

小场景说明文档

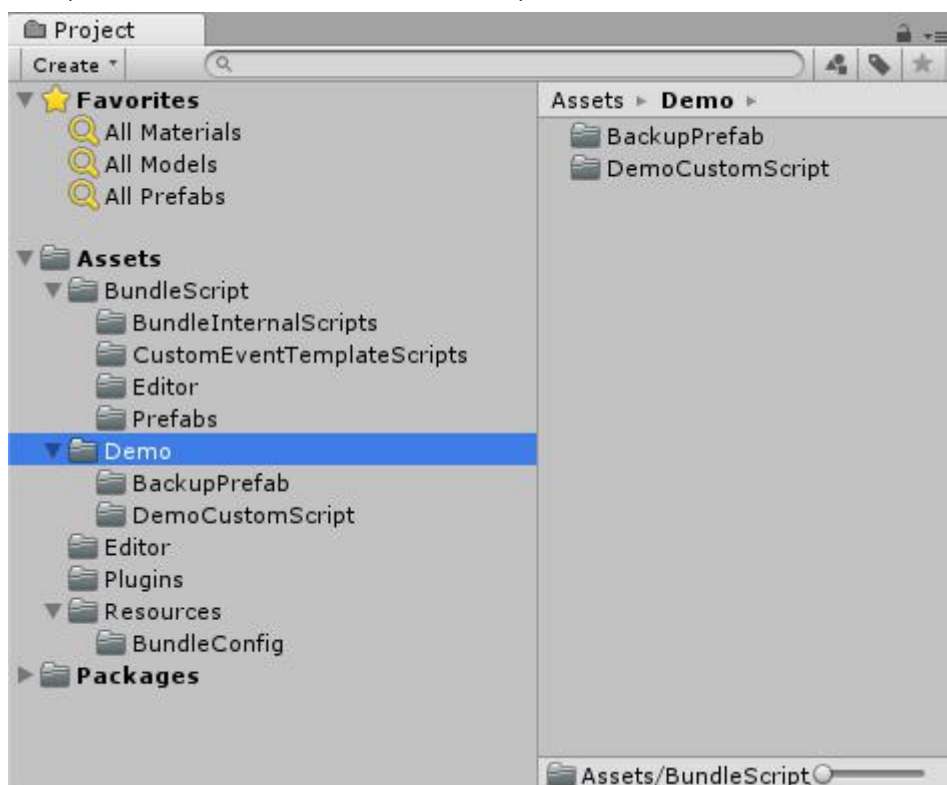
1. 说明

首先我们要求使用者做好一些准备工作以及了解该工具的使用规范，以免造成不必要的麻烦。在 Bundle 文件生成时，由于一些加载和生成配置，我们必须破坏您填入打包的 Prefab，这点您需要特别注意。

2. 文件夹目录及文件准备

如图，文件夹介绍

1. BundleScript 文件夹是 Bundle 工具核心代码，我们不希望您修改他的文件夹位置以及内部结构，CustomEventTemplateScripts 子文件夹下显示自带的模板脚本，一些常用的功能我们会在以后更新中陆续加入
2. Demo 文件夹放置 3 个示例项目，您可以打开查看标准的用法
3. 建议您把准备好的资源整理到一个文件夹下，创建 backup 文件夹以保存 Prefab 源文件(打包过程有可能会破坏选择的 Prefab)



3. 打包配置

此次更新界面也有改动。选择目录禁用粘贴复制，必须通过点击选择保证目录准确性。

- (1) 选择安装路径，需要选择到  Editor 文件夹的上一级(依照

个人安装习惯可能不一样所以需要选择 Editor 上级)。

(2) 脚本打包路径选择栏会在你拖拽 Prefab 后依照物体上的脚本而定, 如果有自定义脚本该栏弹出, 你需要手动选择脚本所在的文件夹。

(3) Bundle 保存路径需要在打包前点击选择, 并且给保存的文件命名(该名称默认为 Unnamed, 当然如果你要进行本地 Bundle 加载它将会是项目名称)

