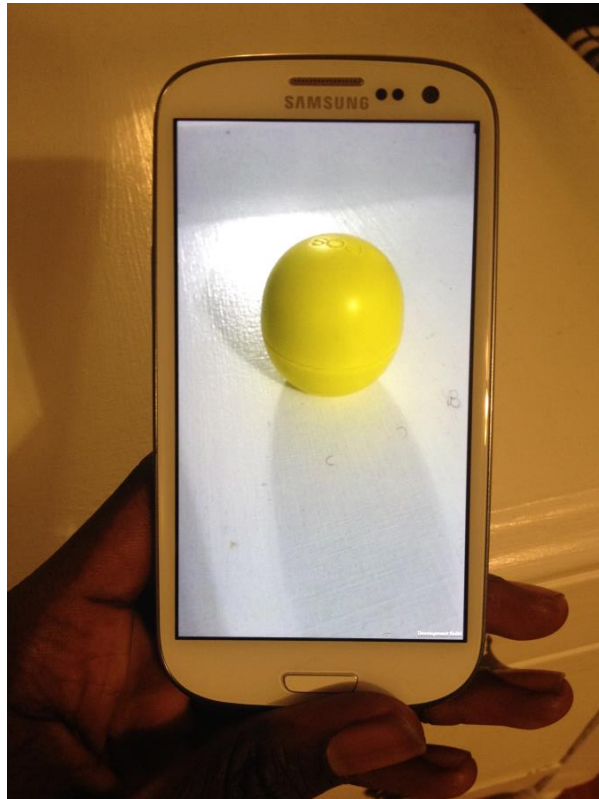


Welcome to NatCam

NatCam is an API that provides a unified and extended interface for developers to control device cameras. Built specifically for iOS and Android, but compatible with all other platforms, NatCam provides you access to intrinsic functions of the camera like focusing, zooming, machine-readable code detection, and so much more, all while providing a high resolution, blazing fast camera preview.

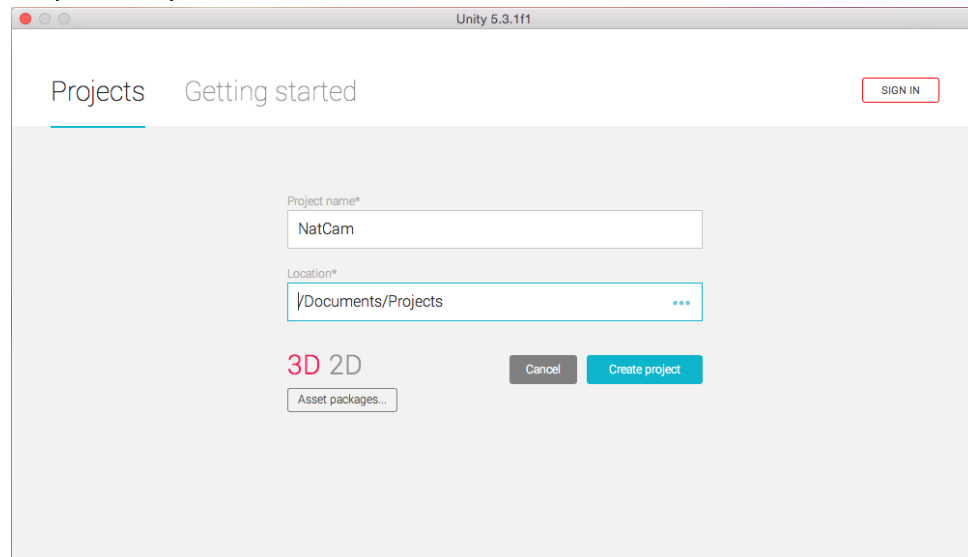
We started building NatCam when we realized how unsatisfactory the WebCamTexture API was especially with resolution and frame rates, coupled with the fact that we needed bar code detection for our app. NatCam has developed greatly since then, but to us, it's still our beta baby.

