**在虚幻引擎中**

**使用git进行**

**版本控制**

**和**

**多人协作**

同乐技术

**目 录**

[参考文档 - 2 -](#_Toc28682)

[设备支持 - 2 -](#_Toc14597)

[Git部分 - 4 -](#_Toc2575)

[安装git - 4 -](#_Toc14129)

[安装git lfs - 4 -](#_Toc8832)

[GitHub - 4 -](#_Toc20000)

[GitHubDesktop - 4 -](#_Toc29060)

[UE本地Git仓库 - 4 -](#_Toc1983)

[Github连接到本地的UE仓库 - 5 -](#_Toc10825)

[多人协作部分 - 7 -](#_Toc8106)

[邀请团队成员 - 7 -](#_Toc17020)

[虚幻工程专用配置 - 7 -](#_Toc17235)

[日常工作流程 - 7 -](#_Toc28249)

[团队协作 - 8 -](#_Toc26704)

[Git插件 - 8 -](#_Toc15938)

[测试文件锁定功能 - 9 -](#_Toc4634)

[分支 - 10 -](#_Toc29180)

[新的合作人员 - 10 -](#_Toc27332)

[锁文件功能测试 - 11 -](#_Toc30479)

[对于新项目的初始化 - 13 -](#_Toc28279)

## 参考文档

<https://www.bilibili.com/video/BV1riKczuEeZ/>

## 设备支持

因为是多人协作，所以需要两台电脑

这里我使用10875一台，作为主要的仓库管理者，命名为Tongle，另外一台12600作为项目的参与人员，命名为Lizzy。

# Git部分

## 安装git

Git客户端：<https://git-scm.com/downloads>

完成后配置git

Git config --global user.name “LaughTogether”

Git config --global user.email “LoongLizzy@gmail.com”

## 安装git lfs

<https://git-lfs.com/>

安装完后，打开GitBash，输入命令git lfs install

## GitHub

注册github账号

我已经有个人账号了，所以注册一个属于企业的个人账号，并将账号登录至10875设备上

账号：

[LoongLizzy@gmail.com](mailto:LoongLizzy@gmail.com)

LizzyLoong@github123

Github提供了10GBLFS存储空间和每月10GB月度流量配额

## GitHubDesktop

下载GithubDesktop

在GithubDesktop上登陆账号

## UE本地Git仓库

将所有内容与UE结合起来

本教程将使用C++。C++项目与蓝图项目内的资源有些不同。

首先创建C++项目，然后在项目的版本控制中选择Git

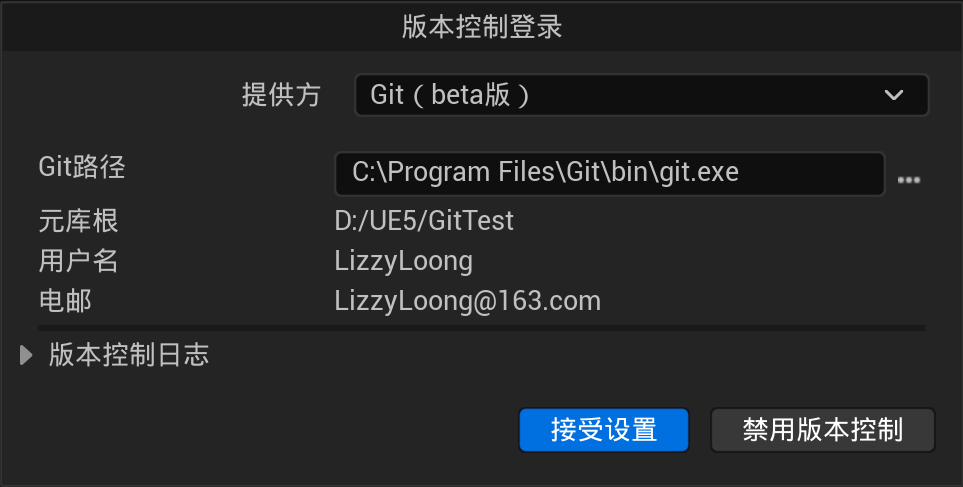
所有全勾上

然后不要点击接受设置，而是点击“用Git初始化项目”



