```
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3  using UnityEngine.UI;
4
5  public class MobileCamera : MonoBehaviour
6  {
7      #region 属性
8      bool bBackCamAvailable;      // 是否存在相机的标识
9      WebCamTexture wctCamera;     // 相机纹理对象
10     public RawImage riFrame;     // 显示图像
11     [SerializeField] AspectRatioFitter arFitter;    // 尺寸适配对象
12     #endregion
13
14     /// <summary>
15     /// 初始化函数
16     /// </summary>
17     private void Start()
18     {
19         bBackCamAvailable = false;
20
21         #region 判断是否存在摄像头
22         WebCamDevice[] devices = WebCamTexture.devices;
23         int nNumCam = devices.Length;
24         Debug.Log("L->:一共有 "+nNumCam.ToString() + " 个相机.");
25         if (nNumCam <= 0)
26         {
27             Debug.Log("L->:没有可用的摄像头.");
28             return;
29         }
30         #endregion
31
32         #region 调用后置摄像头
33         for (int i=0; i< nNumCam; ++i)
34         {
35             if(!devices[i].isFrontFacing)
36             {
37                 wctCamera = new WebCamTexture(devices[i].name, Screen.width,
                      Screen.height);
38                 wctCamera.Play();
39                 riFrame.texture = wctCamera;
40                 bBackCamAvailable = true;
41                 break;
42             }
43         }
44         #endregion
45     }
46
47     /// <summary>
48     /// 每帧调用函数
49     /// </summary>
50     private void Update()
51     {
52         if (bBackCamAvailable == false)
53             return;
54
55         float ratio = (float)wctCamera.width / (float)wctCamera.height;
```

```
56          arFitter.aspectRatio = ratio;
57          Debug.Log("L->aspectRatio: " + ratio.ToString());
58
59          // 上下颠倒
60          float scaleY = wctCamera.videoVerticallyMirrored ? -1f : 1f;
61          riFrame.rectTransform.localScale = new Vector3(1f, scaleY, 1f);
62
63          // 旋转
64          int orient = -wctCamera.videoRotationAngle;
65          riFrame.rectTransform.localEulerAngles = new Vector3(0, 0, orient);
66      }
67
68      /// <summary>
69      /// 回调函数: 退出程序
70      /// </summary>
71      public void onQuitApp()
72      {
73          Application.Quit();
74      }
75
76      /// <summary>
77      /// 回调函数: 切换场景
78      /// </summary>
79      public void onExt()
80      {
81          SceneManager.LoadScene(1);
82      }
83  }
84
```