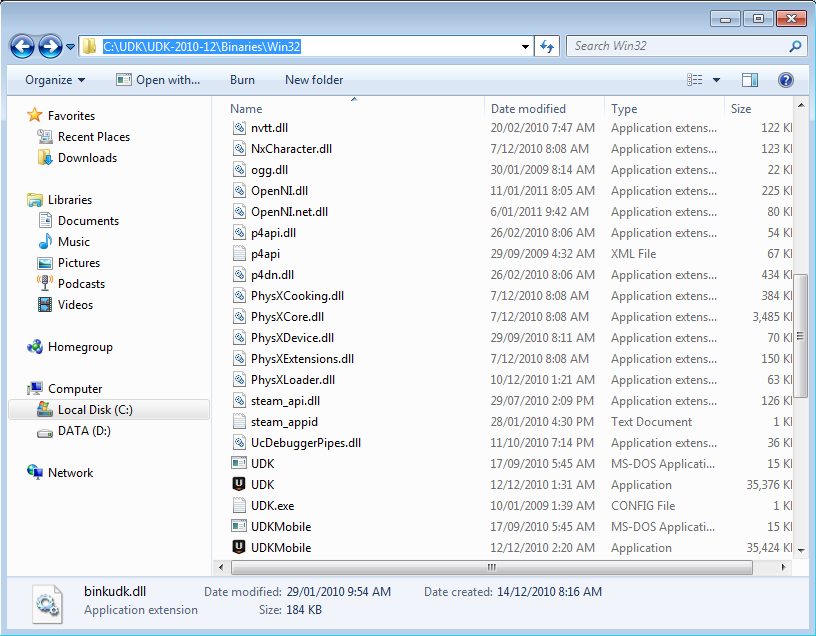


When you run this a small console window should appear and begin installing the OpenNI, NITE and Kinect Driver in the correct order. When this is finished, you can plug your Kinect back into your computer and move on.

**Step 3:**

After you have installed this go to the folder called "**Required dlls**" that is in the same folder as this file and copy all the .dlls in the folder “**To be copied into Win32**” into the following folder:

**(Your current UDK Install Path)/Binaries/Win32**

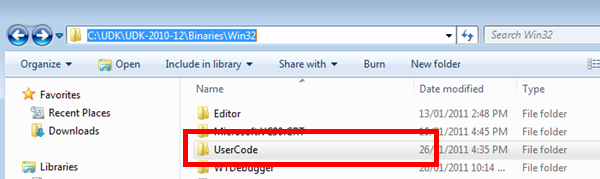
****

**\*ALL\*** of these dll's need to be in the same folder as your UDK.exe. I had massive troubles with this when I first started doing this because of this reason. If UDK spits out error message about Xn... something not being found it is probably because you don’t have these dlls in the directory with the UDK.exe.

**Step 4:**

Copy the NIUI.dll into your “**UserCode”** folder. The NIUI.dll should be in the “**Required Dlls**” folder.

This needs to be the folder that is within the same directory as the Win32 folder.



**Step 5:**

Copy the contents of the folder “**Content/UnrealScript API**” into your projects directory and add them to your current project.

The files needed for the API are:

* **NIUI\_Core:**  The main part of the API. This binds to the NIUI dll and is responsible for updating OpenNI and broadcasting events.
* **NIUI\_CallbackInterface**: An interface that when implemented allows the owner class to tap into the event pump. See NIUI\_SamplePlayerController to show how this can be used.
* **NIUI\_DependencyInterface**: An interface that when implemented the NIUI\_Core will find and notify of creation.
* **NIUI\_SkelControlSingleBone**: Bone control animation node used to link a skeletal bone to an OpenNI bone within an animtree.
* **NIUI\_SkeletalController**: Caches and updates the bone anim nodes. Spawn this in your pawn class and pass the AnimTree template to it when PostInitAnimTree is called.