(4) 手动拖拽你将要制作的 Prefab 到需要打包的 Prefab 栏中, 这将是一个很重要的操作, 对于工具我们也会在你拖拽后进行一系列检查, 如物体是否挂有自定义脚本等, 并且在每进行过一次打包后我们会把他清空, 你需要再次拖拽。

(5) 打包平台选择, 一般是默认勾选。

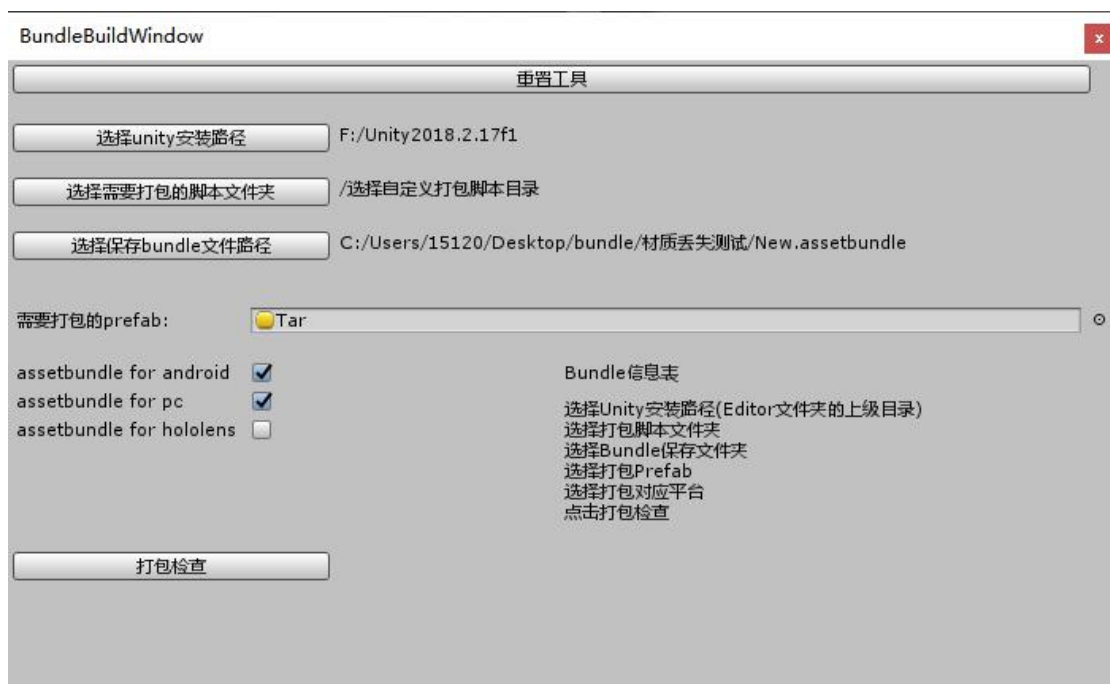
(6) Bundle 信息表, 他会显示当前你可以做的操作或者显示一些提示信息, 当然这些也会在 Unity 的 Console 中提示。

(7) 更新后的工具, 打包前你必须点击打包检查, 检查通过后打包按钮将会显示。如果没用通过检查会弹出一个对话框, 因为在应用里我们需要通过射线来判断一些操作所以需要