Crash Course

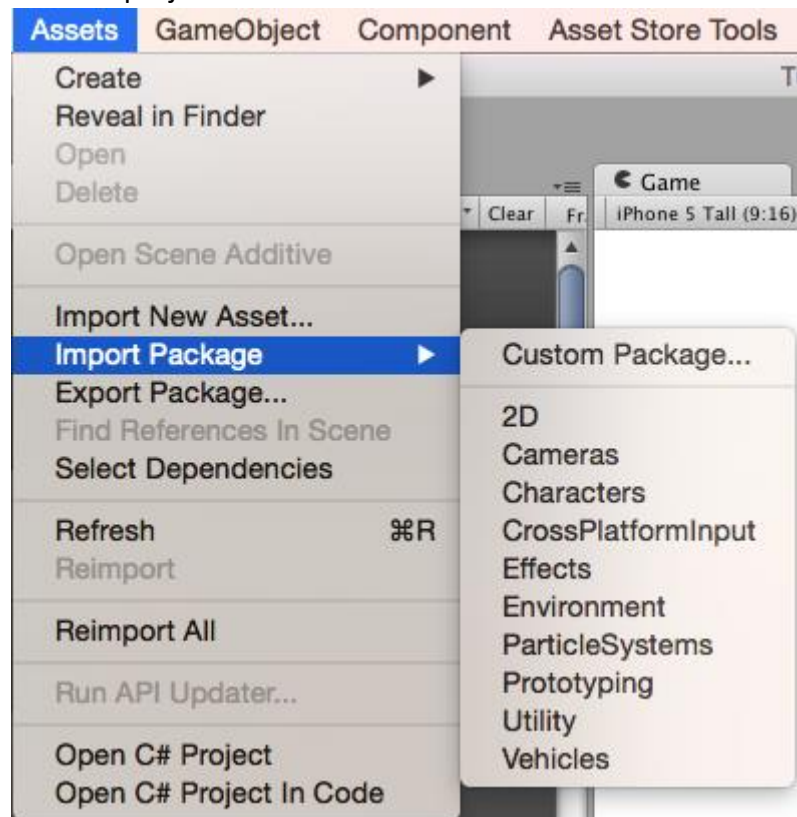


Let us create a quick camera preview app called Unitygram with zooming and focusing ability.

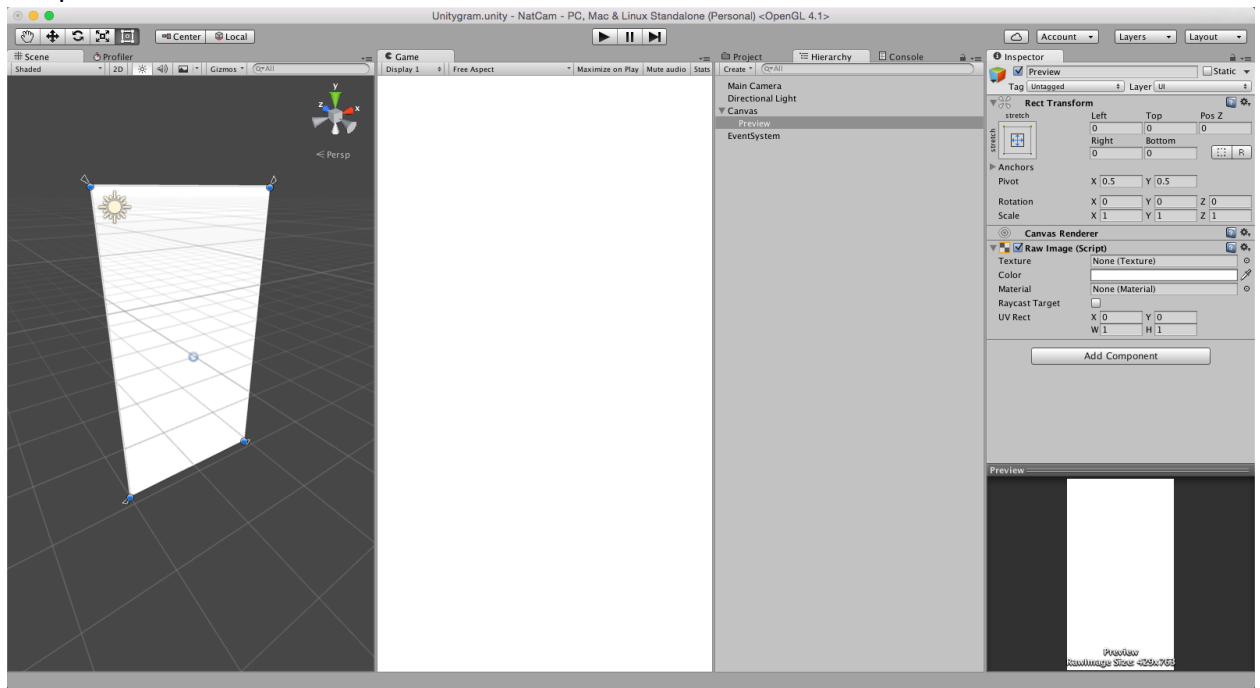
1. First, we open Unity.



