iOS QR for Unity ReadMe V1.0

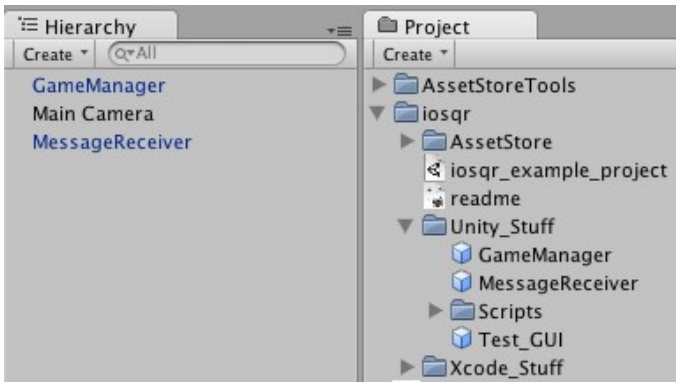
FEATURES:

* QR functionality inside your Unity iOS-App
* Just Drag&Drop
* ready to edit the Script for your needs

This Asset provides the QR functionality inside your Unity iOS-App.
The plug-in is based on zbar and the iOS SDK from Apple.
Requires the **Unity Mobile Version** with a **Pro-License**
and you already have to be an authorised iOS Developer.

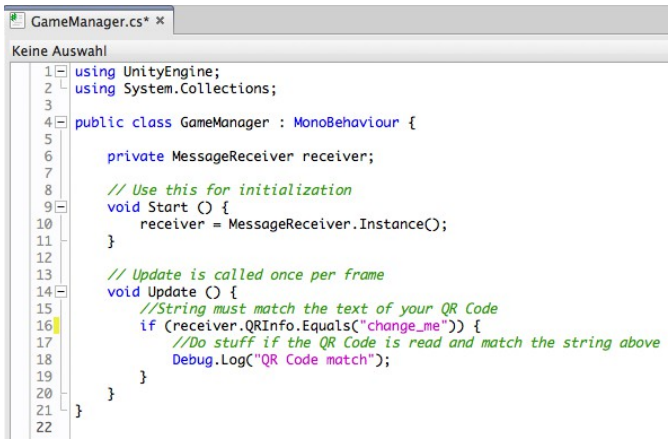
Only seven steps for installing the QR-Code Plugin into your project:

1. Download the Asset and import everything in your Unity App.

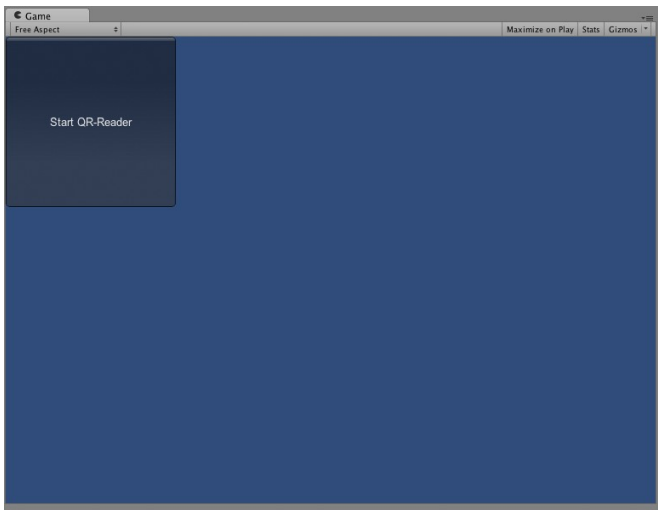


2. Drag&Drop both Prefabs "GameManager" and "MessageReceiver" (located at /Unity_stuff) into your scene hierarchy.
(Do not edit the MessageReceiver GameObject nor the MessageReceiver and the UIBlinding scripts. Otherwise the plug-in will not work. You may change the name of the GameManager GameObject if you want to.)

