NDVIConfig Unity Project Setup

Mark Scherer, September 2018

Setup

Local setup of the project entails the following steps:

1. Clone the [NDVIConfig assets folder](https://github.com/VUSE-Hololens/assets/tree/master/NDVIConfig) of the VUSE-Hololens/assets repository.
2. Open Unity and create a new project locally. Name it NDVIConfig or whatever app name you want.
3. Follow the usual Unity project setup:
   1. Import the Mixed Reality Toolkit package.
   2. Apply Mixed Reality Toolkit project and scene settings.
4. Open Scene1 from the Scenes folder in assets. This will contain ‘prefabs’ of all the gameobjects included in the scene, however many (maybe all?) dependencies are lost and will have to be setup again. A prefab is just a gameobject with predefined dependencies, setting, inspector variables, etc.
   1. Two MRTK files are overwritten by custom files in the NDVIConfig assets folder: SpatialMappingObserver.cs and SpatialMappingSource.cs. Navigate to assets/Holotoolkit/SpatialMapping/Scripts/ and delete the two MRTK versions of these files.
5. Now the prefab dependencies must be reconfigured. This involves dragging components into the gameobject’s inspector panel. This is outlined for each necessary gameobject below:
   1. EFP: The EFP gameobject should contain the following components: EFPDriver.cs, MeshManager.cs, SpatialMappingOberserver.cs and SpatialMappingManager.cs.
   2. Diagnostics: should contain: DiagnosticsControl.cs
   3. Menu: The menu parent gameobject should contain: MenuControl.cs, SliderTester.cs (optional) and ButtonTester.cs (optional).
      1. The MenuBackground child gameobject should contain: Cube (Mesh Filter), Box Collider and Mesh Renderer. These are not custom made components and can be found by clicking ‘Add Component’ at the bottom of the inspector panel and searching.
      2. The child gameobject Content and its children Buttons and Sliders should each contain Object Collection, also found by searching (see 5.c.i).
      3. All further children should be setup correctly already.
   4. DataReciever: should contain ReceiverDriver.cs.
   5. Lighting: should be correctly set up, but if not should contain Light, also found be searching (see 5.c.i).