



## 1. What is Kinect?

Kinect for Xbox 360, or simply Kinect, is a motion sensing input device by Microsoft for the Xbox 360 video game console.

Based upon a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with the Xbox 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands.

## 2. Some Demos

The first demo shows two persons making gestures or dancing. The color of the figure conveys depth information. For each figure, the Kinect OpenNI interface can track a number of body parts and display the match-stick skeleton consisting of tracking points connected by straight lines.

The second demo shows how a robot car can be controlled by gestures. The screen on the right shows the gestures to control the robot car.

## 3. How to install it?

We will use an open source driver provided by OpenNI.

- Please download the source file at [OpenNI stable master tree](#)

Please see ReadMe file(inside downloaded files) for detailed install instructions.

Currently Kinect supports Windows, Linux, Mac, Android.

(Since you must start from scratch to install everything, this approach is the most labor-intensive and prone to mistakes. However, once you are done you have installed the most up-to-date version.)

- Alternative way to install under Windows

1. Download Driver, OpenNI and PrimeSense files from [cs1530 Google Code download page](#)
2. Unzip the zip file and install driver under "BIN" folder
3. Install OpenNI and PrimeSense respectively (Please use "0KOIk2JeIBYCIpWVnMoRKn5cdY4=" as key when prompted)

(Since you are taking advantage of work done by CS1530 students in the past, this approach will save you a lot of time. However the installed version may not be the most up-to-date.)

- [Alternative way to install under Linux](#)

(You can also take advantage of work done by CS1530 students in the past, to install a version under Linux. Again, the version may not be up-to-date.)

## 4. How to test it?

To test it, please finish installation and connect Kinect Device to your computer,

then compile and run sample programs contained in the installation package (under OpenNI/Sample and OpenNI/platform/YOURSYSTEM/build/sample)

For Visual Studio user, there are precompiled VS project in OpenNI/platform/Win32/build/sample

### Frequent Error Messages

#### **"Failed to open XML, Error: Unknown USB Device Speed"**

You need to use the power adapter for the Kinect camera when using a regular USB port on a PC

#### **"error initializing NITE"**

NITE isn't installed properly. Make sure you have used the correct license key. When it requests for the license key use the one provided by Primesense : 0KOIk2JeIBYCIpWVnMoRKn5cdY4=

#### **"Can't create any node of the requested type!"**

NITE isn't installed properly. Make sure you have used the correct license key provided

I keep getting a message telling me I need .NET Framework 4.0.

This error can be fixed by installing with the .exe installer first then reinstalling with the .msi installer. If you use the auto-installer above you should not see this error.

#### **"Kinect Motor Driver not found. LED & Motor control disabled"**

You need to manually install the motor/ LED driver for the Kinect device.

## 5. [How to develop Kinect application using recorded Kinect data?](#)

(You probably do not need to use this option)

## 6. How to present your demo?

You can give a live demo. However, just in case Murphy's Law strikes again, it is always advisable to also pre-record your demo and upload it to Youtube.

kinectxbox



## 7. How to use Kinect Interface?

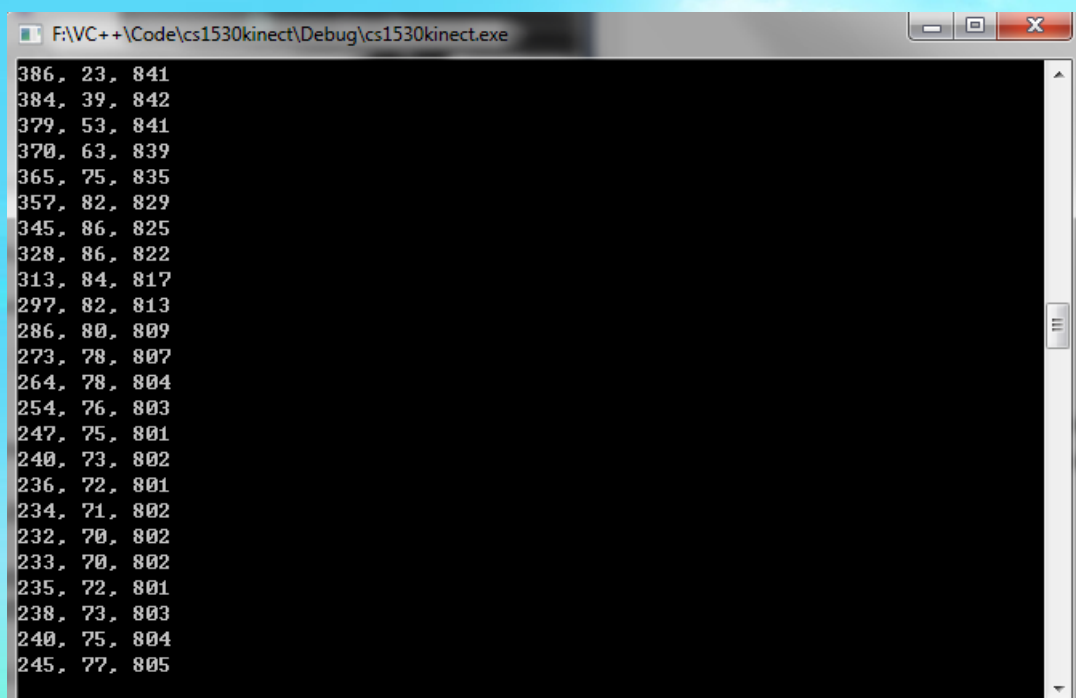
After initial calibration, Kinect is able to track human movement in the form of match-stick skeletons.