2. Import NatCam to the project.



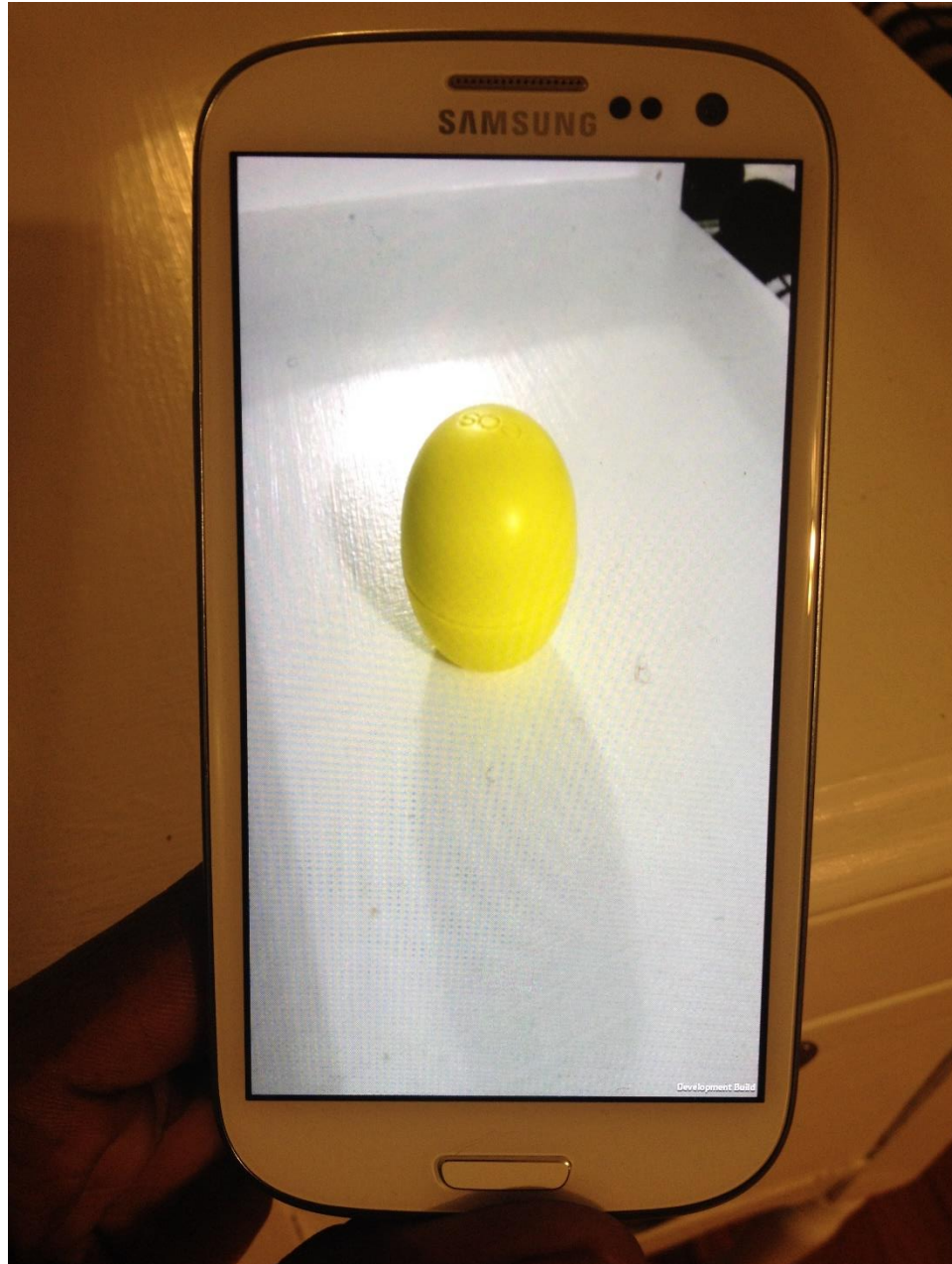
- Now we create a fullscreen UI Panel with a RawImage component so that we can project the preview onto it.



