

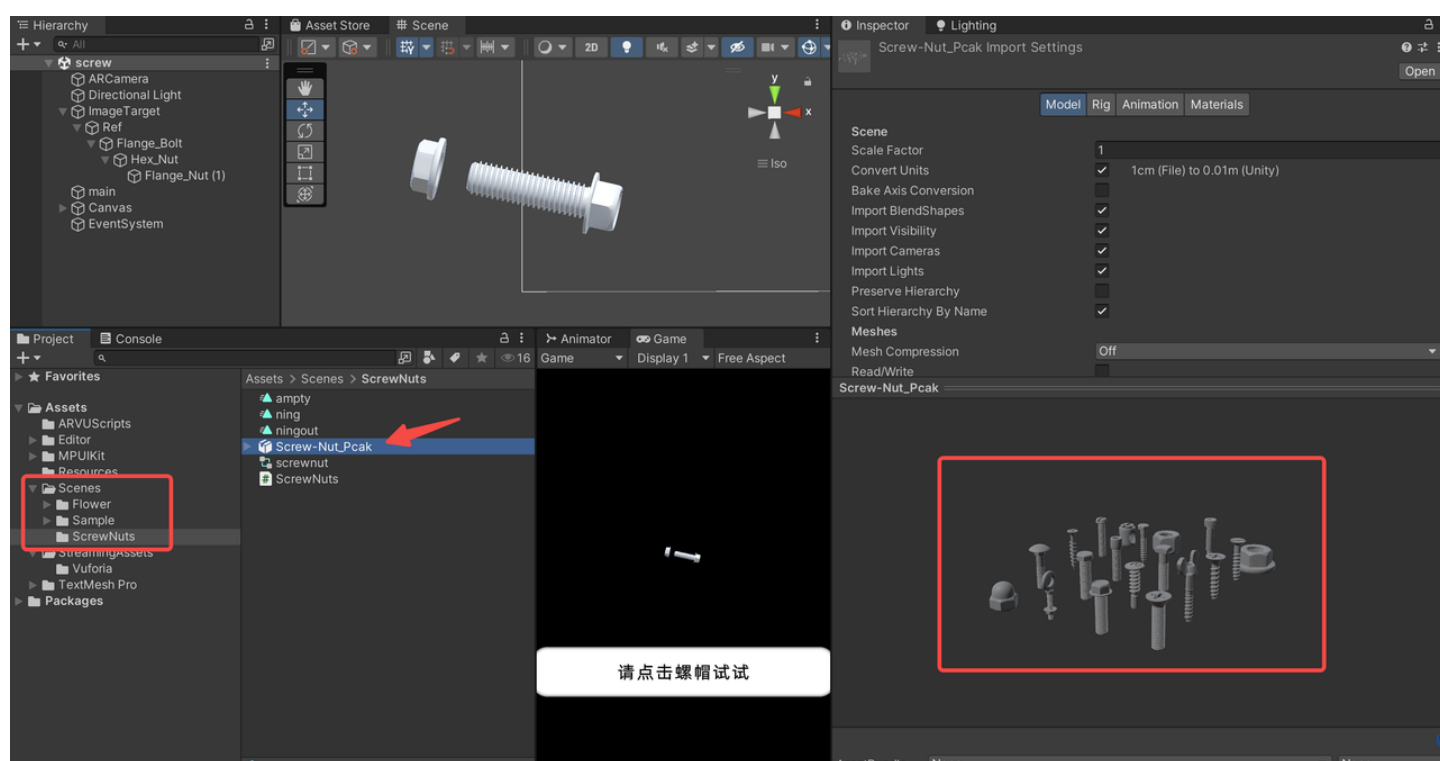
螺丝螺帽AR技术与交互开发文档

1. 总体需求

基于Vuforia图片识别跟踪，实现螺丝螺帽的AR交互，其中当识别成功后，出现模型；丢失后，模型消失。当模型出现后，点击螺帽，触发拧进和拧出动画。

