

How to Build and Update an HLS Streaming AR App in Unity



Unity 2020.1.11f1 (free)

Vuforia (personal: free / basic: \$502 annually)

AVProVideo (\$150 per platform)

DOTween Pro (\$15)

The goal: build an AR app that, when an image is triggered, plays a video that is hosted externally, and pauses the video when the image is no longer tracked.



TABLE OF CONTENTS



| | |
|--|----------|
| 1.0 Preparing your Media for Streaming | 3 |
| 1.1 Converting videos yourself: install homebrew and ffmpeg | 3 |
| 1.1.2 Have ffmpeg convert .mp4 file to packets and an .m3u8 playlist | 3 |
| 1.2 Do it quick and online for like... less than 50 cents | 4 |

| | |
|--|----|
| 1.3 Storing the files in the cloud for streaming | 4 |
| 1.4 Validating a playlist file to make sure it works | 5 |
| 2.0 Vuforia | 5 |
| 3.0 Unity | 6 |
| 3.1 Setting up the App for the first time (skip to 4.0 if app is already set up) | 6 |
| 3.1.1 Unity Build Settings (same settings if building to test, not release) | 6 |
| 3.1.2 Project Window + icons + Hierarchy | 7 |
| 3.1.3 Settings | 7 |
| 3.1.4 Lighting | 9 |
| 3.2 Assets to Install (from Asset Store or External) | 9 |
| 3.2.1 Vuforia | 9 |
| 3.2.2 AVPro | 10 |
| 3.2.3 DOTween Pro | 10 |
| 3.3 Add Materials | 11 |
| 3.3.1 Icons / Logos for game in app stores | 11 |
| 3.3.2 Transparent Material for video player | 11 |
| 3.4 Hierarchy: SETTING UP THE SCENE! (finally~~) | 11 |
| 3.4.1 AR Camera | 11 |
| 3.4.2 Image Target | 11 |
| 3.4.3 AVPro Media Player | 12 |
| 3.4.4 Streaming Video | 12 |
| 3.4.6 Add events to Trigger + Setting Objects Inactive | 13 |
| 4.0 If you are using the pre-built template from GitHub, start here! | 14 |
| 4.1 Logging in | 14 |
| 4.2 Streaming videos | 14 |
| 4.3 Adding new trigger to the hierarchy after everything is already set up | 15 |
| 5.0 Testing on a device simulator | 16 |
| 5.1 In Unity (recommended) | 16 |
| 5.2 Android Studio | 16 |
| 6.0 Building / exporting the app for test / runtime | 17 |
| 6.1 Buttoning up project / reducing file size | 17 |
| 6.2 Build and run / test | 17 |
| 6.2.1 iOS | 17 |
| 6.2.2 Android | 17 |
| 7.0 Preparing for Deployment / testing on a mobile device | 18 |
| 7.1 iOS | 18 |
| 7.1.1 Xcode - testing on a device | 18 |
| 7.1.2 Publishing to the app store | 19 |
| 7.2 Android | 19 |
| 7.2.1 Testing on a device | 19 |
| 7.2.2 Publishing to the google play store | 20 |
| 8.0 FAQs | 20 |
| 8.1 The animations aren't working! (fading in and out/scaling) | 20 |
| 8.2 The logo icons are compressing into weird sizes! | 20 |
| 8.3 Whenever I hit play in editor, everything is flipped horizontally! | 20 |
| 8.4 Why am I seeing an empty rectangle move around while the video is playing in editor? | 20 |
| 8.5 Xcode Errors | 20 |
| 8.5.1 Code signing is required for product type 'Application' in SDK 'iOS whatever' | 20 |
| 8.5.2 If you get a "permission denied" error | 21 |

1.0 Preparing your Media for Streaming

(Any videos that you want to play when a trigger is activated in AR)

Hello! Okay, we'll need to convert your .mv4 video file into something called an .m3u8 Master playlist

Basically what that is is a Playlist and subfolders with several chunks of video and audio inside each one that will load one at a time and streamed seamlessly. It needs to be broken down in chunks so it's not having to prep and load a massive video all at once, if it does it in smaller sizes it can stream more rapidly without a need to buffer for a long time.

