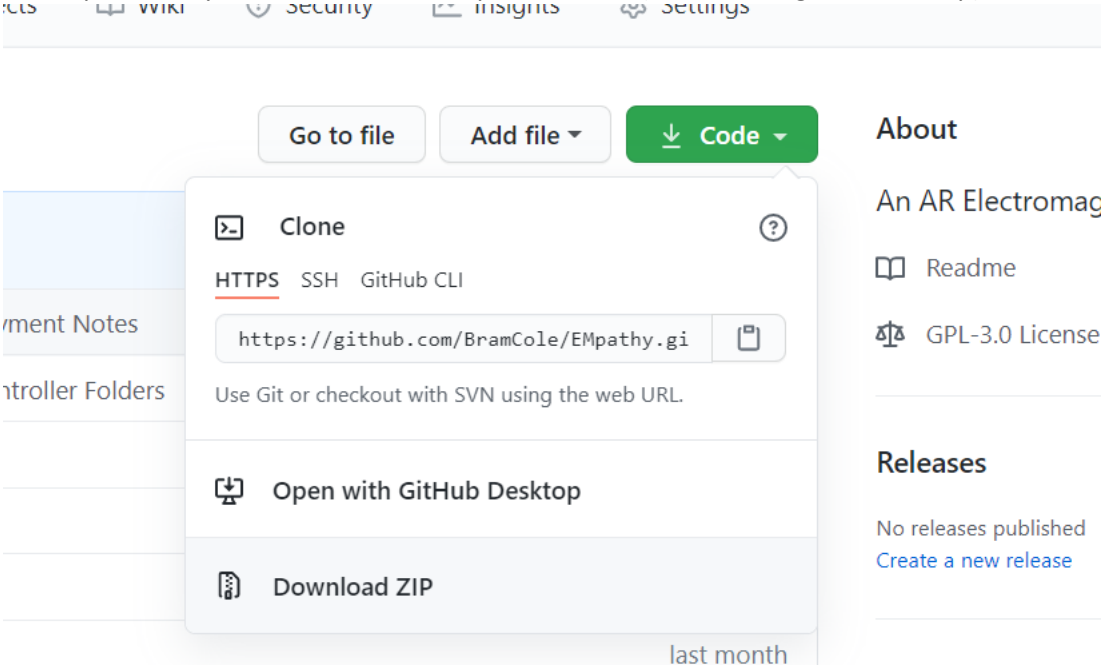
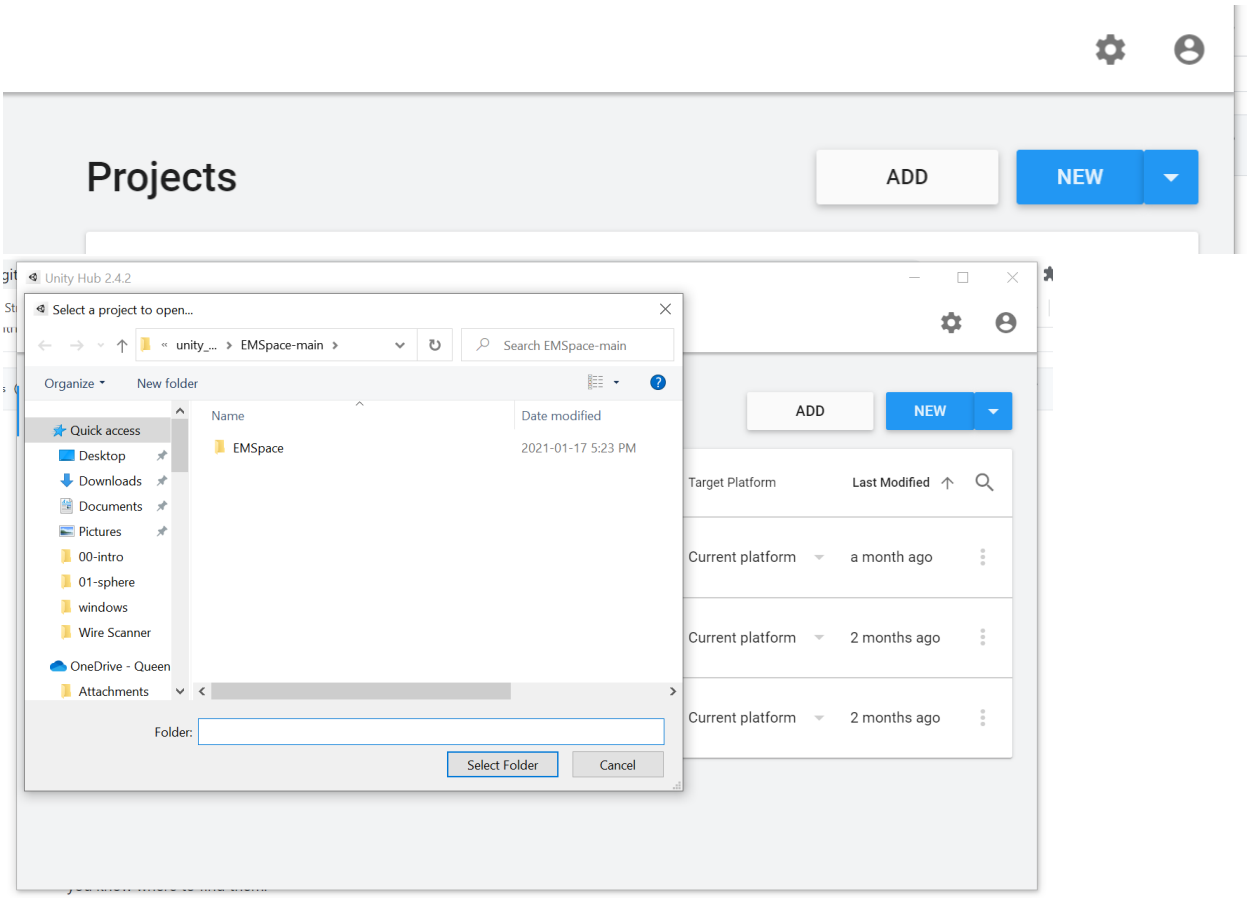
Engineering Tools Assignment Unity and Deployment

