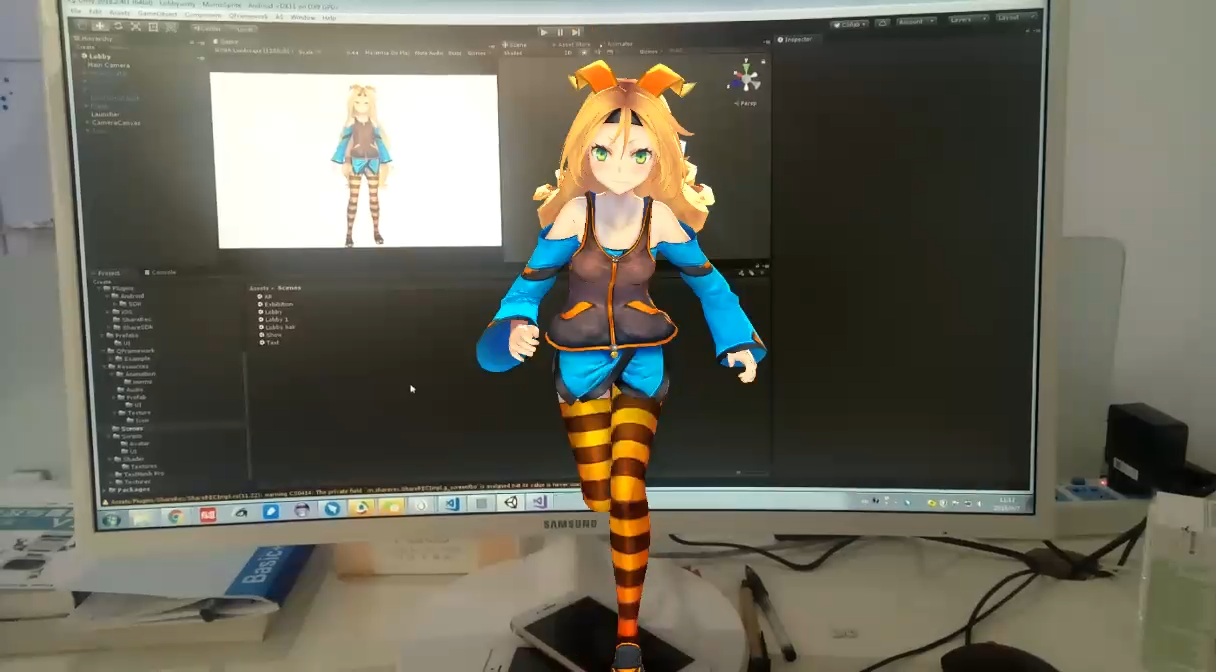
# **AR中的现实背景效果实现**

作者：[ArtStealer](http://gad.qq.com/user/index?id=86976)

链接：<http://gad.qq.com/article/detail/286857>

分享一个简单的功能，AR中都用了摄像机图像作为游戏的背景，今天就来分享一下如何在Unity中实现这个功能。

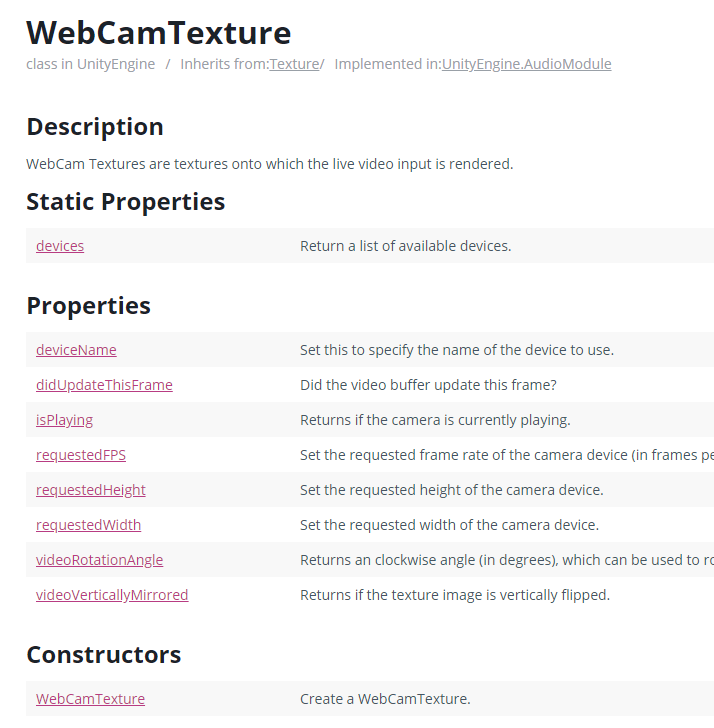
**先上图看看实现效果：**  


好了，看完图一目了然，应该知道是本文实现啥效果了，下面介绍如何实现：

## **基本原理：**

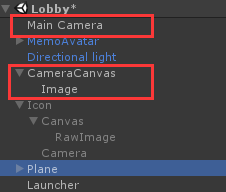
## 用一张image做为整个游戏的背景。然后调用摄像头，每帧将摄像头采集的数据渲染到这张image中。

## Unity中提供了的 [WebCamTexture](https://docs.unity3d.com/ScriptReference/WebCamTexture.html)接口，用来播放视频输入。



好了基础知识点结束完毕，下面看一下具体实现。

Hierarchy中红框：一个 Camera（Main Camera）, 一个Canvas(CameraCanvas), 一个Image；



其中CameraCanvas的属性如下：



至于Image，需要把他设为铺满全屏，具体截图就不展示了。

下面看脚本实现 CameraBGController.cs

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class CameraBGController : MonoBehaviour

｛

WebCamTexture camTexture;

CanvasScaler canvasScaler;

Image image;

void Start()

｛

image = GetComponentInChildren<Image>();

canvasScaler = GetComponentInChildren<CanvasScaler>();

canvasScaler.referenceResolution = new Vector2(Screen.width, Screen.height);

StartCoroutine(CallCamera());

｝

IEnumerator CallCamera()

｛

yield return Application.RequestUserAuthorization(UserAuthorization.WebCam);

if (Application.HasUserAuthorization(UserAuthorization.WebCam))

｛

if (camTexture != null)

camTexture.Stop();

WebCamDevice[] cameraDevices = WebCamTexture.devices;

string deviceName = cameraDevices[0].name;

camTexture = new WebCamTexture(deviceName, Screen.width, Screen.height, 60);

image.canvasRenderer.SetTexture(camTexture);

camTexture.Play();

｝

｝

｝

30行代码搞定。

**剩下就是见证奇迹的时刻了：**

再来个视频：

欢迎关注公众号(ArtStealer)进行技术交流探讨

