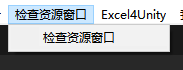
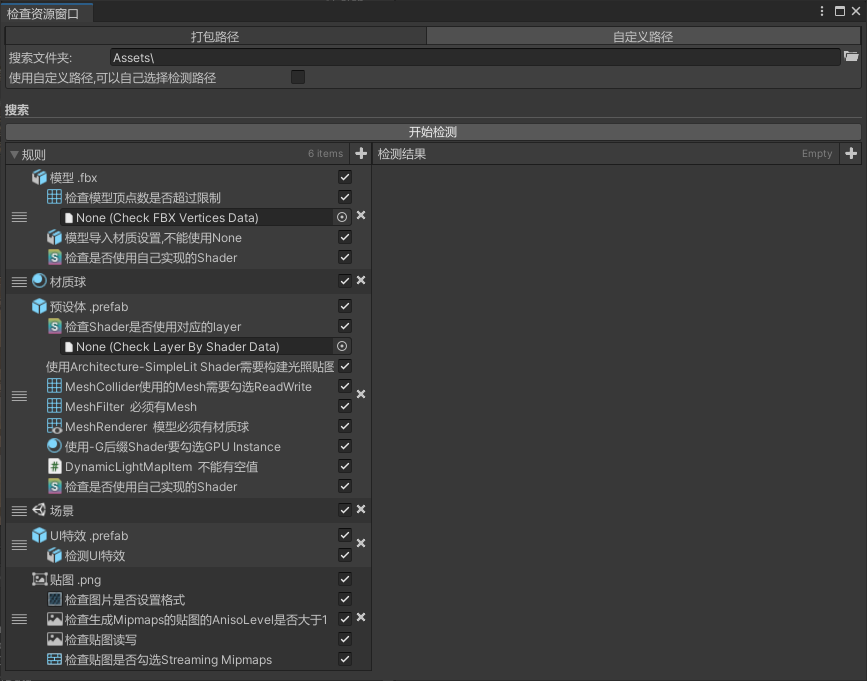
**资源检查工具**