然后就会自动将文件添加到版本控制

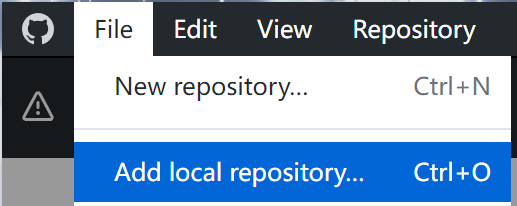
完成后确认Git用户名和邮箱



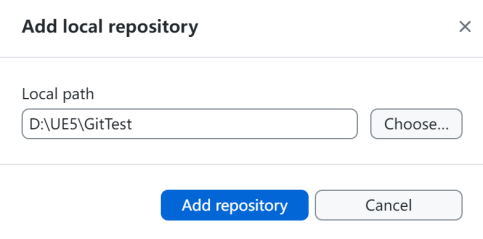
确定后点击接受设置

## Github连接到本地的UE仓库

接下来打开GithubDesktop，添加本地仓库

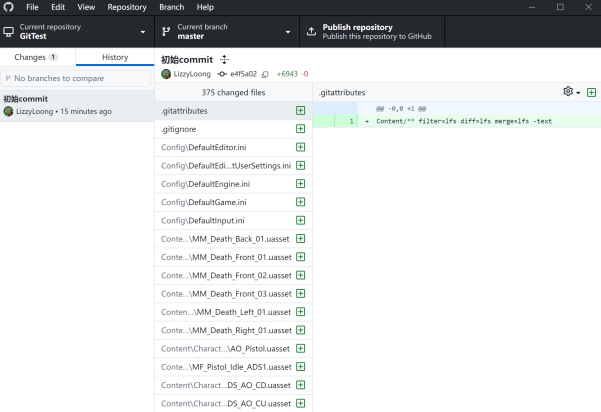
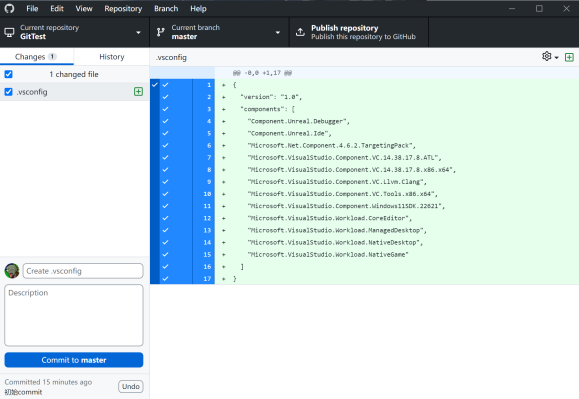