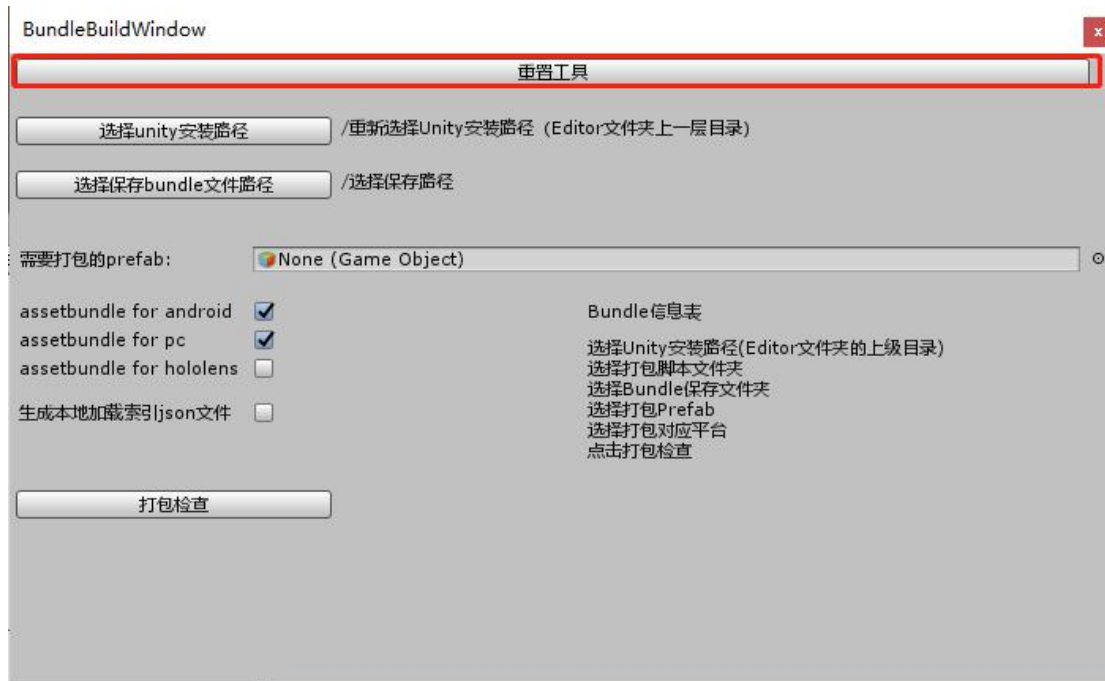


我们创建 Collider, 点击自动创建我们可以帮你快速完成这些操作(需要注意的是这些会自动保存在你选择打包的 Prefab 上), 自行创建需要你自已给 Prefab 创建 Collider, 这可能会很麻烦但是我们面对一些特殊的如粒子特效, Shader, UI 文字等特殊 Mesh 渲染出来的资源时会跟方便, 取消活 x 会忽略, 但也无法进行打包工作。

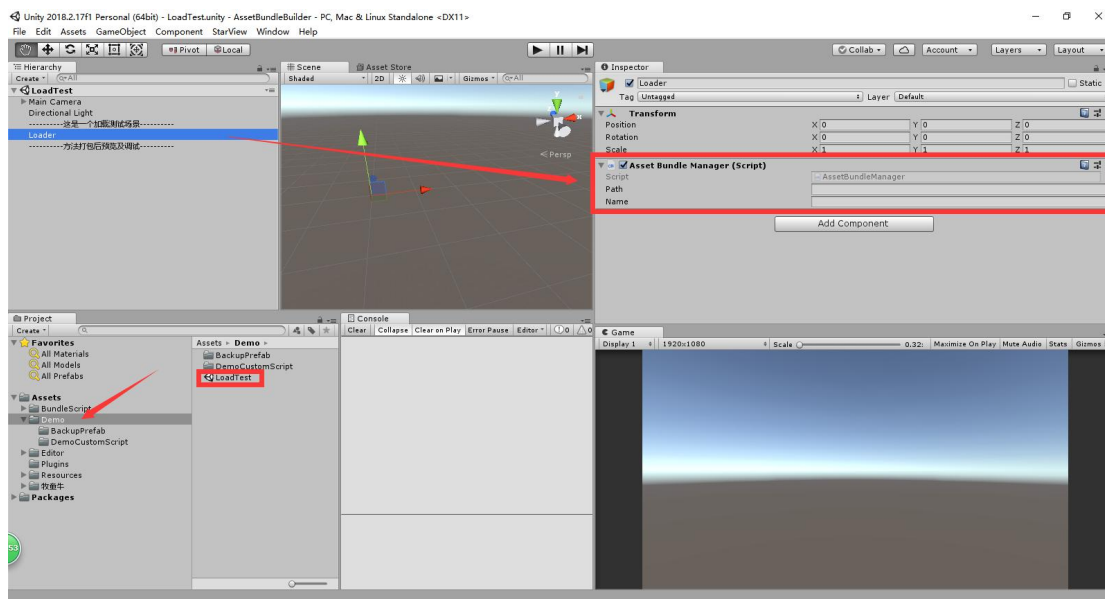
(8) 点击打包, 这可能会需要一些时间, 你只需要等待进度条, 但人性化的一点时完成后我们会帮你打开保存的文件夹。



