

# 1. 编译 UE5 源码

## 1.1 获取源码

### 1.1.1 EPIC 与 GitHub 连接

官方详细教程：[GitHub 上的虚幻引擎 - Unreal Engine](#)

备注：如果进入账户管理界面出现网页内容点击无效的情况，可以先取消已经启用的浏览器插件，然后按“Ctrl+Shift+R”刷新浏览器缓存。

### 1.1.2 从 GitHub 找到源码

UE5 GitHub 源码：[EpicGames/UnrealEngine: Unreal Engine source code](#)

备注：部分情况需要使用梯子科学上网才能进入 GitHub。

如图 1 按顺序点击 Code 和 HTTPS 复制仓库地址。

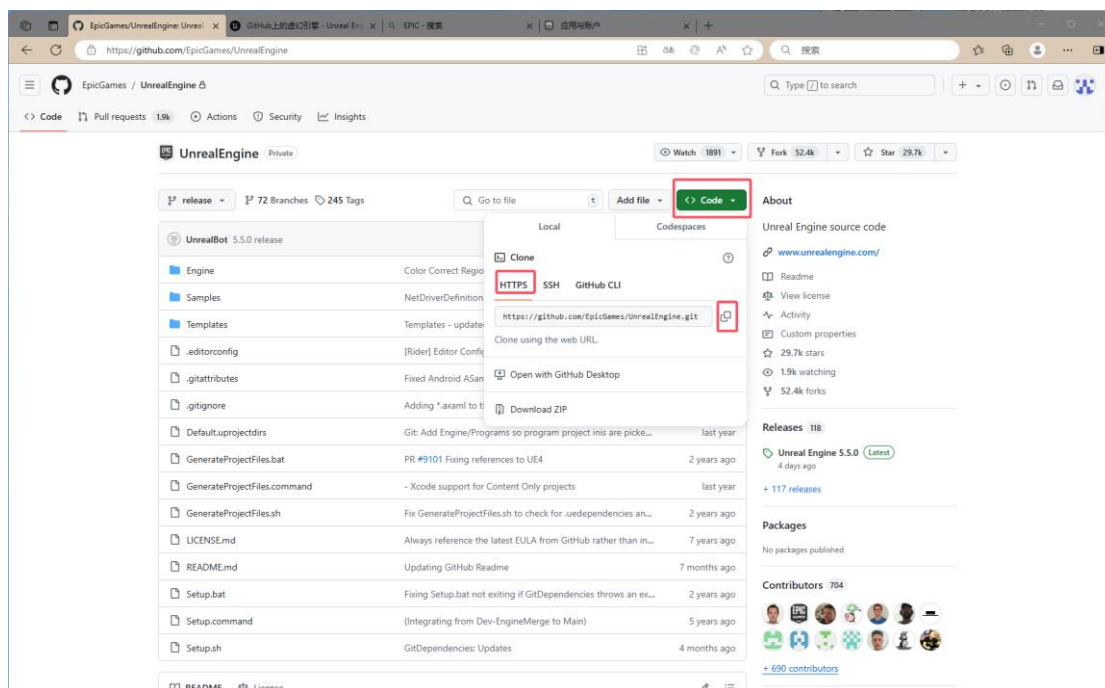


图 1 复制仓库地址

### 1.1.3 克隆仓库到本地

方法 1:

