

# 游戏项目实践 实践课

- 游戏框架及游戏跨平台实践



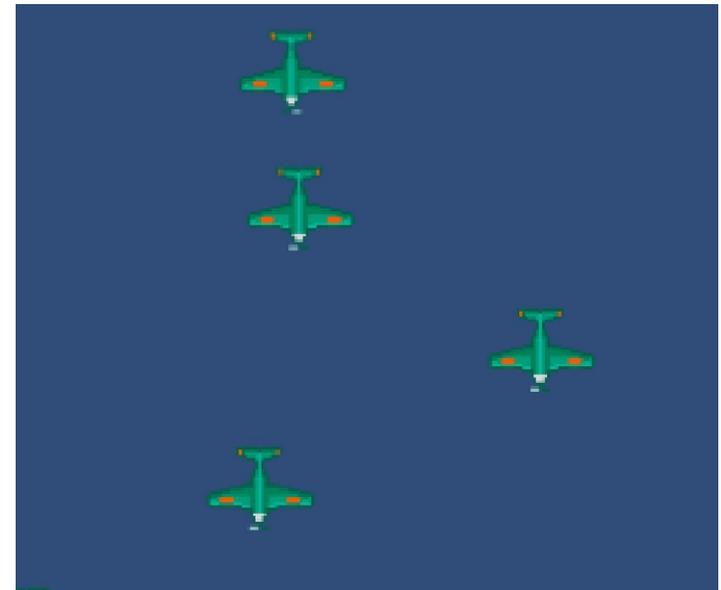
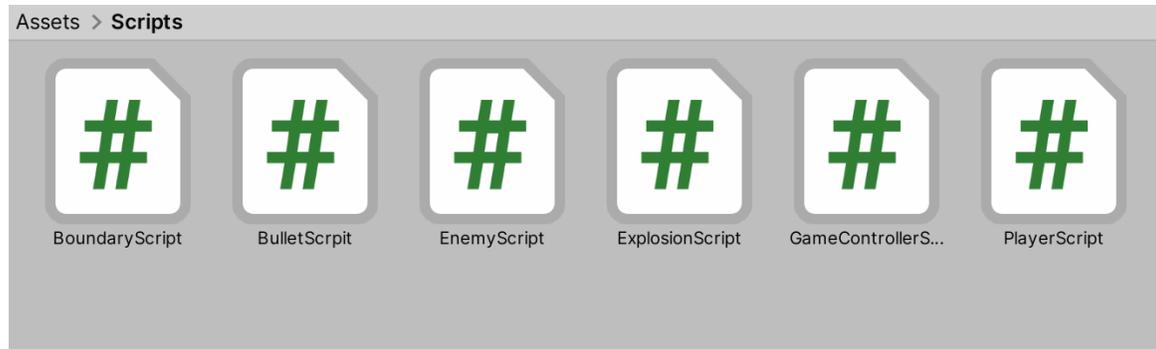
1

# 敌机的出现

# 敌机的出现

## 练习1-1:

1. 导入12-2DFlightShooting.unitypackage
2. 完善GameControllerScript.cs的SpawnEnemy函数，实现敌机的初始位置设置（可随机或遵循一定的规律）
3. 完善EnemyScript.cs代码的Update函数，实现敌机的位置更新



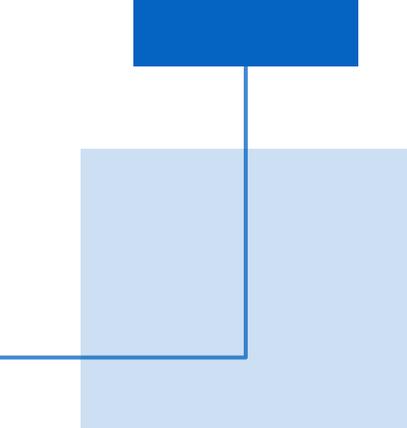


2

统计分数  
游戏结束  
重新开始

### 练习2-1:

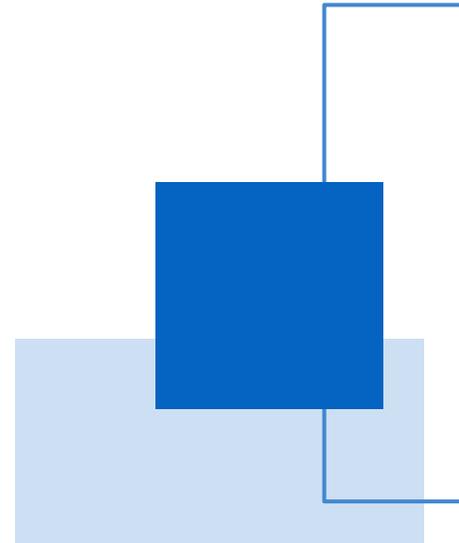
1. 完善GameControllerScript.cs的AddScore及EnemyScript.cs的OnTriggerEnter2D函数，实现分数的统计，子弹碰到敌机计1分
2. 完善EnemyScript.cs的OnTriggerEnter2D函数，实现我机碰撞敌机时显示游戏结束的提示
3. 完善GameControllerScript.cs的Update，实现按下Space键时重新开始游戏