The OpenNI Interface will produce the position of each body part of interest (head, hand, elbow, etc.) in the form of its coordinates (x,y,z). In what follows, updateBodyInfo() pushes the device to grab next frame from camera and update BodyInfo in kinect object, and getBodyInfo() retrieves the BodyInfo from kinect object.

A typical main program may look like the following:

```
int id = 0x01;
Kinect *kinect = new Kinect(id); //create kinect object
BodyInfo body; //declare BodyInfo object
kinect->initKinect();           //init kinect object and start the
device
while(1) {
    kinect->updateBodyInfo();      /*request the device to
get a new frame from camera and update BodyInfo*/
    body = kinect->getBodyInfo();  /*retrieve BodyInfo
from kinect object*/
    if(body.bTracking) /*bTrack indicate whether the device
has confidence about what it is tracking*/
        printf("%.0f, %.0f,
%.0f\n",body.pRHand.X,body.pRHand.Y,body.pRHand.Z);
    /*pRHand.X is X coordinate of right hand*/
    //else
        //printf("No Body Info Updated\n");
}
delete kinect; //clean up
```

The above program produces the following output (the coordinates of successive right hand positions):



- All source code are saved on [google code webpage](#) (See next section for download instruction)

To develop components based upon the Kinect interface, you need to include BodyInfo.h/cpp, Kinect.h/.cpp, KinectFunction.h and Setup\_XML.xml.

Also, make sure your program link to libraries under "lib" folder and add "include" as additional include folder.

- Java Wrapper is also available on "Kinect Java Wrapper". The whole package is an Eclipse sample project.

## 8. Where to download the source code?

We will use google code for version control. [These are detailed instructions on how to access google code svc server.](#)

Our course google code url is "https://cs1530-kinect.googlecode.com/svn/trunk/"

kinectxbox



## 9. How to provide an add-on Kinect Interface?

For some applications it is often more convenient to have both a web interface and an add-on Kinect Interface that provides some of the functions of the web interface. Therefore a user can use the Kinect Interface for certain essential functions and fall back on the web interface for other functions.

For the following example, the source code is available on google code: <http://code.google.com/p/cs1530-kinect/downloads/list>  
A video demo can be found on YouTube: <http://www.youtube.com/watch?v=nmFo8dTNghw>

In the demo, after the Kinect calibration phase a 'start' gesture (a wave of the right hand) will start the application. A 'stop' gesture (a horizontal left hand pointing at a vertical right hand) will stop the application.

When the GestureDetector in the [main program](#) HHAMain.cpp detects a gesture such as the 'start' hand gesture, the sendCommand(ICMD\_START) outputs "HOME HEALTH APPLICATION START" and starts the application. When the 'stop' gesture is detected the sendCommand(ICMD\_STOP) stops the application and outputs "HOME HEALTH APPLICATION STOP".

The user can also use the web interface to do the same. When the user clicks on the "Start Application" button on the web page [test1.html](#), the associated JavaScript function runCmd('start') (see the source for test1.html) sends a 'start' message to c:\Program Files\HomeHealthApp\HHARemoteClient.exe. The [HHARemoteClient](#) sends ICMD\_START through a socket to the main program. The main program HHAMain maintains three threads to interact with Kinect device, display device and HHARemoteClient. Upon reception of ICMD\_START, sendCommand(ICMD\_START) in the main program outputs "HOME HEALTH APPLICATION START" and starts the application. It is the same for the 'stop' command. Please note the web page for this example only works properly on IE browser.

The application developer can add more buttons to the web page test1.html, more commands to HHARemoteClient.cpp, and more processing functions to sendCommand() for a realistic application. GestureDetector() must also be expanded to detect additional hand gestures and body gestures. For this purpose a structure called BodyInfo containing the 3D-coordinates of all body parts that can be tracked by Kinect is provided. The program is [BodyInfo.cpp](#), and the corresponding structure is [BodyInfo.h](#). More information can be found in the following [Readme.txt](#).