21 March 2017

Building a Simple Augmented Reality (AR) Smartphone Application in Unity

Lab 4 PDF (Student Copy)

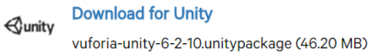
Deliverables

- Send your Unity AR project to charmaine_wee@sutd.edu.sg (email subject: Lab 4 AR Submission)
- Show the instructor the working AR you have built

Guide

Download Vuforia (extension for Unity)

1. Go to https://developer.vuforia.com/downloads/sdk and create a Vuforia account to download the unitypackage.



Download for Unity

vuforia-unity-6-2-10.unitypackage (46.20 MB)

Create License Key (Vuforia)

2. Go to https://developer.vuforia.com/license-manager.
3. Click "Add License Key".
4. Select "Development" as your Project Type.

Creating Markers for your project

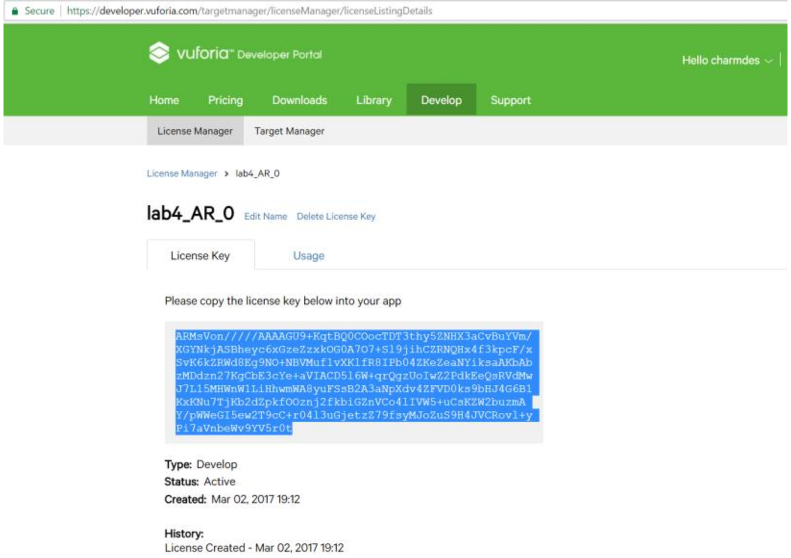
5. Go to https://developer.vuforia.com/target-manager (you might need to sign in to your Vuforia account).
6. Click "Add Database".
7. Select "Device".
8. Click on the new database you've just made.
9. Click "Add Target". Fill in the fields.
10. Click "Download Database (All)".

Creating Project in Unity

11. Create a new project in Unity for this lab assignment.
12. Double-click on the Unity extension file you downloaded in step 1.
13. Accept import of the file into your Unity project.

Add License Key to Unity Project

14. In the License Manager (from step 2), click on the license key you created, and copy the license key.


Secure | https://developer.vuforia.com/targetmanager/licenseManager/licenseListingDetails
vuforia™ Developer Portal
Hello charmdes
Home
Pricing
Downloads
Library
Develop
Support
License Manager
Target Manager
License Manager › lab4_AR_0
Edit Name
Delete License Key
License Key
Usage

lab4_AR_0

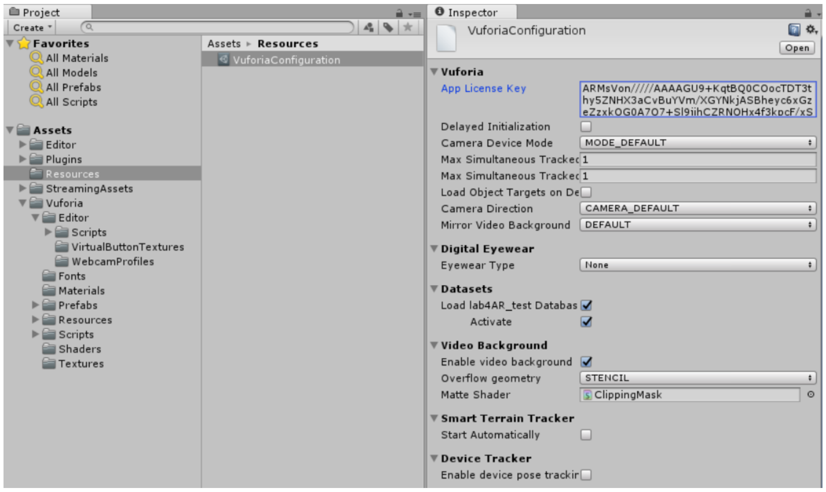
Please copy the license key below into your app

```
ARMsVon/////AAAAGU9+KqtBQ0COocTDT3thy5ZNHX3aCvBuYVm/
XGYNkjASBheyc6xGzeZzxkOG0A7O7+Sl9jihCZRNQHx4f3kpcF/x
SvK6kZRWd8Eg9NO+NBVMuflvXKlfR8IPb04ZKeZeaNYiksaAKbAb
zMDdzn27KgCbE3cYe+aVIACD5l6W+qrQgzUoIwZ2PdkEeQsRVdMw
J7L15MHWnW1LiHhwmWA8yuFSsB2A3aNpXdv4ZFVD0ks9bHJ4G6B1
KxKNu7TjKb2dZpkfOOznj2fkbiGZnVCo4lIVW5+uCsKZW2buzmA
Y/pWWeGI5ew2T9cC+r04l3uGjetzZ79fsyMJoZuS9H4JVCRovl+y
Pi7aVnbeWv9YV5r0t
```

Type: Develop
Status: Active
Created: Mar 02, 2017 19:12

History:
License Created - Mar 02, 2017 19:12

15. In your Unity project, go to Assets>Resources>VuforiaConfiguration and paste the license key in the field labelled "App License Key".



