## How to Run the Example

1. Download and install OpenNI 2 and NITE 2, as described in the next section.
2. Open scene 'SensorKinectOpenNI2', located in Assets-folder.
3. Run the scene. Both avatars are connected to the 1st Kinect user (look at KinectManager-component of MainCamera).
4. Try to change some parameters of KinectManager-component of 'MainCamera' and AvatarController-component of the avatars, and then to re-run the scene.

## Installation of Kinect sensor with OpenNI 2 (Windows)

1. Download and install OpenNI 2, **32-bit** version. Here is the download link: http://www.openni.org/openni-sdk/
2. Download and install NITE 2, **32-bit** version. Here is the download link: http://www.openni.org/files/nite/
3. The drivers for Kinect and PrimeSense sensors are installed during the OpenNI/NiTE installations.
4. Connect the Kinect or PrimeSense sensor to a USB port of your computer.
5. The OS should find and use one of the previously installed drivers. You should be able to see the installed sensor under the PrimeSense-section in Control Panel/Device Manager.

## How to Reuse the OpenNI2 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*' 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 ‘Compute User Map’ and ‘Display User Map’-checkboxes, if you want to see the User/Depth Map on the screen. Check only ‘Compute User Map’, if you want to use the user/depth-texture in your project, but don’t want to display it on the screen.
11. Check ‘Compute Color Map’ and ‘Display Color Map’-checkboxes, if you want to see the User/Depth Map on the screen. Check only ‘Compute Color Map’, if you want to use the color-texture in your project, but don’t want to display it on the screen.
12. Save and run your game.

## References

This example is an extension of 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 and Feedback

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