Each folder is a different quality. The m3u8 file wont switch between qualities. It's just a playlist file, which contains a list of references to different videos. It's the responsibility of whatever is executing the code (in this case, the plugin you're using) to choose between videos.

There are two ways to convert videos into .m3u8 playlists: you can do it yourself, or pay like, 5 cents to have a website do it for you. I recommend not doing this yourself!!!!

1.1 Converting videos yourself: install homebrew and ffmpeg

I DON'T RECOMMEND THIS MY DUDES

This is a ...skill. It takes a LOT of time to come up with the right "recipe" to suit your needs, and to figure out the right bitrate and error margin, and what you know a clients network is set up for, but if you want to do it yourself here's what you can do (only know how to do this on mac:

on a mac, in terminal:

- Terminal, install homebrew - can copy paste this in:
 - `/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
 - `brew install ffmpeg`

1.1.2 Have ffmpeg convert .mp4 file to packets + .m3u8 playlist:

basic example (do this in terminal):

- `ffmpeg -i /Users/[yourusername]/Documents/01-FalconHeavy/FHvid.mp4 -profile:v baseline -level 3.0 -s 640x360 -start_number 0 -hls_time 10 -hls_list_size 0 -f hls FHvid.m3u8`

Heres an example that I eventually came up with:

- `.0390625ffmpeg -i /Users/[yourusername]/Documents/01-FalconHeavy/FHvid.mp4 \`
`-b:v:0 5000k -maxrate 5350k -bufsize 7500k \`
`-b:v:1 2800k -maxrate 2996k -bufsize 4200k \`
`-b:v:2 1400k -maxrate 1498k -bufsize 2100k \`
`-b:v:3 800k -maxrate 856k -bufsize 1200k \`
`-b:a:0 192k \`
`-b:a:1 128k \`
`-b:a:2 128k \`
`-b:a:3 96k \`
`-c:a aac -ar 48000 -c:v h264 -profile:v main -crf 20 -sc_threshold 0 -g 48 -keyint_min 48 -map 0:v -map 0:a -map 0:v -map 0:a -map 0:v -map 0:a -map 0:v -map 0:a \`
`-f hls -var_stream_map "v:0,a:0 v:1,a:1 v:2,a:2 v:3,a:3" \`
`-master_pl_name FHvidmaster.m3u8 -hls_time 4 -hls_playlist_type vod \`
`-hls_segment_filename '/Users/[yourusername]/Documents/01-FalconHeavy/01-FalconHeavy/HLS/file_%v_%03d.ts' /Users/[yourusername]/Documents/01-FalconHeavy/HLS/out_%v.m3u8 \`

1.2 Do it quick and online for like... less than 50 cents

<https://cloud.gencode.com>

The support team there are nice too, lol. They've been helpful to me a lot!

- make an account
- you can select what types of video qualities you want too.
- way quicker and faster and more efficient than doing it yourself.

1.3 Storing the files in the cloud for streaming

Once you've created a .m3u8 file/master playlist either yourself or through [gencode.com](https://cloud.gencode.com) now you need to store it somewhere online where your app can stream it from.

I used cloud.google.com (prefer them over aws. Its free the first year and pretty darn cheap afterwards

- I use their "storage" option
- Create a project/bucket
- inside that bucket, make a folder for your newest video. title it something short
- inside that folder upload the new files that have been prepared for streaming

- set your permissions - I just set everything at public but only I could edit.
- give permission for: Storage Object Viewer and to: allUsers
- (<https://cloud.google.com/storage/docs/access-control/making-data-public>)

Now, once that's all set up, the path to the .m3u8 master file is the link you will enter back in Unity under the Source Path (See section: Setting up the Scene on page 8)