IDE 选用 VS2022，如图 2 点击克隆存储库。



图 2 克隆仓库

如图 3 输入仓库的 URL 并选择将要保存的路径。此处的 URL 为 1.2 节中复制的内容。点击右下角克隆，完成 UnrealEngine 项目克隆。

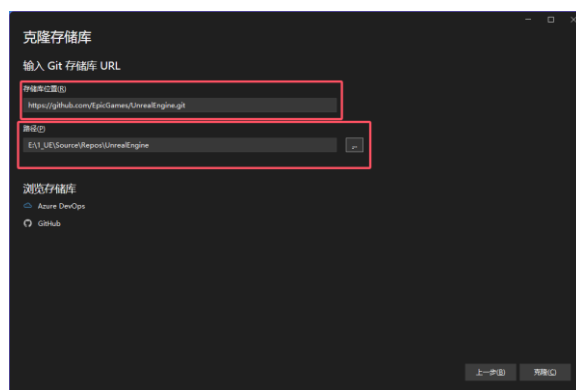


图 3 添加 URL 并选择路径

如图 4 将 VS 与 GitHub 建立连接，完成建立后等待克隆完成。（时间较长）

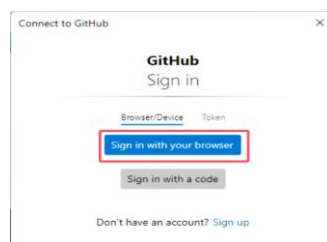


图 4 VS 与 GitHub 建立连接

备注：若出现图 5 的请求，可凭自己意愿选择是否同意 NVIDIA Nsight 采集个人设备信息。

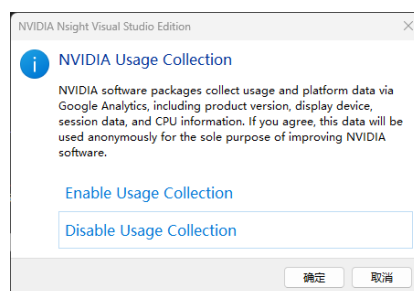


图 5 NVIDIA Nsight 信息采集请求

## 方法 2:

如果方法 1 克隆失败，可以使用 **Git Bash** 将仓库克隆到本地文件目录（如图 2）。由于 UnrealEngine 的项目较大，需要等待一段时间。

备注：Git Bash 需要先在本地安装 **Git**，这里给出官方文档 [Git - 安装 Git](#)，也可使用博客查找其他安装方法。

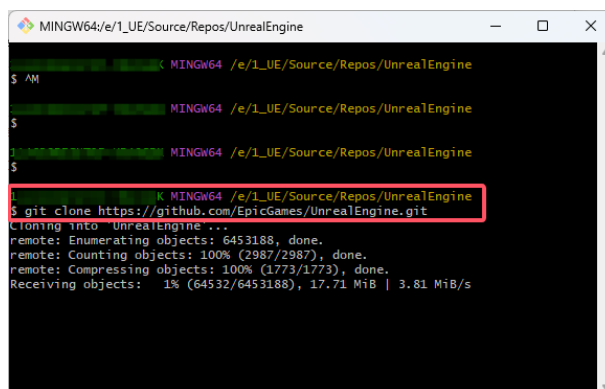


图 6 克隆仓库

## 方法 3:

如果仍旧不行，可以使用方法 3 下载项目到本地（如图 7），然后进行解压缩。

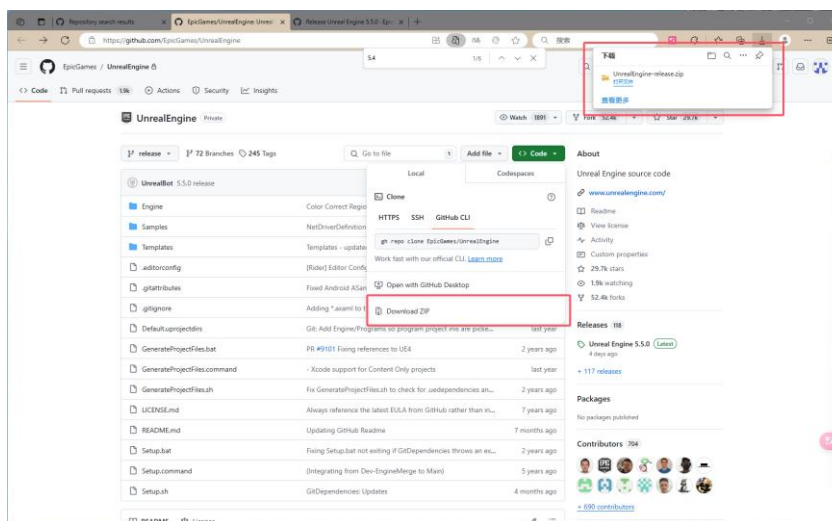


图 7 下载 UnrealEngine 压缩包

## 1.2 编译源码

### 1.2.1 使用 VS 打开项目

打开 VS2022，选择打开本地文件夹即可将在 VS 中查看项目目录，询问是否启用 **cmake**，先选择不启用。



图 8 打开 UnrealEngine

## 1.2.2 安装 UE 依赖并生成解决方案

如图 9，在目录下打开 CMD 命令行，执行 Setup.bat 脚本可以查看安装进度。

备注：中间安装过程需要**管理员权限**，如果没有弹出管理员权限请求，注意查看任务栏是否有权限申请在闪烁。

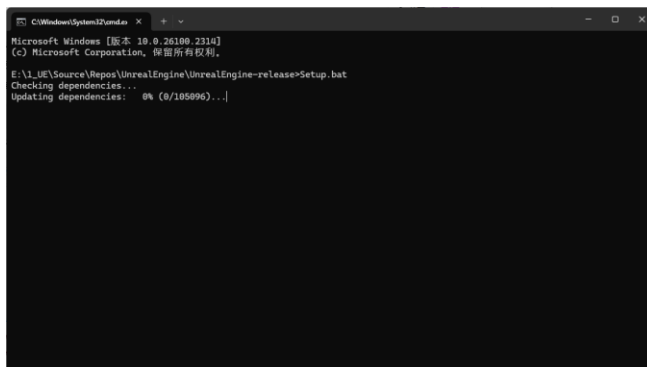


图 9 安装依赖

依赖安装完成后，需要运行 GenerateProjectFiles.bat 生产项目的解决方案，右键以管理员权限运行 GenerateProjectFiles.bat 即可。

