Camera Capture

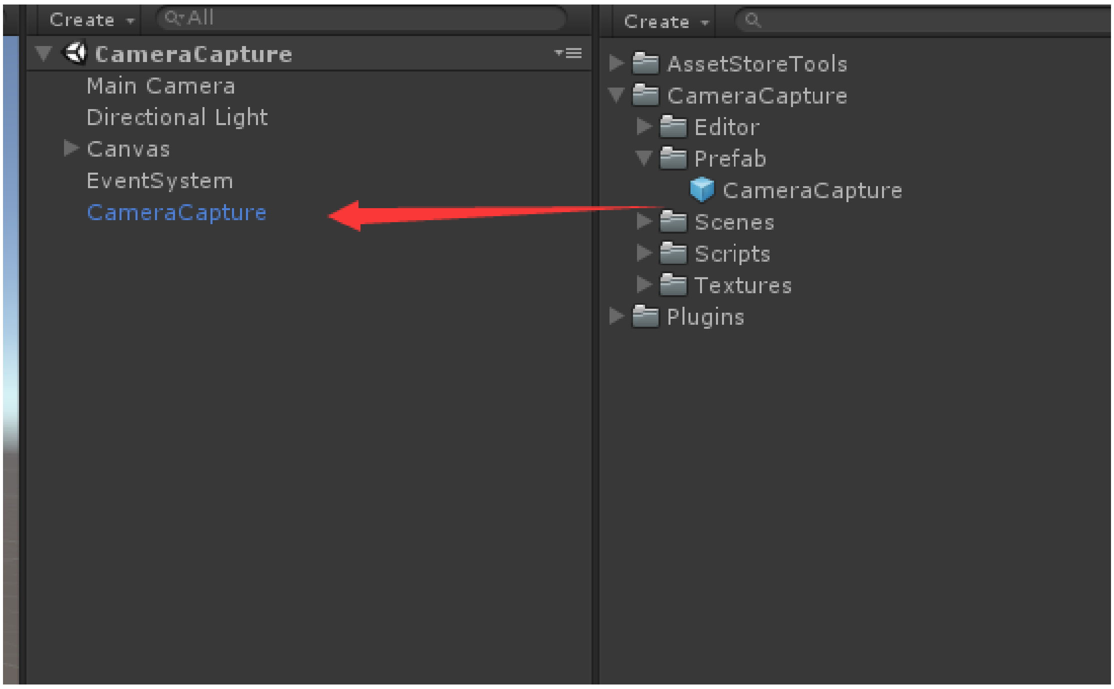
this plugin can easy to integrate with your project, you just need to write a few code to complete your work.it can support take photo and record video by device’ s camera , it available for Android And iOS platform.

How to integaration and use:

Please follow these steps to integrate this plugin with you project.

Step1: Please import this plugin into your project.

Step2: Drag the prefab named “**CameraCapture**” from Assets/CameraCapture/Prefab/ into hierarchy.



Step3:

(1) Camera Capture Features:

1) Get **CameraCapture** reference Object

```
CameraCapture camCap= GameObject.FindObjectOfType<CameraCapture> ();
```

2) Take Photo (it will be saved in gallery)

//this api will open the camera and make you can take photo to save to gallery auto

2.1) *camCap.takePhoto();*

2.2) take photo event:

```
camCap.TakePhotoCompleted += onTakePhotoCompleted;
void onTakePhotoCompleted(string fpath)
{
        // todo something by fpath , you can use the path to load it and preview it
in ui or other.   see the demo sample scene.,
}
```

(2) record video

1) Get **CameraCapture** reference Object

```
CameraCapture camCap= GameObject.FindObjectOfType<CameraCapture> ();
```

2) Record video (it will be saved in gallery)

// the video will be save in the gallery if it success

2.1) *camCap.captureVideo();*

2.2) Record video event:

```
camCap.CaptureVideoCompleted += onCaptureVideoCompleted;
void onCaptureVideoCompleted(string fpath)
  {
        // todo something by fpath , you can use this video path to do what you want,like upload it to webservers , or other .see the demo sample scene.
  }
```

(3) Pick Video and Play

1) Get **CameraCapture** reference Object

CameraCapture camCap= GameObject.FindObjectOfType<CameraCapture> ();

2) pick video to play:

// it will open the video files folders make you can pick the video,and it will play auto

camCap.playVideo();

3) Pick Event:

```
camCap.PickCompleted += OnPickCompleted;
void OnPickCompleted(string path)
{
       // todo something by path, see the demo sample scene.
}
```

(4) Pick Video

1) Get **CameraCapture** reference Object

```
CameraCapture camCap= GameObject.FindObjectOfType<CameraCapture> ();
```

2) pick video:

// it will open the video files folders make you can pick the video,it will return the path;

```
camCap.pickVideo();
```

