

Lab 5: Vuforia + Unity

CS410/510: Virtual Reality

Ehsan Aryafar earyafar@pdx.edu

Sam Shippey sshippey@pdx.edu

Review from last week

- Vuforia: What/why
- Go to <https://developer.vuforia.com/>
- Sign up for an account
- Upload an image under “target manager”
- When choosing an image, select for:
 - Strong lines
 - Good lighting
 - <2Mb size

Review from last week

- Marker-based vs markerless AR applications
 - Marker-based AR: Use image or pattern to orient scene
 - Markerless AR: Map environment to orient scene
- Choosing marker-based vs markerless AR determines many things about your app

Addendum to last week

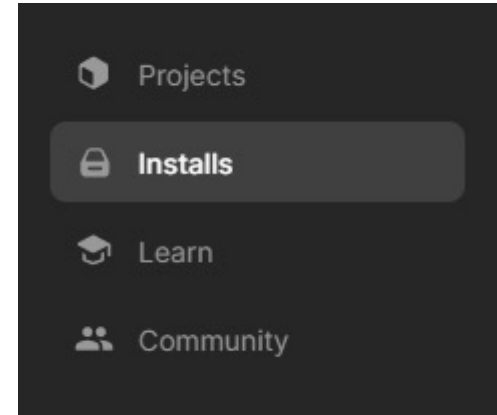
- Limits to environmental understanding are largely performance-based
 - Object recognition/SLAM are hard
 - Phones have limited hardware and poor thermals
 - Poor performance = low framerate = poor user experience

Unity

- Open Unityhub
- We'll set up a very basic marker-based AR app during class
- This is the homework

Editor settings

- If you plan to deploy this to a mobile phone, make sure your platform's build support is checked in the editor you installed.
- To check, go to Installs in the UnityHub.
 - Check for Android or iOS as necessary

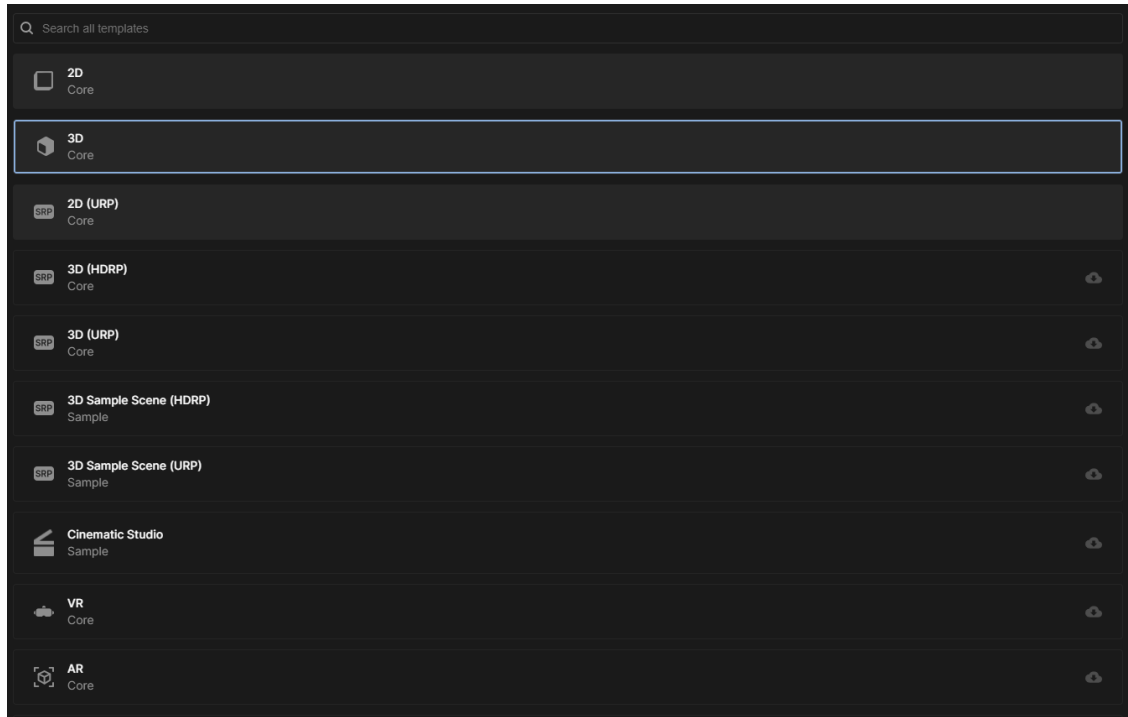


Deployment and testing

- Desktop: Unity play mode will work fine
- Android: Building for Android has worked fine for most people
- iOS: You may experience difficulties building for iPhones without a mac and xcode correctly configured.