1.4 Validating a playlist file to make sure it works

<https://bitmovin.com/docs/player/faqs/how-can-i-validate-my-mpeg-dash-or-hls-stream>

- Get HTTP Live Streaming Tools from apple developer here:
 - <https://developer.apple.com/download/more/?=http%20live%20streaming%20tools>
- in terminal:
 - mediaStreamValidator [path to where your files are stored]
 - ex:
 - mediaStreamValidator http://storage.googleapis.com/whatever/FHVid.m3u8

2.0 Vuforia



developer.vuforia.com

Version 9.6.3

once logged in, in the green bar at the top it will say License Manager or Target Manager

- License Manager: where you can copy the license key into your app (will use this later in SECTION 3.2.1)
- Target Manager > Add Database
 - type: device
 - name: whatever
- once database is created, click on it under Target Manager
 - add new target, click add target:
 - single image, give it a consistent name
- after you add your targets, at the right side of the target manager you'll see "Download Database (all)" and click on it. you'll drag this into your project in section SECTION 3.2.1)

3.0 Unity

Login to Unity:

Unity 2020.11f1

- 3D (not universal render pipeline)

Download

<https://unity3d.com/get-unity/download>

Download Unity Hub

- Installs tab: Add
 - Unity 2020.1.11f1
- once finished, back in installs tab, click on the 3 vertical dots next to 2020.11f1
 - add modules
 - click Android or iOS build support, whatever platform you want
 - make sure everything under the expand arrow is selected

3.1 Setting up the App for the first time

(skip to 4.0 if app is already set up)

3.1.1 Unity Build Settings (same settings if building to test, not release)

iOS:

file > build settings

- left side: iOS
- Scenes in Build: make sure the current scene is checked, otherwise click add open scenes)
- platform: iOS
- run in xcode: Latest version
- run in Xcode as: Debug
- Symlink Unity libraries: checked (this will not be an option if building on pc)
- Development build: check
- Compression Method: Lz4HC

Android:

file > build settings

- left side: Android
- texture compression: don't override

- Build app bundle (google play): check
- compression method: LZ4HC

3.1.2 Project Window + icons + Hierarchy

Deleting unnecessary assets

- example assets > delete folder
- Materials > delete Skybox
- tutorial info > delete folder

Adding assets

- scenes: rename SampleScene to "Main"
 - Scenes > SampleSceneLightingSettings (Rename to MainSceneLightingSettings)
- make folder named icons > add all icons of logo
 - select all icons you just dragged in
 - inspector window (on right) > advanced > non-power of 2: none (this keeps the specific resolution of the icons, otherwise it would resize them to standard sizes)
 - any transparent icons:
 - inspector > texture type: Sprite
 - inspector > advanced : alpha is transparent (check)

Hierarchy

- delete all example prefabs

3.1.3 Settings

edit > project settings > player settings > icon

- Add company name and project name
- under "cursor hotspot" there is a computer icon and an iOS icon. click iOS
 - icon > add icons if they are ready, if not see section 3.3.1
 - Splash Image > logos > + > select your transparent logo

edit > project settings > player settings > resolution and Presentation

- render over native UI: uncheck

edit > project settings > player settings > Other Settings (iOS)

- Rendering
 - Dynamic batching: check
- Identification:

- set signing name/ bundle identifier: com.[your apple developer name].CosmicPerspective, ex: com.BobbyBobberson.ACoolApp
 - the name here has to match the name you are using to sign the app in Xcode or whatever you're using
 - automatically sign: check
- Minimum API level: iOS: 11.0
- Configuration
 - Scripting Backend (IL2CPP)
 - Api compatibility level: .Net Standard 2.0
 - camera use description: "This app requires use of the camera. Please allow in phone settings"
 - mute other audio sources: check
 - unclick optimize mesh data for faster compile times
 - requires persistent WiFi: check
- Optimization
 - Managed Stripping Level: High (can adjust to lower if you don't need to strip for size of the app)
 - Script Call Optimization: Fast but no exceptions (can adjust to slower if wanted)
 - texture mipmap stripping: check

edit > project settings > player settings > Other Settings (Android)

