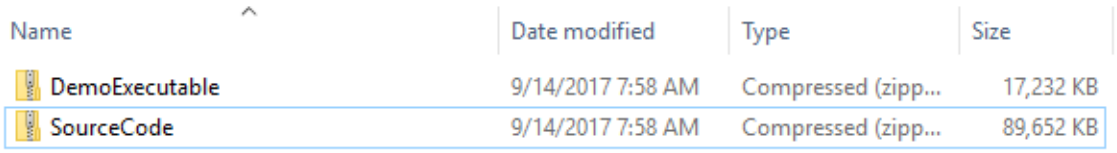
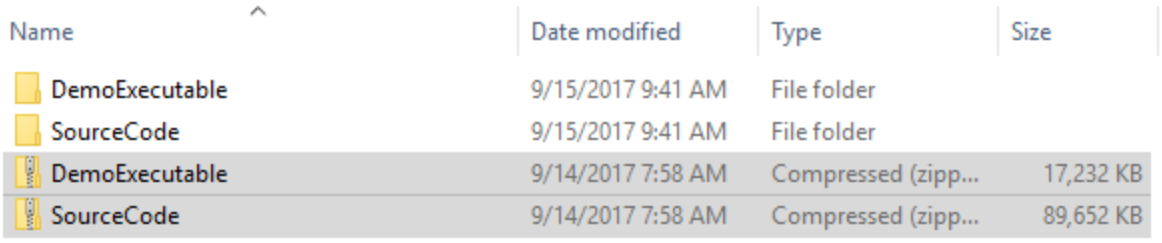
References :
Mapbox : https://www.mapbox.com/mapbox-unity-sdk/docs/00-getting-started.html
VRTK : https://vrtoolkit.readme.io/docs

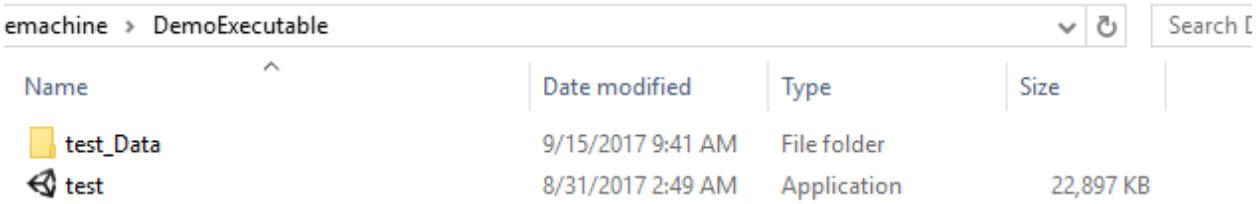
Make new folder
Download files



Unzip files



Before starting to work in Unity try the test app first under Demo Executable



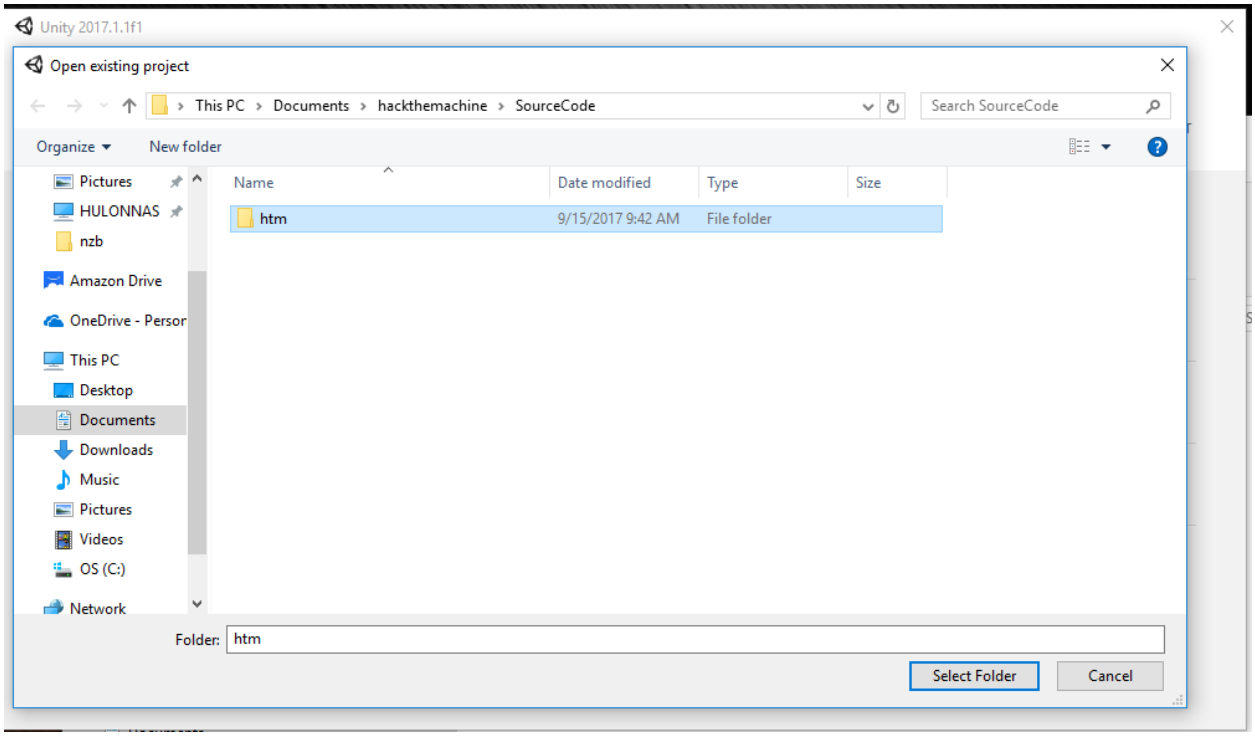
Demo instructions:

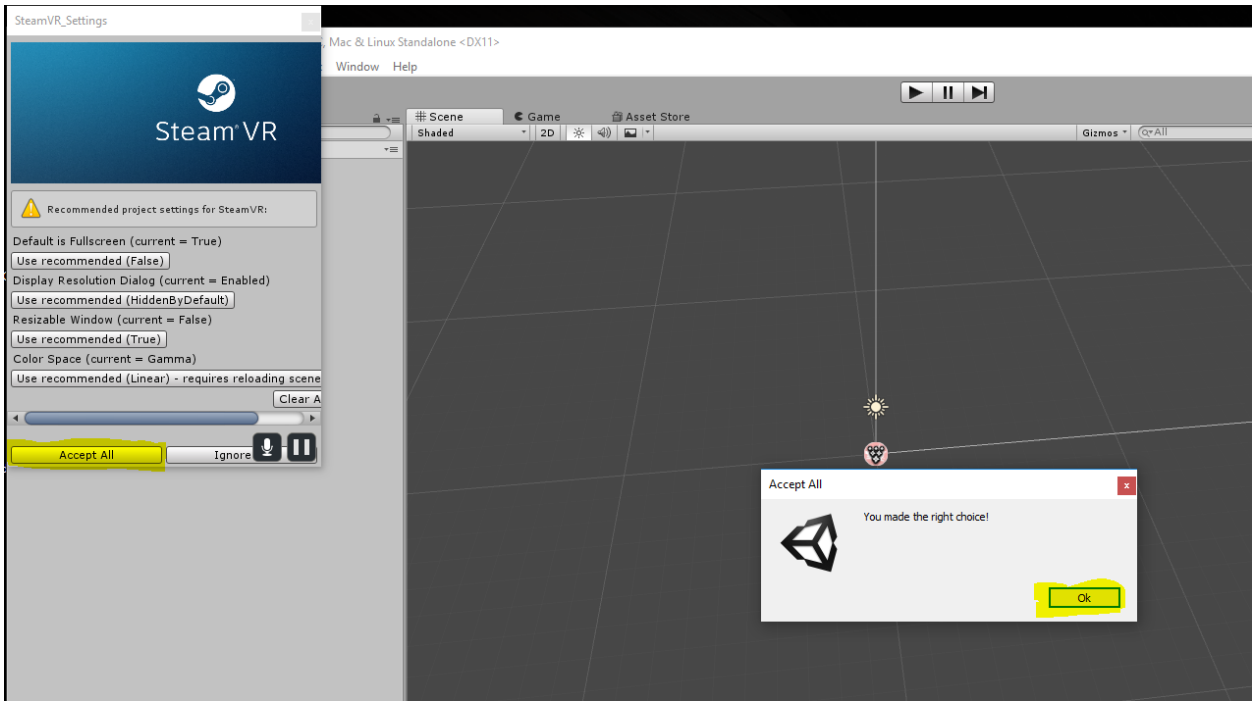
* click on the world to drag it around
* use the scroll wheel on your mouse to zoom in and out
* change the MapBox Style in the dropdown menu to see a few example maps
* click on the waypoints to be teleported to a higher res 3D view of that location
* press space to go back to global view