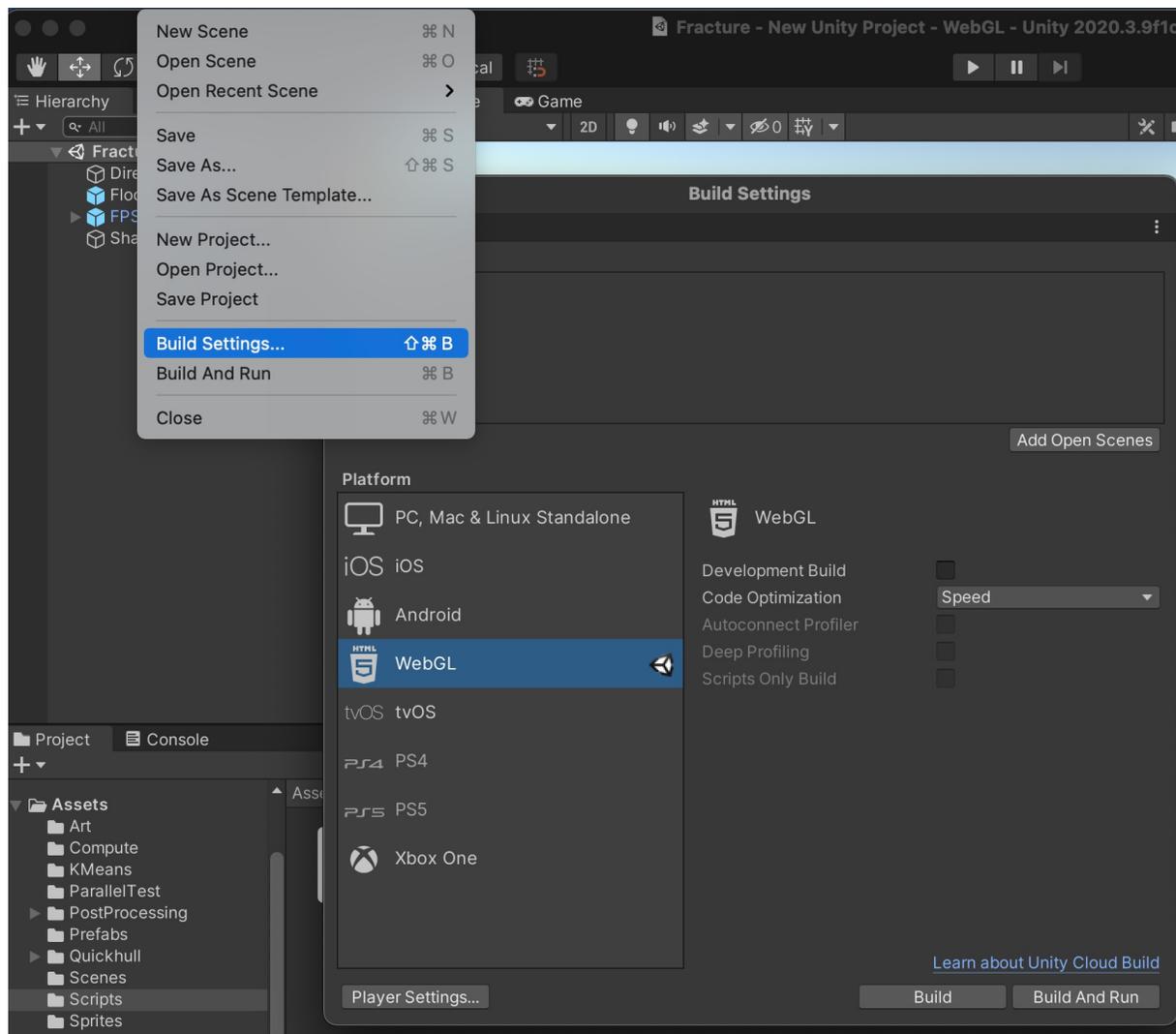
3

# 在浏览器上运行 飞行射击游戏

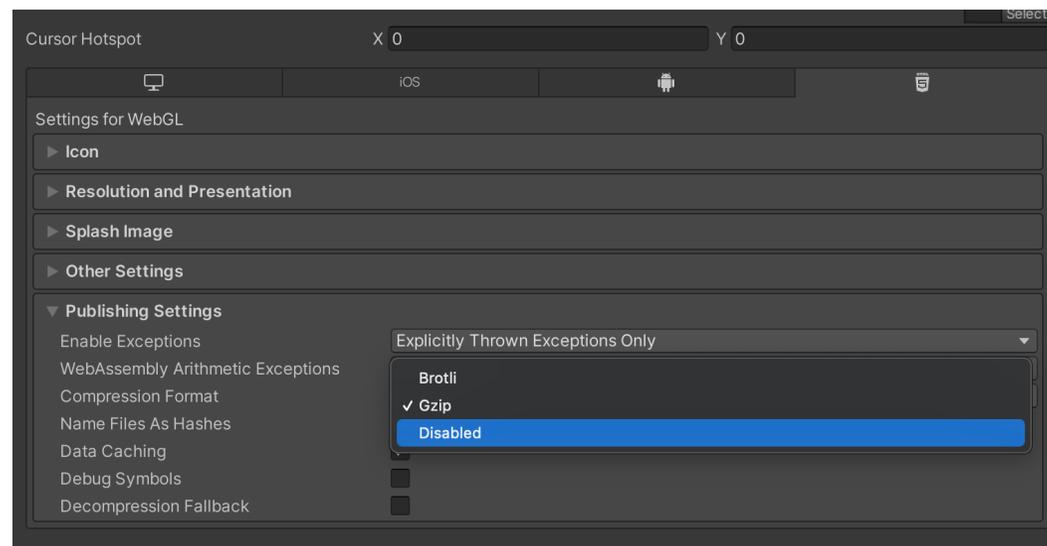
3-1、3-2二选一即可



# WebGL



- 如果提示没有安装Unity的WebGL模块，则要安装
- 直接打开Html通常无法运行游戏，须在文件目录启动Web服务，方式见后页
- 如果浏览器中遇到报错，可能是项目文件中有中文、项目路径中有中文或者发布WebGL的选项需要Gzip要设置为Disable