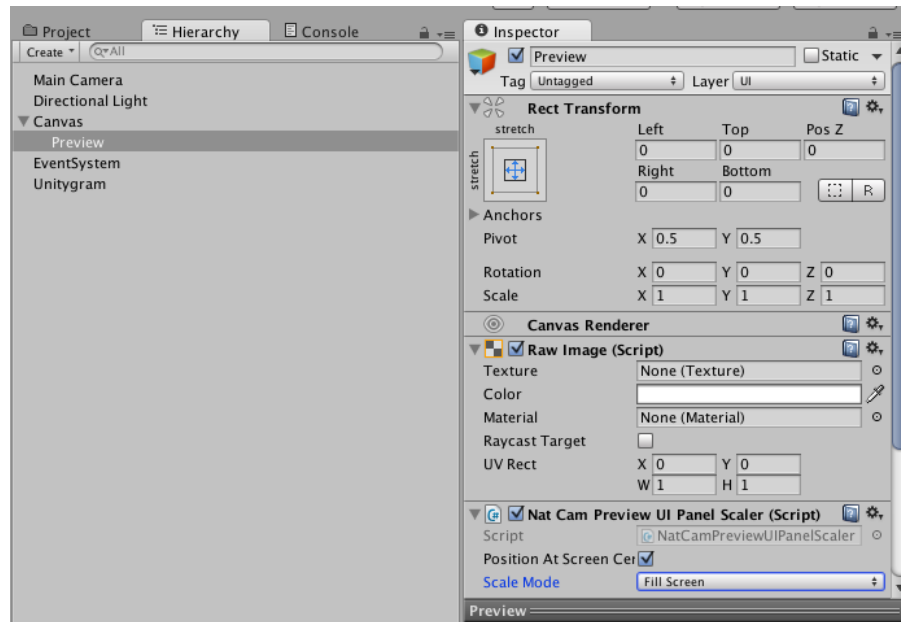
- Let us create a script that will manage Unitygram. We create a script called 'Unitygram.cs' and add NatCam initialization code to the Start method. First, we call NatCam Initialize, then we set the rear camera to be the active camera. Next, we set the resolution of the camera to Full HD (1080p) then call NatCam Play, and set the texture of our UIPanel RawImage to the NatCam Preview texture.

```
Unitygram.cs Assets/NatCam/Example
1
2 0 references
3 public class Unitygram : MonoBehaviour {
4
5     1 reference
6     public RawImage previewUIPanelRawImage;
7
8     // Use this for initialization
9     0 references
10    void Start () {
11        //Initialize NatCam
12        NatCam.Initialize(PreviewType.NonReadable, PhotoSaveMode.SaveToPhotos);
13        //Set the rear camera active
14        NatCam.SetActiveCamera(DeviceCamera.RearCamera);
15        //Set the resolution to Full HD, 1080p
16        NatCam.SetResolution(ResolutionPreset.HD);
17        //Start the camera preview
18        NatCam.Play();
19        //Set the RawImage's texture to the camera preview
20        //We use PreviewTexture instead of Preview to ensure compatibility across all platforms, supported and unsupported likewise
21        previewUIPanelRawImage.texture = NatCam.PreviewTexture;
22    }
23
24    // Update is called once per frame
25    0 references
26    void Update () {
27    }
28 }
```