Getting started

- Create project

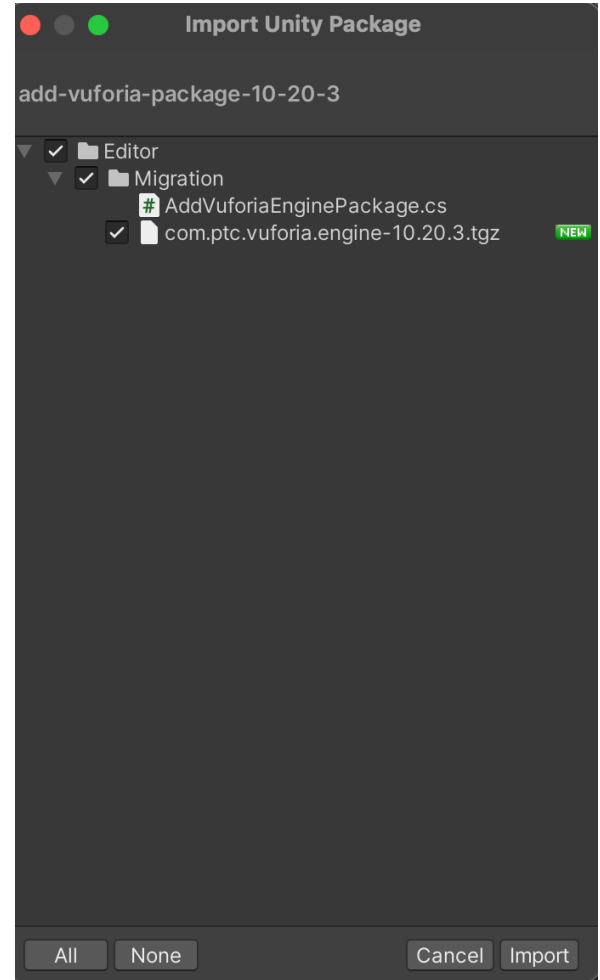


Adding Vuforia support

- Unity packages are distributed as .unitypackage files, usually from developer's websites
- <https://developer.vuforia.com/downloads/sdk>
- Download the .unitypackage file

Adding Vuforia support

- Assets
 - Import package -> custom package
- Find wherever you put the Vuforia package
- Import
- Confirm