因载入其他文件的需要（图片、csv、json等文件），需针对**存有网站文件的本地文件夹**启动一个Web服务  
建立Web服务的方式有多种，练习时建议使用Python3启动Web服务：

## 1. Windows

- **Python3**: `python -m http.server`
- IIS
- Apache
- Mac/Ubuntu
  - **Python3**: `python -m http.server`
  - Apache

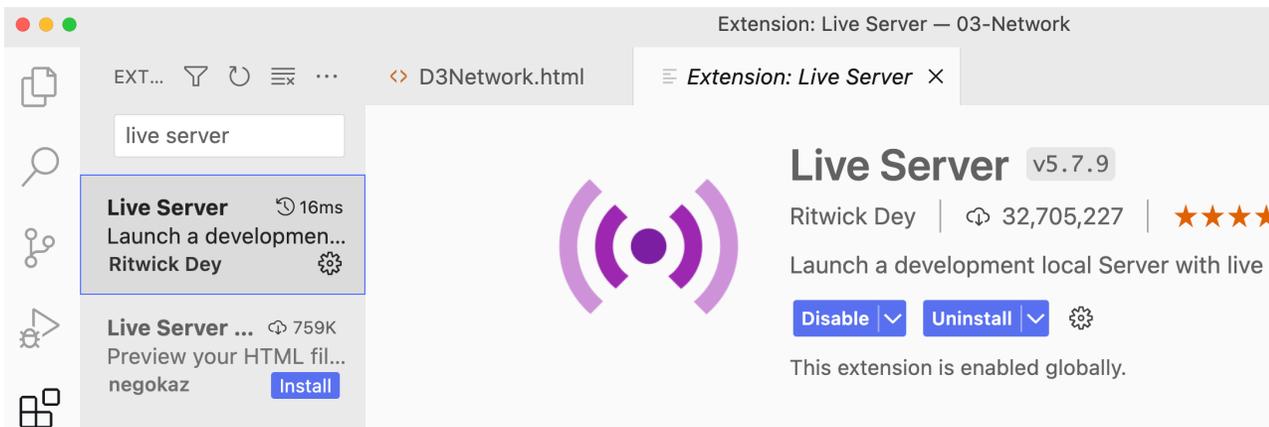
## 练习3-1（使用Python启动Web服务）：

1. 在控制台里进入网页文件的文件夹
2. 输入：`python -m http.server`（假设安装的是python3）
3. 通过`http://localhost:8000`访问网页
4. 如需自定义端口，命令可改为：`python -m http.server 8000`

**注意**：如遇到修改代码后网页内容不更新，请尝试清浏览器缓存或改变端口

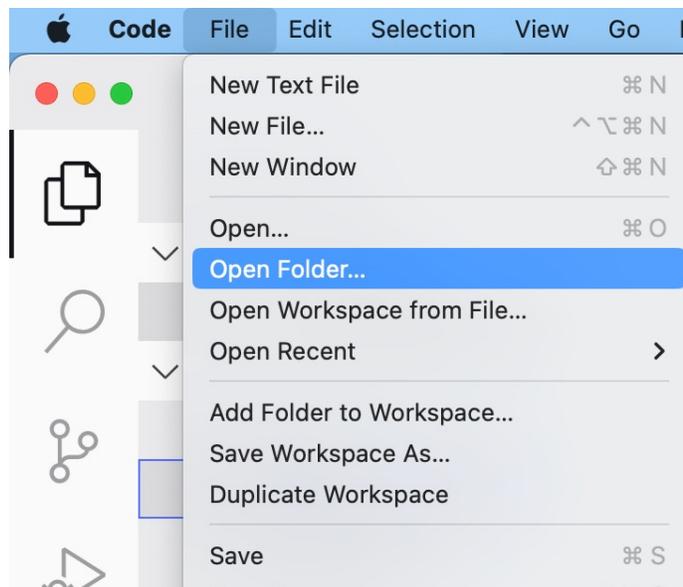
## 练习3-2（使用Visual Studio Code的Live Server启用Web服务）：

