Augmented reality for Unity: the Vuforia package

J. Vezien Nov. 2022

0) Introductory word

Vuforia is a package distributed by the Qualcomm company, which provides an easy way to develop AR applications, especially in Unity.

Vuforia is free for non-commercial use, but requires Internet access for proper functioning. First you will have to **register on the Vuforia developer site**:

https://developer.vuforia.com/

Once you have registered, you have access to an online account, where you can store and manage a number of resources. You will need:

- A development key: this is a string that you will copy-paste in Unity to make Vuforia work. Development keys are free, but deployment keys (to distribute the app) are not.
- Target databases: Vuforia being marker-based, you will manage customized sets for your Unity apps.

We will use both objects below.

1) Install Unity

Starting with Unity.2017.2, Vuforia is supported natively. Current Unity version for the tutorial is 2020.3.11f1 LTS (but should support earlier versions – avoid 2021 for now). At installation time, Unity asks for modules to be optionally installed. Install the Android Build Support (for exporting games to Android platforms such as smartphones), or the iOS Build Support (Iphones and Ipads).

Then go to Vuforia developer website, and download the current Vuforia for Unity package version at:

`https://developer.vuforia.com/downloads/sdk`

This should download a file named `add-vuforia-package-10-11-3.unitypackage`

In your Unity project, import the package (`Assets` → `Import Packages` → `Custom Package`).

NOTE: if there are errors during import (`Project has invalid dependencies: No git executable was found`), you will need to install the git package from http://git-scm.com/download/win, and add your git install path (default is `C:\Program Files\Git\bin`) to the PATH environment variable of your computer. Then restart computer, relaunch Unity, and reimport package. Resolving dependencies may take a few minutes, be patient.

To help learning Vuforia, a number of sample unity assets exist. The best way to start is with the **Vuforia Core Samples** (available in the Unity asset store), which contains a sample for all main Vuforia objects. Core samples should be compatible with the current "Vuforia for unity" package.

2) Target types

The best way to start experimenting with targets is to browse the documentation on the developer site (Developer Library). Available are:

a) **Image targets**: common flat targets made out of one 2D image (from existing image file or captured with camera). Contrary to older packages (e.g. ARtoolkit), images do not have a predefined content (black and white square), they can be anything… well nearly anything. Preferably the target will be rectangular.
b) **Multi-targets**: similar to older ARToolkit multi-targets: a 3D rigid arrangement of image targets placed on a rigid geometry (e.g. a cube).
c) **Cylinder targets**: an image wrapped on a cylinder to provide tracking for things like soda cans. I do not recommend these, they can be touchy.
d) **Model targets:** use existing 3D models (CAD designed) and turn them into 3D markers.
e) **Object targets**: you "scan" an object with a camera, and the collected set of images become the marker. A sort of 3D marker but not made of predefined images.
f) **VuMark** : special kind of image target that can encapsulate meta-data. Uniquely identified, they are a sort of predefined "barcode" markers. You can easily create new ones, and they can contain stuff like a company logo, etc.
g) **Area Targets**: track areas and spaces (i.e. large spaces with no specific objects, like a room). Areas are scanned with specific hardware, e.g. ARKit enabled devices WITH builtin lidar sensors. Uses a separate app. See documentation for details.

Additionally, Vuforia has some built-in ARCore/ARKit functionalities to track *ground planes*, typically the ground or a table top, i.e. a large horizontal surface. Virtual objects can then be placed relatively to the detected plane (on it, above it). See documentation online.

3) Getting started: Image Targets

Let's do a simple image target example.

If you have enabled the vuforia package during installation, a Vuforia item should be present in the GameObject dropdown menu. There you will need an "ARCamera", the main object. This is like a regular camera except it has a script "Vuforia Behaviour" associated. In the Inspector, you can find this script, and open up the "Vuforia Configuration menu" with a button.

The first thing is to copy-paste a valid license key, obtained via the developer site.

