DeepLink Plugin for Android in Unity

By CausalLink (causallink.asset@gmail.com)

Version 1.1

Thank you for purchasing the DeepLink Plugin!

Requirements

Tested and works on Android version 2.3.1 API 9 and higher. To test you need an Android emulator or a physical device.

What can you do with the DeepLink Plugin?

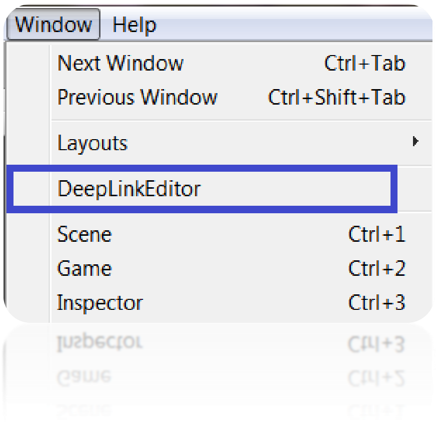
With the DeepLink plugin, you are able to open your android app from a link, and pass its data to your app. You can use a link similar to the following, and use the obtained values to do specific tasks.

myapp://com.companyname.appname?var1=somevalue&val2=anothervalue

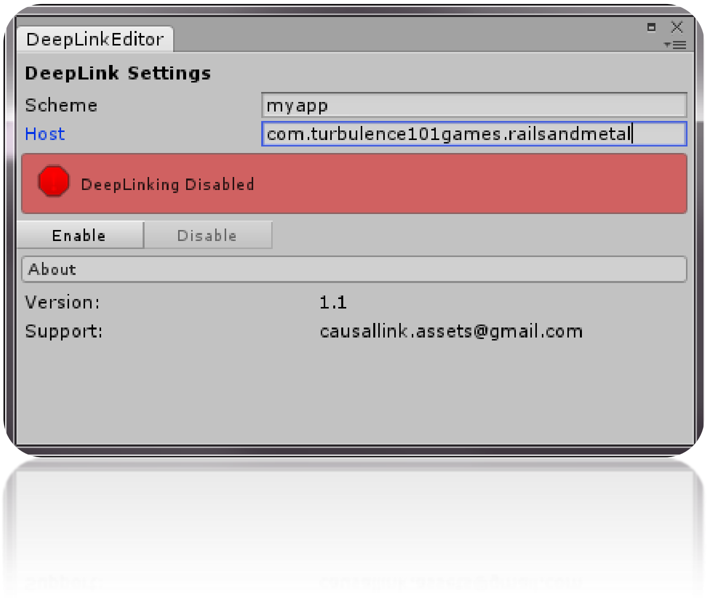
For example, you may grant free in-app currency from a link or give bonus for in-app purchases..etc.