### 1. 安装Live Server插件

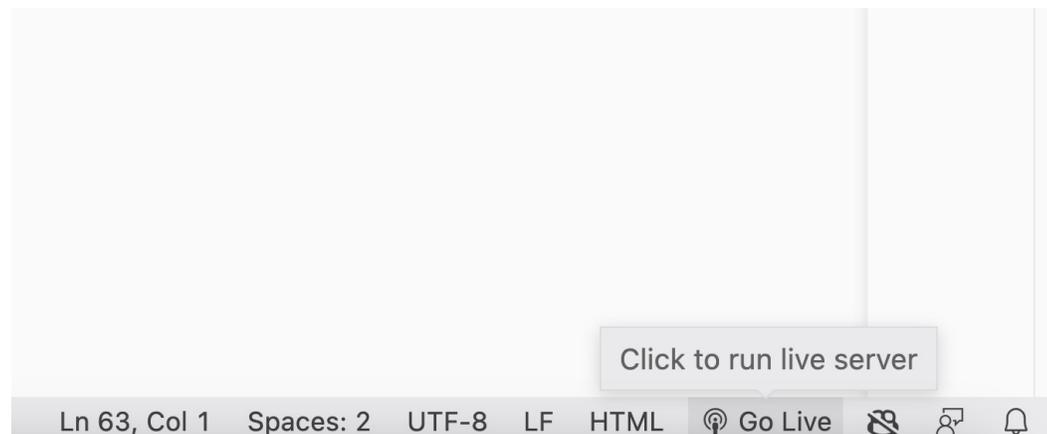


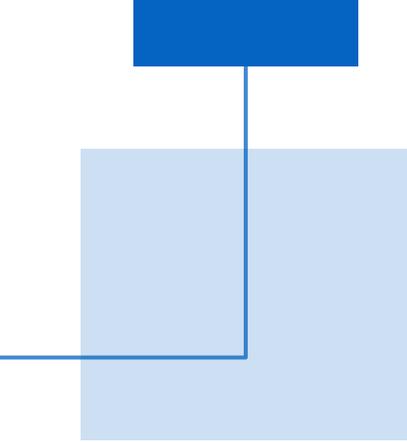
### 4. 浏览器中查看结果

### 2. 打开代码文件夹



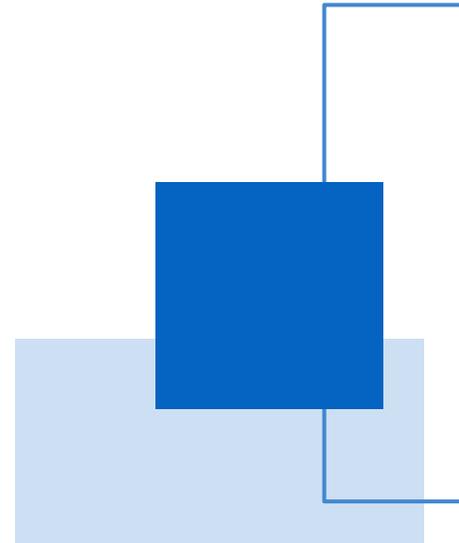