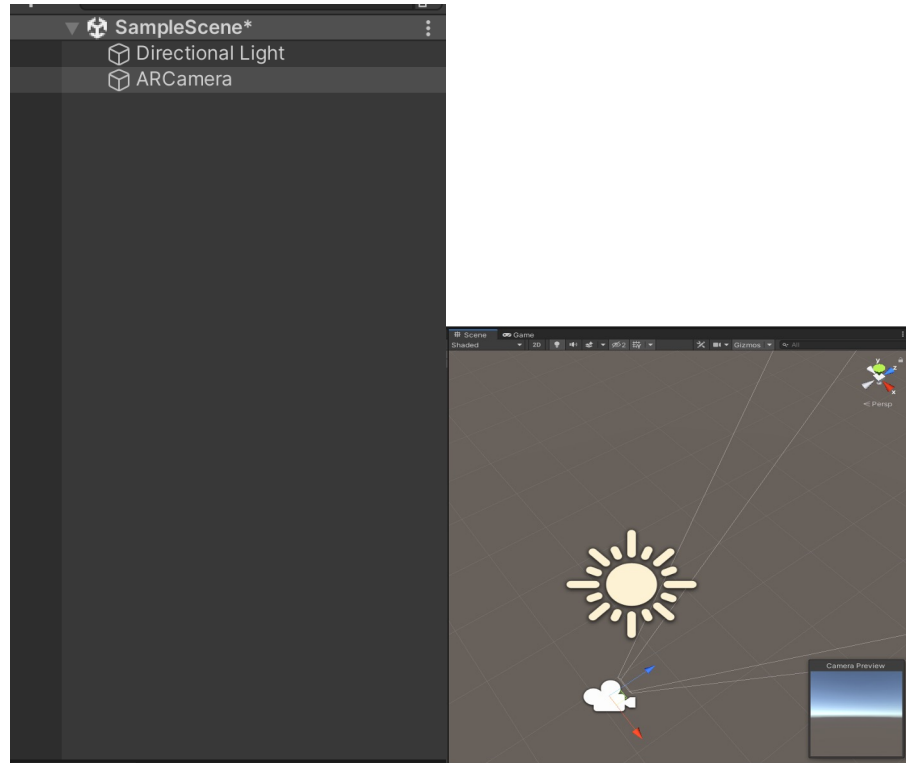


Vuforia Camera

- AR Cameras
 - Vuforia implements its own camera which you have to add in lieu of the normal Unity camera
 - Think of this as a wrapper around a normal camera that:
 - Performs marker recognition
 - Moves the camera to the correct position relative to the marker
- It doesn't matter where you put this
 - All transforms are computed relative to the marker now

Vuforia Camera

- Add the Vuforia camera
- GameObjects
 - Vuforia Engine
 - AR Camera
- **Delete the main camera**
 - **The scene won't work if you leave it there**
- Should look basically the same as the default scene



Adding the Vuforia marker

- Time to add the markers we made last week through Vuforia's developer portal.

Adding the Vuforia marker

- Time to add the markers we made last week through Vuforia's developer portal.
- Go to the Vuforia developer portal
- Make sure your license is still good

The screenshot shows the Vuforia License Manager interface. At the top, there is a green navigation bar with "License Manager" and "Target Manager" tabs. Below this, the "License Manager" section is active, featuring two buttons: "Get Development Key" and "Buy Deployment Key". A sub-header reads "Create a license key for your application." followed by a search input field. Below the search field is a table with the following columns: Name, Primary UUID (with a help icon), Type, Status (with a dropdown arrow), and Date Modified. The table contains one row: "Test license", "N/A", "Develop", "Active", and "Nov 01, 2020". At the bottom of the page, there is a footer with "25 per page" (with a dropdown arrow), "Showing 1-1 of 1", "Last updated: Today 9:42 AM", and a "Refresh" link.

Name	Primary UUID ⓘ	Type	Status ▾	Date Modified
Test license	N/A	Develop	Active	Nov 01, 2020

Adding the Vuforia marker

- Go to the target manager after that
- Select **Download Database (All)**
- Then make sure to select **Unity Editor**


Lab4 [Edit Name](#)

Type: Device

Targets (1)

Add Target

Download Database (All)

<input type="checkbox"/> Target Name	Type	Rating ^①	Status [▼]	Date Modified
<input type="checkbox"/>  20201101_212850_vuforia	Image	★★★★☆	Active	Nov 01, 2020 21:35

Download Database

1 of 1 active targets will be downloaded

Name:
Lab4

Select a development platform:

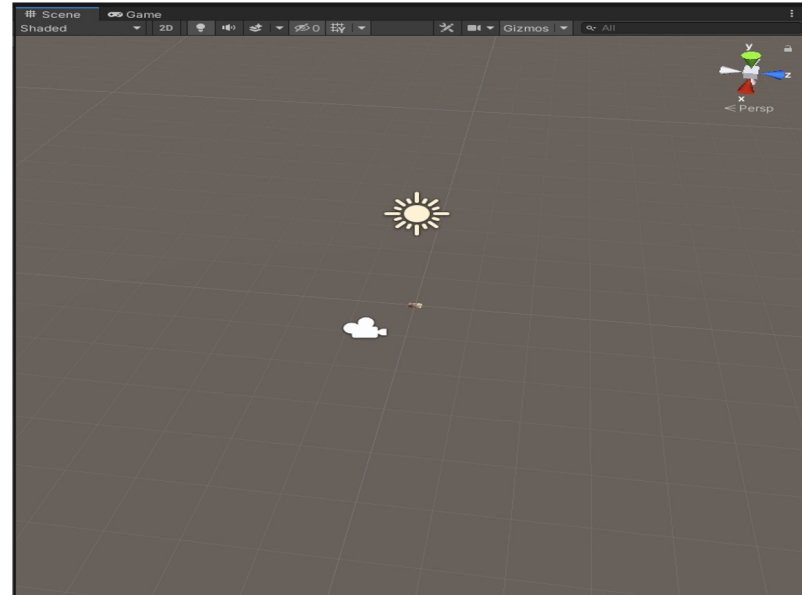
- Android Studio, Xcode or Visual Studio
- Unity Editor