# 打开检测资源窗口





# 配置路径

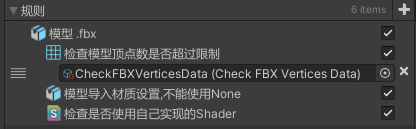
输入要检测的资源的路径

descript

# 搜索规则

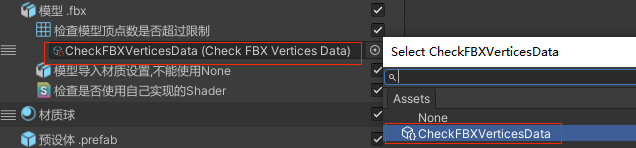
需要检查的资源勾选上。

# 模型

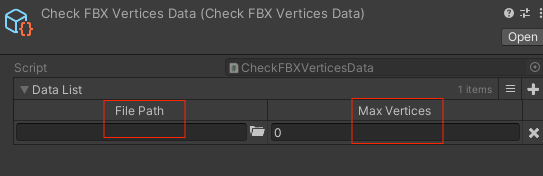


## 1.1 检查模型顶点数是否超过限制

选择asset资产



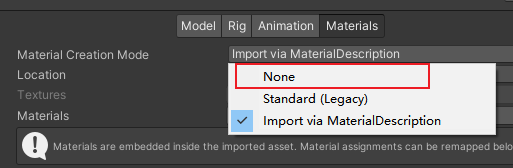
设置要检查的模型路径，以及对应最大限制的顶点数



对应的报错信息为：模型顶点数为：XX超过限制的XX

## 1.2 模型导入材质设置，不能使用None

如图不能设置为None，如果为None将可以被检测出



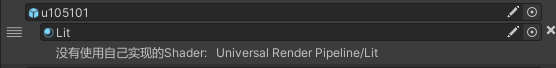
报错为：



## 1.3 检查是否使用自己实现的Shader

如果没有使用指定目录下的Shader，则会被检测出（指定目录为”Asseets/sources/Shaders”和”Assets/Spine”）

报错为：

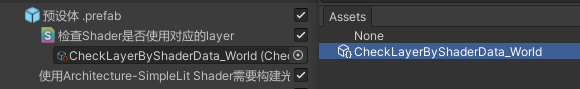


# 材质球

# 预设体

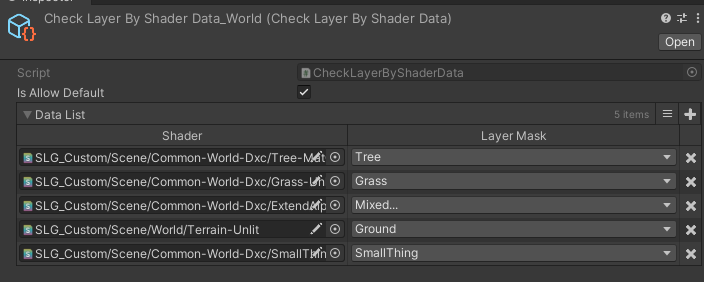
## 3.1 检查Shader是否使用对应的layer

选择asset资产



拖入shader代码，选择相对应的layer层级。

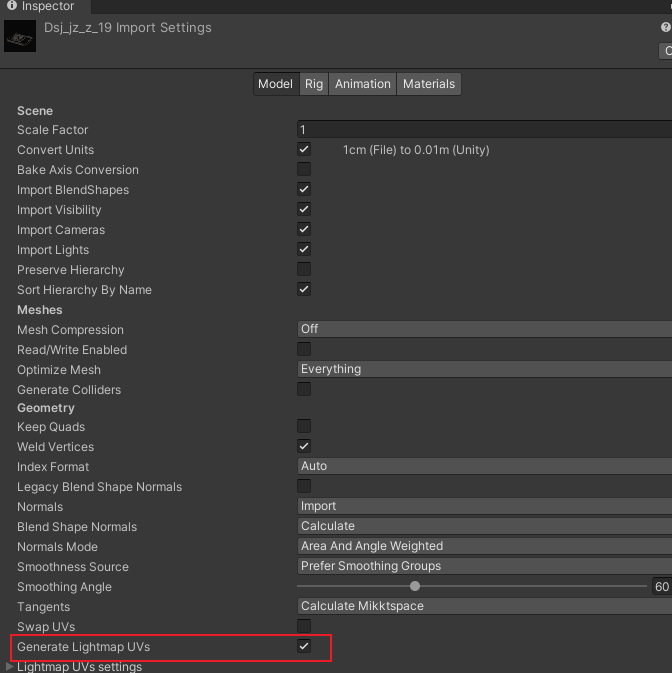
对应的报错信息为：没有使用正确的layer



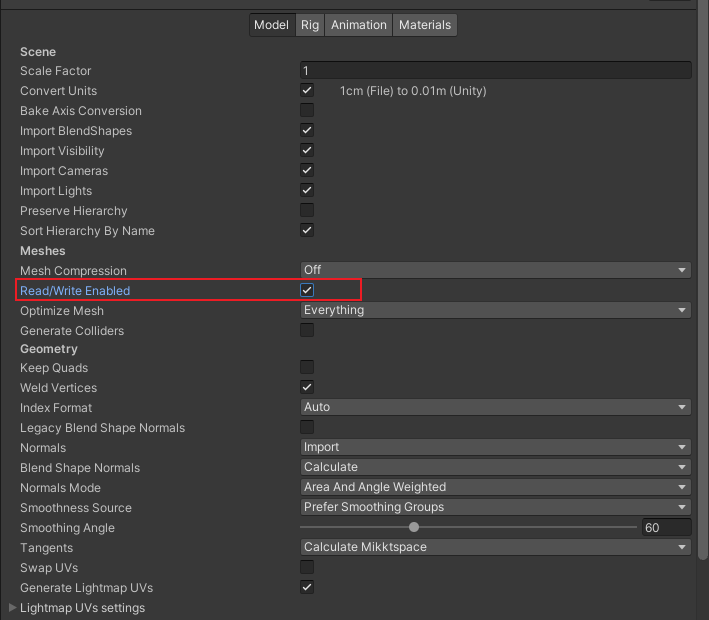
## 3.2 使用Architecture-SimpleLit Shader需要构建光照贴图