### 3. 选中html文件，点击右下角Go Live按钮





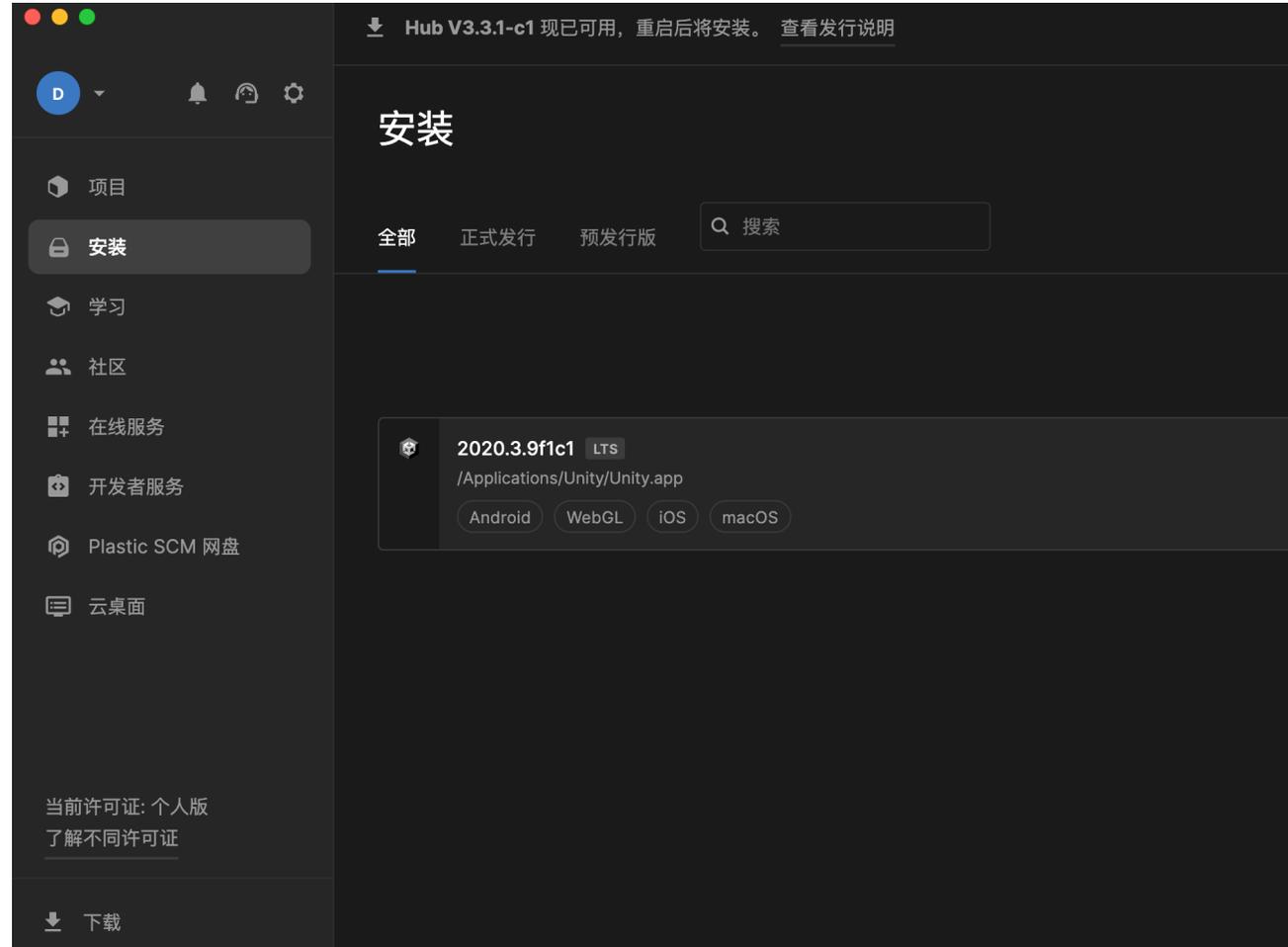
4

# 在手机上运行 飞行射击游戏 (课后练习)

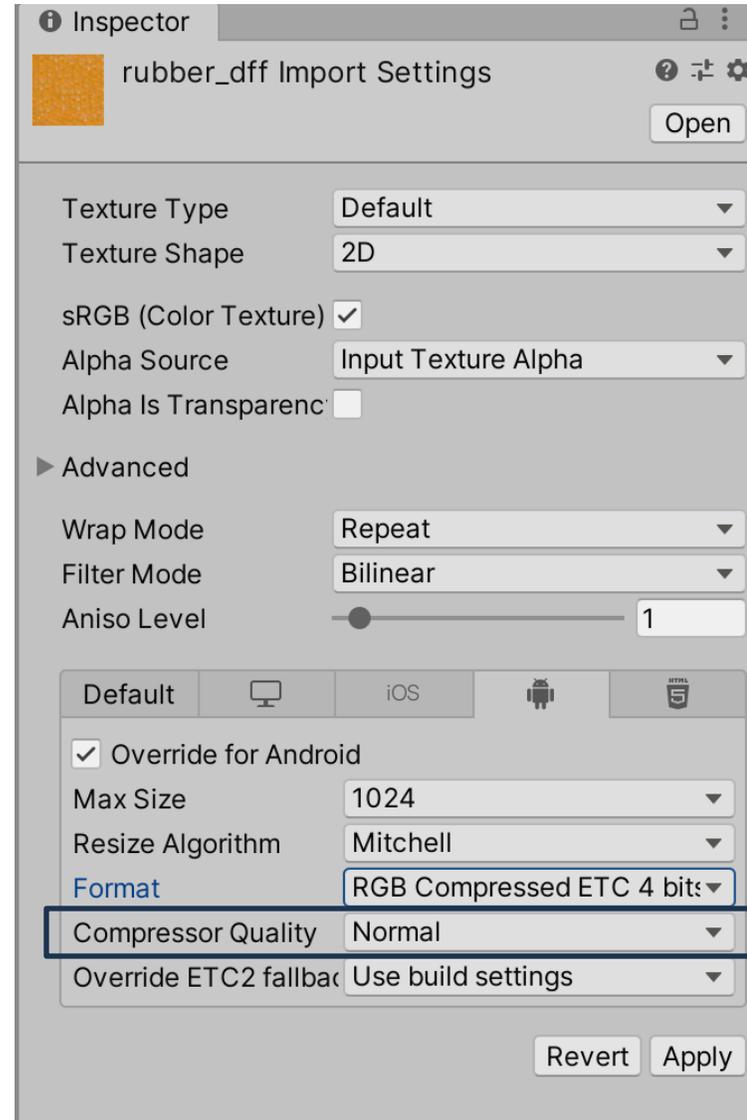
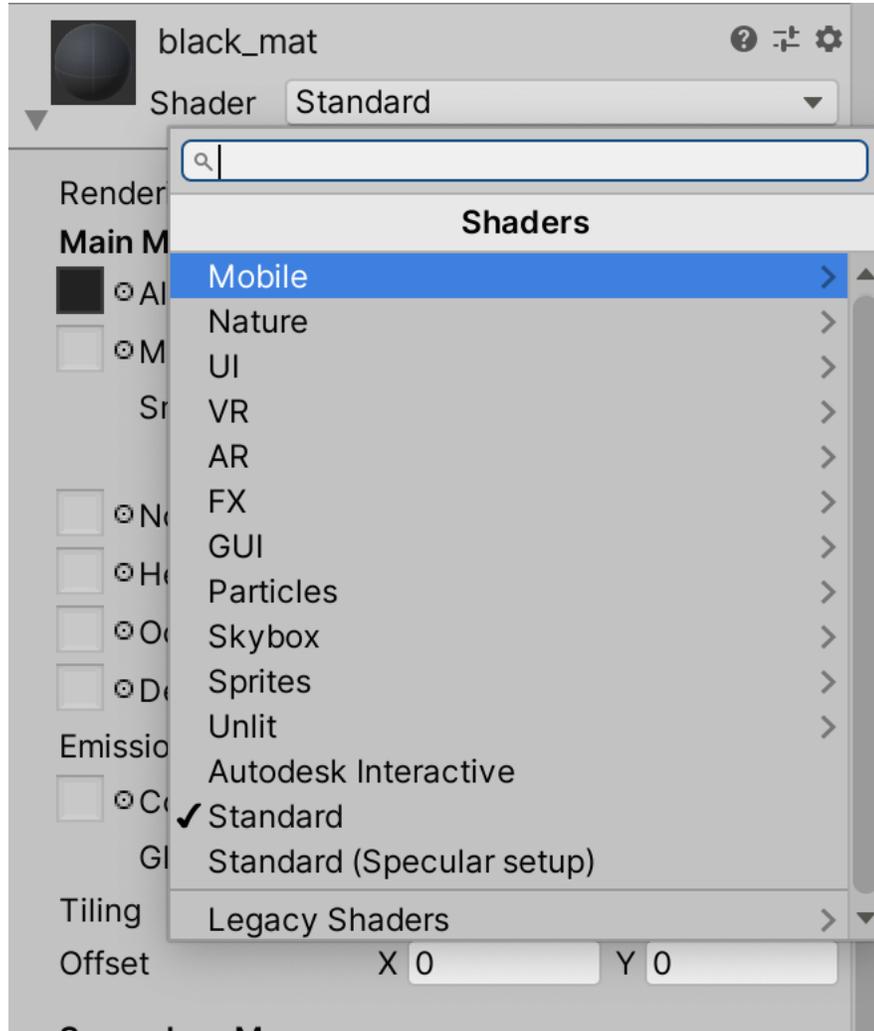