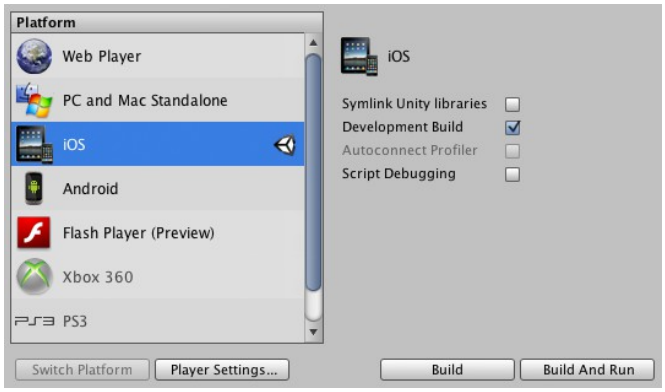
3 Edit the GameManager Script for your needs



3.1 Just replace the string from the if-statement (called "change_me" at the beginning). This string must match the text of your QR Code.



3.2 To start the QR-Reader you must call " UIBinding.ActivateUI(); " (See the "Test_GUI" GameObject and script for reference - Drag&Drop it into your scene if you want a button which starts the QR-Reader)

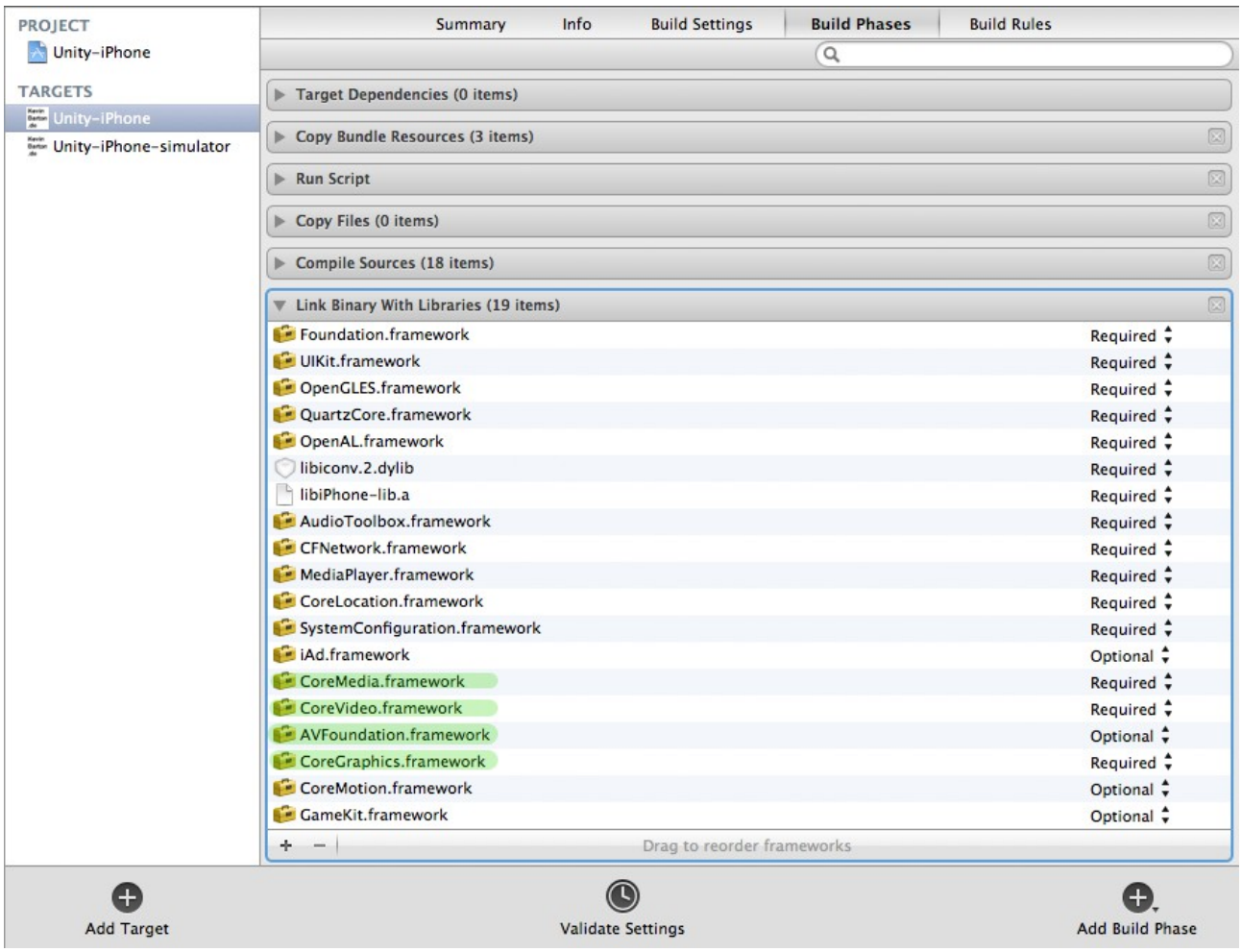


4. If you haven't already, set your project to the iOS Platform (file -> build settings) and confirm by pressing the <switch platform> button

4.1 Once you are done build&run the project on the iOS Device (be sure the settings match your preferences (Edit -> Project Settings -> Player) if you dont have your iOS Bundle Identifier or dont have the unity pro license, the project wont be build).

4.2 Stop XCode from compiling (if you don't stop it you will get some errors. Don't worry about that).

5 Add following frameworks as they are needed by Unity (if they don't exists already): AVFoundation.framework, CoreGraphics.framework, CoreMedia.framework and CoreVideo.framework.



You can add frameworks as follows:

In the project navigator, select your project

1. Select your target
2. Select the 'Build Phases' tab
3. Open 'Link Binaries With Libraries' expander
4. Click the '+' button
5. Select your framework
6. (optional) Drag and drop the added framework to the 'Frameworks' group

6. Drag the folder "iOSUnityQR" (located in /Xcode_Stuff) into your XCode Project. (This folder includes the library and all needed resources.)

7. Build&Run your project in xcode.