How to use the Plugin

1. Import the plugin into unity.
2. Click Windows->DeepLinkEditor



3. Type in the desired scheme and host. This determines how your app's DeepLink will look.



DeepLink Settings

Scheme: myapp

Host: com.turbulence101games.railsandmetal

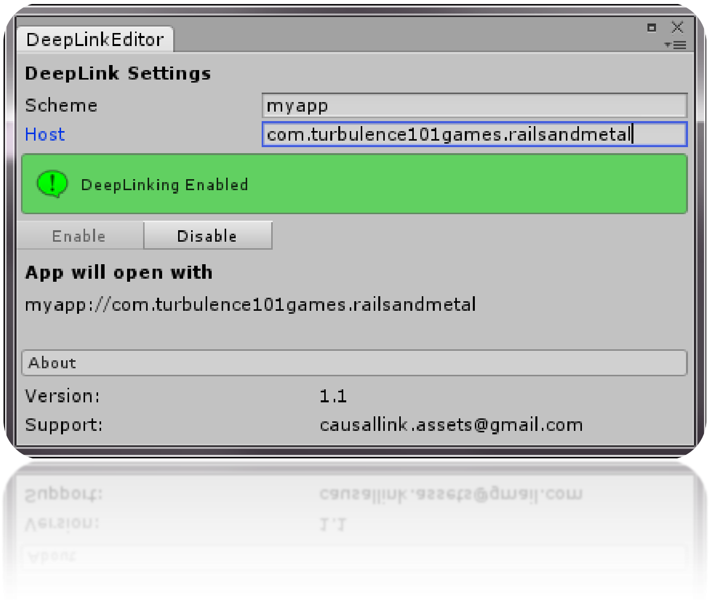
DeepLinking Disabled

About

Version: 1.1

Support: causallink.assets@gmail.com

4. Click on Enable. Now your app can be opened with the DeepLink.



DeepLink Settings

Scheme	myapp
Host	com.turbulence101games.railsandmetal

DeepLinking Enabled

App will open with

myapp://com.turbulence101games.railsandmetal

Version:	1.1
Support:	causallink.assets@gmail.com

5. To Obtain DeepLink's variables, simply use these functions. That's it!

If your URI is as follows:

myapp://com.turbulence101games.railsandmetal?intval=2&floatval=1.1&stringval=Hello!

Note: Before declaring the first variable, there must be a "?", after that, all of the variables must end with "&" except the last variable.

To obtain these values in Unity, use these functions:

```
AndroidDeepLink.GetURL();
AndroidDeepLink.GetValueInInt("intval");
AndroidDeepLink.GetValueInFloat("floatval");
AndroidDeepLink.GetValueInString("stringval");
```

Technical Details

A. IF THE PROJECT DOES NOT HAVE AN *ANDROIDMANIFEST.XML* FILE IN (ASSETS\PLUGINS\ANDROID)

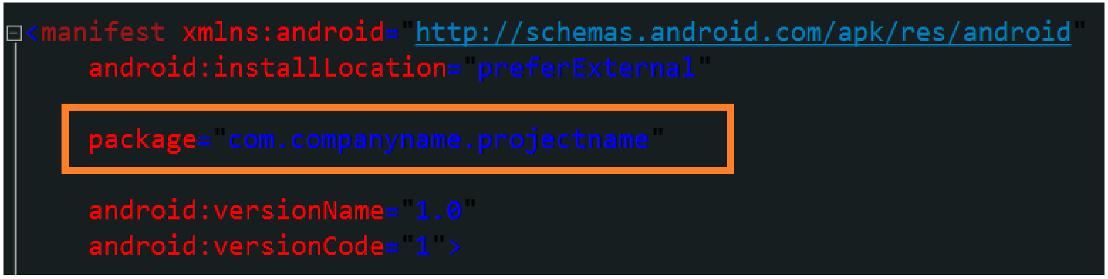
When the Enable button is pressed, AndroidManifest.xml file will be copied from Assets\DeepLink Plugin\ into Assets\Plugins\Android and the following code is copied into the activity that contains "`android.intent.action.MAIN`"

Important: THERE MUST ONLY BE ONE "`android.intent.action.MAIN`" In the Manifest file.

```
<intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="myapp"
android:host="com.turbulence101games.railsandmetal" />
</intent-filter>
```

Values of `android:scheme` and `android:host` will be what you have set in the editor. Above set of code will enable your app to be opened via a DeepLink. In this case, it will open with *Myapp://com.turbulence101games.railsandmetal*

Also, the package name on the manifest will be changed to Project's bundle ID.



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="preferExternal"

    package="com.companyname.projectname"