选择项目文件夹



然后可能会让初始化LFS

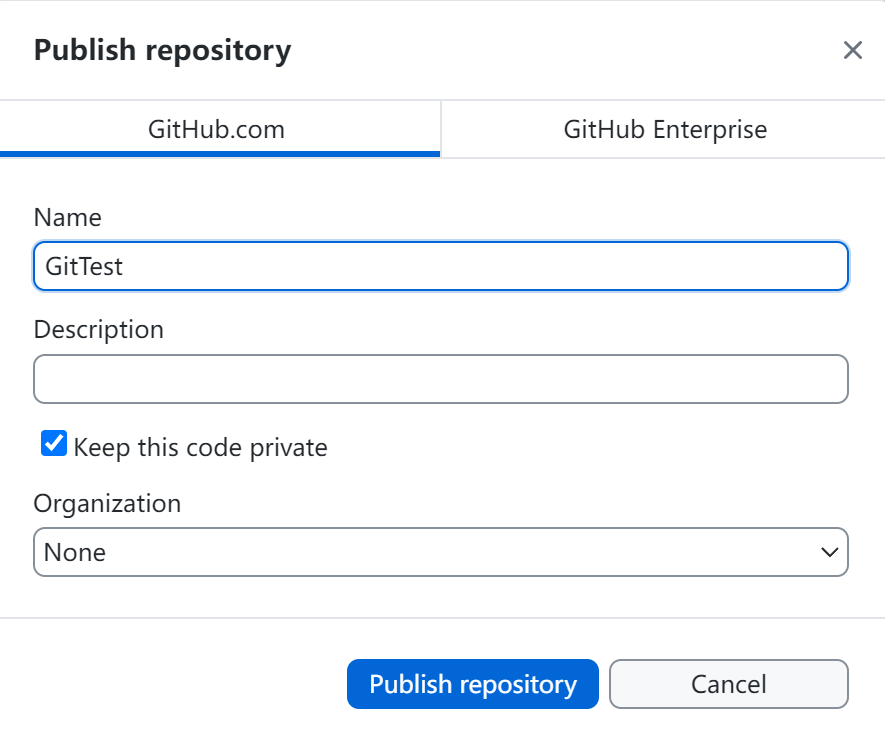
完成后就会看到初始提交



两个图片的区别是左侧的Changes和History

这个初始提交包含了虚幻引擎项目和Git设置，但目前所有内容都存储在本地

要将仓库发布到网上，点击Publish repository



给仓库起个名字和可选描述

这里要注意这个仓库是私有的还是公开的，私有仓库意味着只有受邀协作者才能访问

仓库也要注意是放在哪个组织下面，如果是放在组织下面，个人仓库里是找不到的，必须要到自制仓库里才能找到。

然后这个仓库就会出现在GitHub账户或组织的仓库列表中

（不要点击左侧的commit！！！！！！！！！！更不要点击push！！！！！！！！！！！等着就行）

# 多人协作部分

## 邀请团队成员

在GithubWeb中，在仓库的Settings中找到Collaborators

上述的设置中，我们选择将仓库设置成组织仓库，且是私有。

此时，Tongle用户自然可以全条件的控制这个仓库

但是对于LizzyLoong，并不是主人。所以需要在Collaborators里面设置一下其他参与人员的权限。

## 虚幻工程专用配置

UE项目.gitignore模板：<https://github.com/ProjectBorealis/PBCore/blob/main/.gitignore>

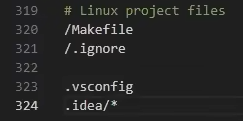
UE专用.gitattributes：<https://github.com/ProjectBorealis/PBCore/blob/main/.gitattributes>

我们特别的需要将文件加上

在把文件下载下来后，直接复制文件里的内容再粘贴到项目的对应文件中即可。覆盖粘贴。

对于gitignore，因为使用的是rider，所以也要添加.idea文件夹和.vsconfig文件夹

在gitignore文件粘贴结果的最后面输入两行



修改完GitIgnore后，在GithubDesktop中就不会显示那些不需要的文件了。

然后再看GithubDesktop，就可以Comment一下Commit，来把改动都保存到本地

然后PushOrigin，来同步到github

## 日常工作流程

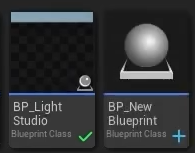
当团队协作时，保持同步更新很重要。这个时候就需要git pull

他会拉取最新版本，包括队友做的所有修改

在UE中，当修改或创建新蓝图时，可以看到可视化提示：

被Git LFS追踪的文件会显示一个勾选标记

将要加入Git LFS的新文件会显示一个加号标记



要直接在编辑器里提交这些改动，去“版本控制”选择“提交内容”

然后写个改动信息的描述然后提交就行

但是注意，这只是在本地提交了文件。如果想把这些改动上传到GitHub在线仓库中，需要打开GithubDesktop手动推送



额外的，如果使用C++代码，无论是添加新类还是修改现有类，UE的Git插件都不会自动检测这些改动。此时需要通过GithubDesktop来处理这些变更。

总上：完整的工作流程为：

**提交工作到本地->将其推动到远程服务器->定期拉取更新保持同步**

按照这些简单的步骤，能减少许多后续的诸多麻烦

## 团队协作

需要有一些额外的设置来保证顺利的工作

假设我们和另一个团队成员共同开发项目，如Tongle和LizzyLoong协作

现在Tongle打开一个蓝图，添加了一个简单的Print，提交并推送

然后LizzyLoong还没有拉取最新更新，打开了同一个蓝图并做修改。此时由Lizzy尝试由UE来提交这些改动时，表面上看起来一切正常，但是当在GithubDesktop中尝试push时，Git阻止了Lizzy。因为其他成员已经抢先推送了修改，此时Lizzy需要先拉取（pull）其他人的变更。此时的问题是：两个人同时修改了一个蓝图文件，而蓝图是二进制文件，git不能合并二进制文件，所以产生了合并冲突。遗憾的是，此时必然会有一方的改动被覆盖。

