How to Use Kinect v2 VR Examples

Prerequisites

In order to run the 'Kinect v2 VR Examples' you need to download the ready-made build of KinectDataServer from the download section here: https://rfilkov.com/2016/05/07/kinect-v2-mobile-vr-examples/ Alternatively, you can run or build the KinectDataServer scene by yourself, if you have the 'Kinect v2 Examples with MS-SDK'-package.

It is required, because the Kinect-related components in the VR package are light-weight. They don't use the Kinect sensor directly, but instead receive the sensor data over the network, from the KinectDataServer. The data-server must be run on the machine, where the Kinect-v2 sensor is connected.

If you plan to build for Android, you also need to install:

1. Android Studio, which can be downloaded and installed from here: http://developer.android.com/sdk/index.html It will also require a recent version of Java SE SDK.
2. Android build support for Unity. To install it, run again the Unity installer here: http://unity3d.com/get-unity/download Then, in the list of components, unselect all and select only 'Android build support'.
3. To enable developer mode on your Android device, open its Settings, tap on 'About device', then on 'Software info'. Then tap on 'Build number' multiple times, until the developer mode turns on. Go back to 'About device'. Select 'Developer options' and make sure that 'USB debugging'-setting is enabled.

If you want to build for iOS, please read the Unity documentation on building for this platform, and make sure you have the appropriate developer license from Apple.

On the Server Machine ('Kinect v2 Examples with MS-SDK')

1. Download and install the Kinect SDK 2.0 from here: https://www.microsoft.com/en-us/download/details.aspx?id=44561
2. Connect the Kinect-v2 sensor to it.
3. Download and run the KinectDataServer from the download section here: https://rfilkov.com/2016/05/07/kinect-v2-mobile-vr-examples/
4. Allow the server to use the machine networks (if you are asked by Windows).
5. Alternatively, if you have the latest version of the "Kinect-v2-Examples-with-MS-SDK"-package, you can open and run the KinectDataServer-scene from there.

On the Client Machine (could be the same as the server machine)

1. Start Unity. Create a new project.
2. Import the "Kinect-v2 VR Examples"-package into the project.
3. Open 'Build settings' and switch to 'Android', 'iOS' or 'PC, Mac & Linux Standalone' platform.
4. Open one of the demo-scenes in Assets/Kinect2MobileVr/DemoScenes-folder.
5. If the server is running on the same WLAN subnet, it should be discovered automatically when the scene runs. You don't need to change any server-related settings in this case.
6. Alternatively, you can manually set the 'Server host' and 'Server port'-settings of KinectDataClient-component of the KinectController-object, if you know the Kinect data server's IP-address and port.
7. Run the scene in the editor to make sure it works. This means the configured server-settings are correct.

8. Repeat steps 3-6 for the other demo scenes.

Short Descriptions of the Available Demo Scenes

1. KinectAvatarDemo1, located in Kinect2MobileVr/DemoScenes/AvatarDemo-folder. This is first-person avatar demo. Move your arms or legs, and try to look at them, to see how the sensor tracks you.
2. KinectAvatarDemo2, located in the same folder. This is third-person avatar demo. Again, move your arms and legs, move or turn a bit left or right, to see your mirrored movements from 3rd person perspective.
3. FlyerGestureDemo, located in Kinect2MobileVr/DemoScenes/GestureDemo-folder. Lean left or right to move the flyer horizontally - left or right. Jump or squat to move it vertically - up or down.
4. KinectGestureDemo1, located in the same folder. This is the discrete gestures' demo. Swipe left, swipe right or swipe up turn the presentation cube in the respective direction.
5. KinectGestureDemo2, located in the same folder. This is the continuous gestures' demo. Use the Wheel-gesture to turn the model left or right, or Zoom-in/Zoom-out gestures, to scale the model. Lower your hands between the gestures, to stop the previous continuous gesture.
6. VisualGestureDemo, located in the same folder. This is a very basic demo that allows you to check how the visual gestures, configured at server-side, get recognized.
7. KinectInteractionDemo1, located in Kinect2MobileVr/DemoScenes/GesturesDemo-folder. Use your left or right hand to control the hand-cursor. Grab an object & drag it around. Open your hand to release it. Try to interact with the UI components, too.
8. KinectInteractionDemo2, located in the same folder. Grab the cube with your left or right hand. Then turn it in all directions, to look at all its sides.
9. SnowflakeShooterDemo, located in the same folder. Look at the falling snowflakes. Close your left or right hand to shoot the snowflake you're looking at. Keep in mind your shooting hand must be high enough, so the sensor could clearly see if it is opened or closed.
10. KinectSpeechRecognition, located in Kinect2MobileVr/DemoScenes/SpeechRecognitionDemo-folder. Say clearly one of the listed commands to control the robot. The grammar is configured at server side.