    android:versionName="1.0"
    android:versionCode="1">
```

B. IF THERE IS AN *ANDROIDMANIFEST.XML* FILE ALREADY IN (ASSETS/PLUGINS/ANDROID)

Important: THERE MUST ONLY BE ONE “`android.intent.action.MAIN`” In the Manifest file.

When the Enable button is pressed, there will be a backup of the original Manifest made in Assets\DeepLink Plugin\Manifest Backup including the date and time. Next it will find the action with “`android.intent.action.MAIN`” and the following code is copied:

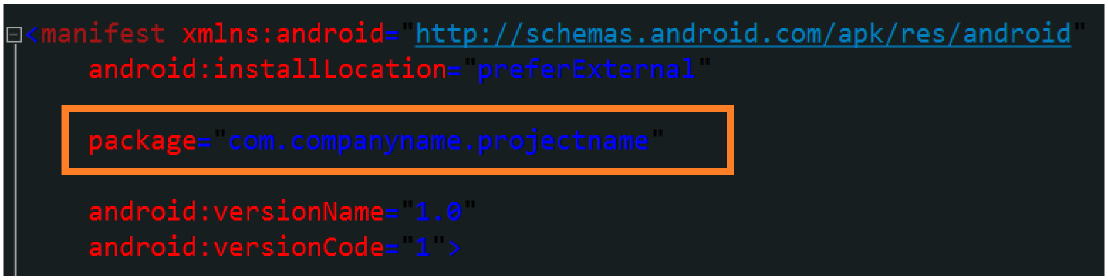
```
<intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="myapp"
android:host="com.turbulence101games.railsandmetal" />
</intent-filter>
```

Values of `android:scheme` and `android:host` will be what you have set in the editor. Above set of code will enable your app to be opened via a DeepLink. In this case, it will open with *Myapp://com.turbulence101games.railsandmetal*

Troubleshooting

App does not open with the Deep Link:

Make sure the package name is equal to the bundle id of the project.



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="preferExternal"

    package="com.companyname.projectname"

    android:versionName="1.0"
    android:versionCode="1">
```

Also make sure that there is only one “android.intent.action.MAIN” line in the Manifest, and also make sure that code similar to following is in that activity.

```
<intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="myapp"
android:host="com.turbulence101games.railsandmetal" />
</intent-filter>
```