2. 模型

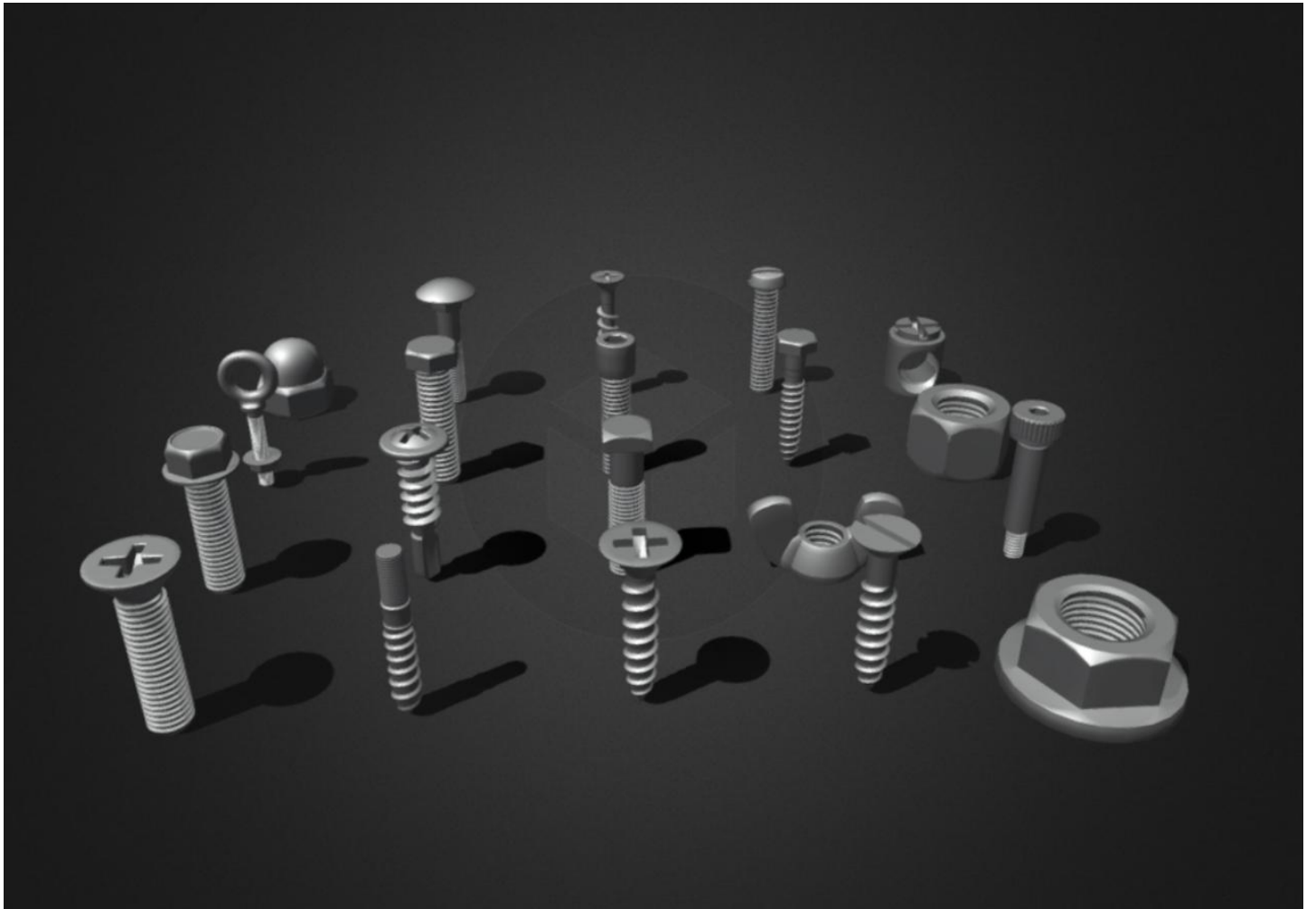
公开网站上下载所得，见下图。



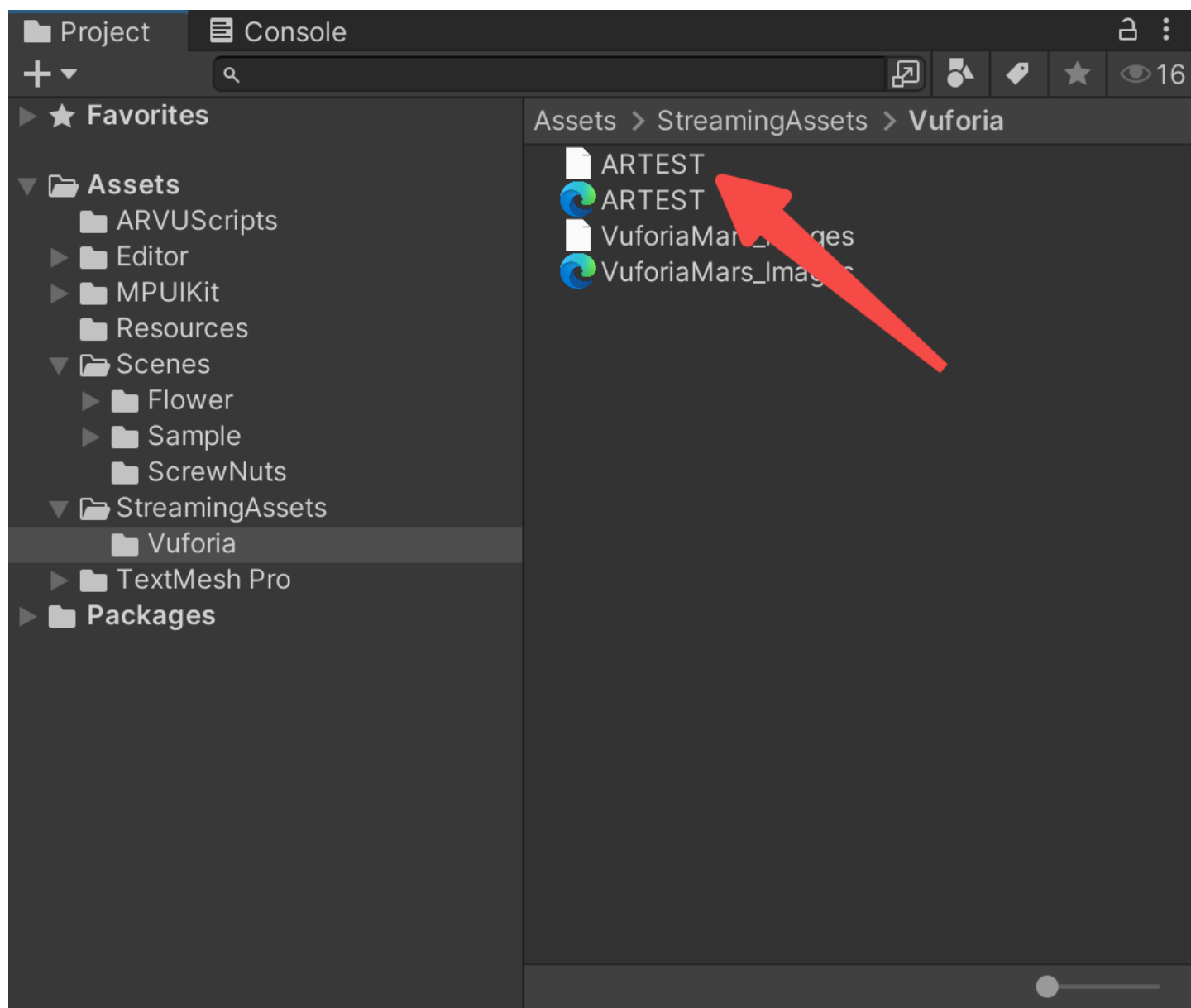
