How to Use Kinect v2 Examples with MS-SDK

1. Download and install Kinect-v2 SDK as described in the next section.
2. Import this package in a new Unity project.
3. If you want to utilize the included shaders instead of CPU image processing, make sure that Direct3D11 is the first option in the 'Graphics API'-list, in Player Settings / Other Settings / Rendering.
4. If you want to utilize the Kinect-v2 speech recognition, download and install the Speech Platform Runtime or SDK, as well as the needed language packs, as described in the next section.
5. Open and run a demo scene of your choice. See Kinect-demos short desriptions below.

Installation of Kinect v2 SDK

1. Download the Kinect for Windows SDK 2.0. Here is the download page: http://www.microsoft.com/en-us/download/details.aspx?id=44561
2. Run the installer. Installation of Kinect SDK 2.0 (or Kinect Runtime 2.0) is simple and straightforward.
3. Connect the Kinect-v2 sensor. The needed drivers are installed automatically.
4. If you want to use the Kinect-v2 speech recognition, download and install the MS Speech Platform Runtime v11 (or Speech Platform SDK v11). Install both x86 and x64-packages, to be on the safe side. Here is the download page: http://www.microsoft.com/en-us/download/details.aspx?id=27225
5. For the Kinect-v2 speech recognition, you also need to download and install the respective language pack. Here is the download page: https://www.microsoft.com/en-us/download/details.aspx?id=43662

Kinect-Demos Short Descriptions

- KinectAvatarsDemo1, located in KinectDemos/AvatarsDemo-folder. Move around to see how the avatars and the cube-man reflect your movements. Try one or more of the suggested gestures.
- KinectAvatarsDemo2, located in the same folder. See how the first-person avatar (and the cube-man) reflects your movements. Look at your arms and legs.
- KinectAvatarsDemo3, located in the same folder. The avatar in this scene utilizes AvatarControllerClassic-component and the OffsetNode-setting (stays related to this object).
- KinectBackgroundRemoval1, located in KinectDemos/BackgroundRemovalDemo-folder. See how the cut-out user's image is mixed with the background texture.
- KinectBackgroundRemoval2, located in the same folder. Check how to set the cut-out user's image as a 2nd background layer and put 3d-objects behind it (the halo), or in front of it (the cubes).
- KinectBackgroundRemoval3, located in the same folder. Check how to use a separate layer and camera, in order to display images, objects and image effects behind the user's silhouette.
- ColorColliderDemo, located in KinectDemos/ColliderDemo-folder. Touch any of the creatures with your hands to make them jump, as a result of the collision event between them and your hand.
- DepthColliderDemo, located in the same folder. Move around to bounce the falling eggs.
- KinectFaceTrackingDemo1, located in KinectDemos/FaceTrackingDemo-folder. See how the Kinect-generated face model overlays your face on the screen. The model can be textured or not.

- KinectFaceTrackingDemo2, located in the same folder. The hat model moves along with your head.
- KinectFaceTrackingDemo3, located in the same folder. See how the user's head can replace avatar's head.
- KinectFaceTrackingDemo4, located in the same folder. See how the head model follows you in the scene, and how your face expressions affect the rigged head.
- KinectFittingRoom1, located in KinectDemos/FittingRoomDemo-folder. Stand in T-pose for calibration. See how the selected clothing model overlays your body on the screen.
- KinectFittingRoom2, located in the same folder. Stand in T-pose for calibration. See how the humanoid model overlays your body on the screen. You can replace it with your own model.
- KinectGesturesDemo1, located in KinectDemos/GesturesDemo-folder. Swipe left, right or up turn the presentation cube left, right or up. You can also check the 'Seated' visual (VGB) gesture here.
- KinectGesturesDemo2, located in the same folder. Use the Wheel-gesture to turn the model left or right, or Zoom-in / Zoom-out gestures, to scale the model. Don't forget to lower your hands between the gestures.
- KinectInteractionDemo1, located in KinectDemos/InteractionDemo-folder. Use your left or right hand to control the hand-cursor on the screen. Grab an object to drag it around. Open your hand to release it. Try to interact with the GUI components, too.
- KinectInteractionDemo2, located in the same folder. Grip the cube with your left or right hand. Then turn it in all directions, to see it from all sides.
- Scene0-StartupScene, located in KinectDemos/MultiSceneDemo-folder. Select 'File / Build Settings' from the menu and add Scene0, Scene1 and Scene2 to the 'Scenes in Build'-list. Then run the startup scene. You will see how to use the Kinect-related components across multiple scenes in a game.
- KinectOverlayDemo1, located in KinectDemos/OverlayDemo-folder. Move your right hand. See how the green ball follows its position on the screen.
- KinectOverlayDemo2, located in the same folder. See how the green spheres overlay the tracked joints of your body, and the lines representing the bones between them.
- KinectOverlayDemo3, located in the same folder. This is a simple 'draw-in-the-air' application. Close your right hand to start drawing. Open it to stop. Press 'U' to undo the last drawn line.
- KinectRecorderDemo, located in KinectDemos/RecorderDemo-folder. Say 'Record' to start recording your body movements, 'Stop' to stop it, or 'Play' to play the previously saved body movements.
- KinectSpeechRecognition, located in KinectDemos/SpeechRecognitionDemo-folder. Say clearly one of the listed commands to control the robot. Repeat it, if needed. Then open the xml-grammar file 'SpeechGrammar.grxml', located in the Assets/Resources folder of the Unity project, and modify the commands, according to your needs. Save the grammar file and run the scene again to try them.
- KinectHeightEstimator, located in KinectDemos/VariousDemos-folder. See how the height of the user can be estimated, based on the Kinect-provided depth information.
- KinectHolographicViewer, located in the same folder. See how the cube perspective changes, when you move left or right to the sensor.
- KinectPoseDetector, located in the same folder. See how the differences in bone angles are calculated and summed up to provide pose matching info.
- KinectSceneVisualizer, located in the same folder. See how the real environment within predefined depth and left/right limits can be mixed with virtual objects.
- KinectUserVisualizer, located in the same folder. See how the user's 3d mesh-model can be added to the virtual scene, and how it interacts with virtual objects in the scene.