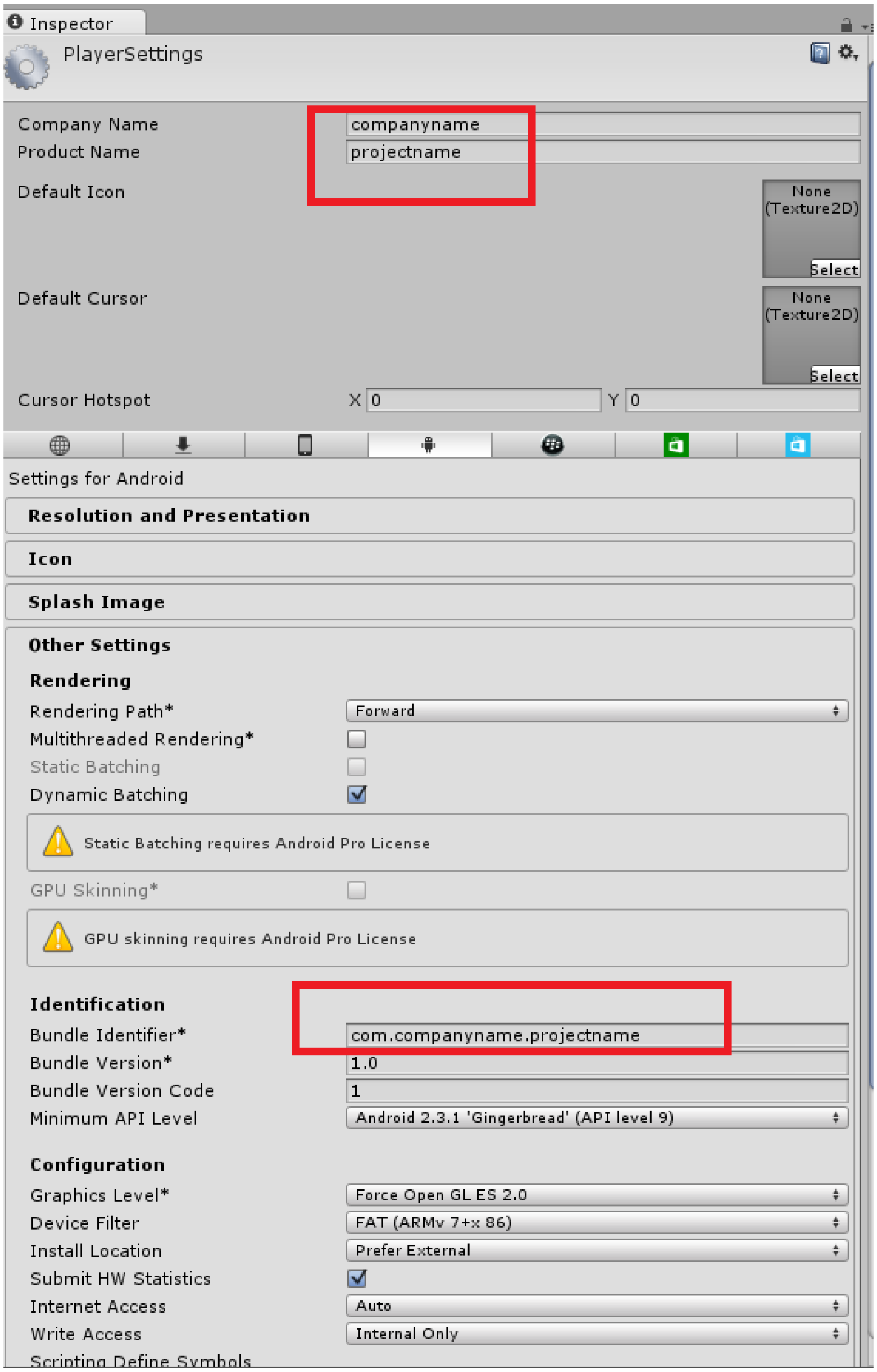
Example:

```
<activity android:name="com.unity3d.player.UnityPlayerNativeActivity"
android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <category
android:name="android.intent.category.LEANBACK_LAUNCHER" />
      </intent-filter>
      <meta-data android:name="unityplayer.UnityActivity"
android:value="true" />
      <meta-data
android:name="unityplayer.ForwardNativeEventsToDalvik"
android:value="false" />
      <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="myapp"
android:host="com.turbulence101games.railsandmetal" />
      </intent-filter>
    </activity>
```

Testing - Testing the example scene

Example in action: http://youtu.be/B8LKkkEvirA

1. Open the example scene, and set the player settings as follows:



Inspector

PlayerSettings

Company Name: companyname
Product Name: projectname
Default Icon: None (Texture2D) Select
Default Cursor: None (Texture2D) Select
Cursor Hotspot: X 0 Y 0

Settings for Android

Resolution and Presentation

Icon

Splash Image

Other Settings

Rendering

Rendering Path*: Forward
Multithreaded Rendering*: □
Static Batching: □
Dynamic Batching: ☑