(10) 新的按钮“重置工具”，该按钮可以在你打包失败，错误提示，无错误提示但在 LoadTest 场景内加载出现触发不正确等非预期时，点击重置工具，方可初始化工具内部数据



(11) LoadTest 加载测试场景，方便在 Unity 中进行打包后的文件进行加载及点击触发测试

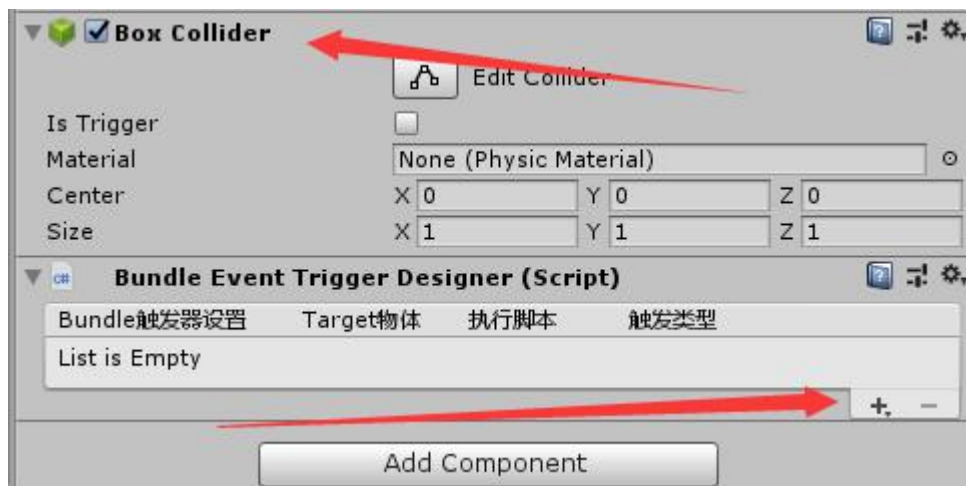


4. 新的组件

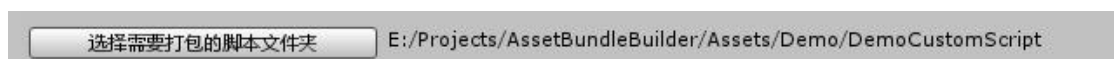
此次更新重构了工具 Bundle 资源的事件触发系统，我们提供类似 EventTrigger 的一个可拖拽选择触发类型的配置事件工具 **BundleEventTriggerDesigner** 设计师组件 AddComponentMenu/Bundle/BundleEventTriggerDesigner 添加，使用注意事项

- (1) 您的 Prefab 的父物体必须是一个空物体，并且上面没有任何其他组件
- (2) 挂载组件的物体，必须是 Prefab 直接子物体，不允许放置在物体下的子物体内
- (3) 挂载组件---->添加事件---->填入被执行物体---->选择触发类型---->保存 Prefab---->打包

- (4) 在您需要设置触发时新建一个空物体，增加 BundleEventTriggerDesigner 组件，组件如图



- (5) 如上图，组件会自动添加 Collider 作为触发响应区域，点击箭头所示 + 号会弹出一个二级菜单，第一栏是自带的模板脚本，第二栏会是您在打包界面选择后的路径下的自定义脚本的选项，选择后及增加一个对应脚本的执行事件



- (6) 如下图配置的意义在鼠标 GazeOn 情况下会在 Cube 物体上执行 SetActive_True 事件



- (7) 您可以点选 '=' 图标对触发事件排序，以及选中后点击 - 号移除
- (8) 点击 '+' 弹出界面的自定义脚本部分显示的脚本是在打包界面选择的文件夹目录下的脚本名称
- (9) 修改 '+' 弹出的二级自定义脚本位置，打开 Bundle 打包界面，拖拽一个有脚本的物体给 Prefab 栏，界面会主动显示“选择需要打包的脚本文件夹”，选择后点击 '+' 弹出的二级菜单即可显示修改后的文件夹下自定义脚本选项