- Rendering
 - Auto Graphics API: check
 - static batching: uncheck
 - dynamic batching: check
- Identification
 - Minimum API level: Kitkat (19)
- Configuration
 - mute other audio sources: check
 - Internet access; require
- Optimization
 - Optimize mesh data: check
 - texture mipmap stripping: check

Project settings > quality > rendering

- Realtime reflection Probes: uncheck
- soft particles: uncheck
- shadow: disable shadows
- shadow resolution: low

Project settings > Graphics > shader stripping

- lightmap modes: custom
 - uncheck all
- fog modes: custom
 - uncheck all

3.1.4 Lighting

window > rendering > lighting > scene

- Realtime lighting: all unchecked
- baked global illumination: unchecked
- lightmapping settings: bounces: none
 - russian roulette start bounce: never
 - filtering: auto
- shadow: disable shadows
- shadow resolution: low

window > rendering > lighting > environment (next to scene at top)

- environment lighting: source: color (select black)
- environment reflections
 - source: custom
 - intensity multiplier: 0
- other settings:
 - fog: uncheck
 - halo strength: 0
 - flare speed: 0
 - flare strength: 0

3.2 Assets to Install (from Asset Store or External)

3.2.1 Vuforia

Download SDK package from <https://www.developer.vuforia.com/downloads/sdk>

- drag and drop that download into the unity asset folder/project section
- when prompted, make sure everything is checked and click import
- you'll get a popup that says "Add Vuforia Engine Package" - click update

Wanna check for updates?

- Window > Package Manager
- At the top, next to the + sign, click the drop down menu: My Registries
- Vuforia Engine AR should now be listed

Walkaround of quarantine issue on macOS Catalina

- If you downloaded the package directly instead of adding the scoped registry through the package manager, you might get a "cannot be opened because the developer cannot be verified" message.
 - to to terminal and add this line:
 - `xattr -r -d com.apple.quarantine <package directory>`

License Key

Project Window

- search "vuforia" > click on vuforia configuration
 - copy license from developer.vuforia.com > login > click on name of project
 - copy license key
- paste licensee into the field on vuforia configuration in unity
- Camera Device Mode: optimize speed or quality (I do speed)
- Max simultaneous devices/objects: as many as you'll have.
- Load target objects: check
- Device Tracker > Track device pose: uncheck
- Android settings : if iOS, disable.

Importing Database

- developer.vuforia.com
 - login > develop > target manager > click on database > download database (all)
 - drag and drop into project window to import

3.2.2 AVPro

window > asset store

- login > download / import (from purchases)
- don't need to import demo folder
- in plugin folder, only need either iOS if mac or Android if android

3.2.3 DOTween Pro

window > asset store

- login > download (from purchases)
- Open DOTween Utility Panel
- Setup
- Update (apply default settings)
- Create ASMDEF > ok

3.3 Add Materials

Make a new folder under Assets named Materials. Put all these materials there.

3.3.1 Icons/Logos for game in app stores

Logos for icons (make a separate folder in the materials section if you want)

- Import/drag all sizes of logos at different resolutions
- make sure they're named correctly
- **this next part is important because unity compresses images weird!!! ******
 - advanced: Non-power of 2: None
 - scroll down to bottom: Default: Compression: None
- **edit > project settings > player settings**
 - add icon to "default icon" box (can drag and drop)
 - under "cursor hotspot" there is a section for iOS and Android. Set the icons to whichever app you are building.
 - override for PC, Mac & Linux Standalone: check
 - icon > add icons

3.3.2 Transparent Material for video player

right click in project folder > create > material

- name: TransparentMat
- shader: AVProVideo > Unlit > Transparent (texture+color+fog+)