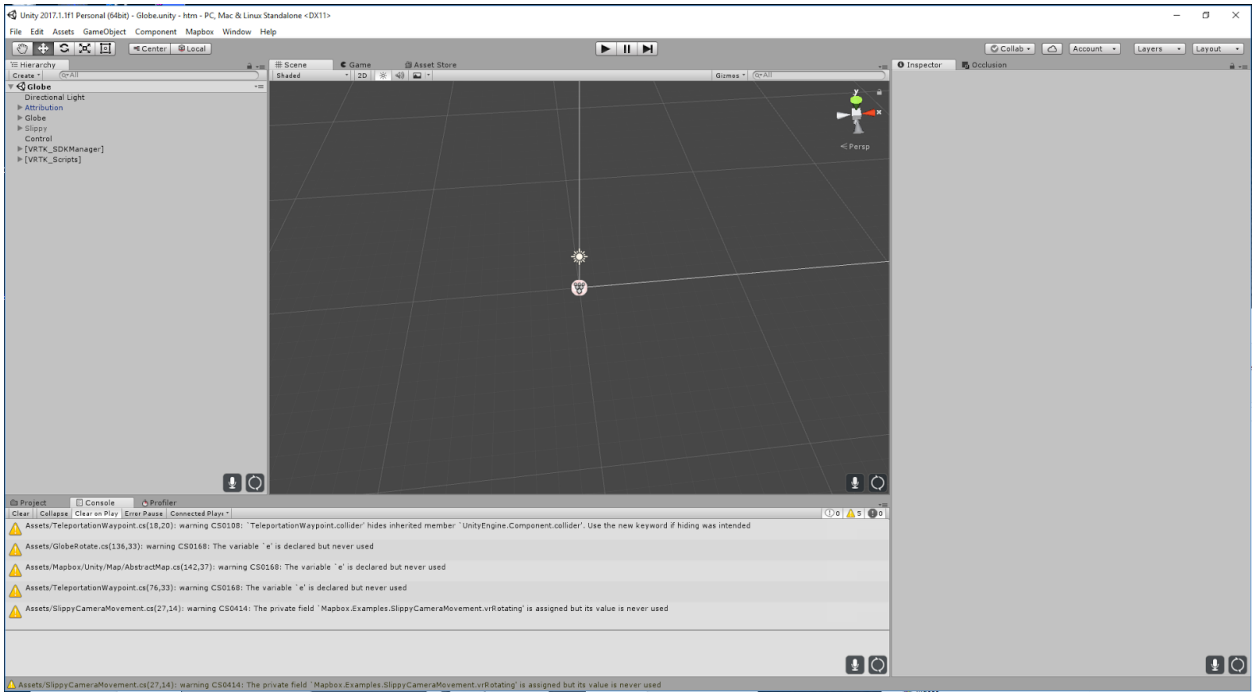
* you can also drag with a Vive controller's trigger button
* press both Vive controllers' triggers and move them towards or away from each other to zoom in and out
* press the Vive controller's top big round button while pointing at a waypoint to teleport there
* you can select MapBox Styles on your left controller in VR, and go back to the Global view (point and click with your other Vive controller)

