CameraShot Plugin

This plugin is to open device camera in image capture mode and video capture mode on Android and iOS.

Captured image and videos will save in default camera folder(i.e. Camera Roll).

CameraShot integration Guide

1). Import plugin to your project

2). Make sure there are following files:

Assets/Plugins/Android/AndroidCameraShot.cs

Assets/Plugins/Android/CameraShots.jar

Assets/Plugins/CamreaShot.dll

Assets/Plugins/CamreaShot_IOS.dll

Assets/Plugins/iOS/ libCameraShot.a

3). Drag “CameraShotListener” prefab from Assets/CameraShot/Prefab/ to your hierarchy

For Android

4). Edit or add

```
<activity
   android:name="com.astricstore.camerashots.CameraShotActivity"
   android:configChanges="orientation|keyboardHidden|screenSize">
 </activity>

 <activity

   android:name="eu.janmuller.android.simplecropimage.CropImage"

   android:configChanges="orientation|keyboardHidden|screenSize">

   </activity>
```

and Provider

```
<provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities="com.test.camerashot.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths"/>
        </provider>
```

in AndroidManifest.xml located at *Assets/Plugins/Android/*

Note: Replace “***com.test.camerashot.provider***” *to* ***“YOUR_PACKAGE_NAME.provider”,*** Suppose your package name is **“com.myproject.myproduct”** then your final line will be

```
            android:authorities=“com.my.myproduct.provider"
```

Your XML will look like

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.test.camerashots"
android:theme="@android:style/Theme.NoTitleBar" android:versionName="1.0" android:versionCode="1"
android:installLocation="preferExternal">

  <supports-screens android:smallScreens="true" android:normalScreens="true"
android:largeScreens="true" android:xlargeScreens="true" android:anyDensity="true" />

  <application android:icon="@drawable/app_icon" android:label="@string/app_name"
android:debuggable="false">

    <activity android:name="com.unity3d.player.UnityPlayerNativeActivity"
android:label="@string/app_name" android:screenOrientation="portrait"
android:launchMode="singleTask" android:configChanges="mcc|mnc|locale|touchscreen|
keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|
smallestScreenSize|fontScale">

      <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
      <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
      <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="false" />
    </activity>

   <activity
     android:name="com.astricstore.camerashots.CameraShotActivity"
     android:configChanges="orientation|keyboardHidden|screenSize">
   </activity>

  <activity

     android:name="eu.janmuller.android.simplecropimage.CropImage"

     android:configChanges="orientation|keyboardHidden|screenSize">

     </activity>

 <provider
            android:name="android.support.v4.content.FileProvider"
            android:authorities=“com.test.camerashot.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths"/>
        </provider>


  </application>
  <uses-sdk android:minSdkVersion="9" android:targetSdkVersion="18" />
  <uses-feature android:glEsVersion="0x00020000" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

For IOS

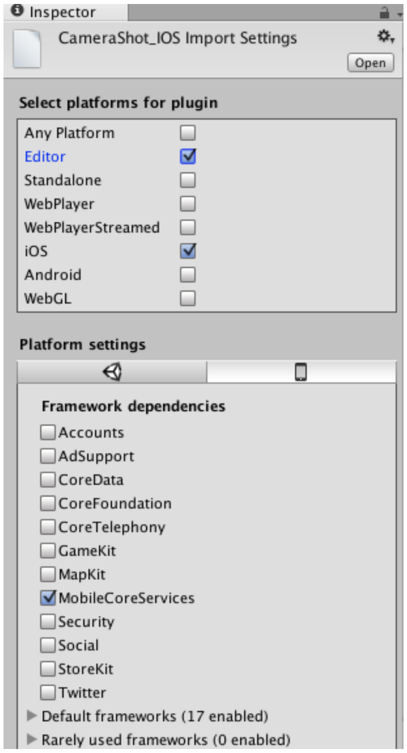
1). Unity 4.x users : *Add “MobileCoreServices.framework” in xCode*

Select on you project in hierarchy > Build Phase > Link Binary with libraries > Click on add (+) button > search MobileCoreServices > select and add

2). Unity 5.x users : *Add “MobileCoreServices.framework”*

- *Select Assets/Plugins/iOS/ libCameraShot.a > Mark “MobileCoreServices“ in framework dependencies.*

- *Select Plugins/CameraShot_IOS.dll and uncheck “**Any Platform” ,** mark **Editor & iOS** only in inspector as shown in image below.*



CameraShot_IOS Import Settings

Select platforms for plugin

Any Platform	□
Editor	☑
Standalone	□
WebPlayer	□
WebPlayerStreamed	□
iOS	☑
Android	□
WebGL	□

Platform settings

Framework dependencies

□ Accounts
□ AdSupport
□ CoreData
□ CoreFoundation
□ CoreTelephony
□ GameKit
□ MapKit
☑ MobileCoreServices
□ Security
□ Social
□ StoreKit
□ Twitter

▶ Default frameworks (17 enabled)
▶ Rarely used frameworks (0 enabled)

If you are working with xCode 8.x.x then you need to add these keys in info.plist

NSCameraUsageDescription
NSMicrophoneUsageDescription
NSPhotoLibraryUsageDescription

You can look here about how to add keys
http://unitydevelopers.blogspot.in/2017/05/add-keys-into-infoplist.html

How to use CameraShot plugin

1). Use below namespace where you want to call use this plugin

using CameraShot;

2). Call following function to launch camera for taking snapshot and save to gallery

Android : *AndroidCameraShot.LaunchCameraForImageCapture(false); //* ***true****, for edit*

IOS : *IOSCameraShot.LaunchCameraForImageCapture(false); //* ***true****, for edit*

3). Call following function to get texture from device camera, image also will save in gallery.

Android : *AndroidCameraShot.Get2DTextureFromCamera(false); //* ***true****, for edit*

IOS : *IOSCameraShot.Get2DTextureFromCamera(false); //* ***true****, for edit*

3). Call following function to launch camera for recording video and save to gallery

Android : *AndroidCameraShot.LaunchCameraForVideoCapture();*

IOS : *IOSCameraShot.LaunchCameraForVideoCapture();*

4). Record video for specific time, its available for Android only

Android : *AndroidCameraShot.LaunchCameraForVideoCapture(int maxDuration);*

IOS : *IOSCameraShot.LaunchCameraForVideoCapture(int maxDuration);*

5). Launch camera with both option (Image capture and video capture), its available for IOS only(Bonus feature)

IOS : *IOSCameraShot.LaunchCamera();*

Events:

```
# Fires when image capture by camera and saved to gallery.

CameraShotEventListener.onImageSaved += OnImageSaved;

void OnImageSaved(string path, ImageOrientation orientation)

{

}
```

```
# Fires when image capture by camera and load to unity, you can get image by tex parameter.

CameraShotEventListener.onImageLoad += OnImageLoad;

void OnImageLoad(string path,Texture2D tex, ImageOrientation orientation)

{

}
```

```
# Fires when video capture by camera , you can get video path by vidPath parameter.

CameraShotEventListener.onVideoSaved += OnVideoSaved;

void OnVideoSaved(string path)

{

}
```

```
# Fires when any error occurs.

CameraShotEventListener.onError += OnError;

void OnError(string errorMsg)

{

}
```

Fires when cancelled by user.

```
CameraShotEventListener.onCancel += OnCancel;

void OnCancel()

{

}
```

- IOS Specific : If you launch camera again then last image/video path will be overwrite, if you save image/video path for future use then it will not work.

For more info you can write me at devesh.pandey19@gmail.com