5. 自定义脚本说明

关于自定义脚本我们做了一些约束和限制，您必须遵守。工程内附带一个自定义脚本模板 [SetScale_Toggle](#)，以供参考。

(1) 条件

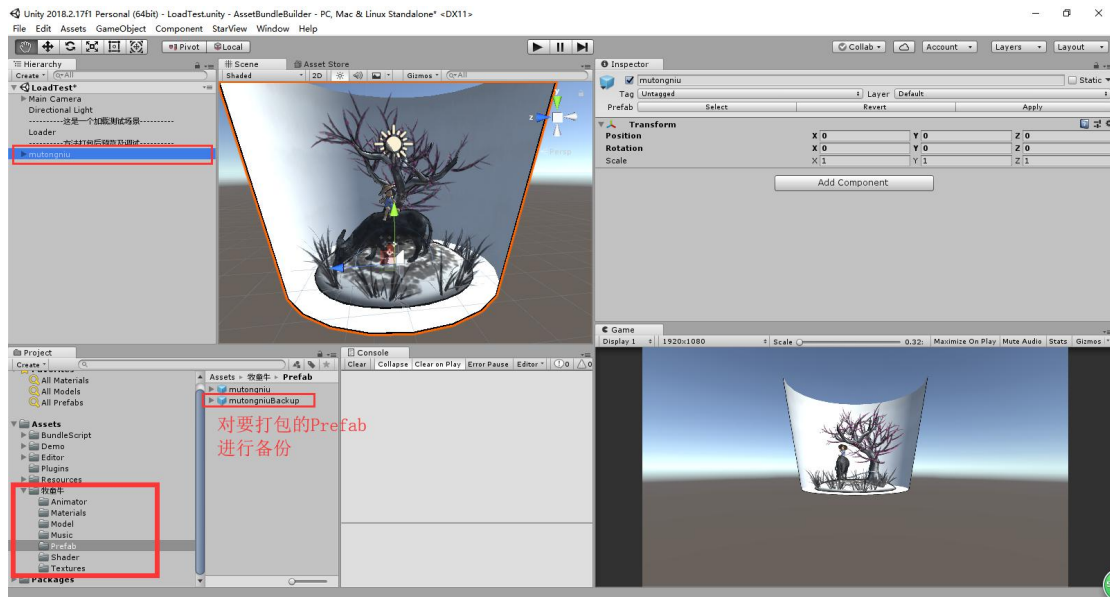
```
/// <summary>
/// 这是一个自定义事件的模板
/// 重复点击以设置物体Scale
/// 自定义脚本必须继承自BundleEventInfoBase类
/// -----必须实现的函数-----
///      OnBundleAction =====> 执行的事件
///      OnReceiveMsg =====> 执行本地事件后其他客户端接收回调，通常和OnBundleAction内一样但不调用SendMsg
///      supportPRS =====> 一个动画，返回值为False即可
/// -----必须在OnBundleAction实现后调用的函数-----
///      SendMsg =====> 执行本地事件后发送消息给其他客户端
/// -----否则-----
/// 1 OnBundleAction中没有调用SendMsg()和传递正确值
/// 2 没有重写OnReceiveMsg函数
/// 会导致多人无法互动
///
/// </summary>
```