Open unity 2017.1.1f1

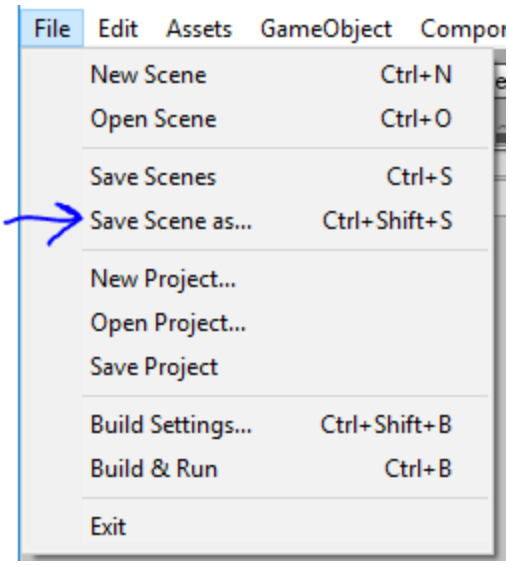
Open sourcecode / htm as folder

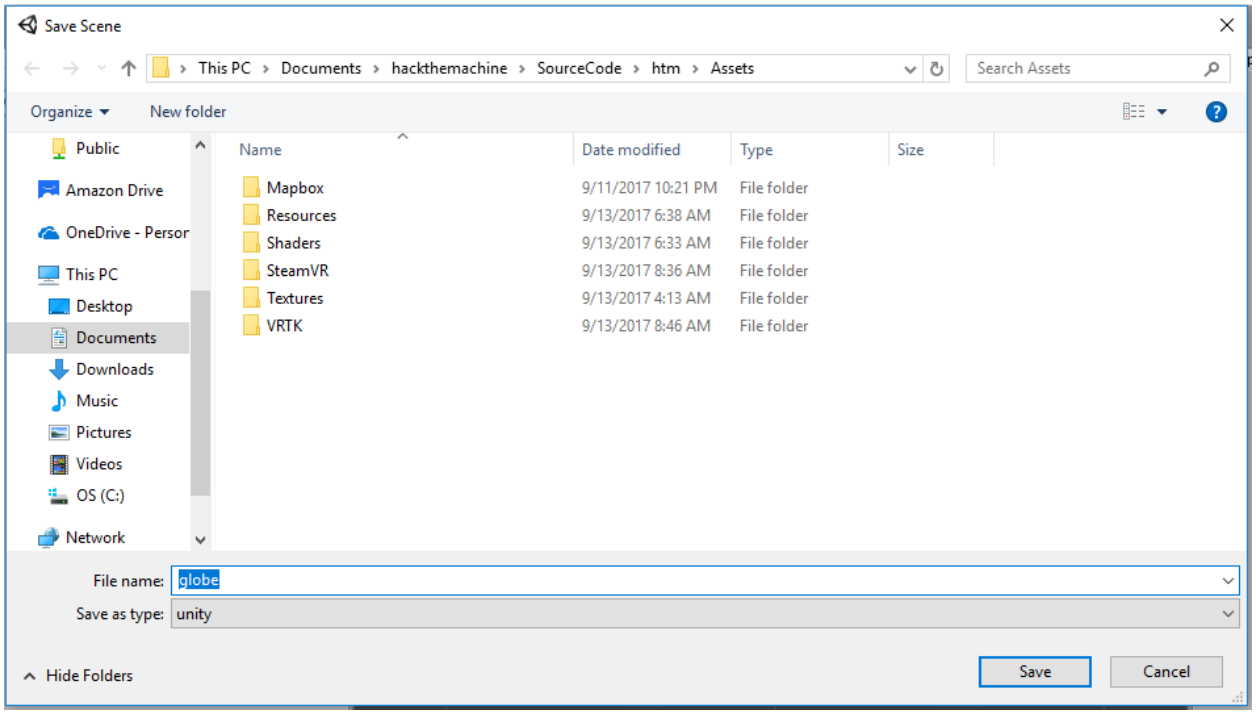

Unity 2017.1.1f1
Open existing project
This PC › Documents › hackthemachine › SourceCode
Search SourceCode
Organize
New folder
Pictures
HULONNAS
nzb
Amazon Drive
OneDrive - Persor
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Network
Name
Date modified
Type
Size
htm
9/15/2017 9:42 AM
File folder
Folder:
htm
Select Folder
Cancel


SteamVR_Settings
Steam VR
Recommended project settings for SteamVR:
Default is Fullscreen (current = True)
Use recommended (False)
Display Resolution Dialog (current = Enabled)
Use recommended (HiddenByDefault)
Resizable Window (current = False)
Use recommended (True)
Color Space (current = Gamma)
Use recommended (Linear) - requires reloading scene
Clear A
Accept All
Ignore
Mac & Linux Standalone <DX11>
Window
Help
Scene
Game
Asset Store
Shaded
2D
Gizmos
Accept All
You made the right choice!
Ok



Save the scene as globe





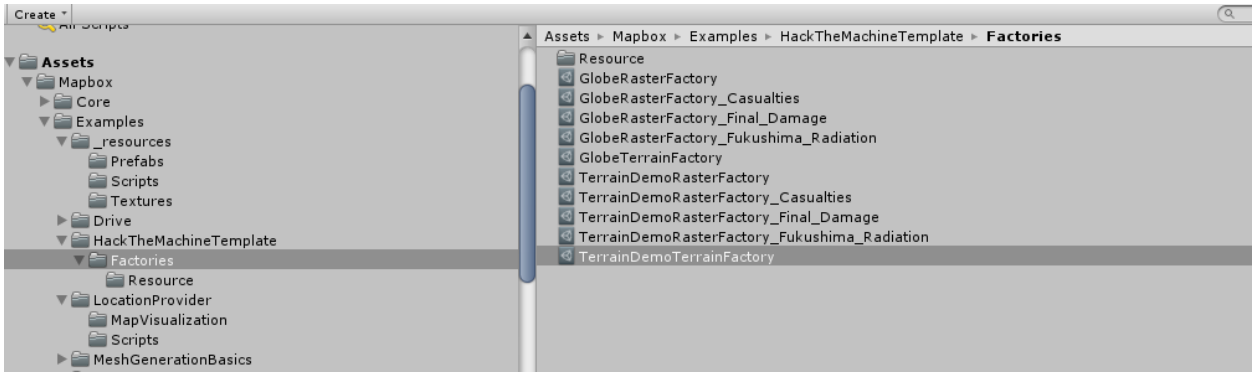
Important Files:
Note that the structures MapBox offers through their SDK take quite a while to load, so unfortunately you will experience delays when shifting between views.

The template project is basically split into the following main components:

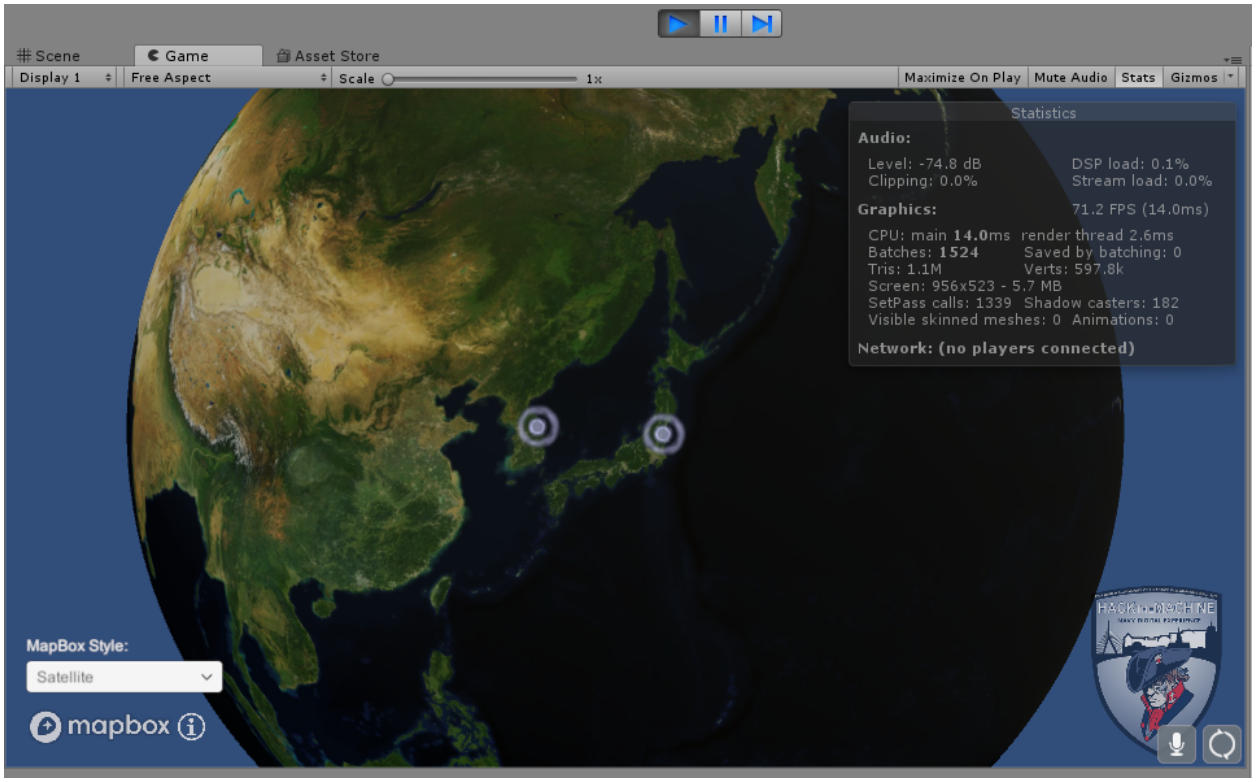
1. Attribution
    a. Required MapBox attribution
    b. HtM logo
    c. MapBox Style dropdown
2. Globe - Handles all spherical representation of globe(s)
    a. Contains GlobeCamera for non-VR globe camera
    b. Multiple DetailGlobeTileProviders that the dropdown will switch between
    c. DetailGlobeTileProvider: built off of MapBox's GlobeTileProvider, but allowing for high detail NSEW bounded area (instead of capping the full globe at certain zoom level, note that MapBox does not support dynamic level of detail)
        i. ExampleMapObjects Holds geo-referenced 3D objects / sprites
    d. GlobeGeoPosition script details where the object should be placed and how it should be oriented

    e. Includes teleportation waypoints (TeleportationWaypoint) that when selected will teleport the user to that GlobeGeoPosition (switching to Slippy maps)
        i. DragSphere is used for interacting with the globe
    f. GlobeRotate script on World GameObject actually handles this
3. Slippy - move around and have tiles dynamically load – note no variable level of detail)
    a. Contains SlippyCamera camera, camera for handling translation tile loading (switches operations based on if VR is enabled or not)
    b. Contains multiple SlippyMaps that are switched between with the style dropdown
    c. ExampleMapObjects contains geo-referenced 3D objects / sprites
4. GeoPosition script details where the object is on the SlippyMap and how it is oriented (note that this differs from the projection GlobeGeoPosition does)
5. Control - master control script for scene initialization and transition handling
6. VRTK - VR library objects
    a. Also includes “VR palette” Canvas in [VRTK_SDKManager] > SDKSetups > SteamVR > [CameraRig] > Controller (left)