3. 技术路线

3.1 AR识别图配置

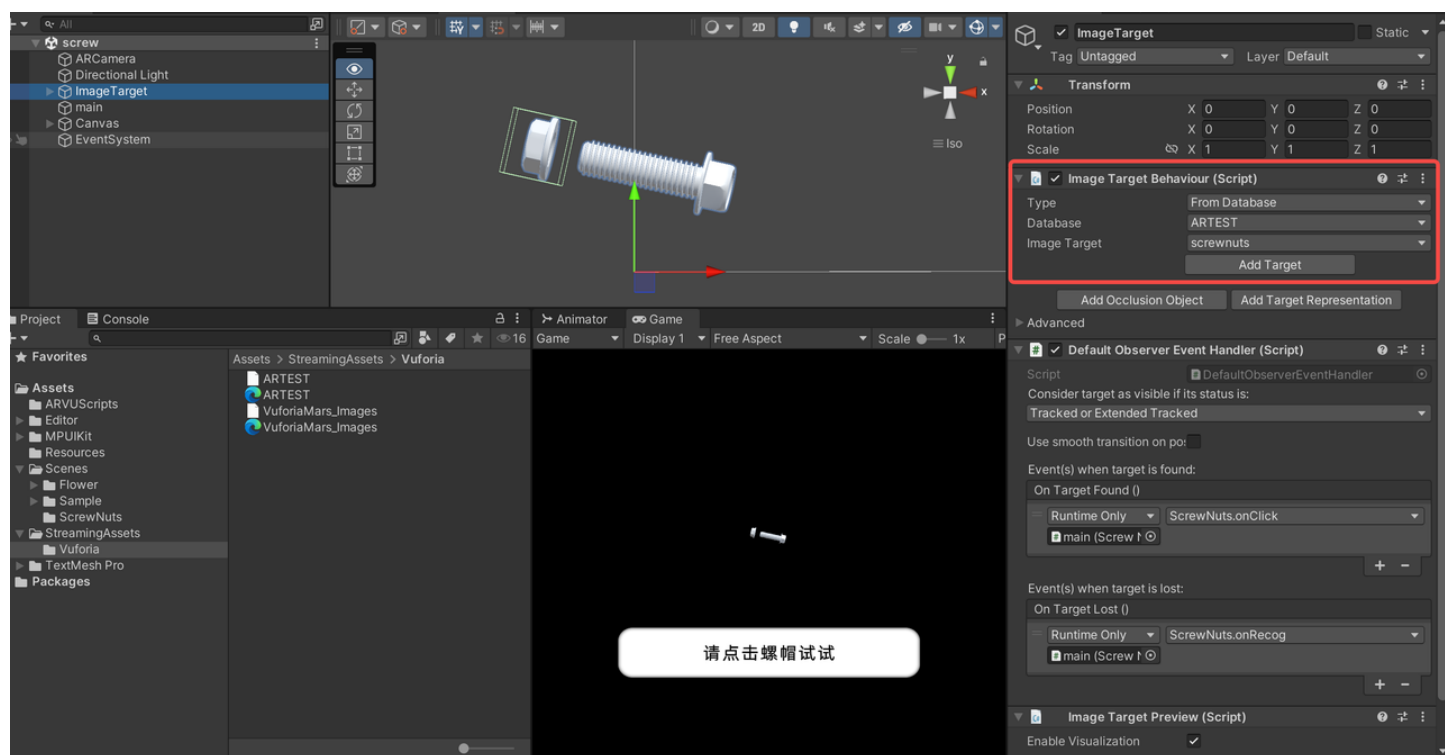
Vuforia图片AR识别和跟踪技术，在其后台配置识别图，成功后并将图片资产导入Unity中。如何配置识别图另见文档。



