How to Run the Demos

1. Download and install Kinect v2 SDK as described in the next section.
2. Open and run 'KinectAvatarsDemo1'-scene, 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.
3. Open and run the 'KinectAvatarsDemo2'-scene, located in the same folder. See how the first-person avatar reflects your movements. Try to see your avatar's arms and legs.
4. Open and run 'KinectGesturesDemo'-scene, located in KinectDemos/GesturesDemo-folder. Swipe left or right with your hand to turn the presentation left or right. You can also try the 'Seated' VGB gesture here.
5. Open and run 'KinectInteractionDemo1'-scene, located in KinectDemos/InteractionDemo-folder. Use your left or right hand to control the hand-cursor on the screen. Use hand grip to grab an object and drag it around. Open your hand to release the object. Try to drag-and-drop objects with your right hand and with your left hand. Then try to interact with the GUI components.
6. Open and run 'KinectInteractionDemo2'-scene, located in the same folder. Use the hand cursor to grab the cube and turn it in all directions.
7. Open and run 'KinectOverlayDemo1'-scene, located in KinectDemos/OverlayDemo-folder. Watch how the green ball follows the position of your right hand on the screen.
8. Open and run 'KinectOverlayDemo2'-scene, located in the same folder. See how the spheres match the tracked joints, and the lines – the bones between them.
9. Open and run 'KinectFaceTrackingDemo1'-scene, located in KinectDemos/FaceTrackingDemo-folder. See how the Kinect-generated face model overlays your face on the screen.
10. Open and run 'KinectFaceTrackingDemo2'-scene, located in the same folder. See how the Viking's hat overlays your head on the screen.
11. Open and run 'KinectFaceTrackingDemo3'-scene, located in the same folder. See how the user's face is tracked and displayed as a texture on screen.
12. Open and run 'KinectSpeechRecognition'-scene, located in KinectDemos/SpeechRecognitionDemo-folder. Say clearly one of the commands to control the robot. Repeat, if needed. Try the other listed commands. Then open the xml-grammar file 'SpeechGrammar.grxml', located **in the root folder** of the Unity project, and modify the voice commands, according to your needs. Save the grammar file and run the scene to try it.
13. Open and run 'KinectBackgroundRemoval'-scene, located in KinectDemos/BackgroundRemovalDemo-folder. See how your image is mixed with the background texture.
14. Open 'Scene0-StartupScene', located in KinectDemos/MultiSceneDemo-folder. Add Scene0, Scene1 and Scene2 to the 'Scenes in Build'-list in 'File / Build Settings'. Then run the startup scene (or make a build and run it). This set of scenes demonstrates how the Kinect-related managers can be reused across multiple scenes in a game.
15. Open and run 'KinectFittingRoom1'-scene, located in KinectDemos/FittingRoomDemo-folder. Stand in T-pose for calibration. See how the clothing models overlay your body on the screen.
16. Open and run 'KinectFittingRoom1'-scene, located in the same folder. Stand in T-pose for calibration. See how the humanoid model overlays your body on the screen.

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/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 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 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

Why Are There Two Avatars in the Scene

The meaning of the two avatars (3D humanoid characters) in the scene is to demonstrate that you can have both – mirrored and non-mirrored movements in your scene.

First, you can have an avatar that mirrors your movement. This is the one facing you in the example scene. As you can see, its transform has Y-rotation (rotation around Y-axis) set to 180 degrees. Also, there is an AvatarController-component, attached to the avatar's game object and its 'Mirrored Movement'-parameter is enabled. Mirrored movement means that when you, for instance, lift your left hand the avatar lifts his right hand and vice versa, like in a mirror.

The second avatar, the one that has his back turned at you, is not mirrored. It reproduces your movements exactly as they are. Your left is his left and your right is his right. Its transform has Y-rotation set to 0 and the 'Mirrored Movement'-parameter of its AvatarController is disabled.

In order to get correct avatar position and movement, first position and rotate the avatar's game object in the scene, as needed. Then attach AvatarController-component to the avatar's game object and set its 'Mirrored Movement'-parameter accordingly.

How to Reuse the Kinect-related Scripts in Your Own Unity Project

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 not copy the libraries you don't want to use.
3. Copy folder 'Standard Assets' from the Assets-folder of the example to the Assets-folder of your project. It contains the wrapper classes for Kinect v2.
4. Wait until Unity detects and compiles the newly copied resources and scripts.
5. Add 'KinectManager'-component to the MainCamera. If you use multiple cameras, create an empty game object and add the KinectManager-component to it.
6. Enable 'Compute User Map' and 'Display User Map'-parameters, if you want to see the user-depth map on screen. Enable 'Compute Color Map' and 'Display Color Map'-parameters, if you want to see the color camera image on screen. Enable 'Display Skeleton Lines' parameter, if you want to see how Kinect tracks the skeletons on the user-depth map.
7. Add the 'AvatarController'-component to each avatar (humanoid character) in the scene that you need to control with the Kinect-sensor. Enable 'Mirrored Movement'-parameter of the AvatarController, if the avatar should mirror user's movements.

8. (Optional) Drag and drop the avatars' game objects from Hierarchy to the 'Avatar Controllers'-list parameter of KinectManager. Otherwise they will be detected automatically to the list, at the scene start-up.

9. You can use the public functions of 'KinectManager', 'InteractionManager', 'FacetrackingManager' and 'SpeechManager' in your scripts. Examples: 'GestureListener.cs' and 'PresentationScript.cs' used by the KinectGesturesDemo-scene, 'GrabDropScript.cs' used by the KinectInteractionDemo-scene, 'KinectOverlayer.cs' used by the KinectOverlayDemo-scene, 'ModelFaceController.cs' used by the KinectFaceTrackingDemo-scene or 'BotControlScript .cs' used by the KinectSpeechRecognition-scene.

Additional Reading

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

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

More Information, Support and Feedback

More information: http://rfilkov.com/2015/01/25/kinect-v2-tips-tricks-examples/

E-mail: rumen.filkov@gmail.com

Web: http://rfilkov.com

Skype: roumenf

Twitter: roumenf