Build for Android Mobile Platforms

To build scenes for Android, you need to have 'Android Studio' and 'Android build support' installed. See the Prerequisites-section above for more information and download links.

1. Select 'File / Build Settings' from the main menu, to open the 'Build Settings'-dialog.
2. Add the scene you want to build into the 'Scenes in Build'-list.
3. Select Android as Platform.
4. Press the 'Player settings'-button and in section 'Other settings' set 'Bundle identifier', according to your (company) name and product name. This is a requirement of the Android SDK.
5. Connect your mobile device and press the 'Build and Run'-button. You will be asked to give a name for the Android's apk-file.
6. If your mobile device is recognized as Android one, Unity will build the apk-file, copy it to the device, install and run it, all in one step.
7. If your mobile device is not recognized, press the 'Build'-button to build the apk-file only. Then copy the apk-file to the mobile device, install and run it manually.

Build for Virtual Reality Platforms

Building for virtual-reality platforms depends on the type of VR-platform:

1. If you build for Unity supported VR platforms, like Oculus or GearVR, you need to do as follows:
    a. Enable 'Virtual Reality Supported'-setting in 'Player settings / Other settings'
    b. For Oculus & GearVR: Place Oculus signature (osig) file for your device into 'Assets/Plugins/Android/Assets'-folder. To generate osig-file for your device, go to: https://developer.oculus.com/osig/ and follow the instructions.
    c. For other platforms: Check if there are any platform-specific requirements.
2. If you build for Google Cardboard, do as follows:
    a. Download and import the Cardboard SDK from here: https://developers.google.com/cardboard/unity/download
    b. Drag the StereoController-script from Cardboard/Scripts-folder to the MainCamera in your scene. Alternatively, you can delete the MainCamera in the scene and replace it with Cardboard/Prefabs/CardboardMain-prefab.
3. Select 'File / Build Settings' from the main menu, to open the 'Build Settings'-dialog.
4. Press the 'Player settings'-button and make sure the 'Default orientation'-setting in 'Resolution and presentation'-section is set to 'Landscape left'.
5. Connect your mobile device and press the 'Build and Run'-button. You will be asked to give a name for the Android's apk-file.
6. If your mobile device is recognized as Android one, Unity will build the apk-file, copy it to the device, install and run it, all in one step.
7. If your mobile device is not recognized, press the 'Build'-button instead to build the apk-file only. The copy the apk-file to the mobile device, install and run it manually.
8. If you build for GearVR, you will be asked to insert the mobile device into GearVR HMD, to see the VR scene. If you build for Google Cardboard, the app will start instantly on the mobile device, in stereo mode. You can then insert the device into the cardboard, to see the VR scene.

How to Use the Mobile/VR Components in Your Own Unity Projects

1. Copy folder 'Kinect2MobileVr/KinectScripts' from the Assets-folder of this package to the Assets-folder of your project. This folder contains all Kinect-related scripts, filters and interfaces.
2. If you build for Oculus or GearVR, copy Plugins/Android-folder from the Assets-folder of this package to the Assets-folder of your project. This folder contains the Oculus signature files.
3. In your scene, create a KinectController-object and add the 'KinectManager'-script as component to it.
4. Add the 'AvatarController'-script as component to each avatar (humanoid character) in the scene, which you want to control via the Kinect-sensor.
5. Add the other needed Kinect components, like InteractionManager, KinectGestures (this is the gesture-manager component), or gesture-listeners.
6. You may also use the public API of 'KinectManager', 'InteractionManager' and other components in your scripts. See the components in the available demo scenes, if you need examples. All Kinect-related components are usually part of the KinectController-game object or the avatar objects.
7. More tips, trick and examples are available here: http://rfilkov.com/2015/01/25/kinect-v2-tips-tricks-examples/

Additional Reading

The following how-to tutorial is also located in the Assets/_Readme-folder of this Unity-package:

- Howto-Use-Gestures-or-Create-Your-Own-Ones.pdf

More Information, Support and Feedback

Web: https://rfilkov.com/2016/05/07/kinect-v2-mobile-vr-examples/
Tip and Tricks: http://rfilkov.com/2015/01/25/kinect-v2-tips-tricks-examples/

Contact: http://rfilkov.com/about/#contact
Twitter: https://twitter.com/roumenf