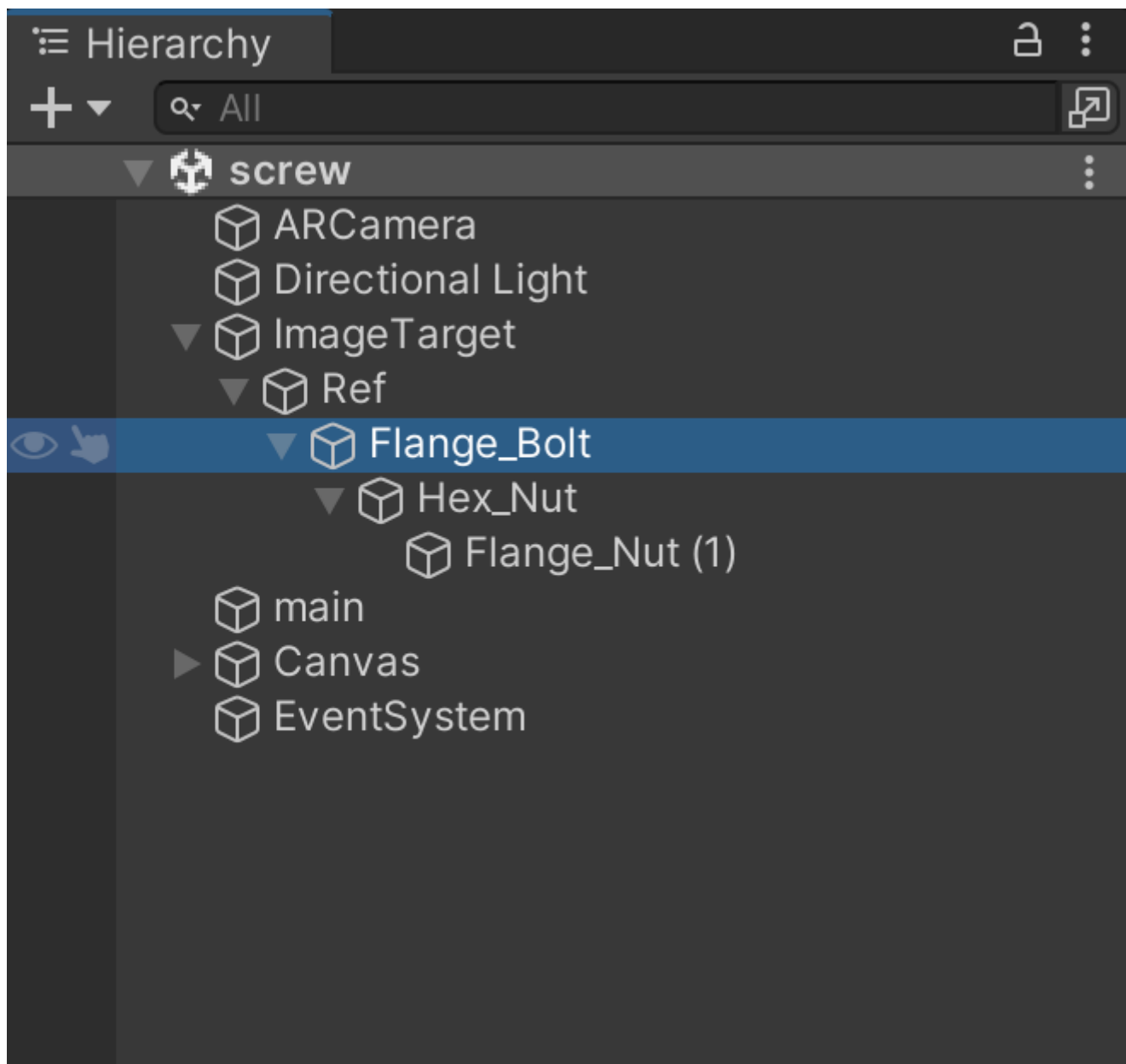
导入的图片资产见下图；



配置识别资产，在场景中创建ImageTarget，在ImageTargetBehaviour中配置该识别图，见下图。

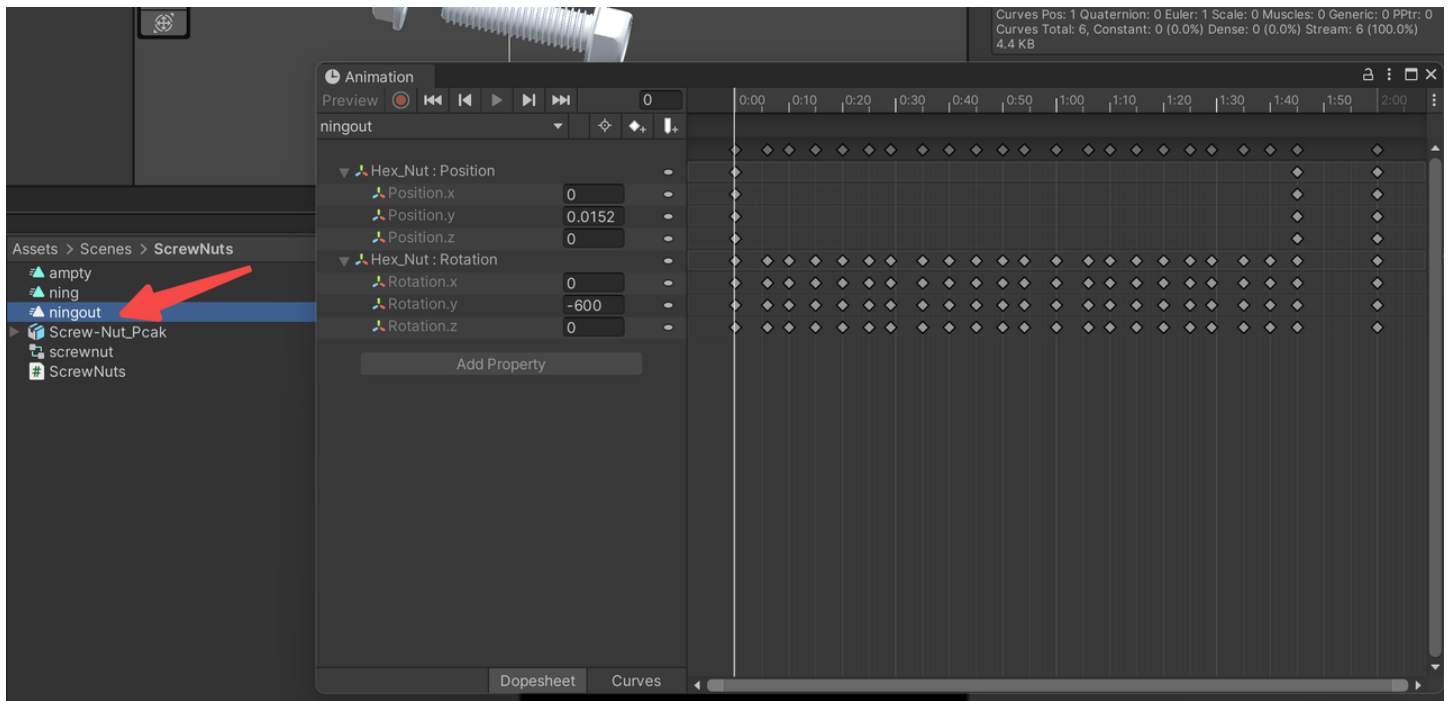


配置完识别资产后，将虚拟模型如下图设置再ImageTarget层级下。

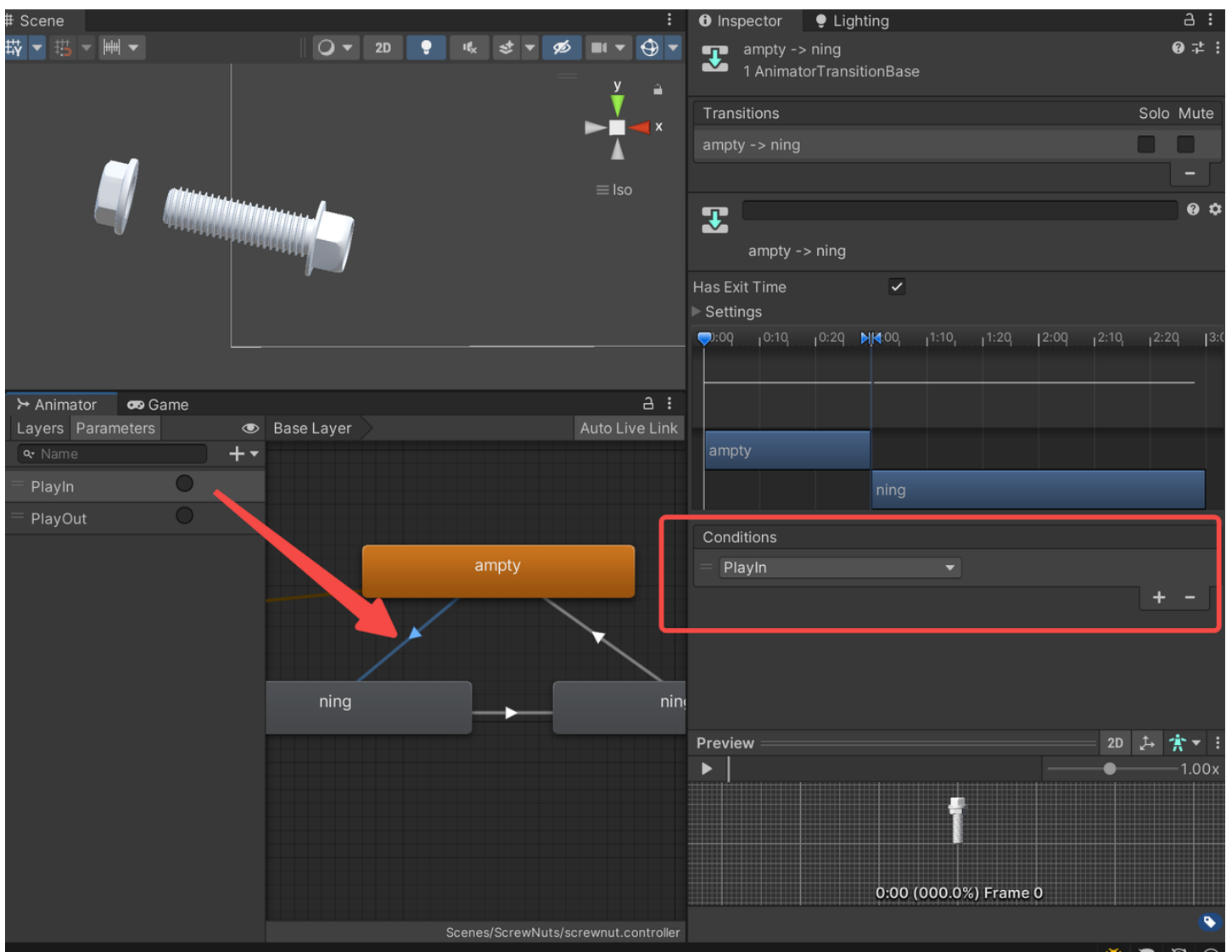