⚠ Static Batching requires Android Pro License

GPU Skinning*: □

⚠ GPU skinning requires Android Pro License

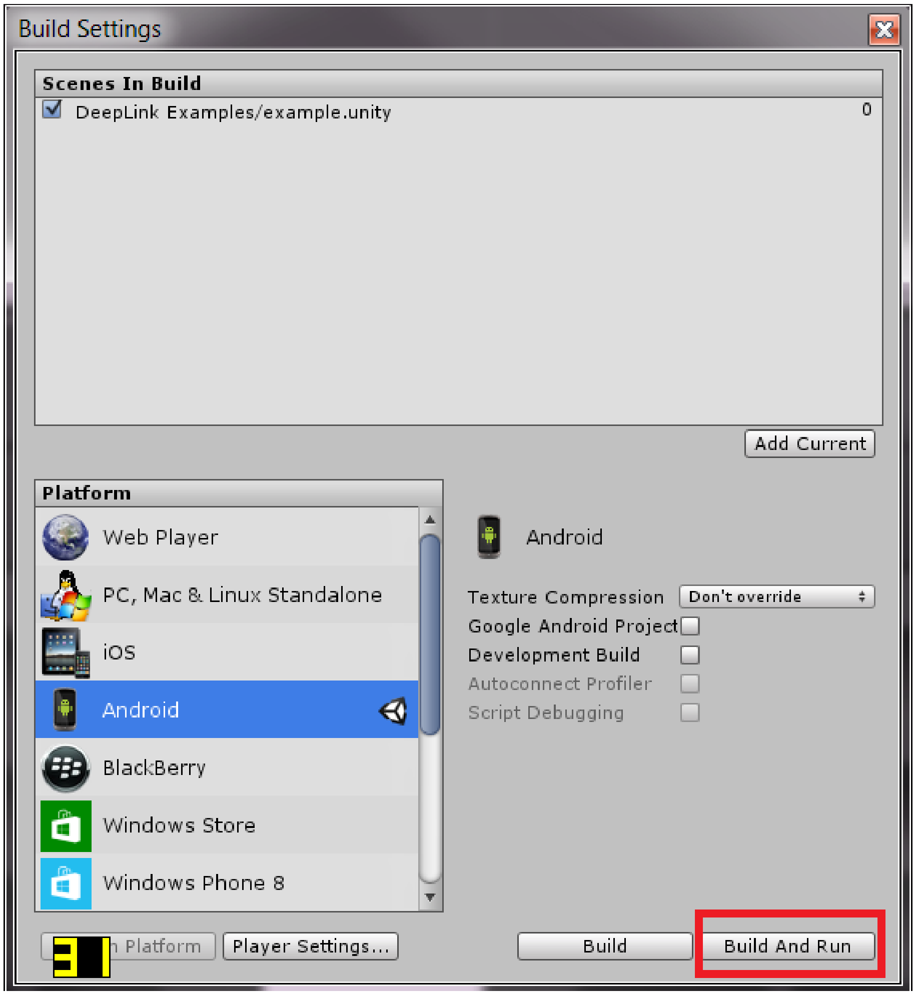
Identification

Bundle Identifier*: com.companyname.projectname
Bundle Version*: 1.0
Bundle Version Code: 1
Minimum API Level: Android 2.3.1 'Gingerbread' (API level 9)

Configuration

Graphics Level*: Force Open GL ES 2.0
Device Filter: FAT (ARMv 7+x 86)
Install Location: Prefer External
Submit HW Statistics: ☑
Internet Access: Auto
Write Access: Internal Only
Scripting Define Symbols

2. Connect your Android Device or turn on your android emulator, then click Build And Run



Build Settings

Scenes In Build

☑ DeepLink Examples/example.unity 0

Platform

- Web Player
- PC, Mac & Linux Standalone
- iOS
- Android
- BlackBerry
- Windows Store
- Windows Phone 8