Mapbox factory files are here:

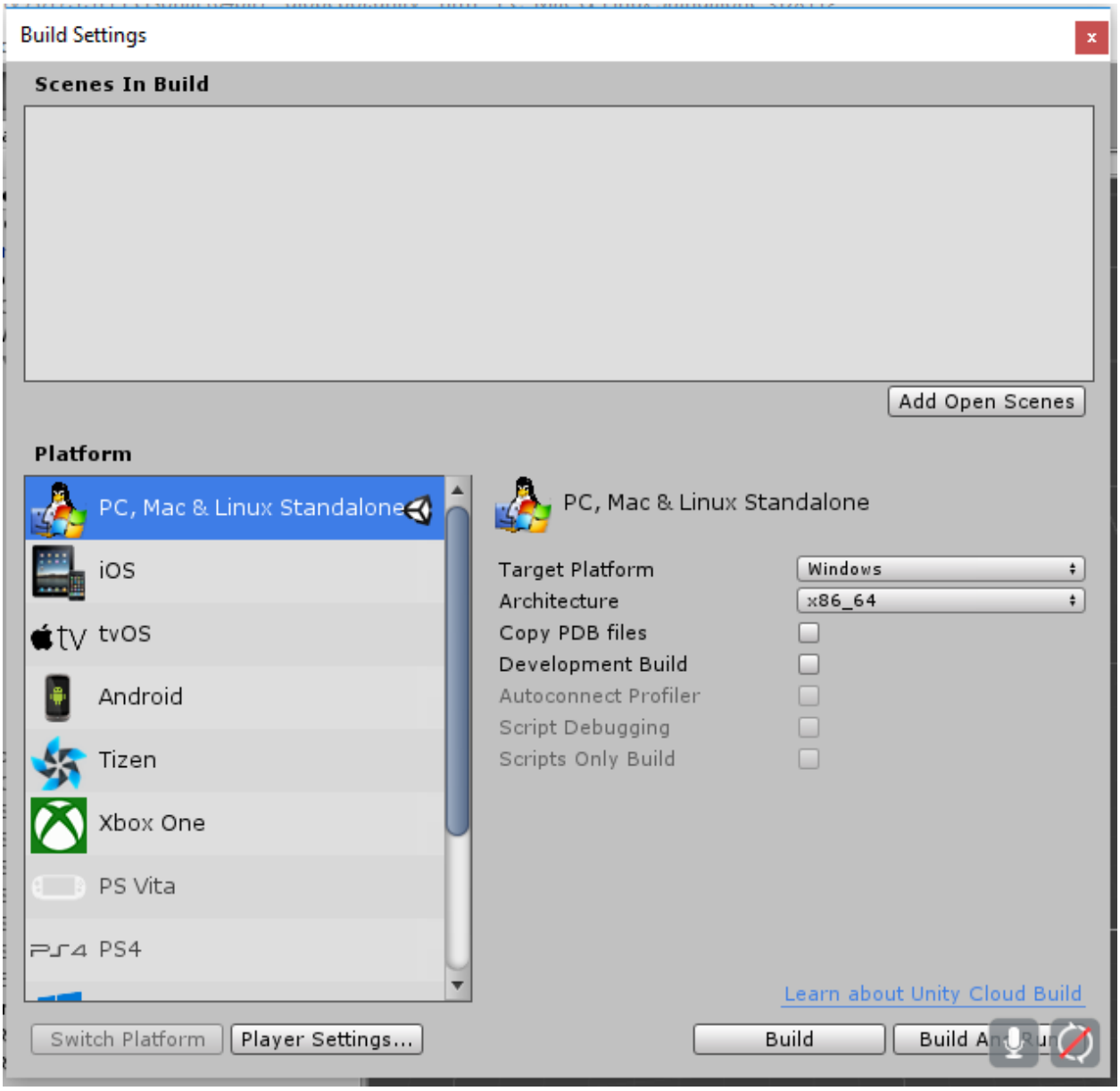


When working on the application you can push the run button to run the application within unity at any time. Click on the button a second time to stop the run. Debugging works while running in this manner



To build the application

File Build Settings -> build and then you can name and execute the build



Lets try to build something useful on top of the baseline. As part of search and rescue it's important to know where hospitals and schools are as they are typically used as shelters. Let's make a new map and highlight where the schools and hospitals are.

First we'll build a new mapbox style with points of interests and a filter with hospitals and schools

Basic-poi

Edit

Published 17 minutes ago • Private •

1 source

Mapbox Streets v7

landuse_overlay z0 - z22
- landuse_overlay_national_park

admin z0 - z22
- admin_country

building z0 - z22
- building

road z0 - z22
- bridge_major
- bridge_major case
- bridge_minor
- bridge_minor case
- road_major
- road_minor
- tunnel_major
- tunnel_minor