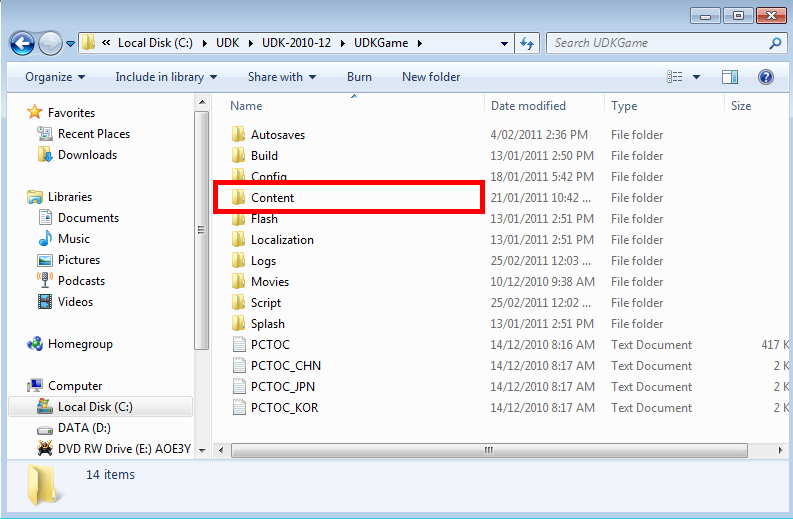
You are best to add these 5 files to your project rather than adding the NIUI project and building it from scratch. The sample project is there so that you can see how to implement the API into your GameInfo, PlayerController and Pawn classes.

**Step 6:**

Find the NIUIConfig.xml file in the content folder within the folder where this document lives and put it into the **UserCode** folder which is where you put the NIUI.dll (See step 3). This is the configuration settings used to create the OpenNI context.

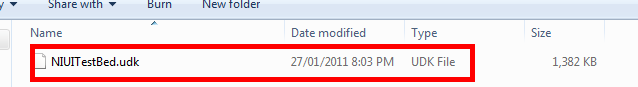
**Step 7:**

Copy the entire folder “**NIUI Demo Content**” from the “**Content/Unreal Packages**” folder into your “**(UDK Directory)/UDKGame/Content”** folder. This is the packages needed for the demo.



**Step 8:**

If you want to play the sample project, you will also need to copy the map called “**NIUITestBed.udk**” that is used in the sample project.



This needs to be copied from the “**Content/Unreal Maps**” folder found within the download into “**(UDK Directory)/UDKGame/Content/Maps”** folder.

**Step 9:**

Take a look at the sample files I have made to show how to set up NIUI in your GameInfo class (see NIUI\_Main), how to tap into the event pump (see NIUI\_SamplePlayerController) and how to set your pawn up to use the skeleton controller(see NIUI\_SamplePawn).

Also, have a quick look at the provided Anim-Tree document in the “**Documentation**” folder so you can get an idea of how to wire up the bone overriding. You need to have an animation that puts your mesh into the OpenNI compliant TPose so that the SkeletonController can cache the rotations. See the sample content packages and the NIUI\_SamplePawn.uc file to see how this is done.

I strongly recommend that you read the “NITE Algorithms 1.3.pdf” that is within the **Documentation/Recommended Reading** folder. This explains some of the key concepts and underlying knowledge that is needed when working with OpenNI.

**Step 10:**

Finally, thanks for taking the time to download my work ☺.

What I have done is far from complete and I will keep building onto the API as I go. This is an ***alpha***version of the API so do not expect the functionality to be rock solid. I have several features in mind for the beta and the final version’s which I am aiming towards.

Also, I currently have not attempted to build an installer for an OpenNI game so I am expecting problems with this. I will be working on a tutorial for this in the Beta version of the API.

If you use this in your projects then please put the OpenNI logo into your splash screen.

If you have a bug, error or suggestion, either send me a PM on the Unreal Forums or leave a message in the thread that has been started in the Released Projects section of the forums. My username is OneThought.