The second thing is to have a valid image database available. There is a default one, the "Mars" database, that you can just use by checking the boxes (images are located in the Assets/Editor/Vuforia/ForPrint subdirectory, you can print them from the files there). Otherwise you can create your own on the developer site, and download it via the "Add Database" button (so you need Internet connection to add new datasets – using them does not require Internet access). Then they will become available for use. Details on how images should be selected and added to a database are found on the developer site (see “Features/Images/Image Targets” page on the Developer Library). You will find that sometimes the image is ranked “ok” by the target creation software, but the detection is not that good in real-life… life sucks.

There are more options in the `ARCamera`, in particular you can choose your webcam and calibrate it if your webcam is not known (although default parameters will work most of the time).

After that, you can add a GameObject of type: Vuforia/Image in drop-down menu. This will create an instance of an ImageTarget in the Unity world. Beware! Default unit is meter, so your Image will likely be very small, zoom in to make sure it is there. The scale factor should be ok, and correspond to the actual size you specified when the target was added in the database.

To link a Unity 3D model to the marker, just import your object in your scene and put it as a “son” of the target in the Unity hierarchy, again with proper scale.

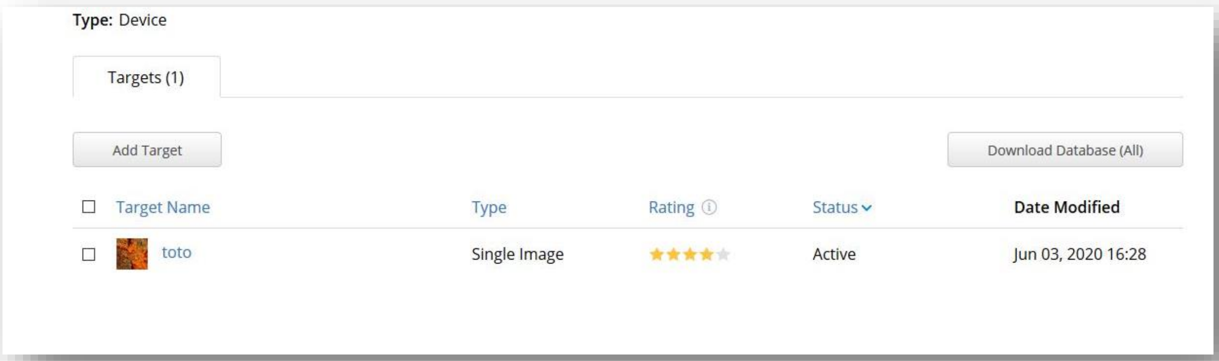
All done ! Now run the Unity project and put the physical image in front of the camera, the object should appear on top of it (if this is where you put it). Of course the object can be animated, like in the ImageTargets scene provided as a sample.

4) Creating a new Database (Image Target)

Instead of using the predefined “Mars” database, let us make a new one. On the developer Website, got to the Target Manager tab (https://developer.vuforia.com/vui/develop/databases), and create a new database (“Add Database” button). Choose a name, and stick to the “Device” type (the database will be downloaded on your local device at runtime). Then “Add Target” of type “Single Image”. You will upload the desired image. There are rules to select a proper target image: such a marker should have textures, colors, and a lot of distinctive features. Try and follow the guidelines of the documentation as stated in:

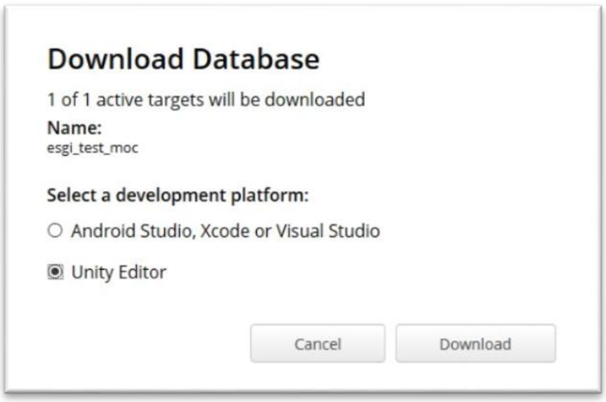
https://library.vuforia.com/features/images/image-targets/best-practices-for-designing-and-developing-image-based-targets.html