Android

Texture Compression: Don't override

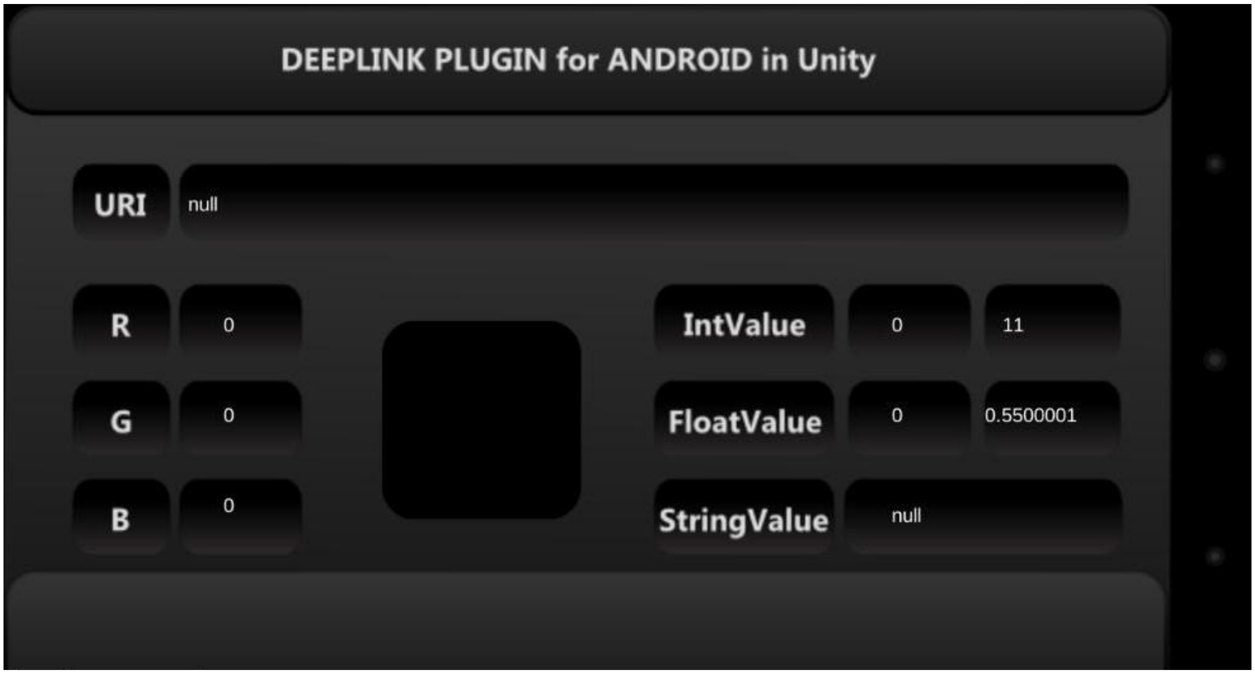
Google Android Project □

Development Build □

Autoconnect Profiler □

Script Debugging □

3. If everything is correct, you will see something like this: Note, There is no values filled: this is because we did not open the app using a link.