3.4 Hierarchy: SETTING UP THE SCENE! (finally~~)

3.4.1 AR Camera

delete any camera already in the scene

- right click in hierarchy > Vuforia Engine > AR Camera
 - make sure it is tagged at top as "MainCamera" (it should be by default)

3.4.2 Image Target

right click in hierarchy > Vuforia Engine > Image Target

- in inspector: Type: change "from image" to "from database"

- database: select
- Image target: select
- (if nothing comes up, its because you didn't do step 3.2.1)

3.4.3 AVPro Media Player

We will need a MediaPlayer for each specific trigger. We wouldn't have one media player streaming all of the videos, and so because we will need many, I name mine specifically so I know which media player belongs to which trigger. EX: if FHTrigger is the trigger, than FHPlayer is the Media player

- **Add: AVPro > MediaPlayer** (can right click in hierarchy)
 - settings
 - source path: Absolute path or URL
 - add link to wherever you have your .m3u8 files ready for streaming in the cloud (see section 1.3)
 - auto open : yes
 - auto play : no
 - loop : yes
 - **add component: apply to mesh**
 - media: current media player (this)
 - default texture (transparent png) (this is the one we made in section 3.3.2)
 - Mesh: [name]Vid (mesh) (will add this in the section: 3.4.4)
 - texture property: _MainTex

3.4.4 Streaming Video

Add as child of Image Target (ex: a child of FHTrigger).

- Right click on Image Target/Trigger > create > 3D > Quad
 - name: rename [name]QuadVid, ex: FHQuadVid or FHVid
 - **Layer: 1: TransparentFX ***** Whatever this is, make sure it is NOT set to 0:Default but to something below it. This is because we are putting a sprite (the patch) in front of it, and the sprite will automatically go behind it otherwise. This changes the sorting order to have the patch visible on top of it. Even if it doesn't look like it in the editor, in runtime the patch will come to the front
 - transform: Roation X:90 (the trigger is
 - transform: whatever size and space you want your video to be! I would make it match the size of the trigger if you want it to cover the trigger completely.
 - Materials > element 0 = transparentMat (from section 3.3.2)
 - lighting
 - Cast Shadows : off
 - receive shadows : uncheck
 - probes

- light probes: off
- reflection probes: off
- **add component: apply to material (should have AVPro logo on it)**
 - drag media player ([name]Player) from section 3.4.3) into media field
- **add component: DOTween Animation**
 - press "preview controls" so it is NOT selected/highlighted green
 - where it says none: click and select: fade
 - duration : 4
 - Delay : 0
 - ease : InSine
 - loops : 1
 - from : 0 (MAKE SURE YOU CLICK THE "TO" Button so it changes to: "FROM")
- **Scroll to bottom of inspector in Material section******
 - shader: AVProVideo / Unlit /Transparent (texture+color+fog+packed....)
 - the two texture boxes should be empty / set to none
- **After this is set up, make sure to drag this game object (this 3D quad in the hierarchy) into these two places:**
 - the [name]Player (from section 3.4.3)
 - under apply to mesh component > Render Target > Mesh: drag into this field

3.4.5 Add events to Trigger + Setting Objects Inactive

After this is set up, go back to the Image Target game object
the [name]Trigger > default trackable event handler:

On Target Found ()

- click the + button
 - drag the [name]Player into this field
 - no function: click > MediaPlayer > Play ()
- click the + button
 - drag the [name]Vid into this field
 - no function: click > GameObject > SetActive(bool)
 - check the little box that appears

On Target Lost ()

- click the + button
 - drag the [name]Player into this field
 - no function: click > MediaPlayer > Pause ()
- click the + button
 - drag the [name]Vid into this field
 - no function: click > GameObject > SetActive(bool)
 - uncheck the little box that appears

When finished, select [name]Vid

At the top of the inspector, there will be a little checked box next to the name.

- UNCHECK THIS.
- It will set the game object inactive. MAKE SURE THE [NAME]TRIGGER and [NAME]PLAYER ARE STILL ACTIVE.
- What we just did above will set the objects active again when tracking is found. if tracking is not found, they will be inactive.