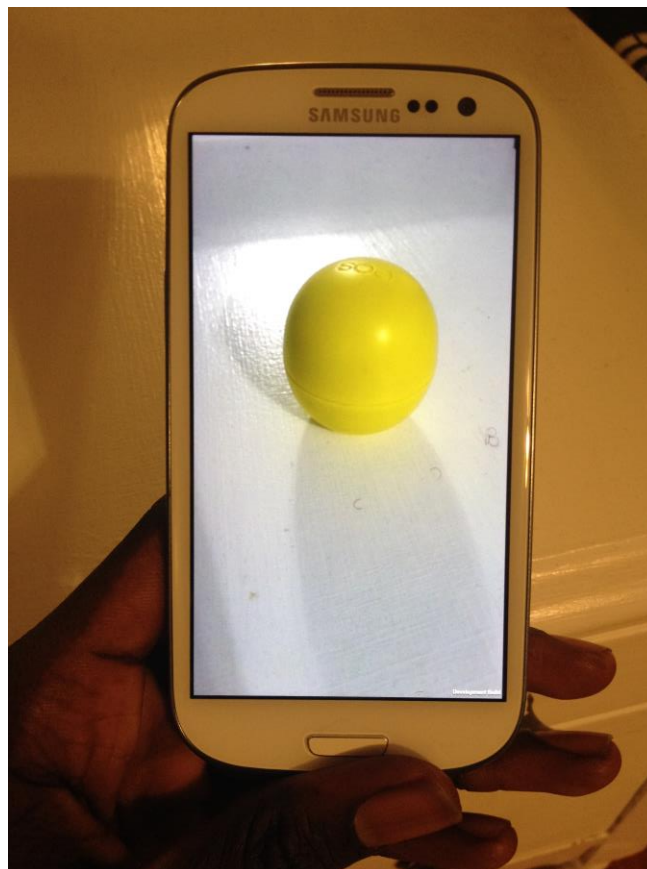
5. Build and run.



6. Notice that the camera preview is stretched. This is because the camera's aspect ratio is different from the screen's aspect ratio. Thankfully, we can fix this easily by adding a [NatCamPreviewUIPanelScaler] component. We will set the ScaleMode to ScaleMode.FillScreen because we want to fill the screen.