DEEPLINK PLUGIN for ANDROID in Unity

URI null

R 0

G 0

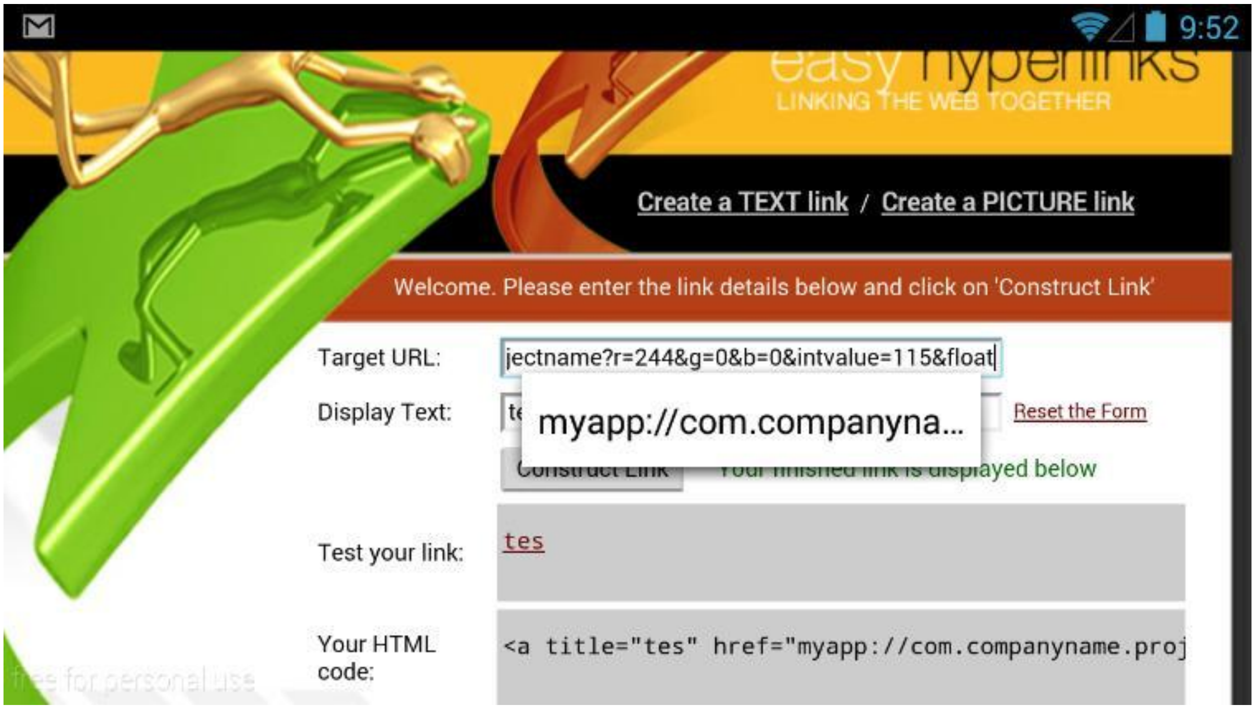
B 0

IntValue 0 11

FloatValue 0 0.5500001

StringValue null

4. To test with a link, you can use any hyperlinktester, here we used http://easyhyperlinks.com/ insert the target url as follows: myapp://companyname.projectname?r=somevalue&g=somevalue&b=somevalue&intvalue=somevalue&floatvalue=somevalue&svalue=somevalue



Welcome. Please enter the link details below and click on 'Construct Link'

Target URL: jectname?r=244&g=0&b=0&intvalue=115&float

Display Text: te

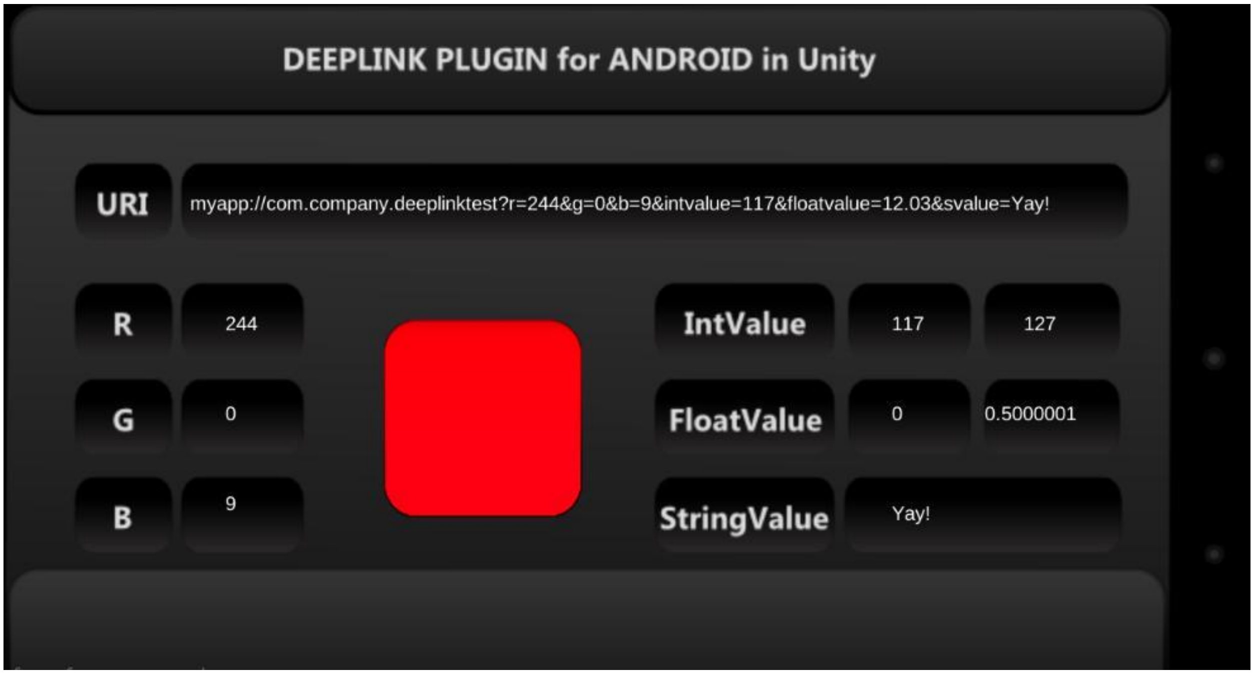
myapp://com.companyna...