Then, select the width of the target AS IT WILL BE PRINTED (beware to deactivate any rescaling at printing time), expressed in meters (usually a fraction, e.g. 0.12), and a name for the target. Once you press “Add”, the image will be added to your database on the website. Reload the page after a few seconds to see is the image was accepted as a proper target: it should have at least a 4-star rating:


Type: Device

☐	Target Name	Type	Rating	Status	Date Modified
☐	toto	Single Image	★★★★☆	Active	Jun 03, 2020 16:28

Finally, download the database AS A UNITY PACKAGE:



Download Database

1 of 1 active targets will be downloaded

Name:
esgi_test_moc

Select a development platform:

- [ ] Android Studio, Xcode or Visual Studio
- [✓] Unity Editor

The database "test" will thus produce a test.unitypackage file, once downloaded. You can add it to your scene with the standard Assets->Import Package command.

The database can now be selected in the list of possible ones in your ImageTarget properties. Of course you can add more than one ImageTarget. A database can also contain other target types:

5) More target types

There are other targets than just simple images. For example, you can create **MultiTargets**, which are composite targets made out of multiple simple ones. These multi targets are described in a special xml file. If your image database (the one you created on your developer site) is named MySetOfImages, then the file is `MySetOfImages.xml`. You have to put it in the repository, and (if needed) reload the image database in your project. Here is what the default file looks like (from the "Mars" database:

```
<?xml version="1.0" encoding="UTF-8"?>
<QCARConfig>
    <Tracking>
        <MultiTarget name="MarsBox">
            <Part name="MarsBox.Left" translation="-
0.0381000000470876694 0 0" rotation="AD: 0 1 0 -90"/>
            <Part name="MarsBox.Right"
translation="0.0381000000470876694 0 0" rotation="AD: 0 1 0 90"/>
            <Part name="MarsBox.Front" translation="0 0
0.0190500000235438347" rotation="AD: 1 0 0 0"/>
            <Part name="MarsBox.Back" translation="0 0 -
0.0190500000235438347" rotation="AD: 0 1 0 180"/>
            <Part name="MarsBox.Top" translation="0
0.05079999938607216 0" rotation="AD: 1 0 0 -90"/>
            <Part name="MarsBox.Bottom" translation="0 -
0.05079999938607216 0" rotation="AD: 1 0 0 90"/>
        </MultiTarget>
    </Tracking>
</QCARConfig>
```

Each subpart is a separate database image, added as a simple target. It is placed in space with a 3D translation (again in meters), and a rotation expressed as Axis/angle (in degree). As in ARToolkit, the resulting multi-target can be 2D or 3D (in the sample it is a box shaped object).

Once it has been created, the multi-target can be used by adding a "MultiTarget" object in your Unity scene and selecting the proper database and target.

Note: you can find the data files in your Unity project in the folder:

```
Your_project_location\Assets\StreamingAssets\Vuforia
```

Note : the default `VuforiaMars_Images.xml` file also contains a CylinderTarget that works similarly, except it is designed for cylindrical objects.

6) Exports

Once your Unity scene behaves as expected you can export the resulting code into the target architecture of your choice (see File/Build Settings menu). For instance, for the Hololens AR device, one would select "Universal Windows". To export on an Android smartphone, select "Android", etc.: building should generate the corresponding executable that you can download on your phone/tablet/gizmo. If things go wrong, try a simple Unity project without AR in it. Usually, AR is not the problem. Possible trouble: minimum version of Android, Android TV Compatibility (should be off), and whatever is in "Player Settings".