## Unity移动端支持包的安装

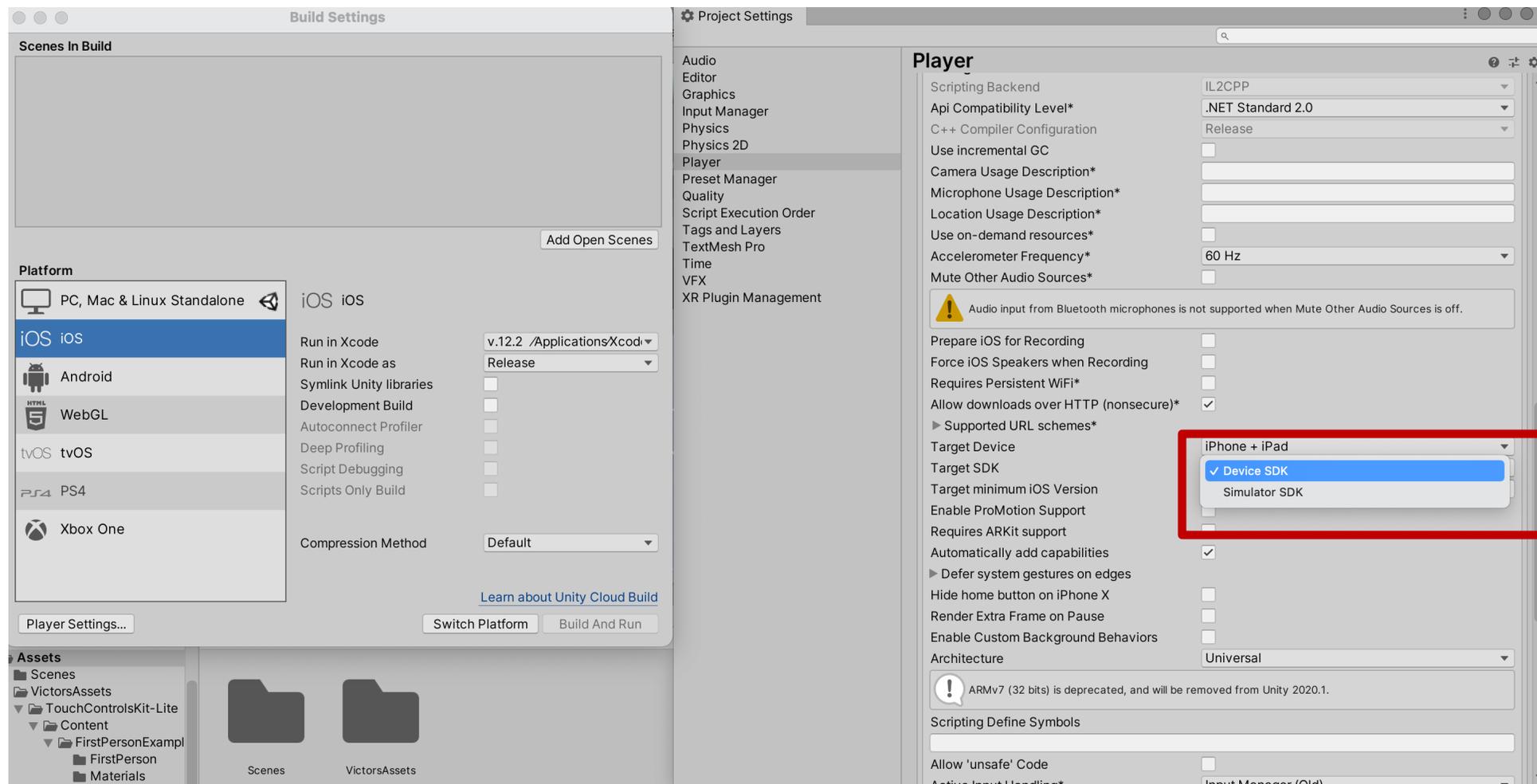
- 如果无法补安装的话，需要卸载Unity重新安装，注意勾选支持Android或iOS以及WebGL



