## How to Run the Example

1. Download and install OpenNI, NITE and OpenNI-Kinect driver as described in the next section.
2. Open scene 'SensorKinectOpenNI', located in Assets-folder.
3. Run the scene. Both avatars are connected to the 1st Kinect user (look at KinectManager-script).
4. Try to change some parameters of KinectManager-script, attached to 'MainCamera' and AvatarController, attached to 'U_Character_REF' avatars, and then to re-run the scene.

## Installation of Kinect sensor with OpenNI 1.5 (Windows)

1. Download and install OpenNI v1.5.4, **32-bit** version. Here is the download link: www.openni.org/wp-content/uploads/2012/12/OpenNI-Win32-1.5.4.0-Dev1.zip
2. Download and install NITE v1.5.2, **32-bit** version. Here is the download link: www.openni.org/wp-content/uploads/2012/12/NITE-Win32-1.5.2.21-Dev.zip
3. Download OpenNI-compliant Kinect-sensor driver. Here is the download link: https://github.com/avin2/SensorKinect/archive/unstable.zip
4. Unzip the Kinect-sensor driver.
5. Connect the Kinect sensor to a USB port of your computer.
6. When the OS asks for a device driver, select the following sub-folder of the unzipped OpenNI-Kinect driver-folder: *Platform/Win32/Driver*.
7. Install the Kinect-driver for OpenNI. Run '*SensorKinect093-Bin-Win32-v5.1.2.1.msi*', located in the '*Bin*'-subfolder of the unzipped OpenNI-Kinect driver-folder.
8. Check your installation. Try to run one or more of the OpenNI examples and one or more of the NITE-examples. They should run normally, without errors.

## How to Reuse the OpenNI Kinect-Example in Your Own Unity Project

1. Copy folder '*KinectScripts*' from Assets-folder of the example to the Assets-folder of your project. This folder contains the 3 needed scripts – *KinectWrapper.cs, KinectManager.cs* and *AvatarController.cs*
2. Copy folder '*Resources*', containing '*OpenNI.xml*' and '*Plugins*', containing '*UnityInterface.dll*' from the Assets-folder of the example to the Assets-folder of your project.
3. Run Unity and open your project.
4. Add 'AvatarController'-script to each avatar (humanoid character) in your game that you need to control with the Kinect-sensor.
5. Drag and drop the appropriate bones of the avatar's skeleton from Hierarchy to the appropriate joint-variables (Transforms) of 'AvatarController'-script in the Inspector.
6. Uncheck 'Mirrored Movement', if the avatar should move in the same direction as the user. Check it, if the avatar should mirror user movements.

7. Add ‘KinectManager’-script to the MainCamera. If you use multiple cameras, create an empty GameObject and add the script to it. Script’s Start()-method initializes OpenNI, Update()-method updates all Kinect-controlled avatars.
8. Drag and drop the avatars from Hierarchy to the ‘Player 1 Avatars’ list.
9. If you need a 2nd Kinect-user to control avatars, check ‘Two Users’ in the parameters of ‘KinectManager’-Script in the Inspector. If this is the case, repeat steps 4-6 for each avatar, controlled by the 2nd user. Repeat step 8 as well, but this time for ‘Player 2 Avatars’ collection.
10. Check ‘Display User Map’-checkbox, if you want to see the User-depth Map after the user calibration completes.
11. Save and run your game.

## References

This example is based on the following example from CMU.edu. A big “Thank you” to their authors:

- http://wiki.etc.cmu.edu/unity3d/index.php/Microsoft_Kinect_-_Open_NI
- https://github.com/OpenNI/UnityWrapper/archive/master.zip

## Support:

E-mail: firu@fhv.at; Skype, Twitter: roumenf