poi_label z0 - z22
- poi_label

waterway z0 - z22
- waterway

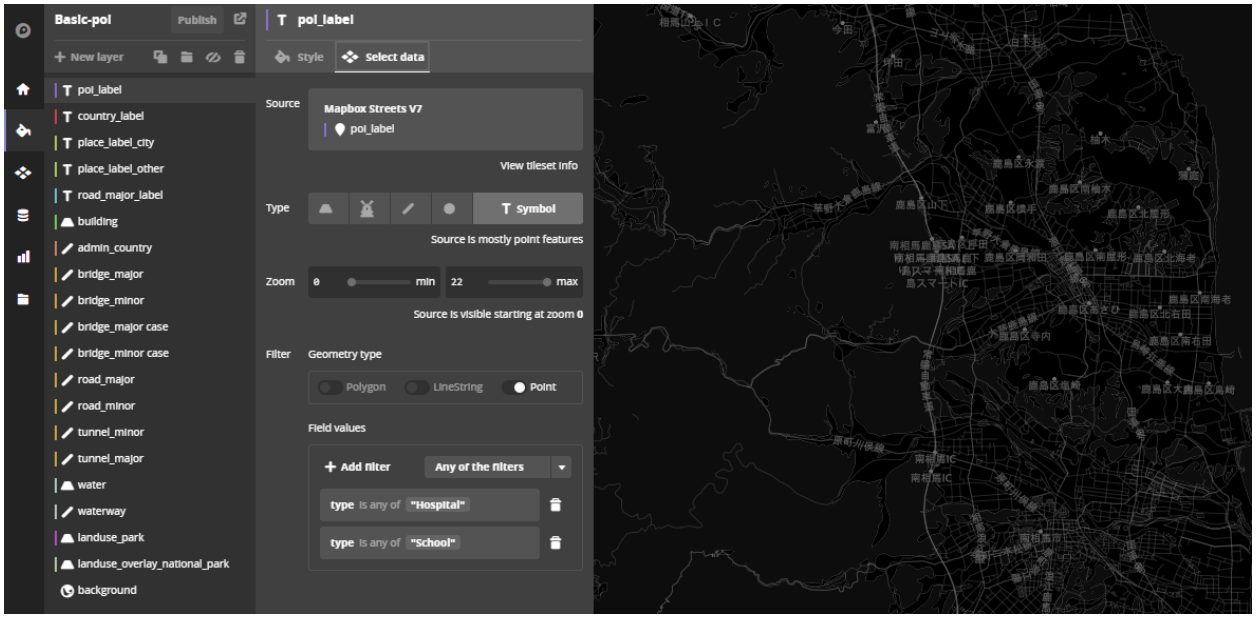
landuse z0 - z22
- landuse_park

road_label z0 - z22
- road_major_label

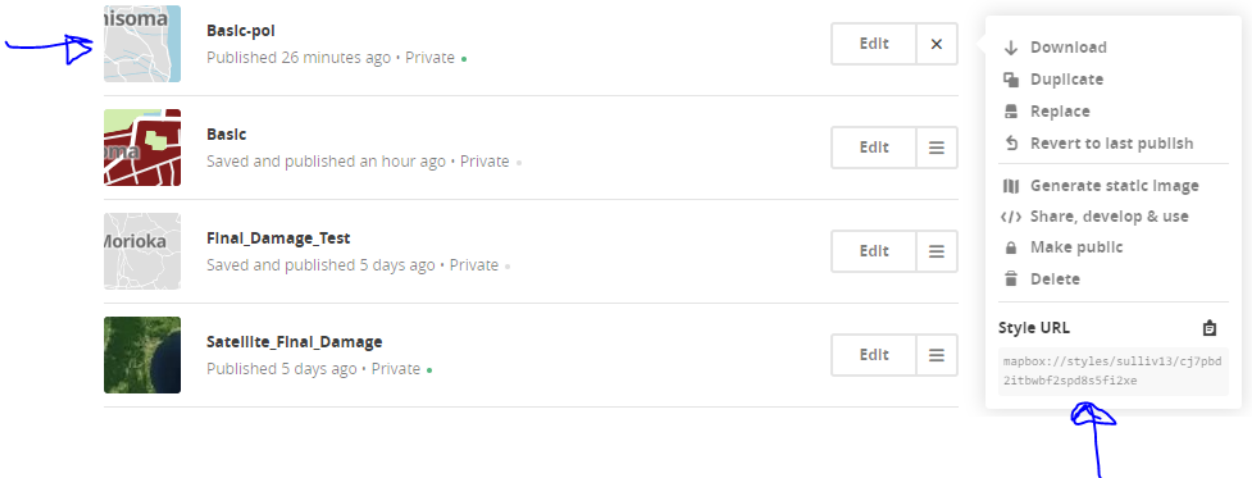
country_label z0 - z22
- country_label

place_label z0 - z22
- place_label_city
- place_label_other

water z0 - z22
- water

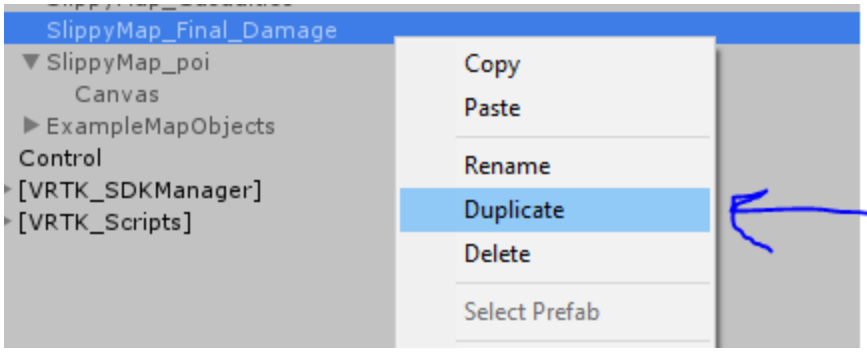


Save and publish the map through mapbox

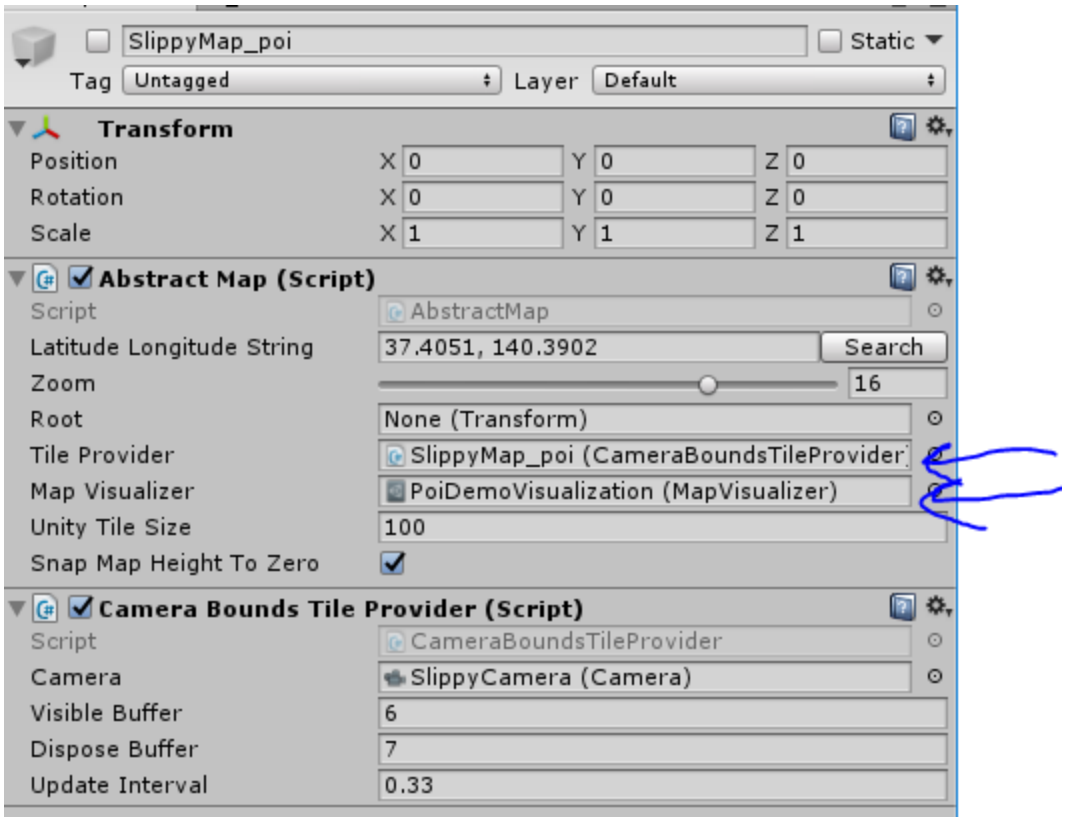


You'll need the style url later