Why Are There Two Avatars in KinectAvatarsDemo1-scene

The presence of the two avatars (humanoid characters) in the scene is to demonstrate different options of their AvatarController-components. These are the components that transfers user's movements to the avatar model.

First, see the avatar that looks at you. It mirrors your movements (for instance your left arm is his right one), and moves around the main camera, the same way you move around the sensor. As you can see, its transform's Y-rotation is set to 180 degrees. The AvatarController-component, attached to this avatar's game object has its 'Mirrored Movement'-setting enabled. It also has 'Pos relative to camera' set to the MainCamera, to make it equal to your movement around the sensor.

The left avatar, the one that has turned his back at you, is not mirrored. It reproduces your movements as they are. Your left is his left and your right is his right. Its transform Y-rotation is set to 0 (in order to be in the same direction as you) and the 'Mirrored Movement'-parameter of its AvatarController is disabled. It also has 'Pos relative to camera' set to None. It moves around its initial position, instead of around the camera.

Also note the 'Player index'-setting of both avatars. It specifies the user each avatar tracks - 0 means the 1st detected user, 1 – the 2nd one, etc. If you want any of the avatars to follow another user, just change the 'Player index' setting of its respective AvatarController-component.

How to Reuse the Kinect-related Scripts in Your Own Unity Project (Windows Builds)

1. Copy folder 'KinectScripts' from the Assets-folder of the example to the Assets-folder of your project. This folder contains all needed scripts, filters and interfaces.
2. Copy folder 'Resources' from the Assets-folder of the example to the Assets-folder of your project. This folder contains the needed libraries and resources. You may skip the libraries you don't want to use.
3. Copy folder 'Standard Assets' from the Assets-folder of this package to the Assets-folder of your project. It contains the wrapper classes for Kinect v2.
4. Wait until Unity detects, imports and compiles the newly detected resources and scripts.
5. In your scene, create a KinectController-object and add the 'KinectManager'-component to it.
6. Enable its 'Compute User Map'-setting, if you want to use the body texture or the user-depth texture. Enable 'Compute Color Map', if you want to utilize the color camera texture. The respective Display-settings may be used to display these textures on the screen.
7. Add the 'AvatarController'-component to each avatar (humanoid character) in the scene that you need to control with the Kinect-sensor. See the previous topic for more information regarding AvatarController.
8. You may use the public API of 'KinectManager', 'InteractionManager', 'FacetrackingManager', 'SpeechManager', etc. components in your scripts. Here you can find many tips and tricks: http://rfilkov.com/2015/01/25/kinect-v2-tips-tricks-examples/

Additional Reading

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

1. Howto-Use-Gestures-or-Create-Your-Own-Ones.pdf
2. Howto-Use-KinectManager-Across-Multiple-Scenes.pdfs

More Information, Support and Feedback

Tip and Tricks: http://rfilkov.com/2015/01/25/kinect-v2-tips-tricks-examples/
Troubleshooting: http://rfilkov.com/2014/08/01/kinect-v2-with-ms-sdk/#ki

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