3) Pick Event:

```
camCap.PickCompleted += OnPickCompleted;
void OnPickCompleted(string path)
{
       // todo something by path, see the demo sample scene.
}
```

(5) Pick Photo

1) Get **CameraCapture** reference Object

```
CameraCapture camCap= GameObject.FindObjectOfType<CameraCapture> ();
```

2) pick video:

// it will open the image files folders make you can pick the images ,it will return the path;

```
camCap.pickPhoto();
```

3) Pick Event:

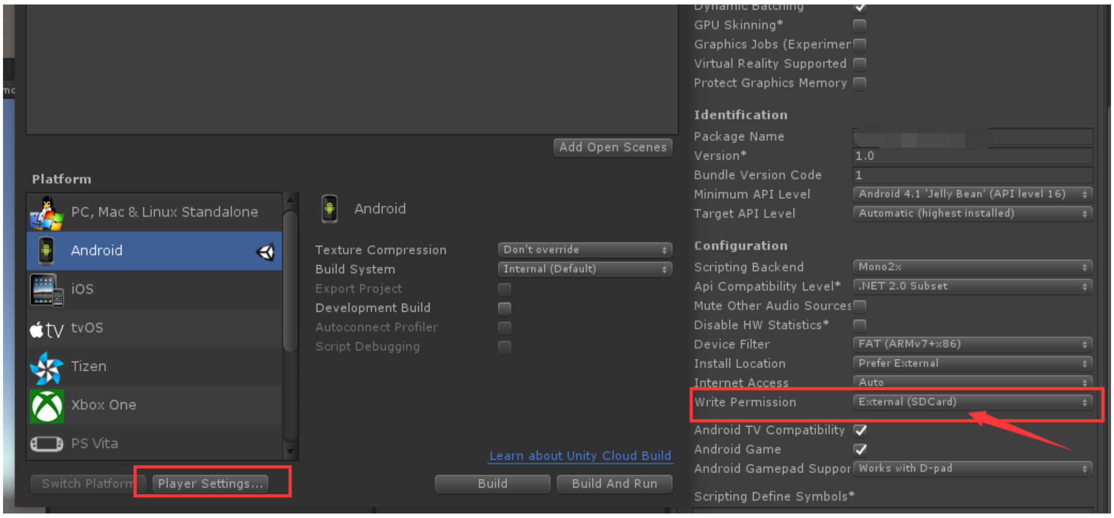
```
camCap.PickCompleted += OnPickCompleted;
void OnPickCompleted(string path)
{
    // todo something by path, see the demo sample scene.
}
```

Step4:

build app

Android:

1，Please set the “**Write Permission**” as “**External(SDCard)**” in the **player setting**.because the plugin need to access the SDCard when taking photo to gallery. If not ,it can’t save image to gallery success



Android

Texture Compression: Don't override
Build System: Internal (Default)
Export Project □
Development Build □
Autoconnect Profiler □
Script Debugging □

Dynamic Batching ☑
GPU Skinning* □
Graphics Jobs (Experimer □
Virtual Reality Supported □
Protect Graphics Memory □

Identification
Package Name: [redacted]
Version*: 1.0
Bundle Version Code: 1
Minimum API Level: Android 4.1 'Jelly Bean' (API level 16)
Target API Level: Automatic (highest installed)

Configuration
Scripting Backend: Mono2x
Api Compatibility Level*: .NET 2.0 Subset
Mute Other Audio Sources □
Disable HW Statistics* □
Device Filter: FAT (ARMv7+x86)
Install Location: Prefer External
Internet Access: Auto
Write Permission: External (SDCard)
Android TV Compatibility ☑
Android Game ☑
Android Gamepad Suppor: Works with D-pad
Scripting Define Symbols*

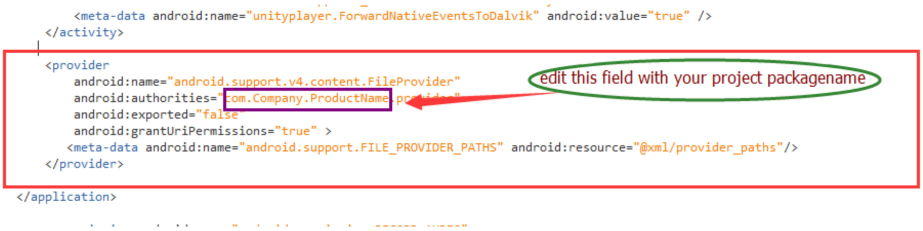
2，Edit the AndroidManifest.xml file (important)

1) Open the AndroidManifest.xml in the Assets/Plugins/Android/ folder. And edit it.

2) Locate at the node <provider></provider>

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

3) Edit the “**com.Company.ProductName**” with **your project packagename.**



```
        <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="true" />
    </activity>

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

Ok, we have provide some sample scenes for you ,you can see them for detail.

For support email me at wiliamheart@gmail.com