使用该shader时，模型导入设置中，需要勾选Generate Lightmap UVs。

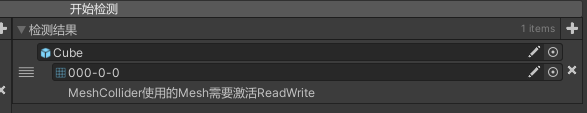
对应报错信息，检测结果中显示为：模型使用Architecture-SimpleLit.Shader,没有勾选生成uv。



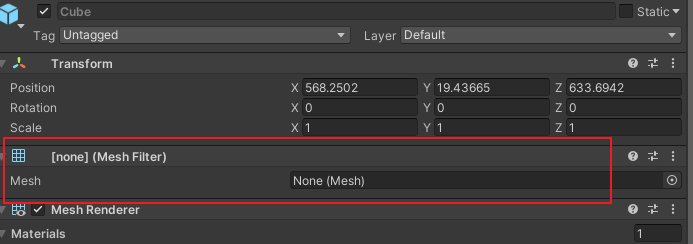
## 3.3 MeshCollider使用给的Mesh需要勾选Read Write



对应的报错信息为：



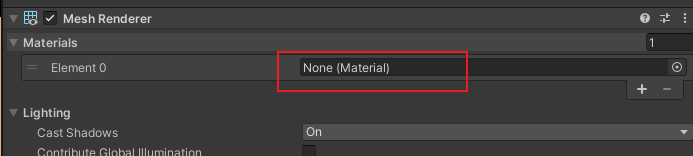
## 3.4 Mesh Fileter必须有Mesh



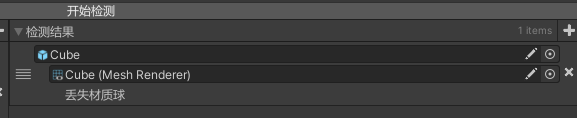
报错为：



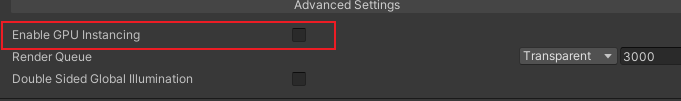
## 3.5 MeshRenderer模型必须有材质球



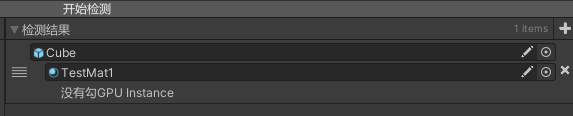
报错为：



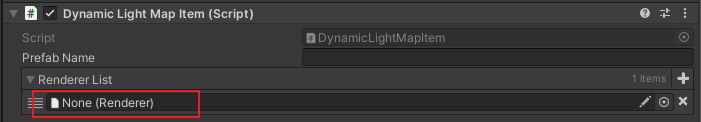
## 3.6 使用-G后缀Shader要勾选GPU Instance



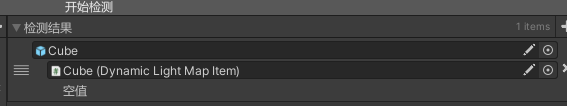
报错为：



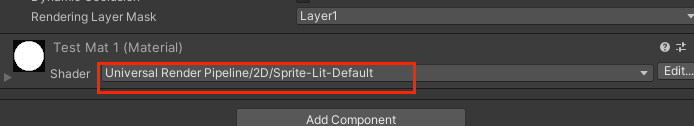
## 3.7 DynamicLightMapItem不能有空值



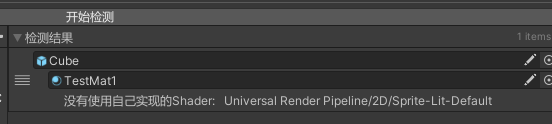
报错为：



## 3.8 检查是否使用自己实现的Shader



报错为：

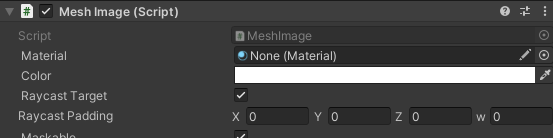


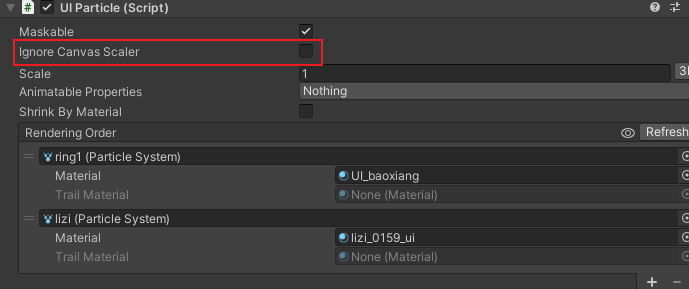
# 场景

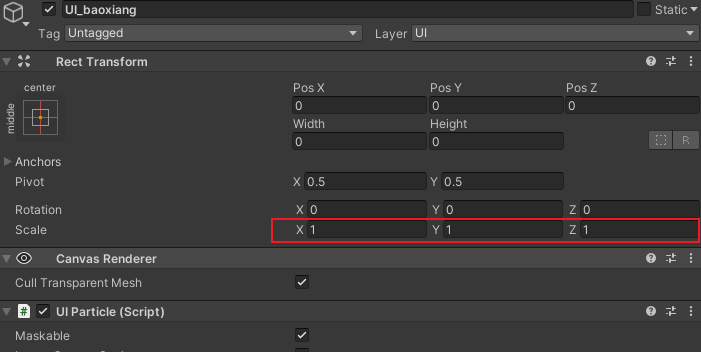