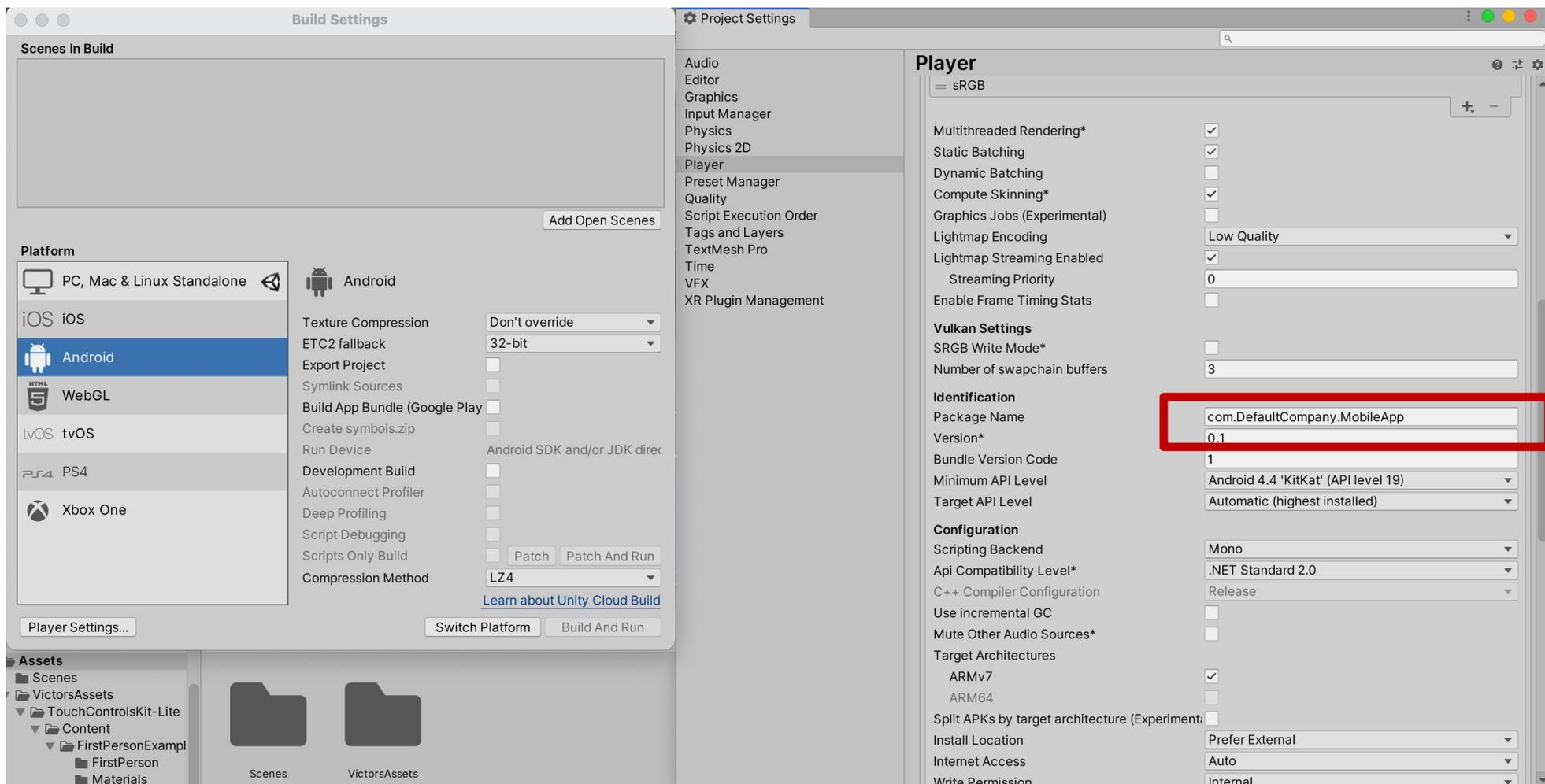
# 移动端图片及材质设置



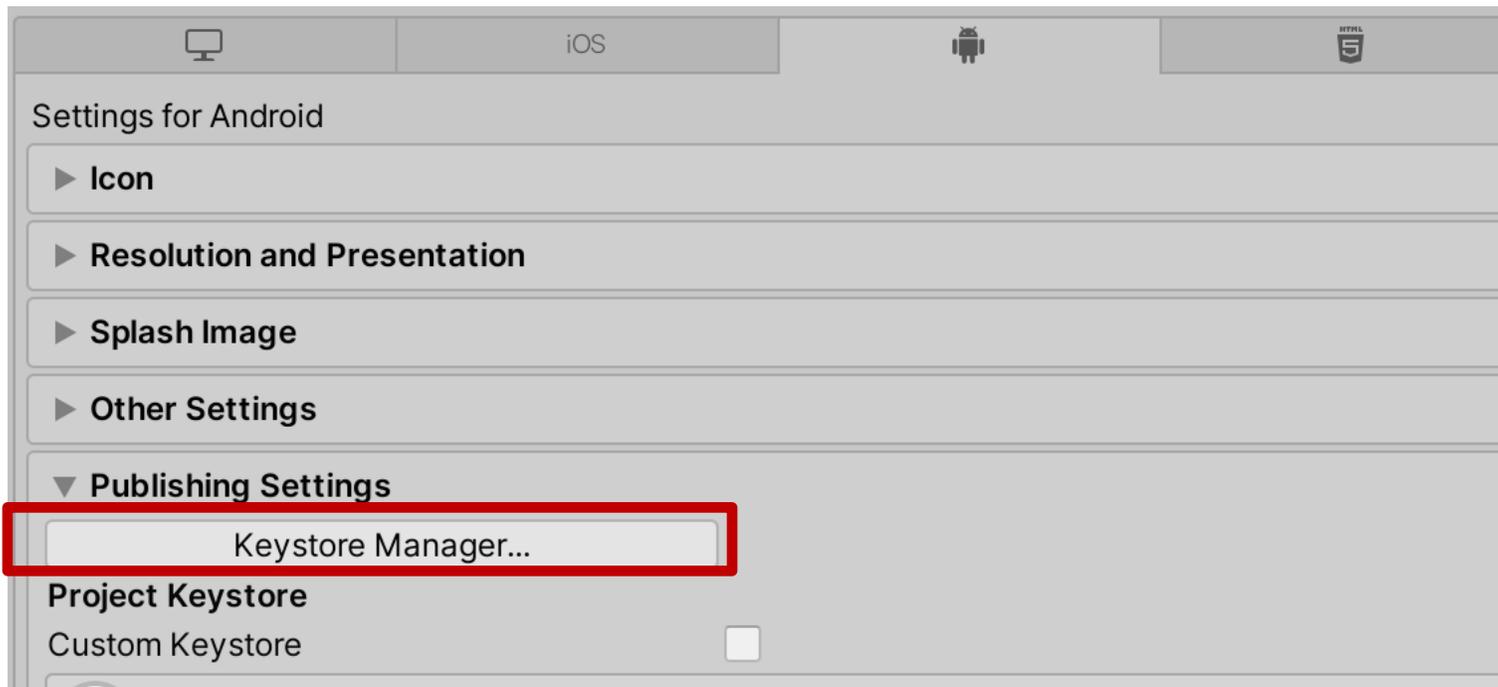
## 真机 (Device SDK) 或模拟器 (Simulator SDK) 发布



## Apk包名的设置，设定为自定义的名字