解决方案生成后，项目的基础目录结构如图 10 所示：

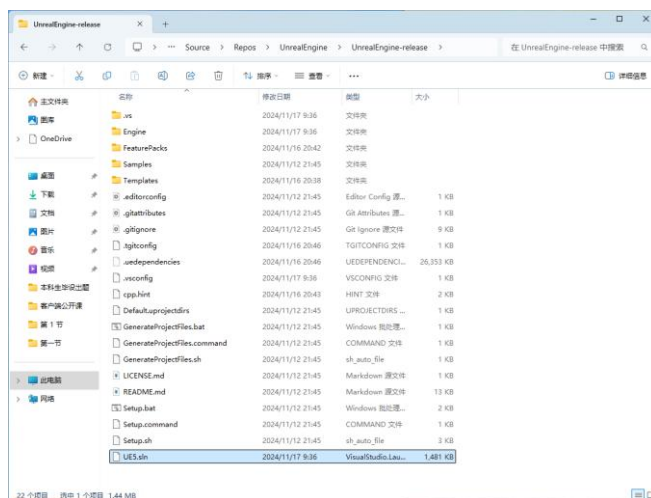


图 10 解决方案基本目录结构

### 1.2.3 编译源码

首先，将 Engine 文件夹下的 UE5 设置为启动项，具体操作如图 11 所示。

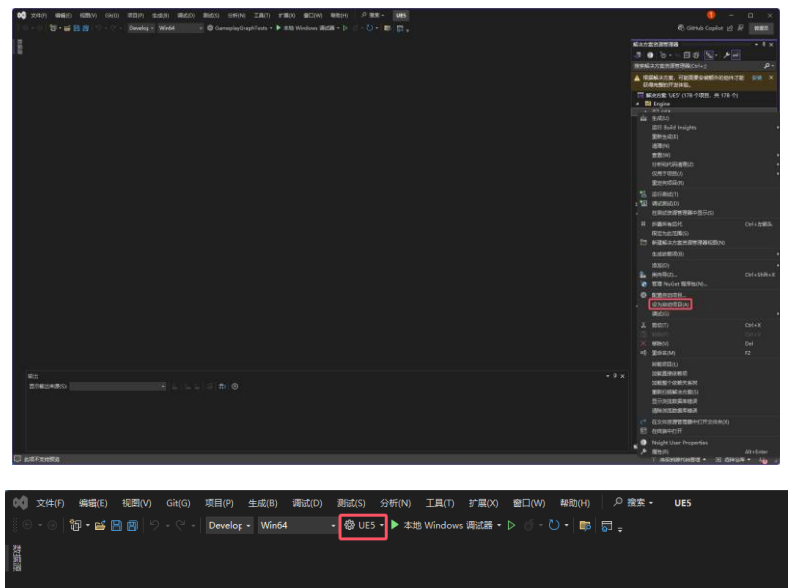


图 11 设置启动项

然后点击本地 Windows 调试器进行调试与编译。得到如图 12 的界面即代表已经开始调试与编译。此过程比较漫长，具体执行时间与硬件设备有关，需要耐心等待。

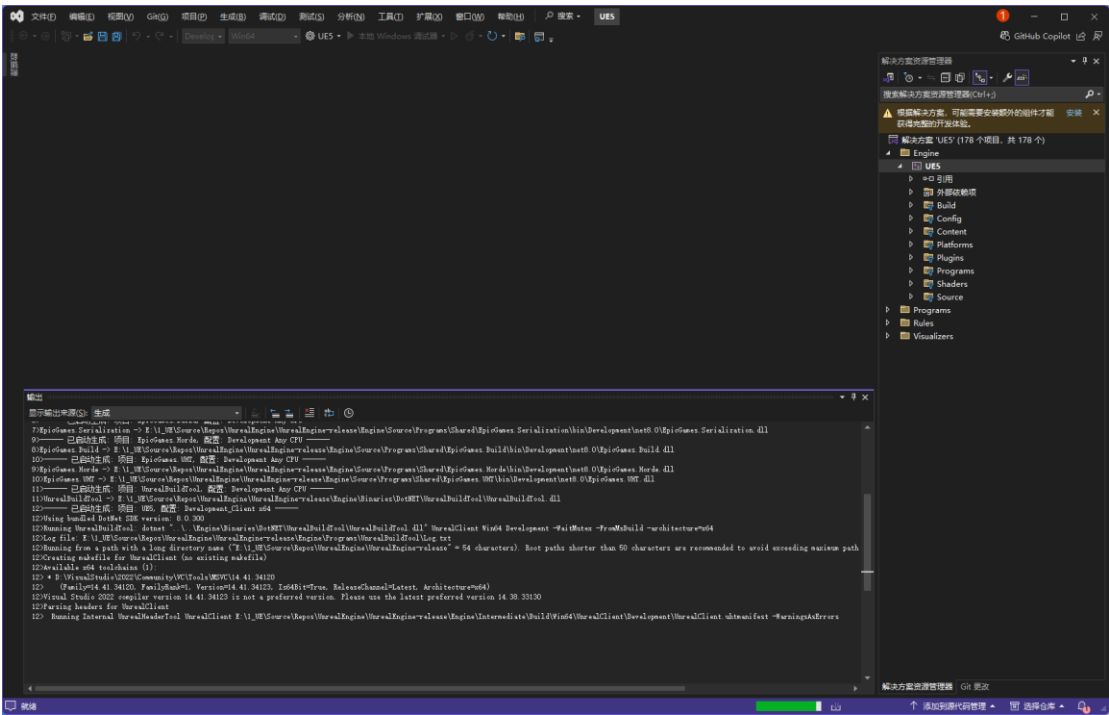


图 12 开始编译

编译完成后 UE 会自动运行，后台 VS 界面如图 13 显示为运行状态，图 14 是 UE 打开后的界面。