This is a guide to using Xcode and unity to physically deploy an application onto an Iphone. This allows us to test our app and it's functionalities on the same product our end user would be using. The procedure will follow chronologically and with pictures when necessary.

1. Register for a Vuforia account, Register for a unity account, Register for an apple developer account if you have a mac. Download Github Desktop to be able to push/pull
2. Download A version of unity later than 2019.4.16f1 LTS this can be done thought the unity hub. You must also download the ios package for unity if you want to deploy on an iphone.
3. Download the latest version of Xcode. You may also need visual studio for mac If you want to look at the code (unity will notify you if you need it)
4. Clone the EMSpace repository. There are many ways to do this. One way is to use github desktop. One way is to download as a zip and extract (then add to github desktop).

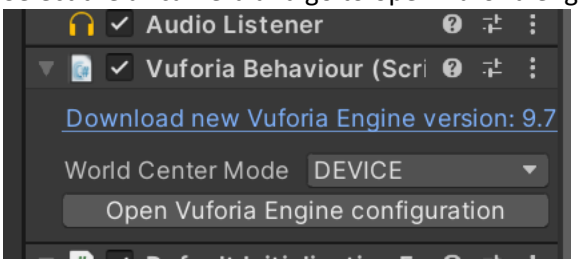


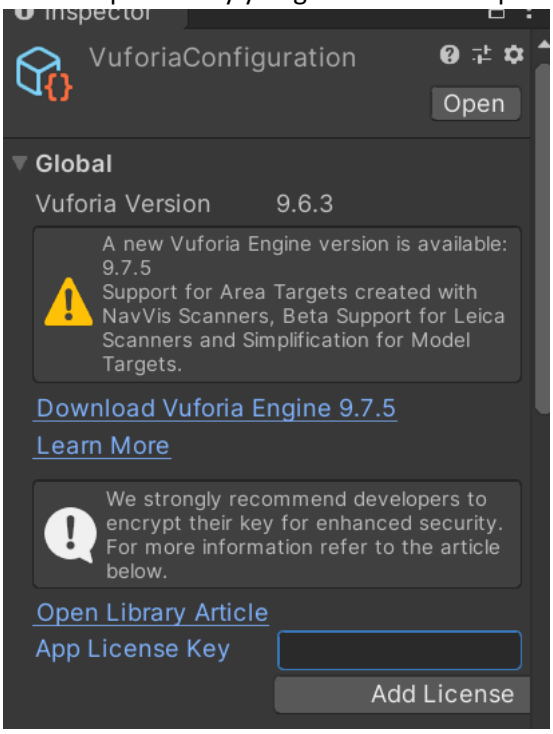
5. In unity Hub there will be an option to add an existing project. Do this and select the EMSpace folder inside the repository