(2) 脚本预览

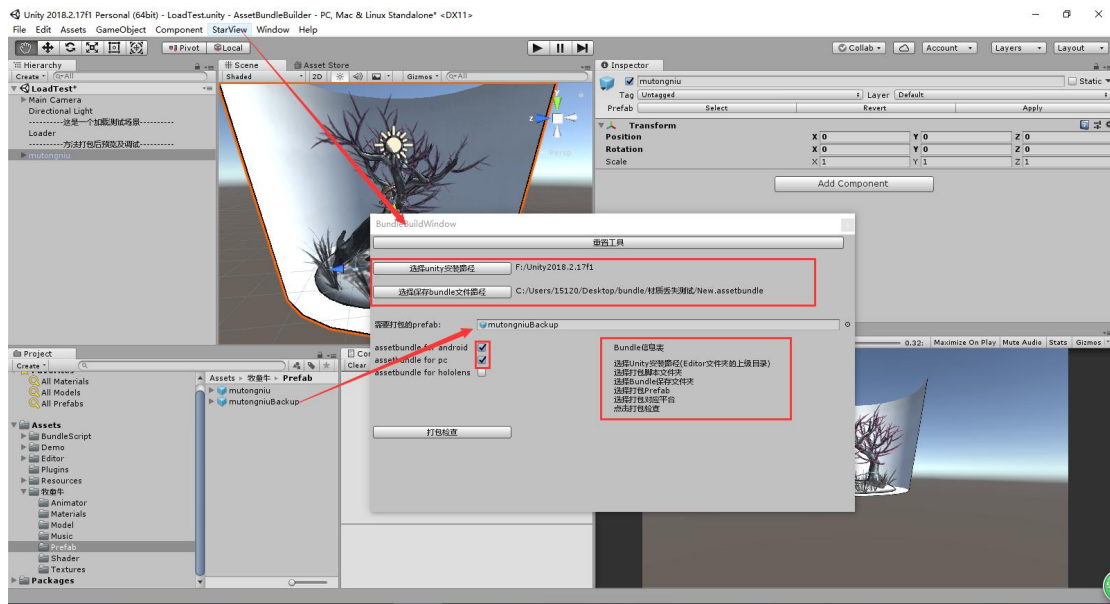
```
20 public class SetScale_Toggle : BundleEventInfoBase {
21
22     /// <summary>
23     /// 自定义值，一般为private
24     /// </summary>
25     private bool scale;
26     void Start () {
27         scale = true;
28     }
29     /// <summary>
30     /// 事件执行主体
31     /// </summary>
32     /// <param name="eventData"></param>
33     public override void OnBundleAction(PointerEventData eventData) {
34         scale = !scale;
35         gameObject.transform.localScale = scale ? UnityEngine.Vector3.one : UnityEngine.Vector3.one * 2f;
36         SendMsg(scale.ToString());
37     }
38     /// <summary>
39     /// 执行本地事件后其他客户端接收回调，通常和OnBundleAction内一样但不调用SendMsg
40     /// </summary>
41     /// <param name="msg">接受参数，需要转换一下</param>
42     public override void OnReceiveMsg(string msg) {
43         scale = bool.Parse(msg);
44         gameObject.transform.localScale = scale ? UnityEngine.Vector3.one : UnityEngine.Vector3.one * 2f;
45     }
46     /// <summary>
47     /// 一个动画，返回值为False即可
48     /// </summary>
49     /// <returns></returns>
50     public override bool supportPRS() {
51         return false;
52     }
53 }
54
```


6. 从 Unity 输出 AssetBuddle 文件

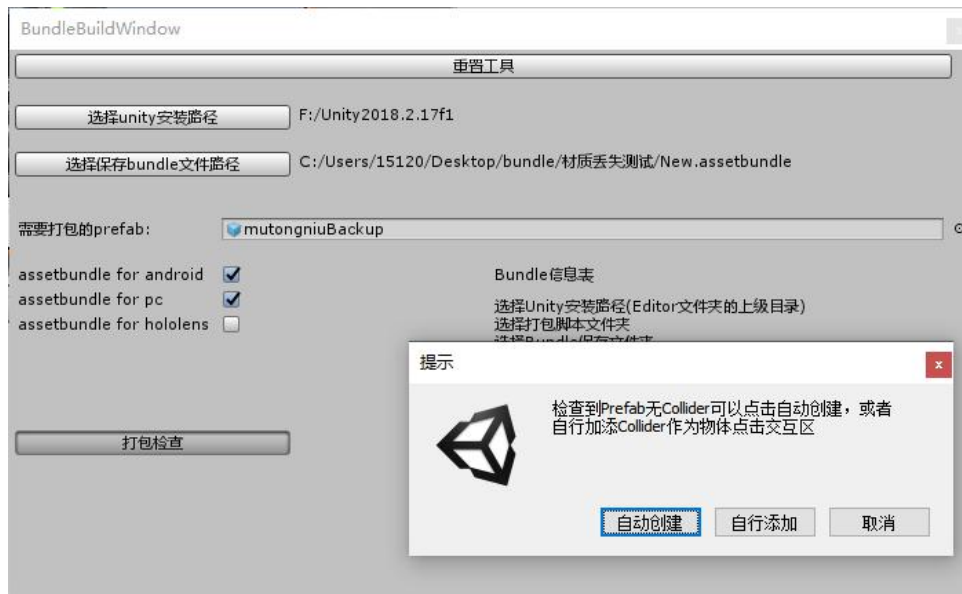
(1) 新建文件夹制作要打包的 Prefab，对其进行备份



(2) 工具栏上 StarView 打开 BundleBuildWindow 窗口，配置好安装目录和文件保存目录，然后点击打包检查，工具会自动进行必要的检查和配置