Your finished link is displayed below

Test your link: tes

Your HTML code: `<a title="tes" href="myapp://com.companyname.proj`

5. Then click on Generate link, and click on the generated link. The result will be as follows, It will open the app with the given data



DEEPLINK PLUGIN for ANDROID in Unity

URI myapp://com.company.deeplinktest?r=244&g=0&b=9&intvalue=117&floatvalue=12.03&svalue=Yay!

R 244

G 0

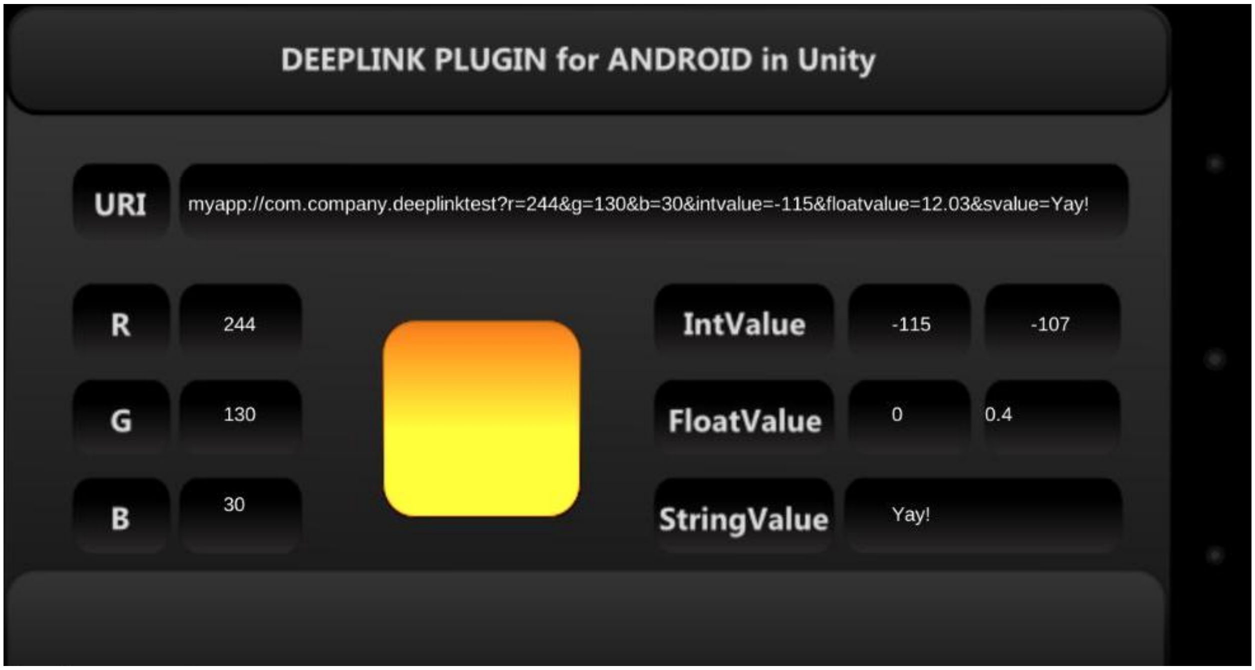
B 9

IntValue 117 127

FloatValue 0 0.5000001

StringValue Yay!

6. When passed different Data



DEEPLINK PLUGIN for ANDROID in Unity

URI myapp://com.company.deeplinktest?r=244&g=130&b=30&intvalue=-115&floatvalue=12.03&svalue=Yay!

R 244

G 130

B 30

IntValue -115 -107

FloatValue 0 0.4

StringValue Yay!