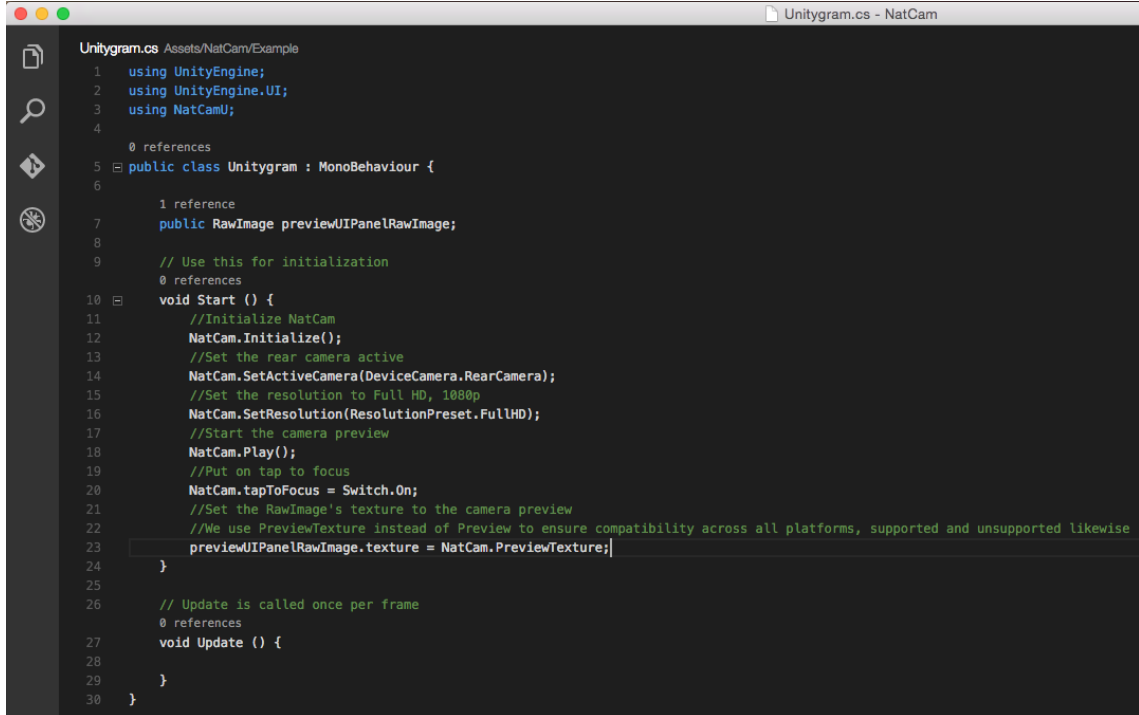


7. Build and run again.



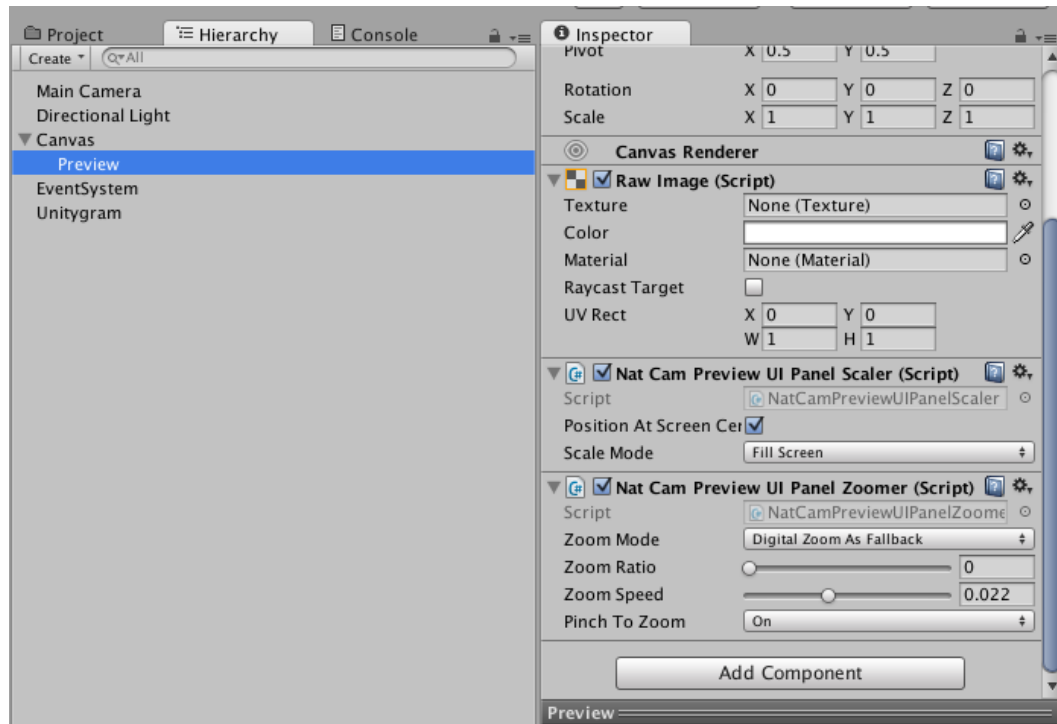
8. Now we want to implement tap to focus. This is as easy as adding one line. In our start method, after NatCam.Play, we will enable tap to focus by adding:

NatCam.tapToFocus = Switch.On.