VuforiaConfiguration

Vuforia
App License Key: ARMsVon/////AAAAGU9+KqtBQ0COocTDT3thy5ZNHX3aCvBuYVm/XGYNkjASBheyc6xGzeZzxkOG0A7O7+Sl9jihCZRNQHx4f3kpcF/xS
Delayed Initialization ☐
Camera Device Mode: MODE_DEFAULT
Max Simultaneous Trackec: 1
Max Simultaneous Trackec: 1
Load Object Targets on De ☐
Camera Direction: CAMERA_DEFAULT
Mirror Video Background: DEFAULT

Digital Eyewear
Eyewear Type: None

Datasets
Load lab4AR_test Databas ☑
Activate ☑

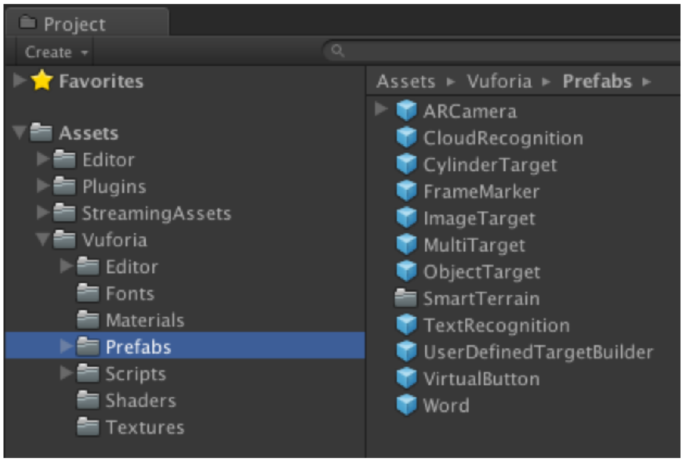
Video Background
Enable video background ☑
Overflow geometry: STENCIL
Matte Shader: ClippingMask

Smart Terrain Tracker
Start Automatically ☐

Device Tracker
Enable device pose trackir ☐

Adding your Target marker to your Unity Project

16. Double-click the downloaded package from step 3 (I.e. import it into your Unity project).
17. Your project structure should now look like this:



Adding AR assets and Prefabs to your Scene

18. Delete the "Main Camera".
19. Add the "ARCamera " in the Prefabs folder into the scene (replacing the deleted camera).
20. Also from the Prefabs folder, drag "ImageTarget" into the scene.
21. Select "ImageTarget" in your scene so that its properties appear in the Inspector tab.
22. In the attached script "Image Target Behaviour", fill in the fields of Database, Image Target and Width.

Adding 3D Objects to the Scene, and Attaching them to Trackables

23. As a test, create a simple Cube object (GameObject > 3D Object > Cube).
24. Add the cube as a child of the ImageTarget object.
25. Adjust orientation/size/position of the cube to your liking, with reference to the ImageTarget object.

App Deployment (for Android and IOS)

Android App Deployment

1. Go to File > Build Settings.
2. Select "Android as your Platform.
3. Click on "Player Settings" and refer to the Inspector tab.
4. Click Other Settings.
5. Set the Minimum API Level to Android 4.1 'JellyBean' (API level 16) or higher.
6. Set Bundle Identifier to a valid name.
7. Save your scene (File > Save Scene).
8. Again, open the File > Build Settings. Ensure your scene is part of "Scenes in Build". If not, click "Add Open Scenes".

You can now build the application. Attach your Android device and then click Build and Run to initialize the deployment process.

IOS App Deployment

1. Go to File > Build Settings.
2. Select "IOS" as your Platform.

3. Ensure the Target Platform is *not* set to armv6 (OpenGL ES 1.1). As of Vuforia version 6.2, the Unity extension supports OpenGL_ES 2 and 3.
4. Ensure the Bundle Identifier is set to the correct value for your IOS developer profile.
5. Save your scene (File > Save Scene).
6. Open the build menu (File > Build Settings).
7. Ensure your scene is part of the "Scenes in Build". If not, click "Add Current" to add the currently active scene.
8. Build the app.

When building and running apps for iOS, Unity generates an Xcode project. It launches Xcode and loads this project. The Vuforia AR Extension includes a PostProcessBuildPlayer script that performs the task of integrating the Vuforia library into the generated Xcode project. This is run automatically when you select Build from within Unity. Be aware that if you manually change the generated Xcode project, you may need to update the PostProcessBuildPlayer script to avoid overwriting your changes.

The generated Xcode project includes a file called AppController.mm. There are Unity provided options in this file to tailor the performance of the app for your own purpose. The PostProcessBuildPlayer script sets the THREAD_BASED_LOOP as a default because it gives the best visible performance with the samples provided alongside the Vuforia AR Extension. Consider changing these options to whatever gives the best performance for your own application.

- End -