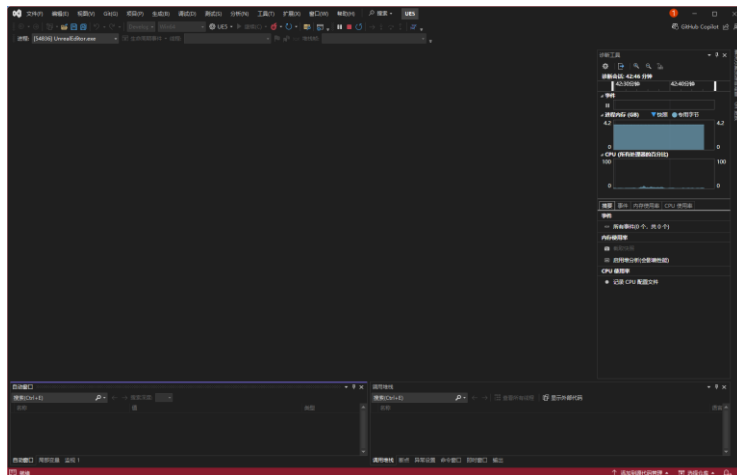


图 13 VS 运行界面

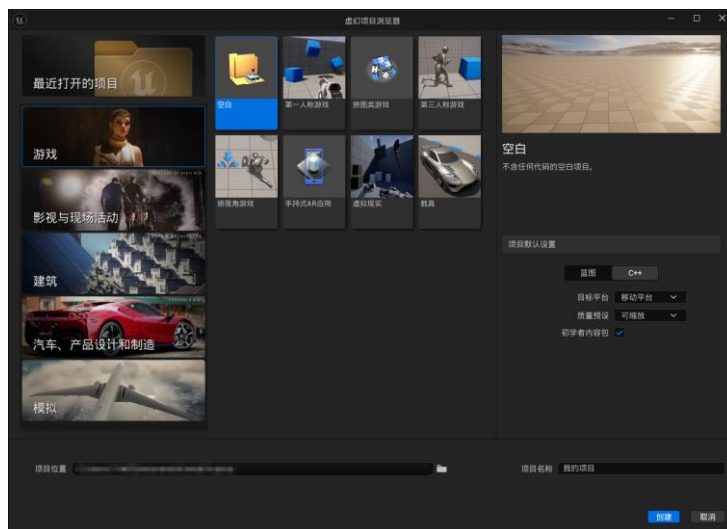


图 14 UE 引擎界面

## 1.2.4 UE5 非 IDE 启动

在项目根目录下寻找如图 15 中 Engine\Binaries\Win64 文件夹内的 UnrealEditor.exe，双击即可启动。

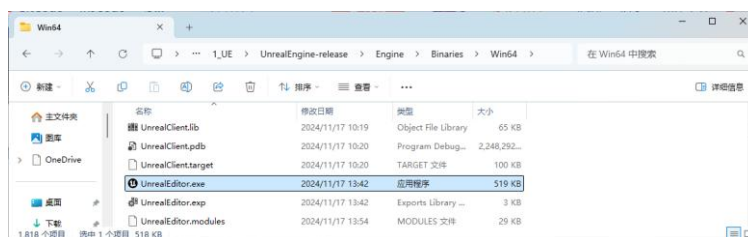


图 15 应用程序启动 UE5

### 1.3 创建 C++工程

在引擎界面设置如图 16，选择 C++工程，并且将目标平台修改为移动端，为了提高后续测试 Android 打包功能的速度，这里将质量预设调整为“可缩放”。最后调整项目位置和项目名称，点击创建。