Cancel

Download

Adding the Vuforia Marker

- Back in the Unity editor, go to
 - GameObject
 - Vuforia Engine
 - Image target
- Should be really really small by default

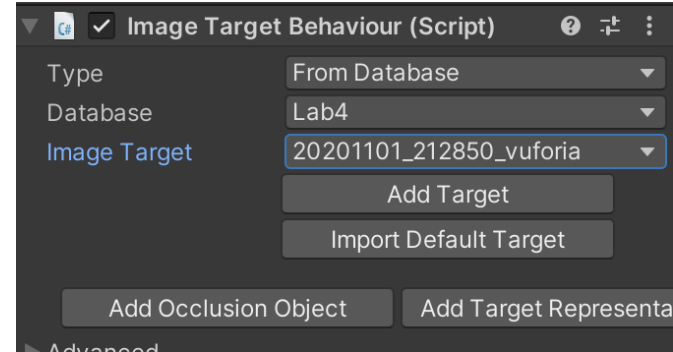


Adding the Vuforia Marker

- Change to our database
- Assets
 - Import package
 - Custom package
 - Find the .unitypackage (the database) you just downloaded

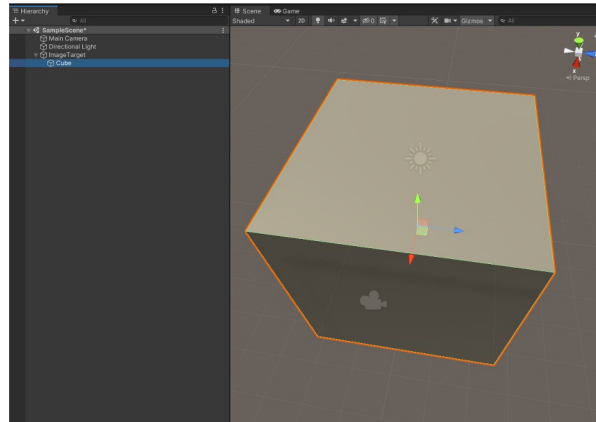
Adding the Vuforia Marker

- Configure the image target
- Select the image target in the hierarchy
- Look at the inspector, find Image Target Behaviour
- Change the “database” and “image target” attributes to point to what you want them to.



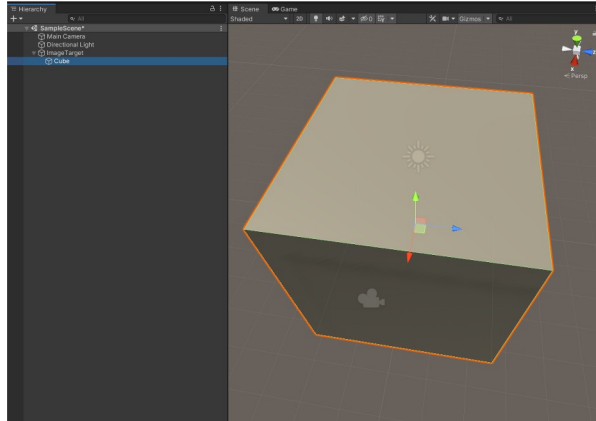
The actual app part

- You now have a kind of working app!
- Add some assets, **as children of the image target**.
- I'm uncreative, so I've put a cube here.
- You can hit the play button to test it



The actual app part

- You now have a kind of working app!
- Add some assets, **as children of the image target**.
- I'm uncreative, so I've put a cube here.
- You can hit the play button to test it



Further reading

- Vuforia is easy to use, but certainly not the only AR platform!
- If you'd like to experiment more with Marker based AR, consider using Three.JS + AR.JS, A-Frame, AWS Sumerian, or any one of around a dozen constantly shifting proprietary solutions.
- If you'd like to experiment with markerless AR, try the same set of things, plus Unity's AR Foundation.