# UI特效

## 5.1 检测UI特效

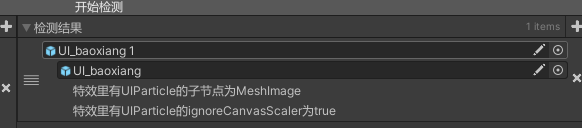
UI特效里有UI Particle组件的节点，其子节点不能有MeshImage组件、参数Ignore Canvas Scaler不能为true、节点的localScale不能为0







报错为：



# 贴图

## 6.1 检查图片是否设置格式

注意此功能检测的路径，必须符合一下两种格式

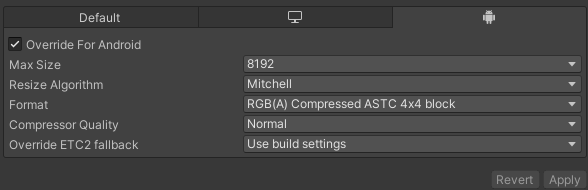
Format："Astc4","Astc5","Astc6","Astc8","Astc10","Astc12","RGBA32",

Size： "64","128","256","512","1024","2048","4096","8192","16384",

1、.../Format/Size 例如：”Assets/sources/Tmp/Astc4/4096”

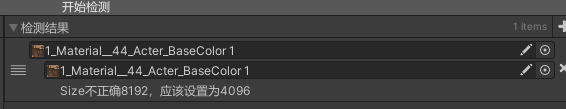
2、.../Format 例如：”Assets/sources/Tmp/Astc4”

检查贴图的大小和格式

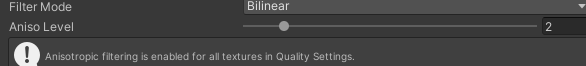




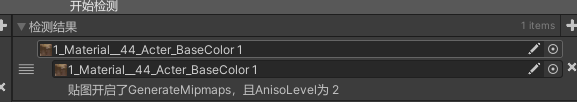
报错为：



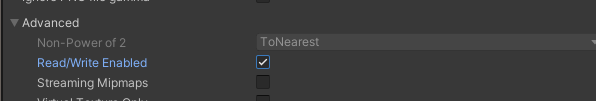
## 6.2 检查生成Mipmaps的贴图的AnisoLevel是否大于1



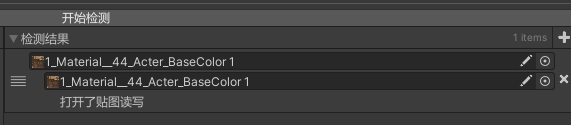
报错为：



## 6.3 检查贴图读写



报错为：



## 6.4 检查贴图是否勾选Streaming Mipmaps

如果勾选了Genenrate Mip Maps，却没有勾选Streaming Mipmaps，报错