Notice: The QR-Reader only works on the device. Be sure you are not running the simulator target.

Update Version 1.4

You now have more control about how the plugin works. The new lines of code in the QRGUI.cs look like follows:

```
19          if (GUI.Button(new Rect(10, 70, 70, 30), "QR Start!")) {
20              Debug.Log("Clicked the button with text");
21              UZBarReaderViewController zBar = new UZBarReaderViewController();
22              zBar.cameraDevice = kCameraDevice.ZBAR_CAMERA_DEVICE_REAR;
23              zBar.symbolType = kScanSymbolType.ZBAR_I25;
24              zBar.configOpt = kScanConfigOptions.ZBAR_CFG_ENABLE;
25              zBar.configSymbolValue = 0;
26              zBar.cameraFlashMode = kCameraFlashMode.ZBAR_CAMERA_FLASH_MODE_AUTO;
27              zBar.showsZBarControls = true;
28              UIBinding.ActivateUI (zBar.getZBarInfos());
```

line 22: This line expresses which iPhone camera you want to use. Change the line to *„zBar.cameraDevice = kCameraDevice.ZBAR_CAMERA_DEVICE_FRONT;“* if you want to use the front camera.

Line 23: This line controls which type of code you are able to scan. See the following list for more information:

http://zbar.sourceforge.net/iphone/sdkdoc/ZBarSymbol.html

3.8.4. Constants

line 26: This line controls the flash light. You can change it to „on“, „off“ and „auto“ which is the default setting

line 27: Disables or enables the zBar controls

line 28: Activates the QR-Reader. This line has always to be at the end.

For further support please visit:
www.iosqrforunity3d.com
For support via E-Mail:
daubit@daubit.org
www.daubit.org