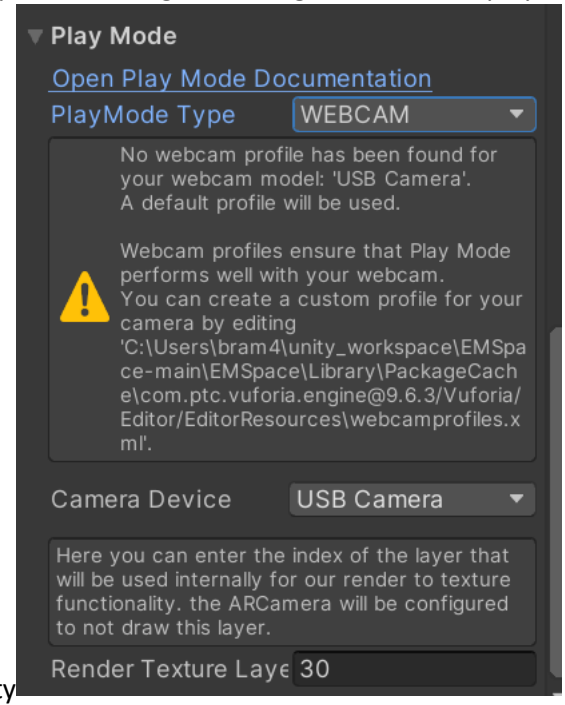


6. Once EMSpace is open go to Vuforia and click get development key this key can be named whatever you want.

7. Download Vuforia for unity. Once the download is complete you can physically drag the package into unity to install it into your project.
8. Select the ar camera and go to open Vuforia engine configuration in the inspector

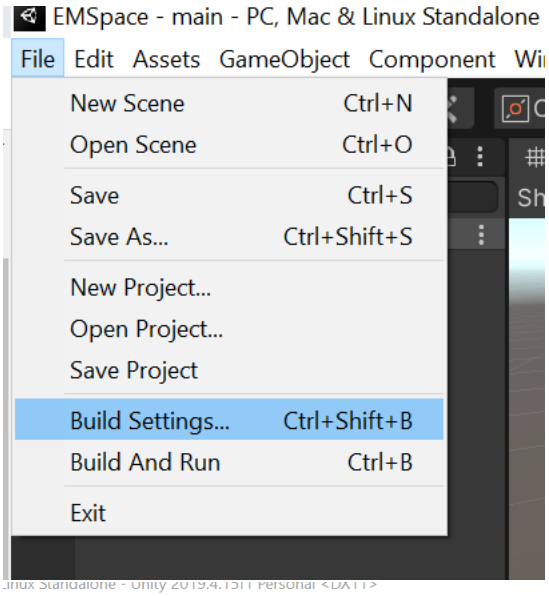


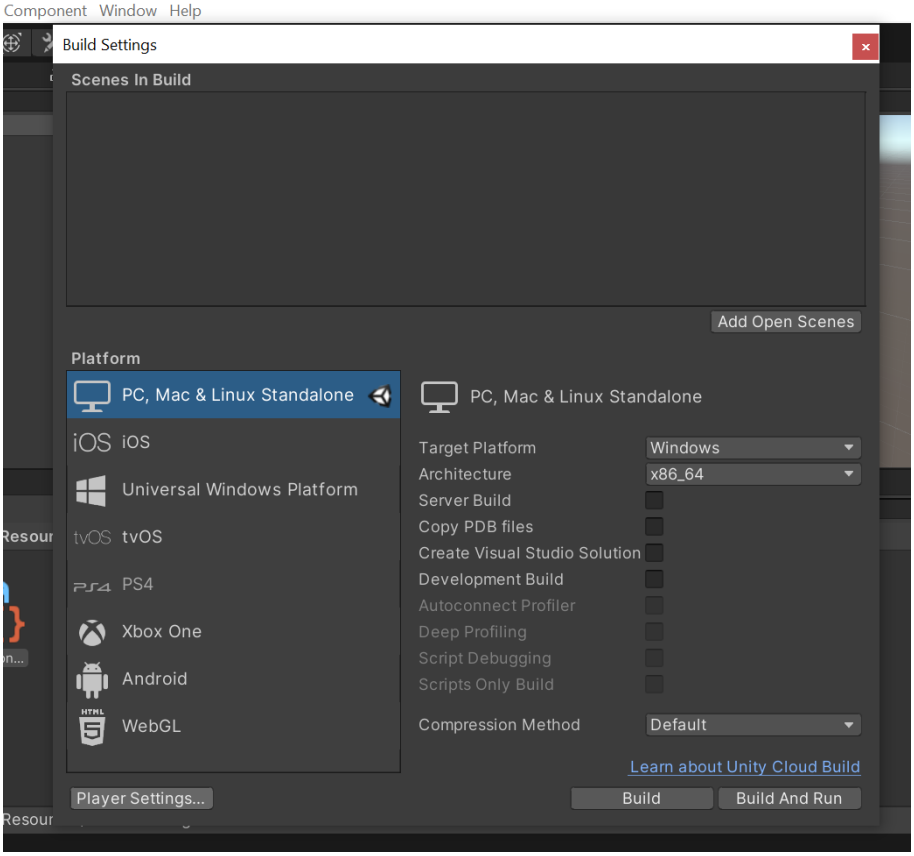
9. Using the development key you got from Vuforia paste the app licence key into the app licence key tab 