4.0 If you are using the pre-built github template, start here!

Open the app in unity. Project window > assets > scenes > Main

4.1 Logging in

Make sure you have an account with Unity and Vuforia

4.2 Streaming videos

Storage

I currently have all of this set up and it is set to stream from my own cloud storage bucket. I should have included a zip file of all of those video playlists, so now you'll need to set up your own cloud storage account to host them if you don't have one already.

- See section 1.3 and 1.4 on uploading to cloud storage and validating the links to make sure they work!

Updating the video path urls to their new storage urls

- Open up Unity and in the hierarchy window, you'll see lots of game objects called [name]Player
 - ([name] is just the catchall for different names))
 - click on each of the different players, and under Media Player > Source Path
 - (Right under it, it should say Absolute Path or URL). In the text field below that, please update each streaming video playlist url to the new one.

4.3 Adding up new triggers to the hierarchy after everything is already set up

Update the number of tracked targets

- in the Project window search bar, type in vuforia. click on Vuforia Configuration
 - Max Simultaneous Targets / Objects : update as you see fit.

Preparing and storing the new video to be streamed with the new trigger

- see section 1.0

Uploading new trigger:

- developer.vuforia.com login
 - click on project name
 - in the green bar, select Target Manger (license manager is default selected)
 - select database
 - add target
 - click whatever type of target you want
 - name it [name]Trigger for ease
 - click add
 - select target manager again > the database > download database (all)
 - this is on the right side
 - download and drag into project window of unity

Copying the [name]Trigger and [name]Player from the hierarchy to use as a template for the new trigger

- select, copy and paste two [name]Trigger and [name]Player games objects that have the same [name] and paste it back into the hierarchy
- rename both parents and children: change the names to [new name]Trigger and [new name]Player
 - in children: change [name]Vid to [new name]Vid

[new name]Trigger

- Image Target: Select new target name
- Name: change the name to the new name
- See section 3.4.2 if you need more details about the [name]Trigger game object

[new name]Vid

- Transform: adjust the height/width/scale as needed to fit the size of the new trigger image. This vid game object will play the streamed video, and it needs to be slightly bigger than the trigger image. Once you've got it, hit the play button and adjust as needed. You might have to adjust again once you test out the app on a mobile device
- see section 3.4.4 if you need more details about the [name]Vid game object
- if you see an empty rectangle that jumps around, thats okay! it shouldn't show up in deployment. I bought the iOS and Android license for AVProVideo and the sdk is

recognizing that it isn't being run on iOS or Android but instead on a PC and thus is trying to throw up a watermark. Once it's deployed it'll be gone.

[new name]Player

- Source Path: please update to the exact URL of where your streaming video is stored externally
- Auto Open: checked
- Auto Play: unchecked

Test it out in the editor!

5.0 Testing on a device simulator

5.1 In Unity (Recommended)

Device Simulator

- Window > Package Manager
 - click gear icon > advanced project settings
 - Advanced settings: enable preview packages: check
- Window > package manager
 - Scroll until you see Device Simulator (you should see a yellow box next to it that says "Preview")
 - click install
- Window > general > device simulator
 - now you can click between the simulator and the game window when you press "play".
 - after hitting play, just click on the simulator
 - top left corner of simulator: select the type of device you want to simulate

5.2 Android Studio

I used to use this to test/simulate on different types of virtual android devices before the unity simulator - BUT!!!! I don't think there is any camera in the emulator. some android images contain default images for when you use the camera. Some android images contain default images for when you use the camera. Check the extended controls > camera section. I'm just leaving this here if there are ever scenes that are not AR based and for future reference.

