The PrimeSense technology that drives Microsoft's Kinect for Xbox 360 is undoubtedly the next big thing. To make prototyping for the Kinect more accessible, here is a tutorial to get the Kinect to send sensor data to Flash on the Mac. The sensor data sent in this example is position info of one hand (x,y,z) and the wave gesture at 30fps.

The tutorial is written and tested for Flash developers that know very little or nothing about C++ and the Command Line Interface. In this example, Kinect uses a socket server to speak to Flash. Any other technology that supports a socket connection (HTML, Processing, Max/MSP, Silverlight, etc) could use this same setup to get Kinect info.

We are not responsible for any problems that make arise while following this tutorial. However, we will do our best to respond to questions within the comments section below and will update the tutorial if needed.

Kevin Ewe / Adam Glazier @ IDEO
-- 02-02-11

0. What is Terminal?
If your not familiar with Terminal, it is an application located in /Applications/Utilities that allows you to type Command Line code. Command Line code does just about anything you can imagine, but for this tutorial we will use it to install some things and run applications.

You might ask yourself "can't I just double click on and app to open it?" or "can't I just double click on the installer to install it?". The answer is no. Terminal will install and open things that would not be possible the normal way (with a mouse in the Finder). Don't worry though because we'll hold your nooby hand step by step.

Note that when you see a path that looks like this "/Users/<user-name>/Projects/Kinect", this implies that you will change the path in order to fit your personal folder path (eg: /Users/joeybuttercup/Projects/Kinect).

1. Install latest version of Xcode
http://developer.apple.com/technologies/tools/xcode.html

2. Get Homebrew
If your not familiar with Homebrew (http://mxcl.github.com/homebrew), it makes installing, updating and managing necessary components much easier. Having Homebrew installed will likely come in handy in the future for other projects.

To install Homebrew, open Terminal and type:
```
ruby -e "$(curl -fsSLk https://gist.github.com/raw/323731/install_homebrew.rb)"
```

Note that some installs within this tutorial will take several minutes and have little feedback within Terminal. Just be patient and wait for a sign that it's complete (usually a new blank line with "localhost:~ user-name$" and a flashing cursor) before quitting Terminal or giving up.

A common roadblock to installing Homebrew is that you might need to set folder permissions for Homebrew to install correctly. To change permissions, open Terminal and type the code below and then try to install Homebrew again (step 2.2).

```
sudo dscl /Local/Default -append /Groups/staff GroupMembership $USER
```

In case of other errors, see the full set of install instructions at: https://github.com/mxcl/homebrew/wiki/Installation

3. Get Dependencies & Tools

3.1 Install Git
If your not familiar with Git, it is a free and open source distributed version control system. It's commonly used with a hosted source code control service such as github in order to have the most recent code library upload and/or download updates. Having Git installed will likely come in handy in the future for other projects.

Install Git by typing this into Terminal:

```
brew install git
```

3.2 Install cmake by typing this into Terminal:

```
brew install cmake
```

If you have trouble installing cmake, you might need to rename the conflicting expat.framework. To do this, type this into Terminal:

```
sudo mv /Library/Frameworks/expat.framework /Library/Frameworks/expat.framework.old
```

then entered your password

3.3 Install freenect and libusb-freenect
Install libfreenect formula to Homebrew by typing these commands into Terminal:

```
cd /usr/local/Library/Formula
curl --insecure -O "https://github.com/OpenKinect/libfreenect/raw/master/platform/osx/
homebrew/libfreenect.rb"
curl --insecure -O "https://github.com/OpenKinect/libfreenect/raw/master/platform/osx/
homebrew/libusb-freenect.rb"
brew install libusb-freenect
brew install libfreenect
```

4. Download Software

At the moment, this tutorial relies on a compiled binaries from the unstable branch (http://www.openni.org/downloadfiles).