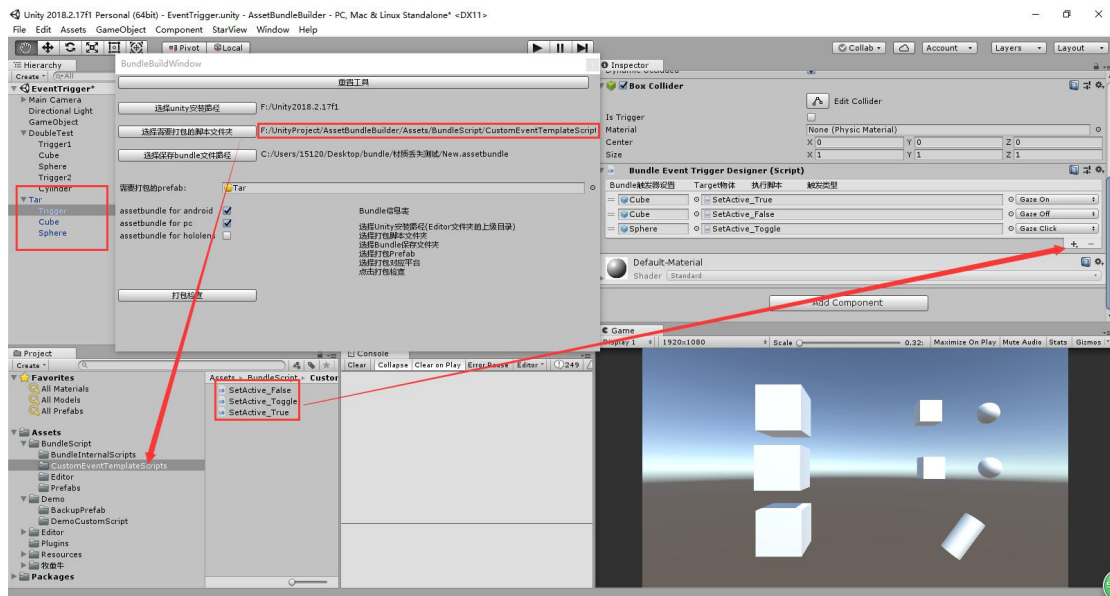
(3) 点击打包检查可能会弹出提示框，会有 3 个选项“自动添加”，“自行添加”和“取消”，为了在眼镜端实现可交互，我们在制作 Bundle 时必须为所有带有 Render 的物体挂载 Collier 在应用内可以内 GazeOn 捕获到以便做触发交互，自动添加一般是可以满足需求，但对于 ParticleSystemRenderer 粒子特效等无形的 Render 支持不是很好，需要制作者自行添加 BoxCollider 或者在自动添加后调整其大小，在点击添加后程序会对原有的 Prefab 修改。