10. If you scroll down there are different play mode settings. Selecting simulator as a play mode type will allow you to play the sim in unity 

11. Go to file build settings. A bundle identifier should be set. If not set the company name to EMWare and the product name to EMPathy. The bundle identifier should change to com.EMWare.EMPathy if not change the identifier to this (since we are deploying on personal phones it doesn't actually matter what bundle identifier you use). A camera

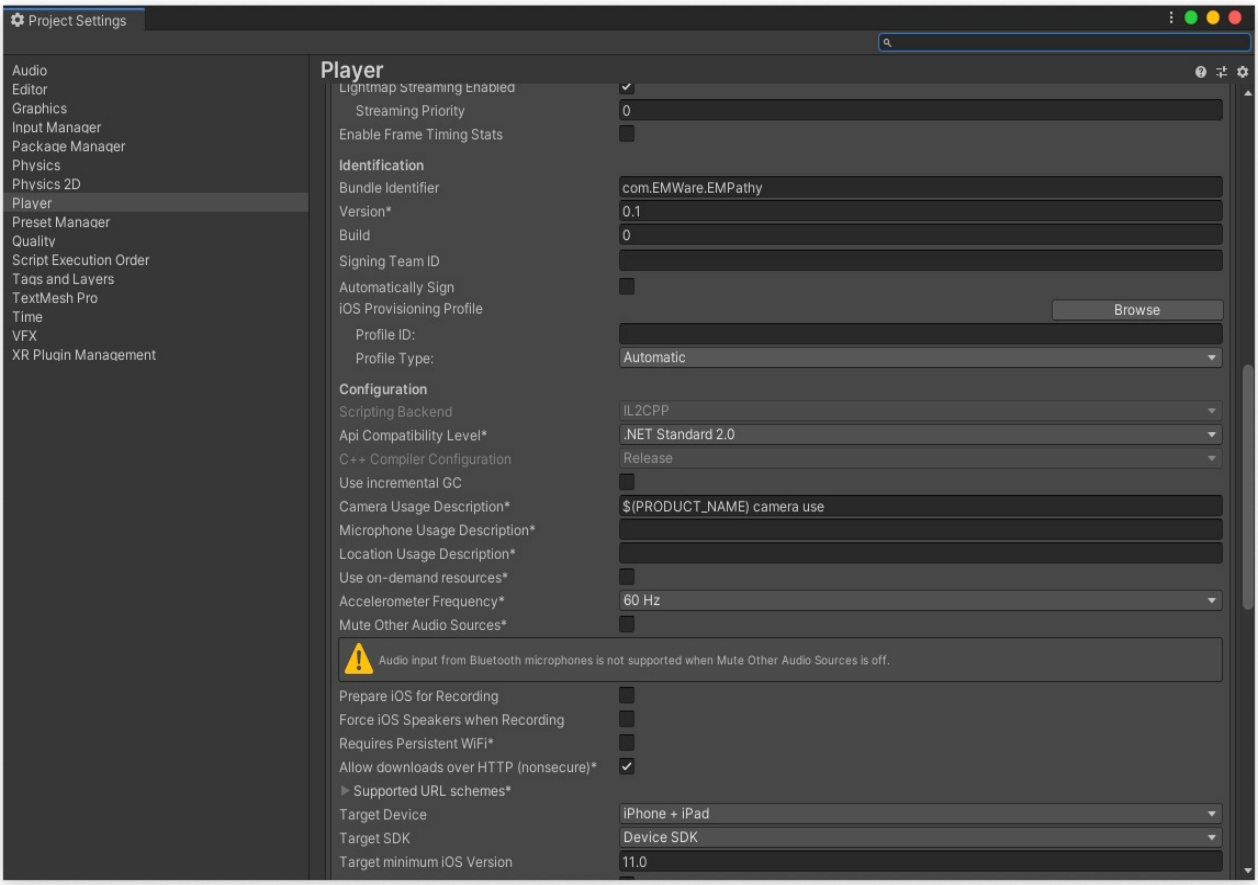
usage description must be set in the player settings under configurations change it to `$(PRODUCT_NAME) camera use`