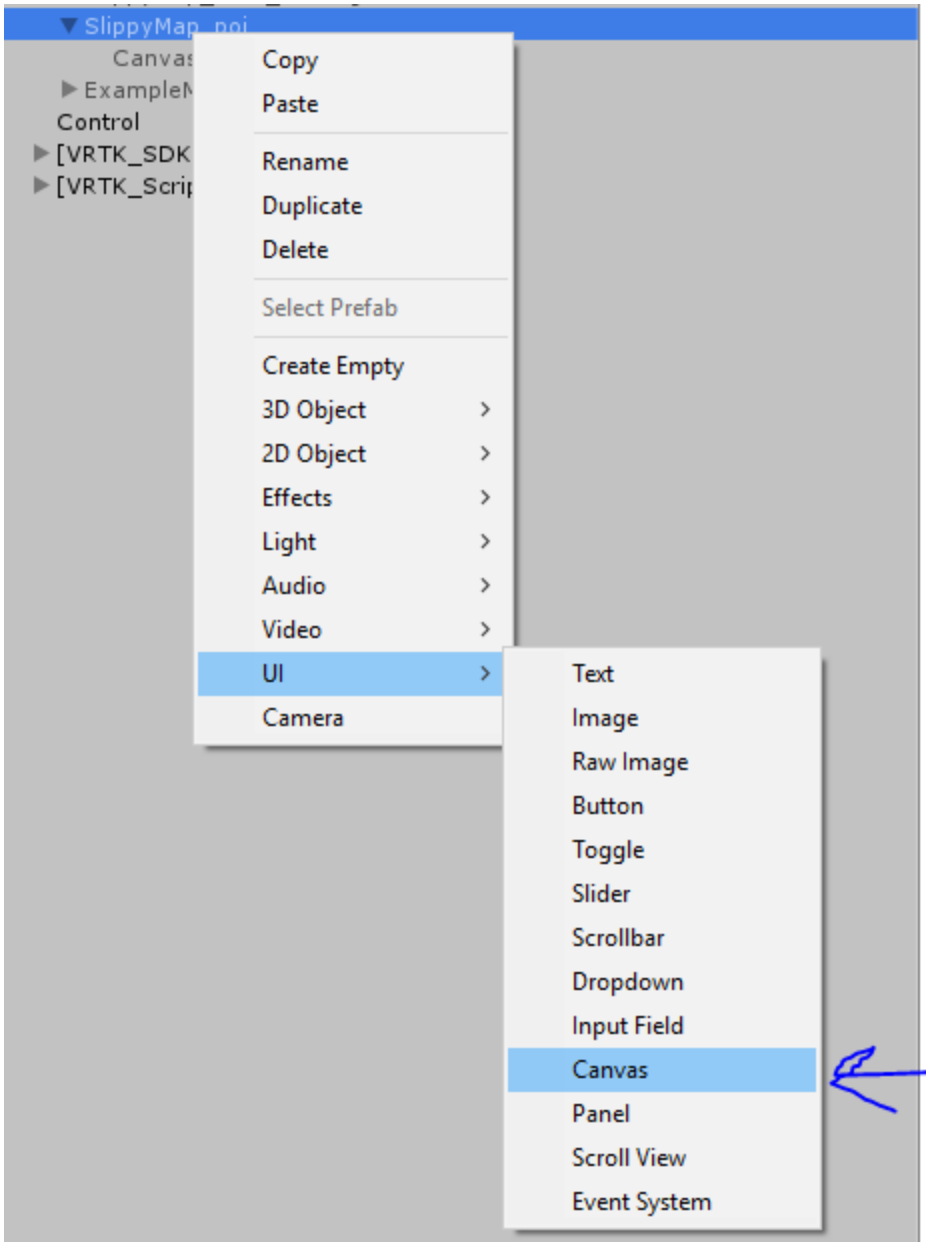
Next in Unity we are going to add a new slippy map so we can explore the hospitals and schools when zoomed in.
Lets copy an existing slippymap



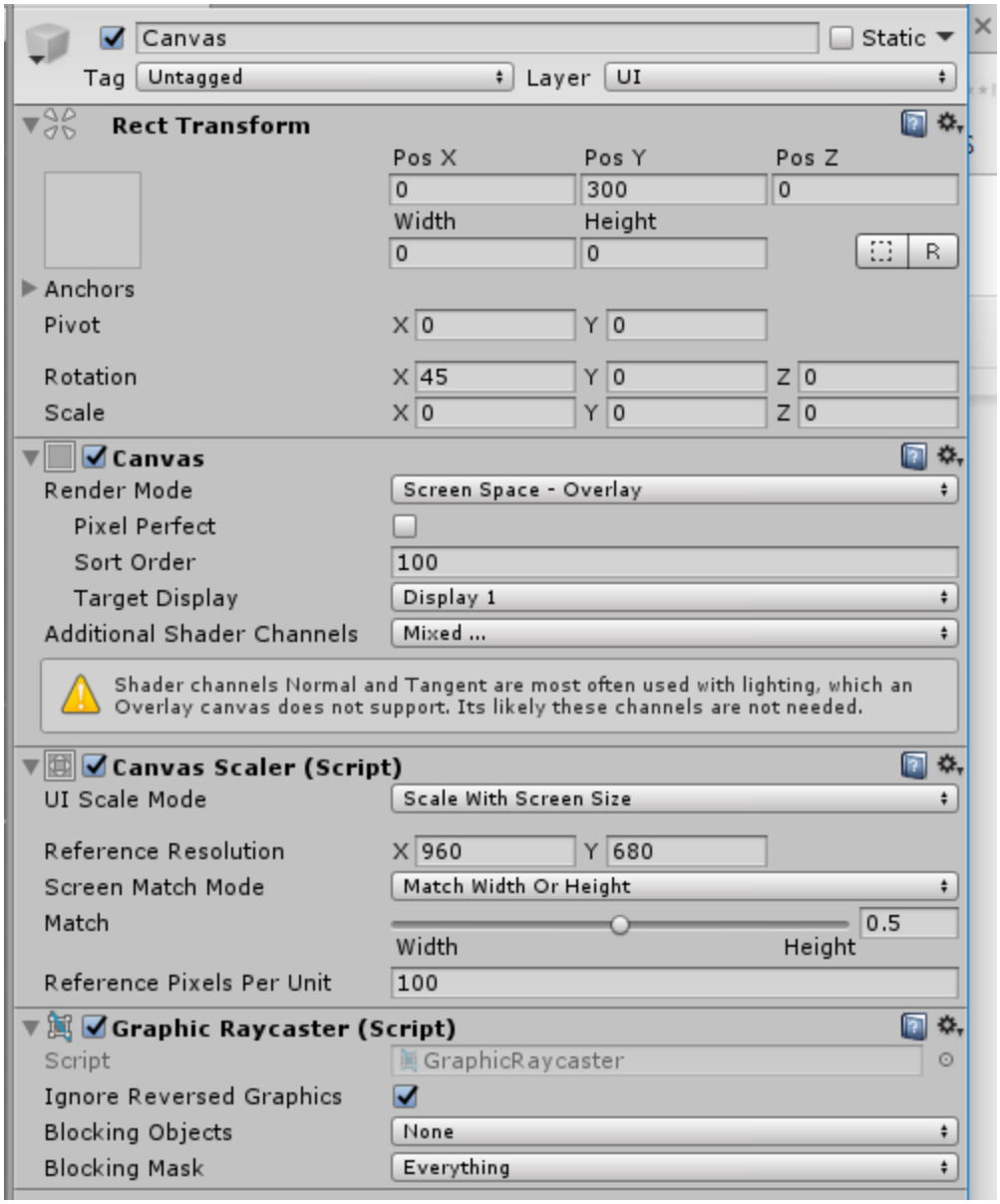
Lets call it SlippyMap_poi
Change the Tile Provider to the new slippyMap you just duplicated and change the visualizer to PoiDemoVisualization under mapbox/examples/MeshGenerationPois



Under SlippyMap_poi create a new Canvas Asset (this is for the POI icons (I think ?))