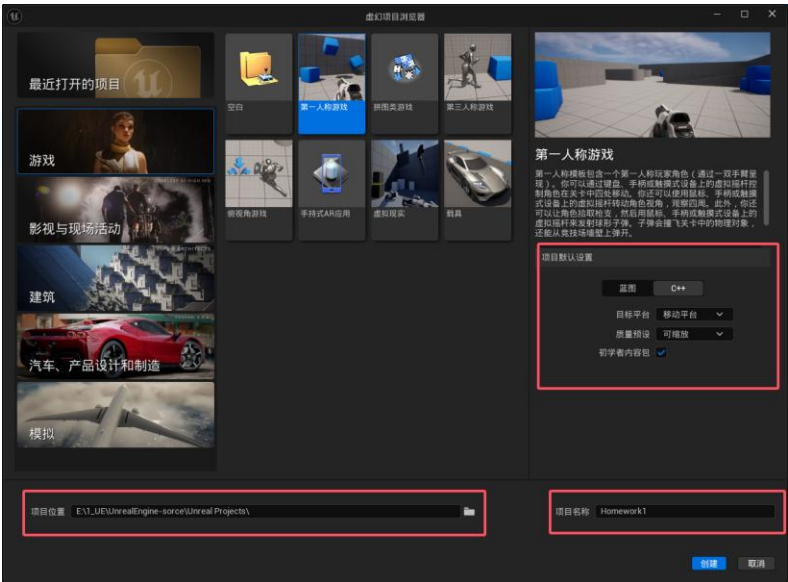


图 16 项目初始设置

创建完成后会出现图 17 的提示，直接点击安装，并选择所有需要的组件，等待安装完成。

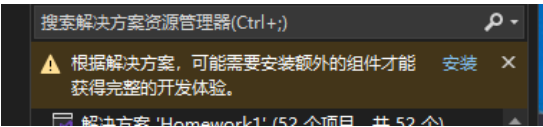


图 17 额外组件安装

安装完相关组件后，将创建的 C++工程设置为启动项，可以按照图 18 依次测试相关的集成配置是否能够实现，如有问题可查看相应的解决方案。

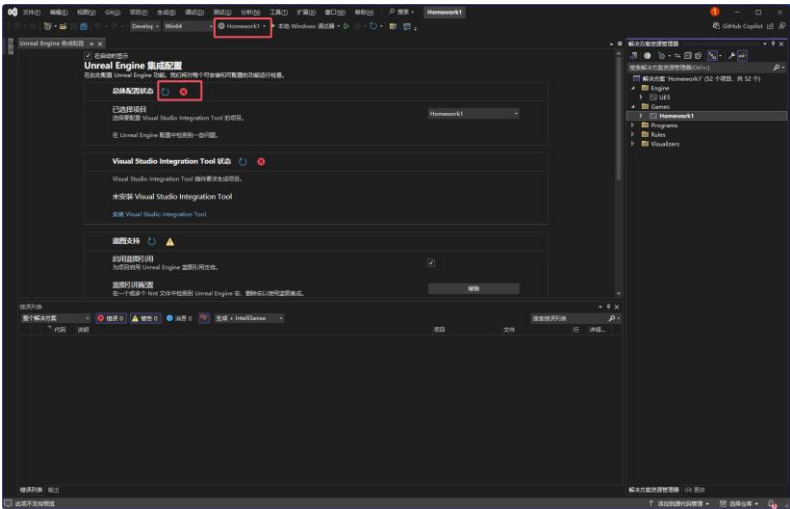


图 18 配置测试



针对图 18 中出现的问题，可以按照图 19 的方式解决：打开 Visual Studio Installer，选择修改，在工作负荷选项中找到 C++游戏开发，勾选 Visual Studio Integration Tool 和 HLSL（其他默认选项不做修改），并选择框中任意一个 SDK，点击右下角的修改。等待修改完成，对 VS 进行更新，重启电脑。



图 19 安装 Visual Studio Integration Tool

重新打开图 16 创建的解决方案在调试中选择开始执行(不调试)运行 C++工程，等待编译完成即可启动 UE 编辑器界面。出现如图 20 的结果即已完成项目的生成。



图 20 打开项目

## 2. Android 平台打包

### 2.1 安装 Android Studio

安装前需要关闭虚幻编辑器和 Epic Games 启动程序。

如图 21 在 [Android Studio 下载文件归档](#) | [Android Developers](#) 中找到 **Android Studio Flamingo | 2022.2.1 Patch 2 May 24, 2023**，并选择对应的操作系统进行下载安装。



推荐使用 Chrome 浏览器，安装时最好使用默认路径“C:\Program Files\Android\Android Studio”。

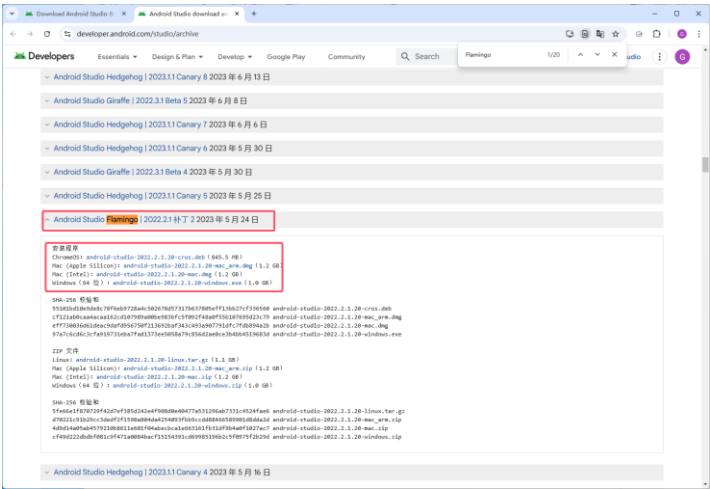


图 20 Android Studio 下载文档

下载完成后，安装 Android Studio，具体执行过程参考：[如何为你的虚幻引擎开发环境设置 Android SDK 和 NDK](#) | [虚幻引擎 5.5 文档](#) | [Epic Developer Community](#) | [Epic Developer Community](#)。

## 2.2 安装时注意事项

### 2.2.1 卸载重装相关

如果安装失败除了卸载 Android Studio 的操作，还需要将“C:\User\用户名\AppData\Roaming\Google”内的如图 21 中的两个文件夹全部删除，防止第二次安装自动读取原有的配置文件。