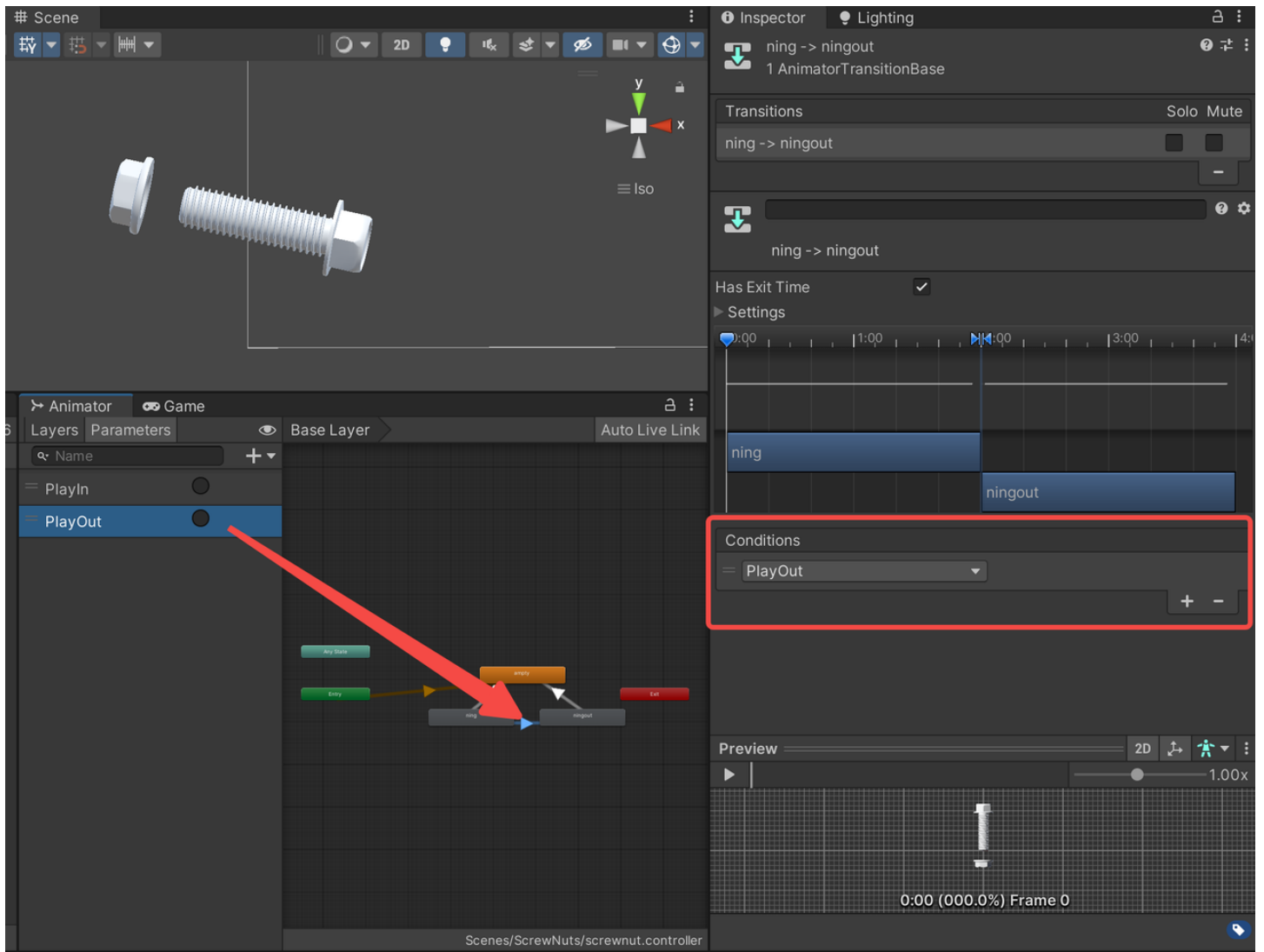
3.2 制作拧进和拧出动画

动画状态机Animator见下图配置。



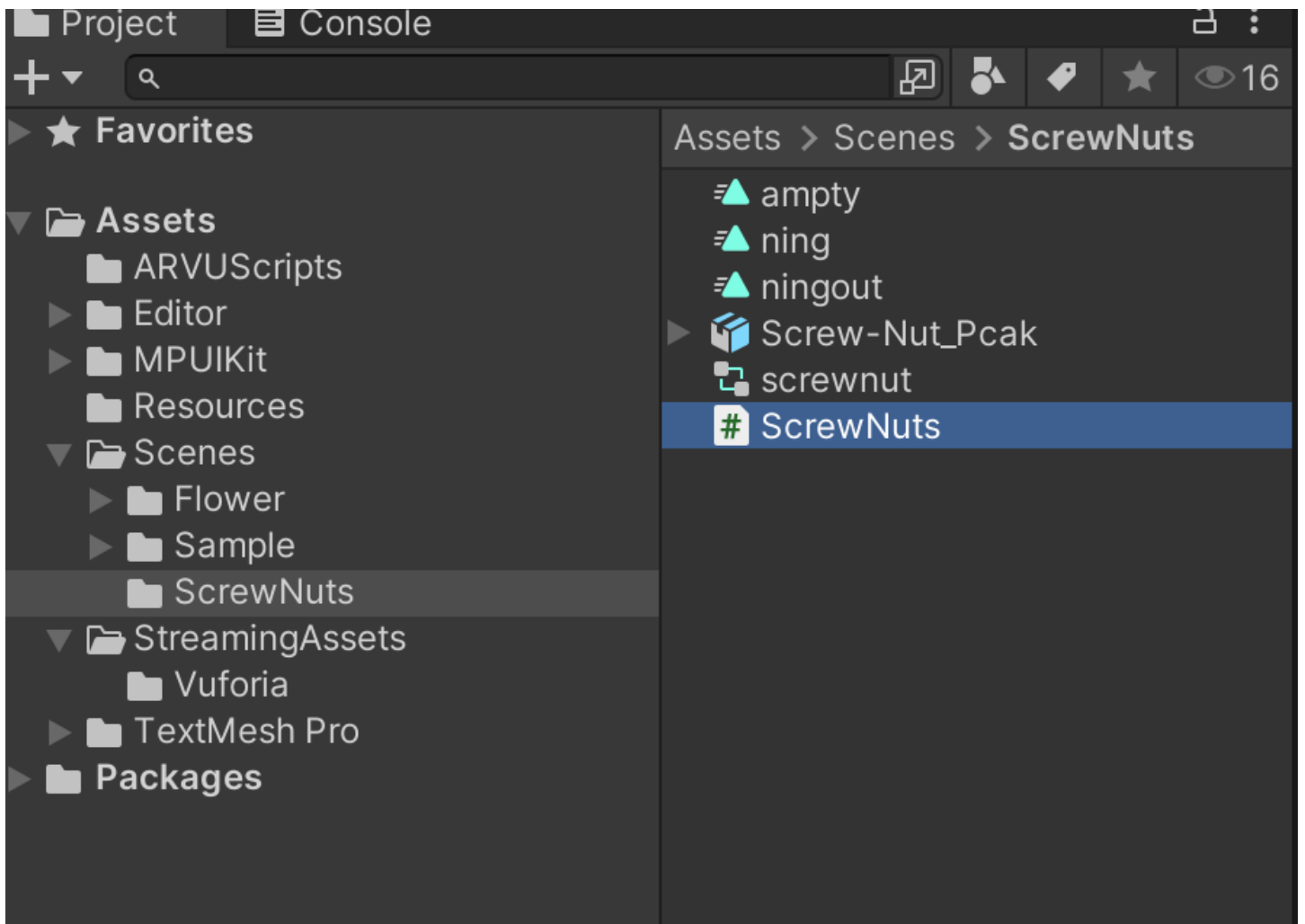
其中三个动画状态设置播放逻辑：默认模型的初始状态是ampty动画；设置PlayIn触发参数，当触发该参数可切换拧进动画（ning）；设置PlayOut触发参数，当触发该参数可切换拧出动画（ningout），播放完成后自动切换到初始状态。





3.3 点击交互逻辑

ScrewNuts脚本，实现点击螺帽模型交互。脚本位置见下图



脚本核心逻辑见下图

代码块

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ScrewNuts : MonoBehaviour
6  {
7      public Transform target;
8      /// <summary>
9      /// 动画的时长计数，默认值大于动画时长
10     /// </summary>
11     private float starttimecount = 10;
12     /// <summary>
13     /// 标记动画状态，记录是否是拧进或拧出动画，true是拧出，否则是拧进
14     /// </summary>
15     private bool onPlayBack = false;
16     /// <summary>
17     /// 动画时长，在播放拧进或拧出动画时，不允许点击交互。
18     /// </summary>
19     private float animtime = 2.1f; // 略大于动画时长
```

