Holo SDK for Industry/Industry+ and Leonardo headsets Dev Manual

Sommario

1. Purpose

The Holo SDK is a set of Unity scripts and native libraries to make it usable all the main features of AR headset by Youbiquo Srl.

2. Current Version

This manual refers to the Holos SDK version 0.98.1.

3. Requirements

The Holo SDK has been developed and tested with Unity 2019.4.20f1. Later versions of 2019.4 LTS branch of Unity should work.

In order to use AR capabilities, please use Vuforia SDK (version 9.8.5 tested and working).

4. The Holo SDK

The Holo SDK is a set of assets inside a Unity project.

4.1. Setup

1. Create an empty Android project
2. Close unity
3. Replace the Packages and Project Settings folders of the project with the ones provided with the Holo SDK ones.
4. Import the Holo SDK unitypackage in the project.
5. A sample scene is present in the directory Assets/Holo SDK/Scenes you can use to study the framework.

4.2. Using the SDK components

The Holo SDK includes a prefab in the directory Assets/Holo SDK/Prefabs called HOLO SDK. This prefab is already present in the sample scene described above.

The Holo SDK prefab consists of a set of game objects and components ready for use, that takes advance of all the capabilities of the headset.

In most cases, all useful components have already been added to this hierarchy, but you can add any Holo component manually using the Inspector of the Unity editor. All the components are included in the Holo_SDK.dll file. All the components have the prefix “Holo” in the names. A complete list of the components is reported in the DOC folder of the SDK.

Figure 1 - Holo SDK Hierarchy

4.3. Main components

4.3.1. Head Rotation

The HoloHeadRotation component allows to reproduce the rotation of the user head to any game object in scene. By default the main camera has a HoloHeadRotation component, so any change of view of the user is reported to the main virtual camera in the 3D scene. More info about the component options are reported in the DOC folder.

4.3.2. Gaze Pointer

Holo SDK a gaze pointer which allows th user to interact with interact with 3D and UI game objects in scene.

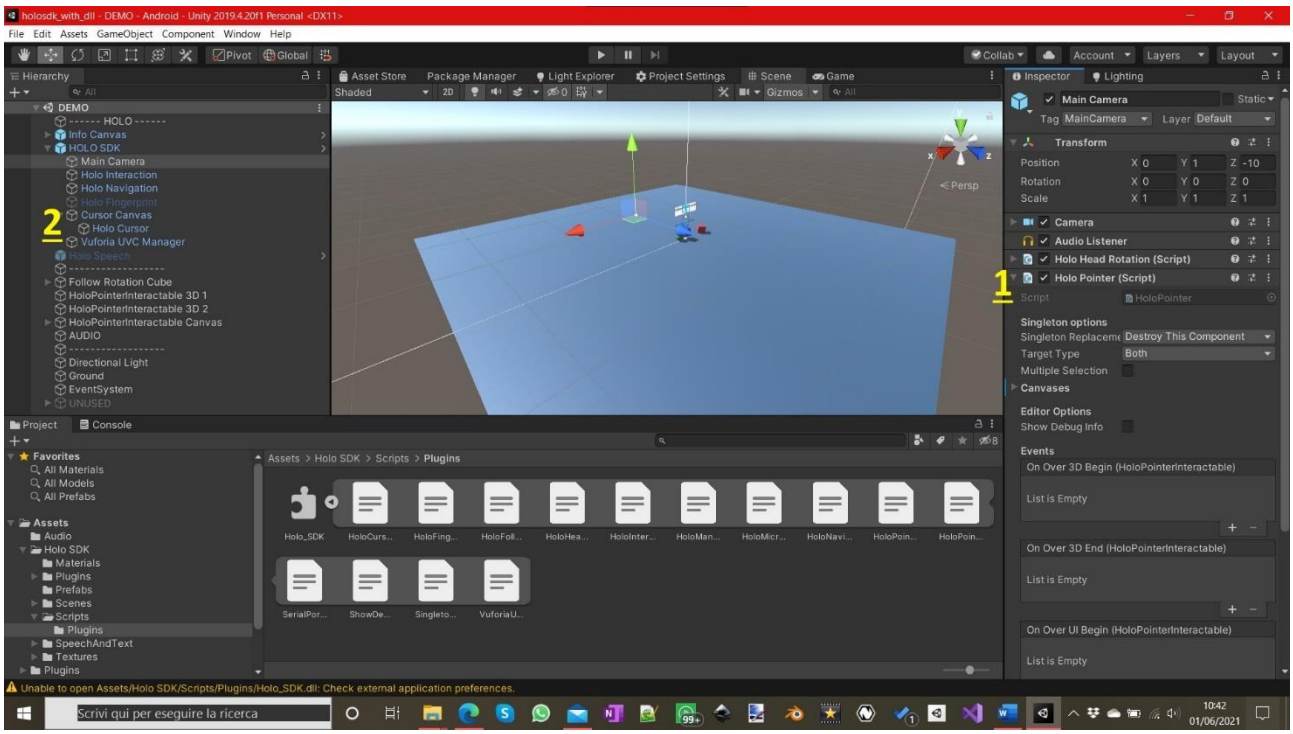

Figure 2 - Gaze pointer system

It is made up of two main components:

The Holo Pointer (1 in the figure above)

This component allows the user to select and interact with game objects in scene.

By default, it is not possible to interact with each game object in scene, but there are 2 conditions to be met:

1. The user should set the Target Type in the HoloPointer component. It can be 3D, UI or Both (default) to make it work with 3D objects and UI items.
2. The object the user wants to interact with, should have a component derived from the abstract class HoloPointerInteractable. This derived class should implement all the abstract methods of the superclass. In the Scripts folder you can find the HoloPointerTargetTest script that works with both 3D and UI items.

The Holo Cursor (2 in the figure above)

The HoloCursor component can be used as a marker that you can control. Basically it is a UI sprite that can change color and size. You can control it through the HoloPointer events exposed in the editor (OnOverBegin, OnOverEnd, etc.).

4.4. UX

The UX is implemented from the HoloInteractionManager on the Holo Interaction game object.

Figure 3 - UX component

It contains several options to customize the interaction between the user and the system. The user can for instance enable/disable double tap, triple tap, customize the tap, etc. For further info please move the mouse on the fields of the component in the Unity editor (to read the tooltips) and consult the DOC.

4.5. Navigation

The Holo Navigation component contained in the Holo SDK hierarchy contains basic behaviours about navigation. Actually, it allows the use to choose if the double tap implements the back operation. Moreover, it exposes 2 events to be invoked when back and quit are performed. At the moment, the default behaviour consists of a Debug.Log.

4.6. Fingerprint

This functionality is not fully implemented still.

4.7. Other components

- ShowDebugGUI: a helper component used to write advanced logs on the screen.
- Switcher2d3d: a component that changes the working mode of the headset.
- HoloFollowRotation: allows a game object to follow any rotating target (for example the camera) to keep it visible. It uses some threshold angles and speed.

To get more info about these components, please refer to the DOC.

5. Using Vuforia AR

The Holo SDK is compatible with Vuforia SDK.

5.1. Setup

1. Import the Vuforia package from the package manager (version 9.8.5 tested and working). After import, check that VUFORIA_ANDROID_SETTINGS is present in Project Settings/Player/Other Settings/Scripting Define Symbols. If not, add it manually.
2. Deactivate game object HOLO SDK-> Main Camera and copy the Holo components to the Vuforia main camera.
3. Make sure that the Vuforia UVC Manager game object is active in the HOLO SDK hierarchy.

Figure 4 - The Vuforia UVC Manager