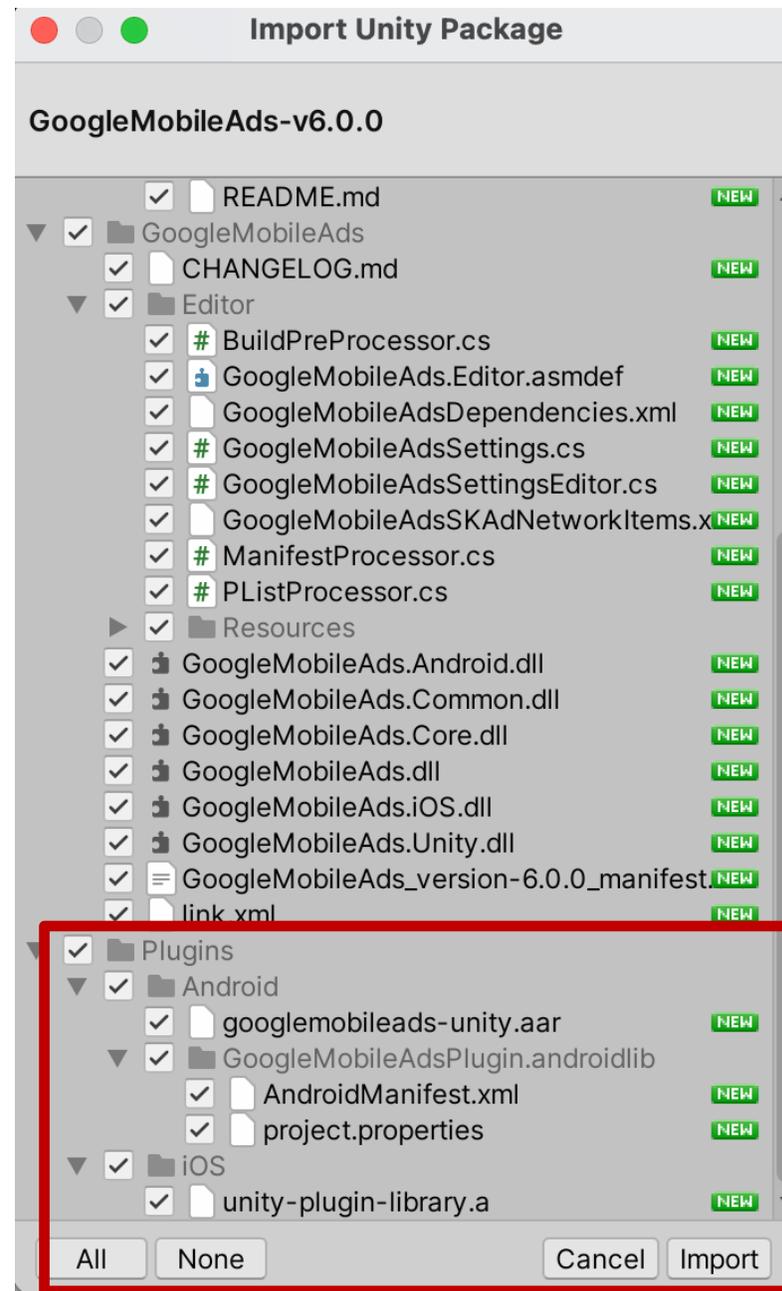
## 如果要发布到安卓商店，需要进行签名的设置



## Plugins

- Android
- iOS

Unity中和Android及iOS相关的插件文件，  
统一放到Plugins目录下的制定文件中



| Q&A