AndroidStudio2022.2	2024/11/18 18:47	文件夹
consentOptions	2024/11/18 16:37	文件夹

图 21 配置文件

### 2.2.2 组件无法安装

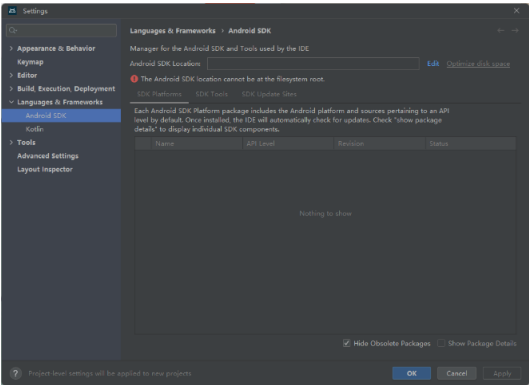


图 22 无法配置 SDK

在进行安装过程中，如果出现“Unable to access Android SDK add on list”不要去设置代理，直接点击取消，继续进行后续步骤。

如果安装完成后，去 SDK 管理器中修改 SDK 相关配置时出现如图 22 没有任何内容的情况，这可能代表安装过程中组件安装失败。此时可以删除图 21 中的两个配置文件，重新打开 Android Studio，选择自定义安装。如果在自定义安装过程中出现图 23 中无法选中组件的情况，那便是组件安装失败。此时可以检查网络，是否能够连接到 Google，并确保自己不是在某一“内网”，必要时可以使用流量进行安装。

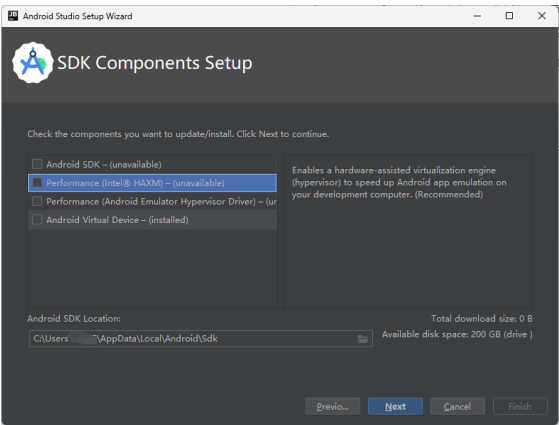


图 23 组件无法安装

## 2.3 安装 Android NDK

打开目录“./Engine/Extras/Android”，运行图 24 中对应系统的脚本安装 Android NDK。（Windows 执行.bat 文件）。

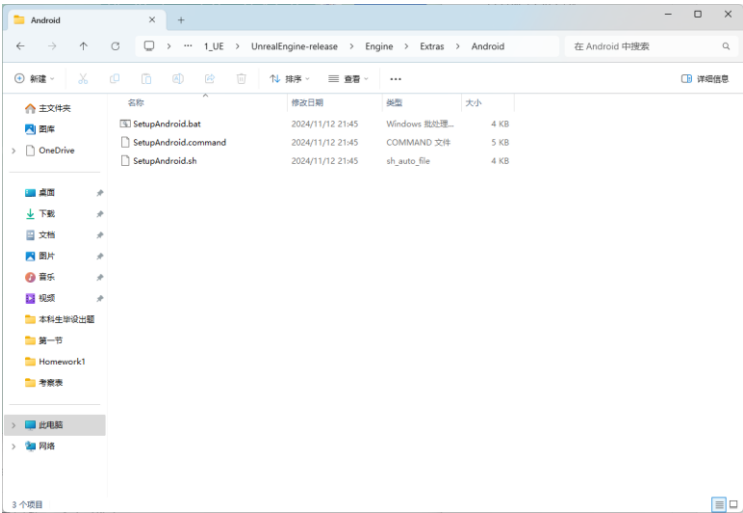


图 24 脚本目录

运行脚本时若出现如图 25 的错误，主要原因是未重置环境变量，需要按照安装文档的要求重新配置环境变量。如：将 JAVA\_HOME 对应的值改为“C:\Program Files\Android\Android Studio\jbr”。

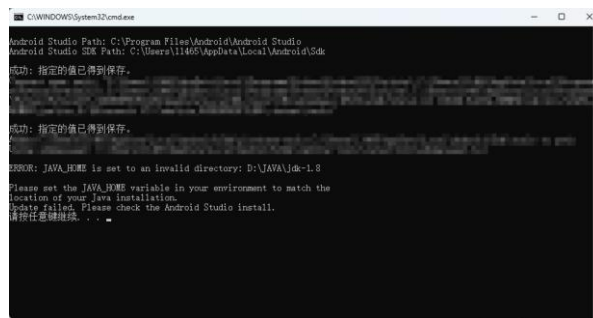


图 25 环境变量错误

如果出现图 26 的错误，代表无法连接到 Google 的相应网页，解决方案是切换非内网环境，必要时可开启流量热点。

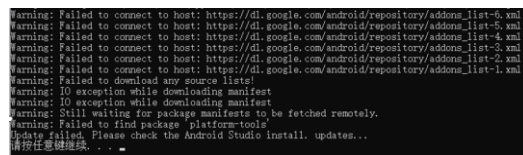


图 26 网络错误

执行成功会得到如图 27 的界面，重启计算机令其生效。其安装目录为“C:/Users/[Username]/AppData/Local/Android/SDK/ndk/”。