Player settings in the bottom left corner of this window

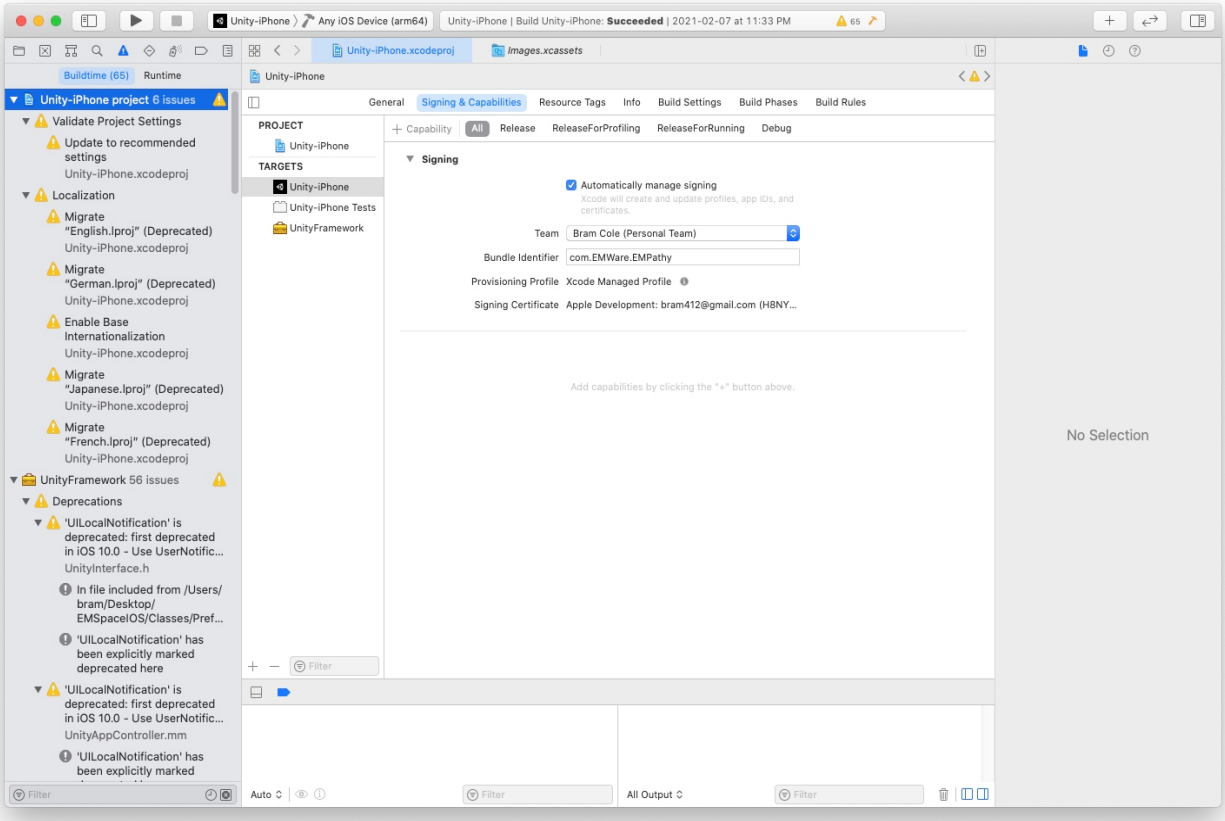
12. Vuforia also wants a minimum iOS version of 11.0 so change that in the player settings also.



Here we can see the bundle identifier being set. The camera usage description and the version at the bottom.

13. Once done hit build and run (make sure you build to ios) and it should take you to Xcode. Once in XCode there will be a little window in the left that shows files. Click the file that looks like a blueprint and that says the name of your project. This will bring you to XCode project settings. Go to signing and check automatically manage signing. After

choose a team.



14. To connect your phone, you must enable a developer mode both on Xcode and on IOS. **Xcode and your phone usually need to be updated to the latest version to run.** To run select your device to the right of the stop button and run.

This tutorial video is helpful if you are having problems exporting from unity to Xcode.

The unity documentation is a good place to start with unity. Learn how to place objects and how the scene builder works. Learn how cameras works and how to move around in play mode.

Here is the section for scene view navigation:
https://docs.unity3d.com/Manual/SceneViewNavigation.html

Aside from placing the AR camera Vuforia's features would be rarely used for this project so focus on unity.

Scripts are a way to extend Unity's functionalities and are written in c#. The way c# integrates with Unity makes it a little different from regular c#. The scripting API for unity can be found here https://docs.unity3d.com/ScriptReference/ . The docs for C# here: https://docs.microsoft.com/en-us/dotnet/csharp/ .