Thomas Brown

02/23/17

UWB\_RoomTexture Project Status

Current status:

* Framework for enabling / disabling voice commands – **NEEDS TESTING** (Speech recognition unavailable on desktop – must investigate)
* Framework for enabling / disabling gestures – **NEEDS TESTING**
* PhotoCapture framework for taking pictures / location data:
  + Start camera mode – **NEEDS TESTING**
  + Photo capture – **NEEDS TESTING**
  + End camera mode – **NEEDS TESTING**
* Voice commands recognized – **NEEDS TESTING**
* Gestures recognized – **NEEDS TESTING**
* Framework for generating textures – **NEEDS TESTING**
* Framework for saving textures – Tested
* Framework for loading textures - Tested
* Framework for manipulating textures – Tested
  + Flipping of textures capability – Tested
  + Alpha value clipping of textures – Tested
* Framework for recognizing camera location – **NEEDS TESTING**
* Framework for saving camera location (“CameraLocation”) data – Tested
* Framework for loading camera location data – Tested
* Framework for dynamically generating Materials – Tested
* Framework for dynamically generating Projectors – **NEEDS TESTING**
* Framework for dynamically generating Texture2D arrays for use with sample shader – Tested (with one texture/camera location combination)
* Framework for automatically changing the layer of Meshes / RemoteMapping constructed meshes – Tested
* Framework for automatically assigning the correct Material / shader to a mesh / RemoteMapping constructed mesh - Tested
* Ability to dynamically assess most recent texture capture in Unity Editor – **NEEDS TESTING**
* Shader (Sample) – **NEEDS TESTING**
  + Blending of pixel values – **Needs implementation**
* Shader (Original / Unity Projector) – **NEEDS TESTING**
  + Alpha clipping of items (i.e. no mipmapping haze by distancing from texture) – **NEEDS RETESTING**
  + Blending of items (BlendOp Max) – **NEEDS RETESTING**
* Ability to package / unpackage AssetBundles for:
  + Textures – **Needs to be implemented**
  + CameraLocation files (as TextAssets) – **Needs to be implemented**
  + Scripts – **Needs to be implemented**
  + Shaders – **Needs to be implemented** (Possible code stripping of shaders – must investigate feasibility of transferring custom shaders through AssetBundles)

What needs to be tested:

* General understanding of Hololens usage / setup
* **PhotoCaptureFrame**
  + Based off of forums and code found online, UnityEngine.VR.WSA.Webcam references the Hololens camera
  + PhotoCaptureFrame should provide location data of an image
  + PhotoCaptureFrame should provide the image used as the raw texture
* **Shader**
* Voice commands / Gestures
* Can be made into an application for Hololens with no issue
* Plays nice with the **UWB Network Manager** (after AssetBundle capability is added in – needs to be differentiated from a specific UWB Network Manager – just need function calls available)

Current issues:

* Windows 10 installation being slow / difficult (need to see if laptop is bloated after installation)
* **Must manually set read/write capabilities for textures in Unity**
  + ImportSettings code breaks Unity – must investigate
* **There is NO way for Unity to generate layers on the fly so users must create that layer by themselves at startup if using the Unity Projectors route (only currently applies if more than 30 textures exist in the clipped texture file AND this may get phased out later anyways)**
* Method for getting default resolution of Hololens camera breaks Unity
  + May be a specific issue with Unity and LINQs or may be an issue with not having the Hololens plugged in
    - Workaround = Hardcode the resolution and camera’s Field of View according to documentation found online for its default
    - Workaround can be easily adjusted by changing the Resolution() method in Camera file of Constants namespace
* Deleting a texture does not delete its counter-part camera location file – should create helper prefab to manage deletions appropriately (easy to do manually, but might as well implement)
* Extreme difficulties getting GitHub to properly sync project like other projects currently up – Applied differences to file and generated .gitignore, but it’s still utterly destroying my project upon download – gonna have to see if there’s something I’m doing wrong for this
  + Using .unitypackage to reimport package on laptop to showcase changes in the meantime
* Have to work on a paper for my class that is gonna eat up a lot of free time

Perceived issues:

* May be some bugs in code for the Hololens specific stuff – should be very easy to fix
* May be some bugs in understanding of how Hololens camera saves stuff / makes it available – moderately easy to fix depending on level of discrepancies
* AssetBundles might mess up shader – must decide what to do
* AssetBundles packaging / unpackaging not currently implemented
  + Expecting to be able to send an AssetBundle to another client with a specific destination folder
  + Expecting to be able to have some tag / name for the AssetBundle recognizable by a script to unpack the AssetBundle appropriately
  + If these aspects of AssetBundle are unavailable, may have to rework project for good communication of necessary assets
* Must return Hololens by Monday or Tuesday after work
  + Must grab Taran or Keng’s phone number to contact them after arriving on campus
  + Limited time available to program reasonably (I reserve Sundays for this project, but paper may have to take priority depending on quality of work done for it by that point)
* Application.datapath may have to be changed to Application.persistentDatapath for the root folder being referenced to allow proper referencing as an application
* Using things from the UnityEditor namespace might cause issues with creating and running the application on the Hololens?