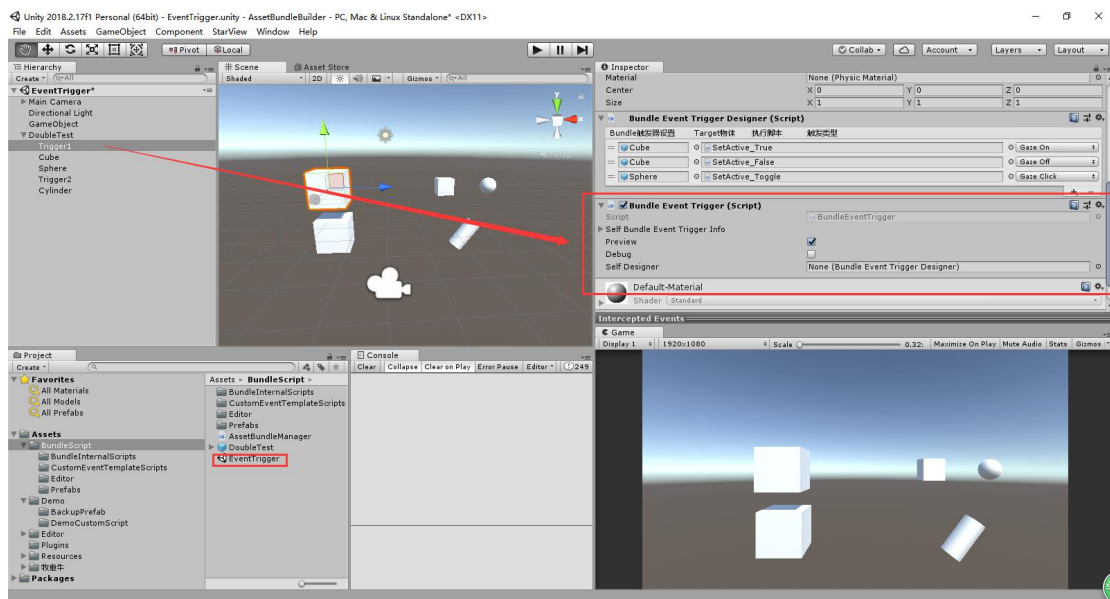
(4) 打包, 打包检查成功后即可开始打包, 打包过程和项目内容和项目内无关内容有关, 这可能是个漫长的过程。



(5) 对有自定义触发事件的物体制作时, 将带有脚本的 Prefab(或者 Demo/BackupPrefab/Tar 复制一份) 拖入需要打包的 Prefab 选择槽中, 会弹出“选择需要打包的脚本文件夹”图片中选择 Asset/BundleScript/CustomEventTemplateScripts, 新建一个 Cube 作为事件触发器(或者在模型上添加 MeshCollider 后添加组件), 添加 BundleEventTriggerDesigner 组件后该物体会自动添加一个 BoxCollider 作为触发器的触发范围(当然你也可以调整其大小), 触发器必须是单独的一个物体不容许和其他触发器或者模型嵌套



(6) 示例项目 EventTrigger 场景附带了一些给开发人员内部测试的例子,当然此文件夹不可以被挪动和删除,包含了所有工具的内容。该场景示范了如何在制作过程中测试自己的触发事件而不需要在 Loadtest 场景把打包的内容加载出来在进行测试,只需要在原有的 Trigger 上挂载 BundleEventTrigger 组件,勾选 Preview 值运行即可测试。不过测试完不要忘记删除 BundleEventTrigger 组件再打包。



7. 问题处理

1. 材质丢失

关于 bundle 工具打包后在 view 眼镜端及 PC 端加载材质丢失问题说明

- (1) 目前支持 Stander 和自定义的 Sander, 如果有材质丢失问题请根据这种内容向开发人员反馈。

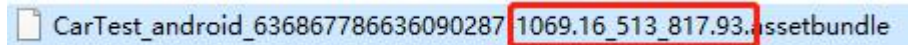
(2) 对于自定义 shader 注意选择 CompiledCode 类型为 AllPlatforms

2. 加载中等待时间长

大于 3 分钟可能已经报错，请检查制作 Bundle 在 Loadtest 场景加载测试，如有错误请在打包时按照规范制作。

3. 加载后看不到物体

(1) 在制作 Bundle 的 Prefab 时要注意比例尺，打包后的 Bundle 文件名后面数字代表了该物体的实际大小(单位：米)，如下图已经不适用于眼镜端，加载后物体太大或者太远



CarTest_android_636867786636090287_1069.16_513_817.93.assetbundle

(2) 注意 Prefab 的中心点位置，加载后 Pin 放置位置根据物体的 Bounds 计算，物体中心点偏移导致计算得出的 Bound 太大也会导致物体里头部太远看不到

4. 其他问题

目前在对模型资源的测试没有发现问题，我们当前能想到的情况也都写在代码里，当你遇到特殊资源时，程序可能会无法正常打包，请尽快联系我们，我们会尽快增加对特殊资源的处理，祝你使用愉快。