README --- HoloLens D3D Keyboard v1.2d

Overview: This Unity Asset Package features a full-size 3D keyboard for Microsoft HoloLens app developers, to be used in D3D build, for in-App username / password / url entries. Organized as Drag-and-drop prefabs. Hand-crafted keyboard includes: uppercase, lowercase, digits, and most symbols available on a standard English keyboard. Special keys: "⇧": Shift; "□": Space; "⇦": Backspace; "⏎": Return, starting on a new line. Non-special keys can be easily customized. Built-in with Show/Hide (green "done") button and Move/Pin voice commands for HoloLens. Four original recordings of key-typing sound.

A test scene is included for demonstration purposes using Gaze, Gesture, and Voice on HoloLens.

Please note: A recent version of HoloToolkit is included in the test scene for demo purposes only and is licensed by Microsoft under MIT license, but this Unity Asset Package may not contain the latest version of HoloToolkit. You may download the latest version: http://github.com/Microsoft/HoloToolkit-Unity but there is no guarantee that the test scene or scripts will work with the latest version of HoloToolkit. Version 1.2d supports Unity 2017.2.1f1 and HoloToolkit / MixedRealityToolkit v2017.2.1.0. Version 1.2d fixed breaking changes which occur when upgrading Version 1.2c to Unity 2017.2 or newer.

Setup Instructions:

1. Import this Unity Asset Package into a blank new project.
2. Open "KeyboardOne.unity" scene file under the main folder named "D3DKeyboard".
3. File > Build Settings > Click on "Universal Windows Platform" from the list and then "Switch Platform" button.
4. Select from dropdown lists on the right: SDK = "10.0.14393" or higher, Build Type = "D3D".
5. Click on "Add Open Scene" button, and make sure the scene "D3DKeyboard/KeyboardOne" is checked.
6. Click on "Player Settings", and make sure "Virtual Reality Supported" is checked under "XR Settings", and "Windows Mixed Reality" appears in the list of "Virtual Reality SDKs".
7. Under "Publishing Settings", make sure "Microphone" is checked.
8. Click on "Build" button, and create a new folder named "App" and click "select folder".
9. Unity will start building the project.
10. Once Unity finishes building the project, go to "App" folder, and open the solution file named "D3D Keyboard.sln" in Visual Studio 2017.
11. Please make sure your computer and your HoloLens are connected to the same Wifi network.
12. Select from drop down lists: "Release", "x86", "Remote machine", and enter the IP address of your Microsoft HoloLens. Then click on Build > Deploy Solution.

Typing:

1. Look at a key steadily such that the circular "Gaze" cursor is on the key, and perform an air tap "Select" gesture. There should be a typing sound and color change as feedback.
2. The typed character will be added to the Text field of the Text Mesh component of gameObject "InputTextDisplay" under "keyboardone" prefab, and thus displayed above the keyboard.

Voice commands:

1. “Move keyboard”: Keyboard will be placed at a distance of 1.5 metre in front of you, while facing you and following your movement.
2. “Pin keyboard”: Keyboard will stop moving and stay where it was in space.
3. “Scale up keyboard”: Keyboard will scale up by 10% (i.e. new scale = old scale * 1.1)
4. “Scale down keyboard”: Keyboard will scale down by 9.09% (i.e: new scale = old scale/1.1)
5. “Leave note”: A copy of current text will be created and left in the holographic space.
6. “Remove note” while gazing at the top row of a previously left note: The note will be removed.

Special keys:

“⇧”: Shift between lowercase and uppercase keyboard displays and inputs;

“□”: Space;

“⇦”: Backspace;

“⏎”: Return, starting on a new line;

“done” green button: Shows / Hides the keyboard.

Scripts: under “Scripts” folder

1. KeyboardGG.cs: This script is attached to each gameObject representing a key, handing user interactions with each key
2. KeyboardMain.cs: This script is attached to the top level gameObject “keyboardone”, handling keyboard level changes such as shifting between uppercase and lowercase displays, showing and hiding the keyboard, and moving and pinning the keyboard.

Prefabs: Keyboard prefabs are in the “Models” folder

Sounds: Key-typing sound files are in the “Sounds” folder

Other: Courier font is used for the display of characters overlaying each key. This is recommended for equally spaced display of characters, to match the positions of the gameObjects of keys.

Customization:

Customization to the special keys would require changes to the scripts (which is not part of the scope of this documentation).

To customize a non-special key (i.e. keys other than “⇧”, “□”, “⇦”, “⏎”, and “done” green button):

1. Locate the key which needs modification under the gameObject named “keyboardSet” within the hierarchy of “keyboardone”. The gameObjects for non-special keys are named with 5 characters: “key”+ 1 character for lowercase + 1 character for uppercase.
2. Change the lowercase and uppercase character part of the name to the characters of your choosing. E.g. “keyaA” to “keybB”
3. Locate the original lowercase and uppercase characters in the Text field of the TextMesh component of gameObject “keyboardDisplayL” and “keyboardDisplayU”, respectively, under “keyboardSet” gameObject and change them to the desired lowercase and uppercase characters respectively.