- Drag and drop your .apk file (you will have to build it first, see section 6.0) from unity into android studio
 - once it loads in and you see the green hammer icon up top right, click on it to build
- tools > ADV Manager

- +create Virtual Device (add whatever devices you want to test on. I usually do the earliest (in this case, Kitkat API 19), the latest, and one in the middle)
- press the green triangle icon next to the device you want to test on
 - when it loads that emulator (give it a minute) flip it to the position you want it to be on using the white and grey buttons to the side)

6.0 Building / exporting the app for test/runtime —

6.1 Buttoning up project/reducing file size:

Doing this reduces bloat and the end file size

- edit > project settings > graphics > Shader Stripping
 - light map modes > custom
 - untick all boxes.
 - fog modes > custom
 - untick all boxes; instancing variants > strip unused
- if you are building on a mac select symlink unity libraries. lol including this so i can remember in the future.
- can delete any "demo" folders in AVPro / Demigant / StreamingAssets Vuforia folders

6.2 Build and Run/Test

6.2.1 iOS

file > build settings > save it as whatever you want

- Scenes in Build: make sure the current scene is checked, otherwise click add open scenes)
- platform: iOS
- run in xcode: Latest version
- run in Xcode as: Debug (if debug), Release (if release)
- Symlink Unity libraries: check (only if you're building on a mac)
- Development build: check (if debug), unchecked (if release)
- Compression Method: Lz4HC

6.2.2 Android

file > build settings > save it as whatever you want

- Scenes in Build: make sure the current scene is checked, otherwise click add open scenes)
- platform: Android
- texture compression: don't override
- ETC2 fallback: 32-bit
- Development build: checked if debug, unchecked if release
- Compression Method: Lz4HC

7.0 Preparing for Deployment / testing on a mobile device

You'll need to change the bundle identifier to your developer name in editor > project settings > player > other > identification: com.[your developer name].NameOfYourApp

7.1 iOS

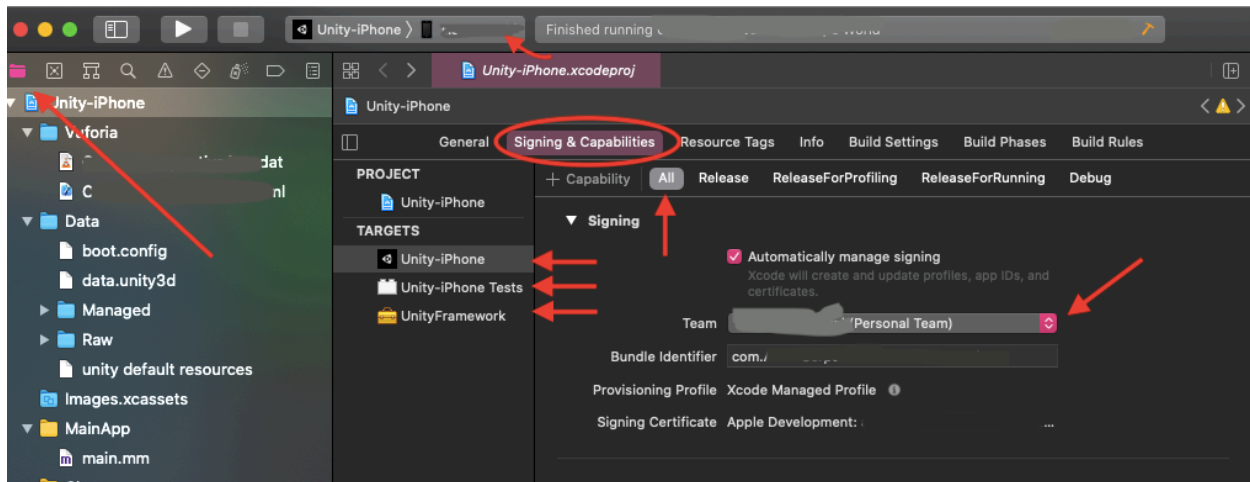
7.1.1 Xcode - testing on a device

(see image on next page)