Download OpenNI:
http://www.openni.org/downloadfiles/openni-binaries/latest-unstable/24-openni-unstable-build-for-macosx-10-6-universal-x86x64-3264-bit-v1-0-0/download
Install by:

```
cd into the openni folder e.g. …./OpenNI-Bin-MacOSX-v1.0.0.25
sudo ./install.sh
```

Download Middleware:
http://www.openni.org/downloadfiles/openni-compliant-middleware-binaries/latest-unstable/45-primesense-nite-unstable-build-for-for-macosx-10-6-universal-x86x64-3264-bit-v1-3-0/download

```
cd into the NITE folder e.g. ../Nite-1.3.0.18
sudo ./install.sh
```

Download Sensor:
http://www.openni.org/downloadfiles/openni-compliant-hardware-binaries/latest-unstable/38-primesensor-module-unstable-build-for-macosx-10-6-universal-x86x64-3264-bit-v5-0-0/download
*****You can possibly skip this step and download the uncompiled version of sensor and use that instead***

Download Uncompiled Sensor:
The precompiled sensor binary didn't work for us, so we will have to compile this in step 5.
https://github.com/ros-pkg-git/Sensor

Download Flash / PrimeSense Sample:
https://github.com/drspin/kinect-socket

Unzip and copy all folders to your working folder (e.g. /Users/user-name/Kinect).

5. Compile Sensor:

Navigate to the Sensor folder (should be in your working folder) by typing this into Terminal:

```
cd /Users/<user-name>/Kinect/ros-pkg-git-Sensor-894cea0/Platform/Linux-x86/CreateRedist
```

Run the RedistMaker script by typing this into Terminal:

```
./RedistMaker
```

This will compile everything and create a Redist package in the directory:
/Users/<user-name>/Kinect/ros-pkg-git-Sensor-894cea0/Platform/Linux-x86/Redist

It will also create a distribution in the directory:
/Users/<user-name>/Kinect/ros-pkg-git-Sensor-894cea0/Platform/Linux-x86/CreateRedist/Final

Navigate to the new Redist package by typing this into Terminal:
cd /Users/<user-name>/Kinect/Platform/ros-pkg-git-Sensor-894cea0/Linux-x86/Redist"

Install sensor by typing this into Terminal:
sudo ./install.sh

6. Copy the .xml files to /Data directory
You'll need to copy all the .xml files from Data/ from the github download to your /Nite-1.3.0.18/Data directory.

7. Plug-in the Kinect or PrimeSense hardware:
To be completely clear, you will need the USB adapter for the Kinect. The USB adapter comes with the Kinect box, but not with the Xbox 360 / Kinect bundle. For the bundle, you will have to buy the adapter online.

8. Try a Sample
Before we get Kinect sending data to flash, make sure Kinect is working itself. There were a lot of steps above, so let's make sure it all works before blaming it on Flash.

Plug in your Kinect to the USB port and also make sure to plug in the power adapter, otherwise it will not work. Note that if you put your computer to sleep with the Kinect connected, it will likely freeze upon waking up and any unsaved data will be lost.

Navigate to a sample directory by typing this into Terminal:
cd /Users/user-name/Kinect/Nite-1.3.0.18/Samples/Bin

Run a sample by typing this into Terminal:
./Sample-Boxes
Perform the focus gesture and you should be able to see a slider bar at the bottom of the boxes.

If you see errors or nothing at all occurs, go back step by step to see if you missed something in this tutorial or ask a friend to figure out what went wrong.

9. Get Kinect talking to Flash
To get Kinect data piped to Flash, a socket connection needs to be established. To do this, we have a JavaScript file socket-server.js that opens and establishes the connection, then you open our SingleControl_Flash sample app called SingleControl, then open Flash.

Note: You will need flash player version 10.1

Before starting anything, you will need to change the privacy settings within Flash to allow the socket connection to work. To do this, go to this url:
http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html

Then select "Add Location" from the "edit locations" dropdown. Then browse for the Flash SWF file that will utilize the socket server and make sure it has a green checkbox for "Always allow".

Navigate to the JavaScript file by typing this into Terminal:

```
cd /Users/user-name/Kinect/SingleControl_Flash
```

Note that the next three steps **must be done in this particular order.

1) Run the socket-server.js by typing this into Terminal:

```
node socket-server.js
```

2) Run the SingleControl sample app by typing this into Terminal:

```
./SingleControl
```

3) Open the Flash SWF file that will talk to the socket server. (note: currently, if you exit the flash app, most likely the node socket server will die, just start it all up again. We'll write better code soon!)

You're Alive!

If all of the steps above went properly, you should be seeing a dot tracking to your hand. Now you're setup to use Flash to rapid prototype wild ideas that would normally take 2-10 times longer in C++ and OpenGL.

As mentioned above, if you would like to use this setup (Kinect / OpenNI / Socket Server) with technologies other than Flash, repeat Step 8 and replace the Flash stuff with code from another technology that supports socket connections (HTML, Processing, Max/MSP, Silverlight, etc).

We're excited to make prototyping for PrimeSense (Kinect) more accessible, so please share back how your using this within the comments.