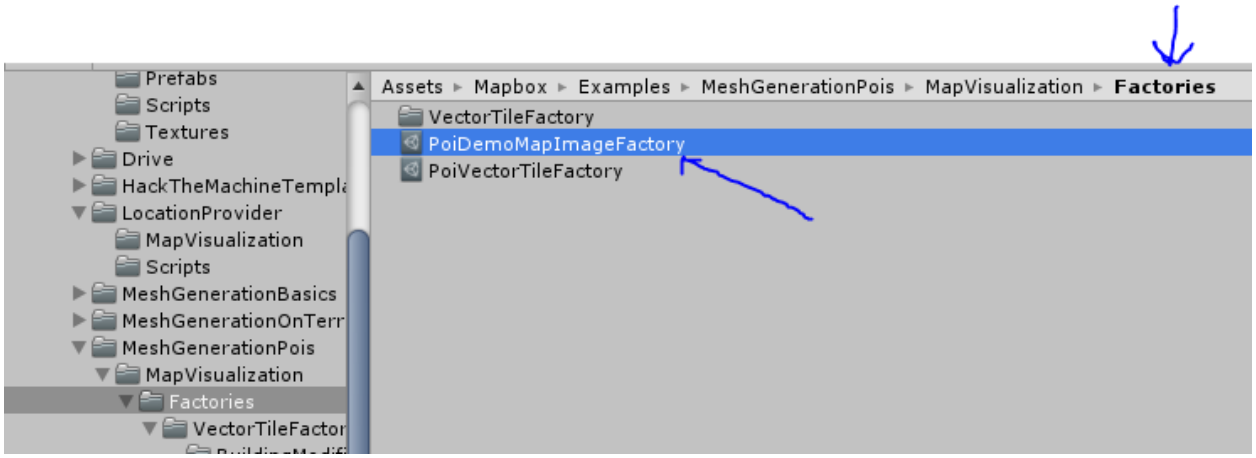


Make sure the canvas has these settings to make it work with VR

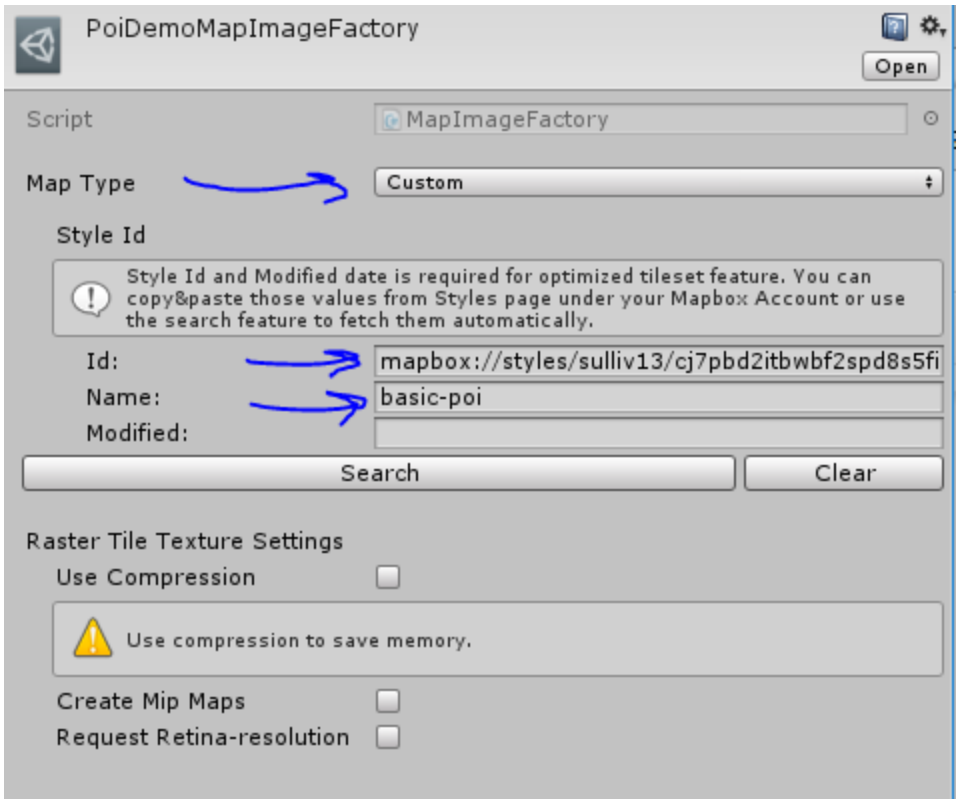


Alright now we will change the map style . Copy the map style url from earlier

We will modify the map image factory here:

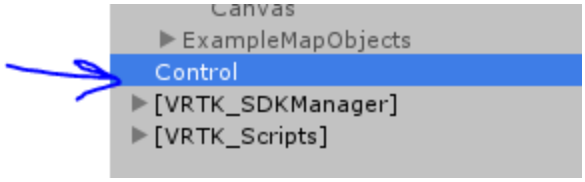


Paste in the url after changing the selection to custom

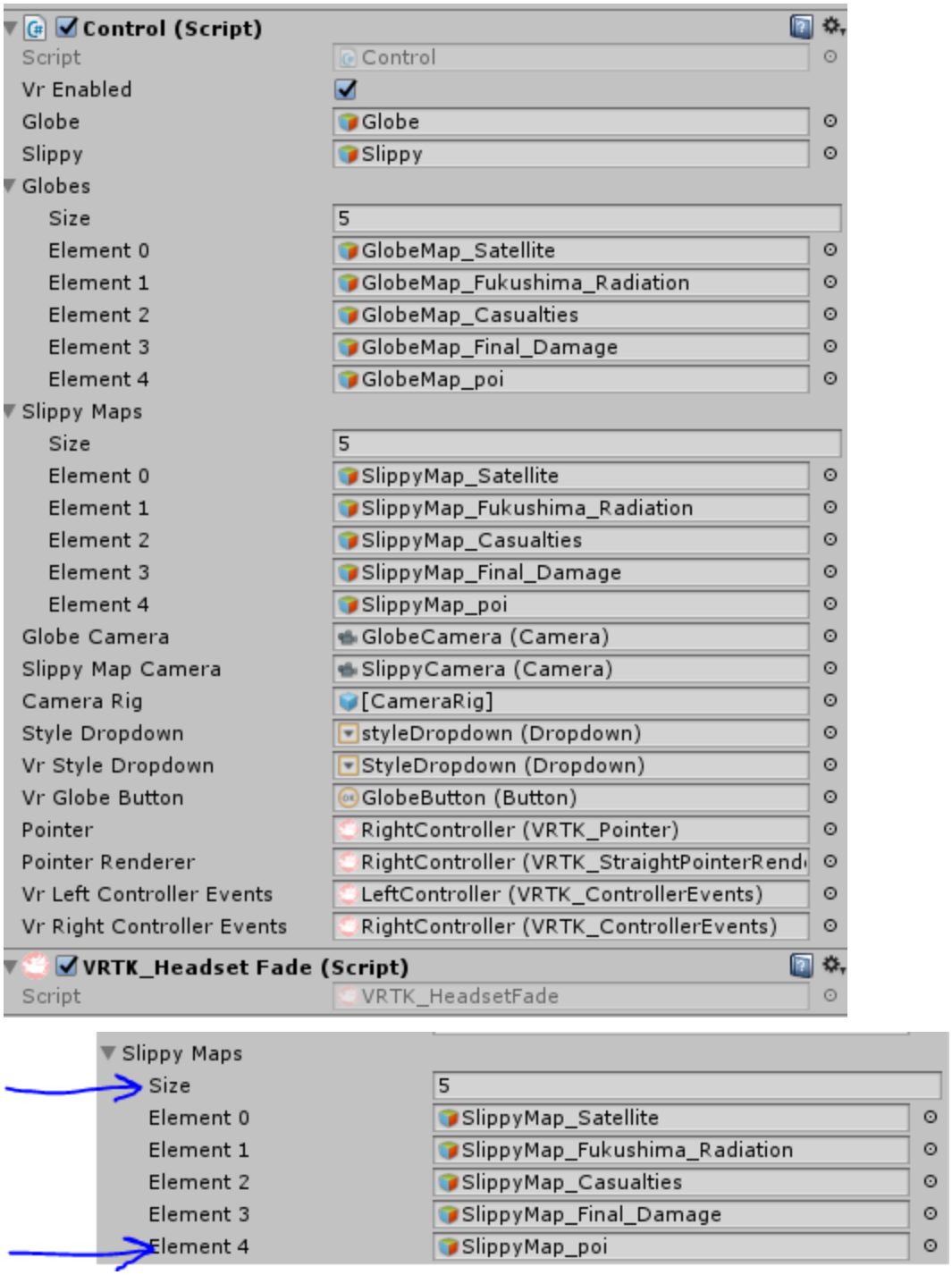


Next we will have to add poi as a layer in the drop down box so we can switch to that layer in vr

Select the Control asset

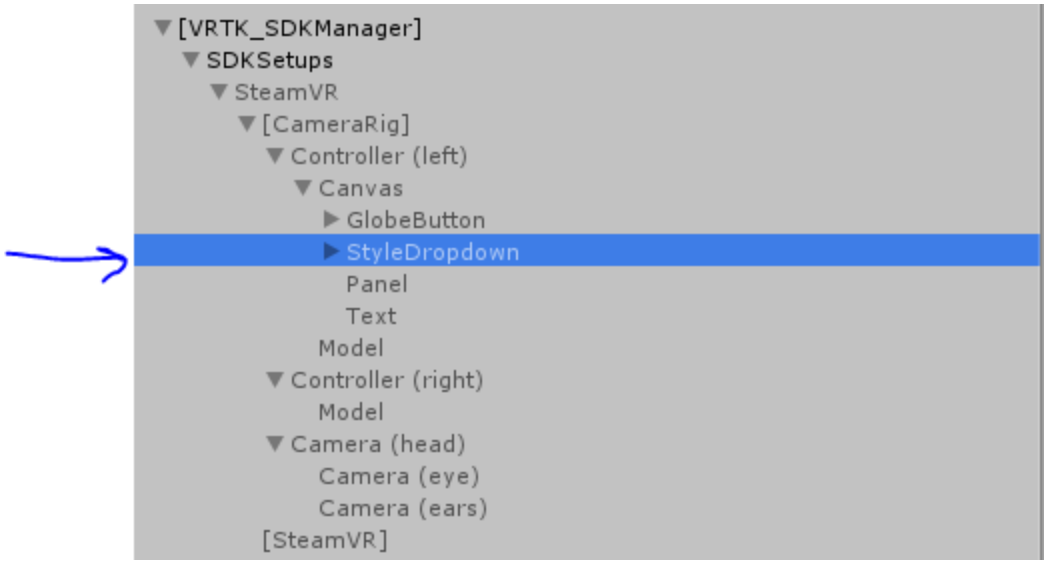


And add the SlippyMap poi as a new element in the drop down

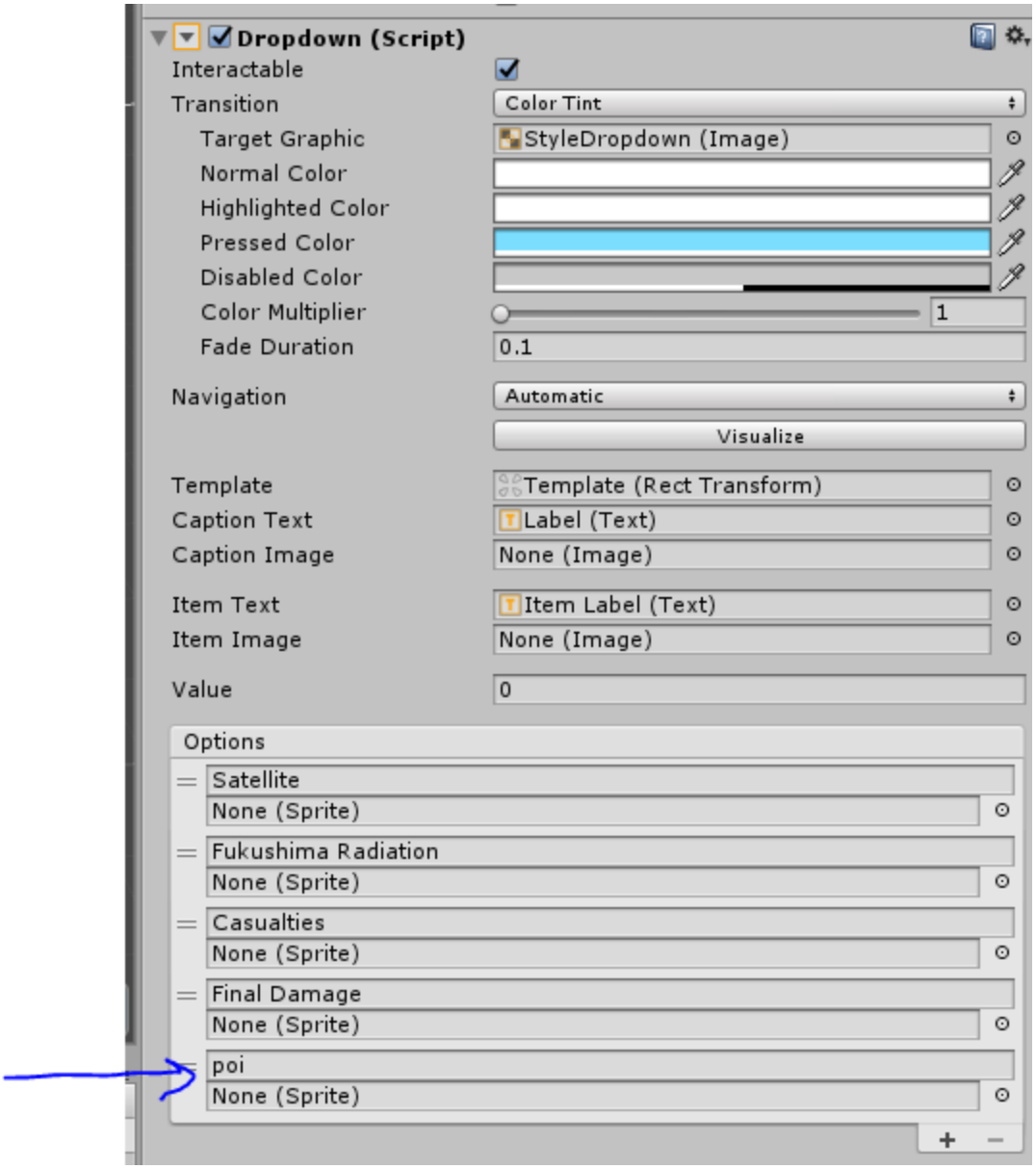


Now we have to add the text to the drop down menu in the ui

Select StyleDropdown under VRTK_SDKManager



Add the text to the dropdown



Test it out on the VR headset. You should see the drop down for POI and it will load a new map