```

20
21     /// <summary>
22     /// 提示识别
23     /// </summary>
24     public Transform mpImageRecog;
25     /// <summary>
26     /// 提示点击
27     /// </summary>
28     public Transform mpImageClick;
29
30     // Update is called once per frame
31     void Update()
32     {
33         // 动画中计时, 未达到时间不允许交互
34         if (starttimecount < animtime)
35         {
36             starttimecount += Time.deltaTime;
37             return;
38         }
39
40         if (Input.touchCount == 1)
41         {
42             //获取屏幕点击
43             var touch = Input.GetTouch(0);
44             var state = touch.phase;
45             //屏幕点击开始时
46             if (state == TouchPhase.Began)
47             {
48                 //检测是否点中螺帽模型
49                 var ray = Camera.main.ScreenPointToRay(touch.position);
50                 if (Physics.Raycast(ray, out RaycastHit hitInfo, 1000))
51                 {
52                     //判断是否点钟
53                     if (hitInfo.transform.name == target.name)
54                     {
55                         if (starttimecount > animtime)
56                         {
57                             var animator =
hitInfo.transform.parent.GetComponent<Animator>();
58                             //通过触发参数控制动画播放
59                             animator.SetTrigger(onPlayBack?"PlayOut":
"PlayIn");
60                             onPlayBack = !onPlayBack;
61
62                             starttimecount = 0;
63                         }
64                     }

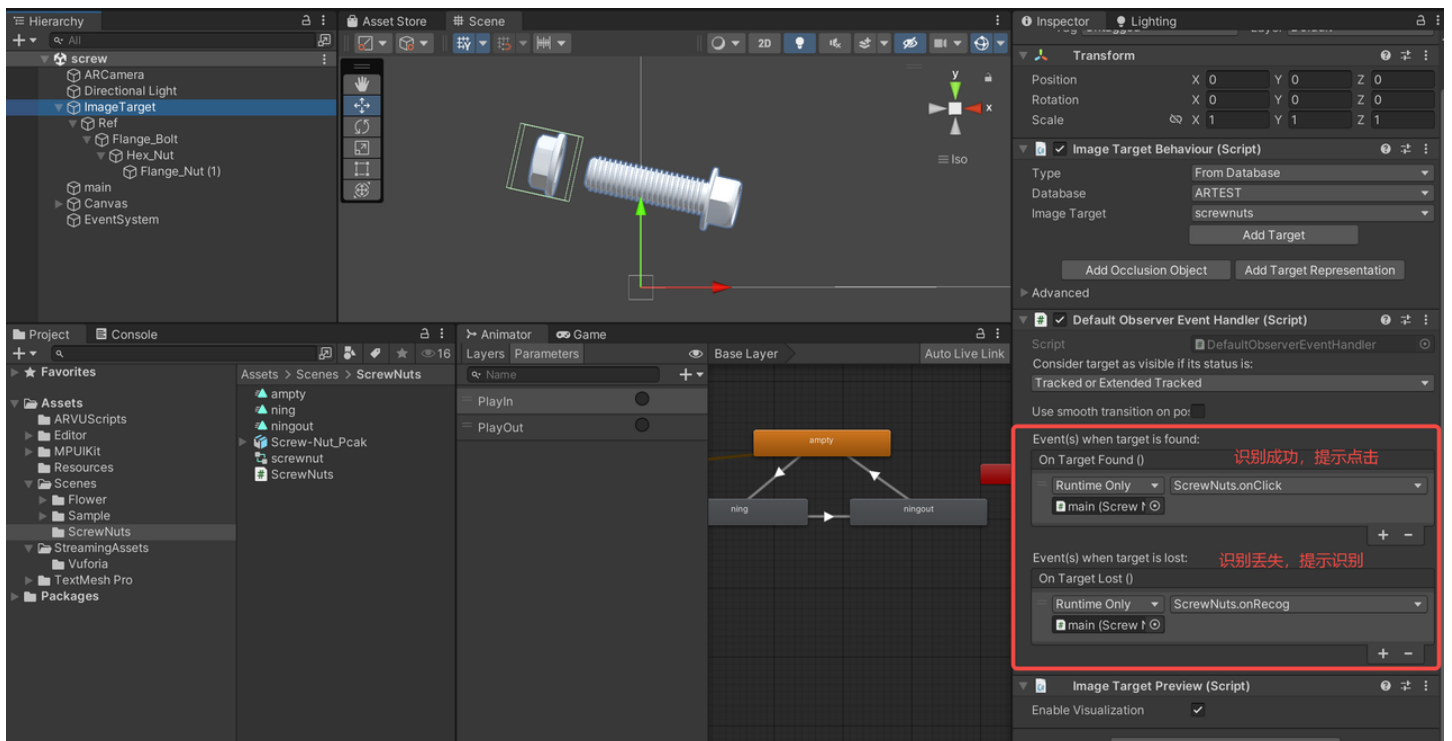
```

```

65         }
66     }
67 }
68 }
69
70
71 /// <summary>
72 /// 提示识别
73 /// </summary>
74 public void onRecog()
75 {
76     mpImageRecog.gameObject.SetActive(true);
77     mpImageClick.gameObject.SetActive(false);
78 }
79 /// <summary>
80 /// 提示点击
81 /// </summary>
82 public void onClick()
83 {
84     mpImageRecog.gameObject.SetActive(false);
85     mpImageClick.gameObject.SetActive(true);
86 }

```

配置提示识别和点击逻辑。



3.4 发布

导出安卓端APK文件进行测试。

4. 交付材料

识别原图1张；

测试APK一个；

unity工程文件一个（unitypackage形式）；

测试视频效果一个；

技术文档一份，介绍技术路线基于交互逻辑；