Copy that folder that you just built and send it however you want to a mac computer to use Xcode. If you don't have a mac, there are a few solutions for you to get Xcode on a pc. see codewithchris.com/xcode-for-windows/

open it and click on Unity-iPhone-Xcodeproj

- it should open in Xcode
- Click on folder icon in top left corner
 - in the second window pane, you'll see PROJECT with Unity-iPhone under it, and TARGETS, with a few targets under it.
- click each of the targets and select "signing & capabilities" to make sure you are signed in on all three.
 - select your "team" + check Automatically manage signing
 - if you don't have a team set up yet:
 - Xcode > preferences > accounts > + button on bottom left to add your apple id > select Apple ID > follow instructions
 - if testing: click on "debug" (note: there are some weird bugs with the latest few Xcode's and using it to run an app in debug mode unless you have a developer account, its REAL frustrating).
 - connect your phone via usb to laptop



- make sure your phone is selected in the place to test the app (red arrow in pic, my phone is called Hope World)
- if releasing, use your pro developer account (can get at developer.apple.com)
- Press the "play" icon (sideways right triangle at top left)
- usually on your device you'll have to enable permissions to use the app:
 - on iPhone: settings> general > scroll to the bottom, Device Management > "Trust this developer" or whatever it says

IF YOU GET ERRORS IN XCODE, SEE SECTION 8.5

7.1.2 Submitting your app to the app store

This describes the process much better than I could

<https://codewithchris.com/submit-your-app-to-the-app-store/>

another tutorial that can explain it better than me

<https://learn.unity.com/tutorial/publishing-for-ios#>

7.2 Android

7.2.1 Testing app on a device

This tutorial explains it better than I could:

<https://learn.unity.com/tutorial/building-for-mobile#5c7f8528edbc2a002053b4a2>

7.2.2 Deploying to the google play store

<https://play.google.com/console>

- Make an account
- pay the \$25 dollar fee to publish an unlimited number of android games to Google Play
- Upload your game to the console and fill out the details, then hit publish!

8.0 FAQs

8.1 The animations aren't working! (fading in and out/scaling)

If animations aren't working properly its because the material settings and shaders are improperly configured. check out section 3.4.4

8.2 The logo icons are compressing into weird sizes!

If logo icons are coming out weird sizes its because ...
check out section 3.3.1

8.3 Whenever I hit play in editor, The target won't leave the screen even when I drag it off screen, and the camera moves weirdly!

I think the AR camera is set up to track faces by default! So go into the vuforia configuration (assets > resources > VuforiaConfiguration) and under Device tracker uncheck Track Device pose or anything you think might be making it track your face.

8.4 Why am I seeing an empty rectangle move around where the video is playing?

If you see an empty rectangle that jumps around, thats okay! it shouldn't show up in deployment. I bought the iOS and Android license for AVProVideo and the SDK is recognizing that it isn't being run on iOS or Android but instead on a PC and thus is trying to throw up a watermark. Once it's deployed it'll be gone.

8.5 XCODE ERRORS

8.5.1 Code signing is required for product type 'Application' in SDK 'iOS whatever'

<https://stackoverflow.com/questions/37806538/code-signing-is-required-for-product-type-application-in-sdk-ios-10-0-stic>

general: each of the targets: signing & capabilities : "all" NOT release, or ReleaseFor Profiling, etc, but ALL**** its important that that one is selected. go through each of the targets and make sure they are all signed. if there is an error, it will be next to the one that isn't signed.

8.5.2 If you get a "permission denied" error

there are 3 different ways you can try and solve.

look at the error and copy the file name or path that leads to the file folder

open up terminal terminal

((**DONT DO ANY OF THESE WITHOUT THE FILE NAME AFTER IT!!!! you are changing permissions so DONT just be like "chmod 777" and hit enter - baaaaad! you don't wanna give permission to everything lol)))**

try one of these three things:

- `chmod +x` (paste file name here, without parentheses)
- `chmod -R 755` (paste path up to problem folder name here, without parentheses))
- `chmod 777` (paste file name here without parentheses)