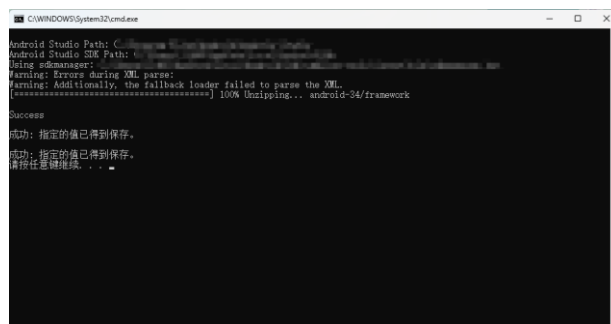


图 27 安装成功

## 2.4 打包项目

如图 28，在左上角点击 平台->Android-> 打包项目，并调整纹理格式。其中 ETC2 为所有设备均支持的项目，在此处选择该格式。

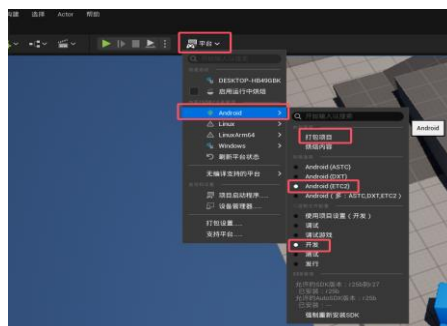


图 28 打包项目

点击打包项目后，开始生成 APK 文件，该文件会存储在项目文件夹中，可

按照图 29 的方式自定义新建文件夹。

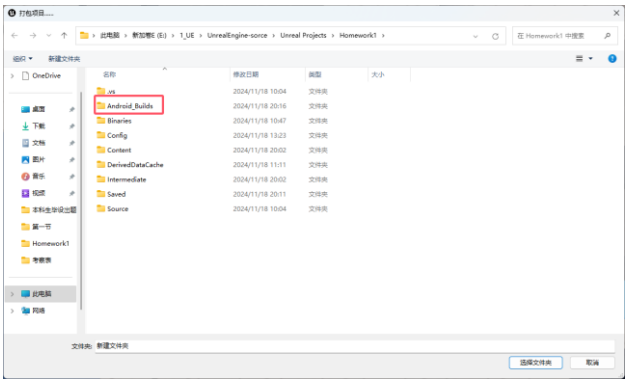


图 29 打包文件地址

当项目右下角出现图 30 的内容几代表项目开始打包。运行完成后此处会提示打包完成。

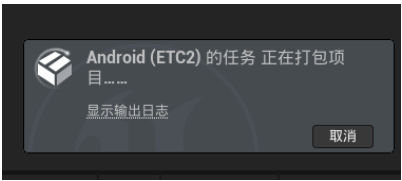


图 30 开始打包项目

点击“显示输出日志”可以查看打包进程如图 31 所示。

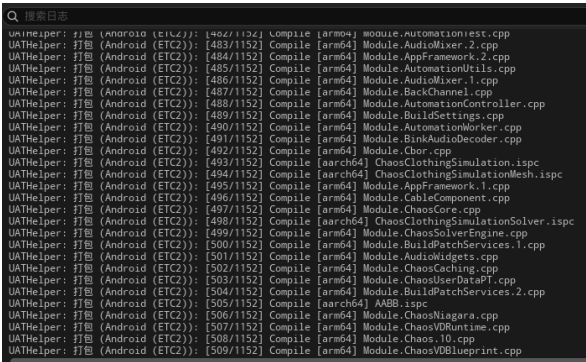


图 31 打包日志

打包过程中出现图 32 的错误，是 gradle 包下载超时。可以终止打包过程，将 gradle-8.7-all.zip 下载到本地，将压缩包存储到如下地址完成 gradle 的下载 “C:\Users\...\gradle\wrapper\dists\gradle-8.7-all\一堆字符串”