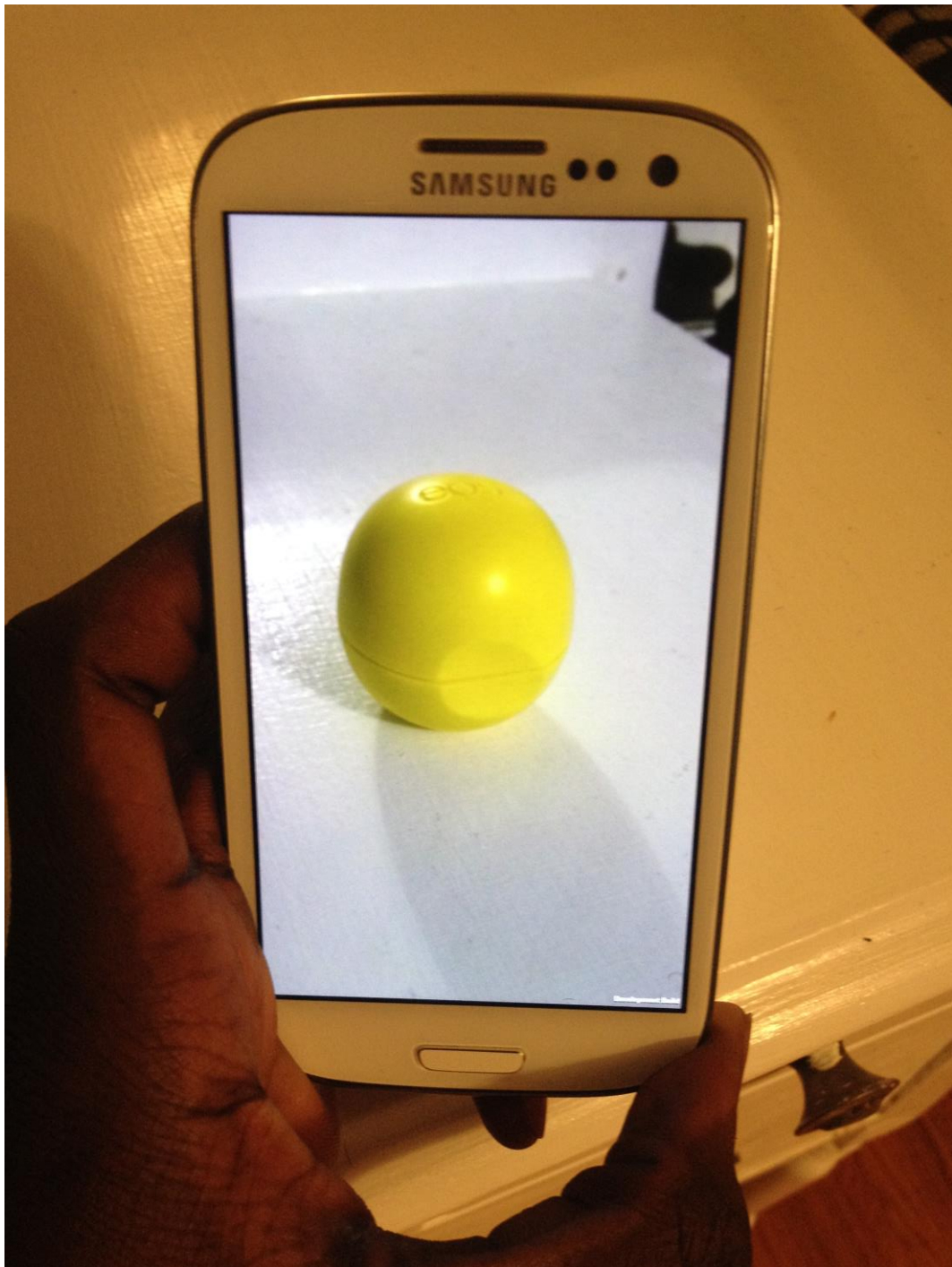


```
Unitygram.cs Assets/NatCam/Example
1  using UnityEngine;
2  using UnityEngine.UI;
3  using NatCamUI;
4
5  0 references
6  public class Unitygram : MonoBehaviour {
7      1 reference
8      public RawImage previewUIPanelRawImage;
9      // Use this for initialization
10     0 references
11     void Start () {
12         //Initialize NatCam
13         NatCam.Initialize();
14         //Set the rear camera active
15         NatCam.SetActiveCamera(DeviceCamera.RearCamera);
16         //Set the resolution to Full HD, 1080p
17         NatCam.SetResolution(ResolutionPreset.FullHD);
18         //Start the camera preview
19         NatCam.Play();
20         //Put on tap to focus
21         NatCam.tapToFocus = Switch.On;
22         //Set the RawImage's texture to the camera preview
23         //We use PreviewTexture instead of Preview to ensure compatibility across all platforms, supported and unsupported likewise
24         previewUIPanelRawImage.texture = NatCam.PreviewTexture;
25     }
26     // Update is called once per frame
27     0 references
28     void Update () {
29     }
30 }
```

9. Let us now add zooming. We want to zoom the preview when the user pinches their screen. Thankfully, NatCam has a `NatCamPreviewUIPanelZoomer` component which we will add to our UI Panel. We want to be able to zoom on all devices, including those which don't support hardware optical zoom. Thus, we will use `ZoomMode DigitalZoomAsFallback` which will fallback to shader-accelerated digital zoom on devices which don't support optical zoom. We will also put on the `pinchToZoom` switch.



10. Let us build Unitygram and run it.



Make sure you visit the scripting reference to learn more about using NatCam.