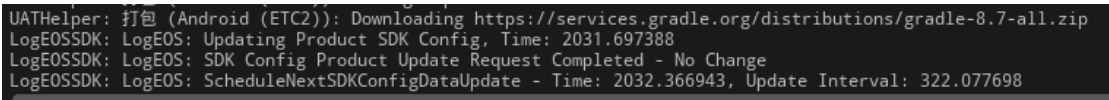


图 32 gradle 下载错误

### 3. Android 手机测试

打包完成后会在 2.4 中设定的文件夹得到如图 33 的内容。

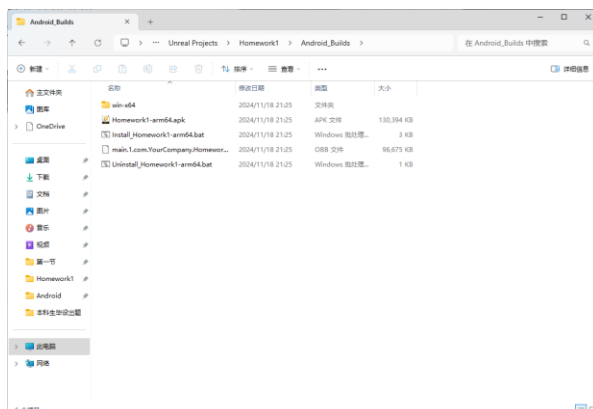


图 33 Android\_Builds 文件目录

将手机连接电脑，并允许电脑修改手机文件，可得到如图 34 所示界面

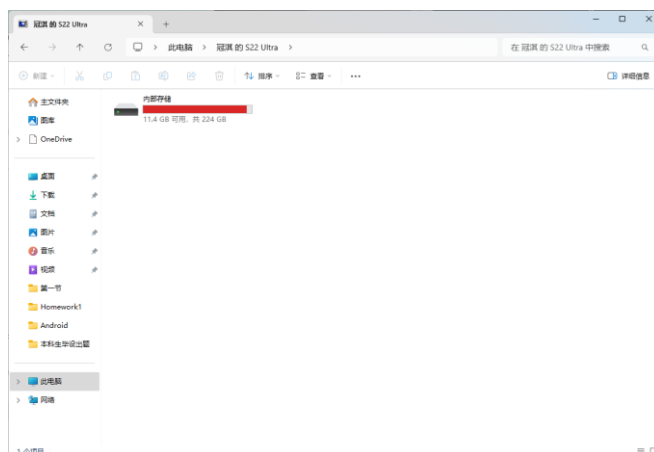


图 34 手机连接电脑

启动手机开发者模式并启用 USB 调试。三星手机的具体操作图 35：在软件信息中多次点击变异编号启动开发者模式 -> 在设置中找到开发者选项 -> 启动 USB 调试

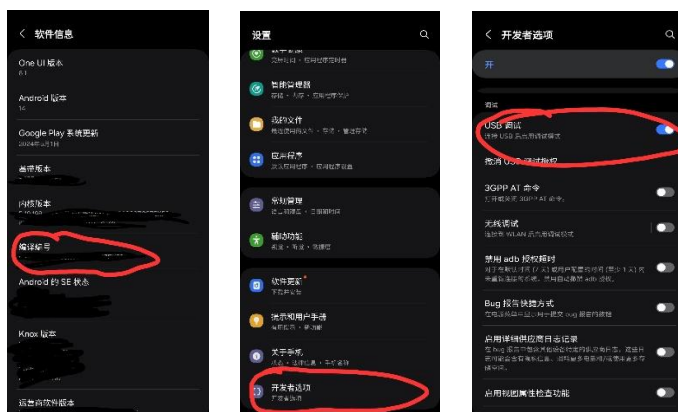


图 35 启动开发者模式与 USB 调试

运行图 36 中的脚本文件，将项目部署到手机上。

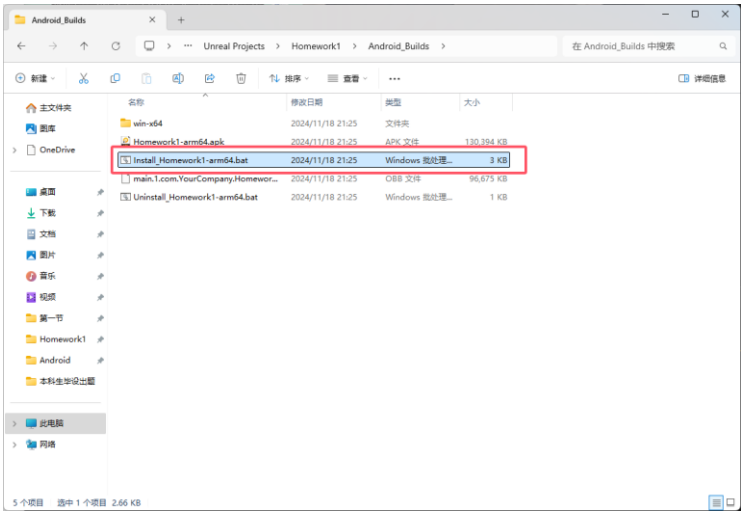


图 37 Android\_Builds 文件目录

图 37 展示成功部署到 Android 平台的截图，并展示运行的画面。

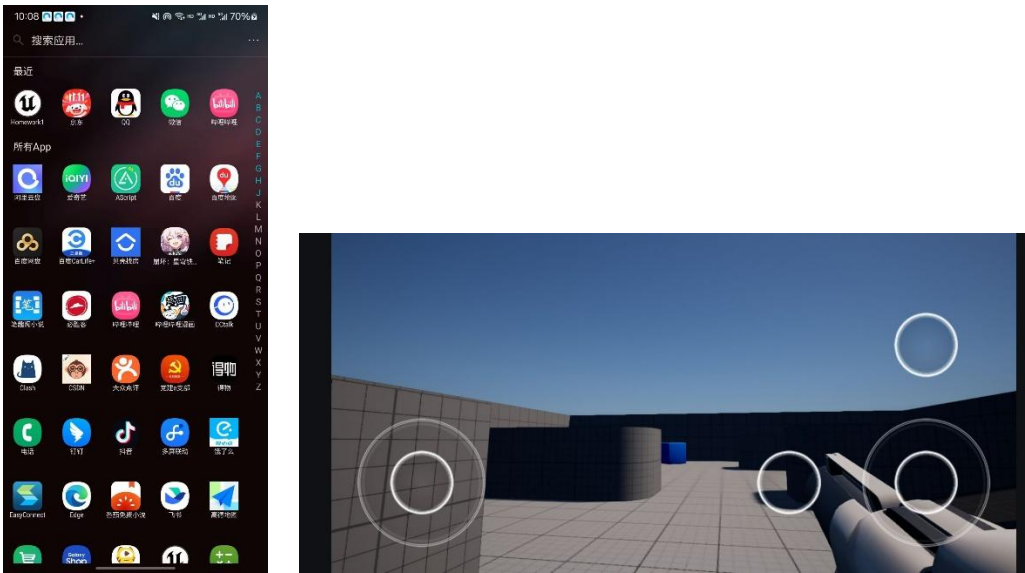


图 37 实际效果