解决办法就是文件锁定功能，来保证在同一时间，只有一个人能编辑特定文件。

启用文件锁定后：

1. 系统会提示其他用户该文件已被占用
2. 当远程仓库存在更新是也会发出警告

但UE的默认Git插件并不原生支持文件锁定。这里我们就要用到UEGitPlugin，一个Github上的社区插件，提供了完整的GitLFS支持。

## Git插件

虚幻Git插件：<https://github.com/ProjectBorealis/UEGitPlugin>

这个插件的重要作用是加上文件锁。

安装方式有2种：集成到引擎全局、仅限当前项目使用

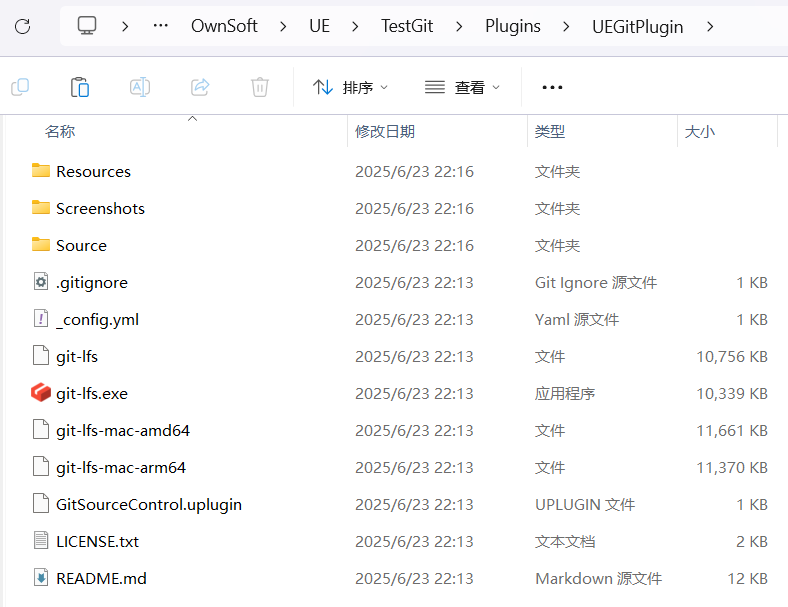
项目级安装步骤：

打开上述github网站后下载zip压缩包

解压后找到项目目录，打开或创建Plugins文件夹

在Plugins文件夹内创建文件夹UEGitPlugin

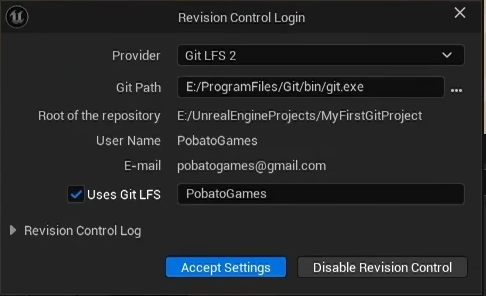
把解压后的文件们放到这个新的文件夹内



注意关注图中的路径，TestGit是项目，uproject文件就在TestGit文件夹内

然后重启UE，重启后再次进入版本控制设置

这次选择GitLFS2选项，此时会要求输入Git LFS专用用户名



务必保持该用户名与默认Git用户名保持一致。如果两者不同可能导致文件锁定失效。

接受设置即可（然后可能需要登陆）

接下来更新.gitattribute文件

在插件的github页面提供了完整版本

<https://github.com/ProjectBorealis/PBCore/blob/main/.gitattributes>

直接复制替换

然后在GithubDesktop中提交并推动变更（插件和attributes的变更），这样团队成员也能获得正确配置。

## 测试文件锁定功能

当Tongle打开一个蓝图并修改其中的内容并保存时，编辑器会要求先签出该文件（Check out）。在版本控制术语中，签出文件意味着声明你要编辑他，让其他人知道这个文件正在被修改。

签出文件时，系统会为其他人记录锁定状态。

此时在Lizzy这边看，就能看到该蓝图当前被Tongle签出，明确告知该文件正在被其他人编辑。

如果首次修改时没有提示签出，可以在资源浏览器中右键该文件选择“ReversionControl--RevertWirtableFile”（版本控制--恢复可写文件）。因为有时新增文件可能存在Bug，即使提交后扔被标记为可写状态。

现在Tongle完成修改，在版本控制中选择“签出已修改文件”，通过提交操作来签入文件。签入意味着释放文件锁定，使其可供他人再次签出。

UEGit插件会自动Push这次提交，我们不需要手动在GithubDesktop上操作。

当Lizzy回到编辑器时，蓝图文件上会出现黄色提示图标，表示远程仓库有更新版本。只需要打开版本控制菜单，里面有Pull选项，就能下载最新版本消除警告，还能确保我们始终基于最新文件工作。

## 分支

当前的配置支持在单一分支流畅协作。

Git中的分支就像是项目的平行版本，允许开发新功能或实验性改动，而不影响主代码库

要创建新分支，在GithubDesktop中前往分支选项，选择创建分支



创建新分支后发布分支以同步到远程服务器

但是UEGitPlugin存在一个限制：无法自动检测其他分支的文件更新

要解决这个问题，需要配置状态分支（StatusBranches）。这个进阶功能的设置方法参考插件的git页面

## 新的合作人员

当我们有了一个新的合作人员，怎么将其拉入到我们的项目中呢？

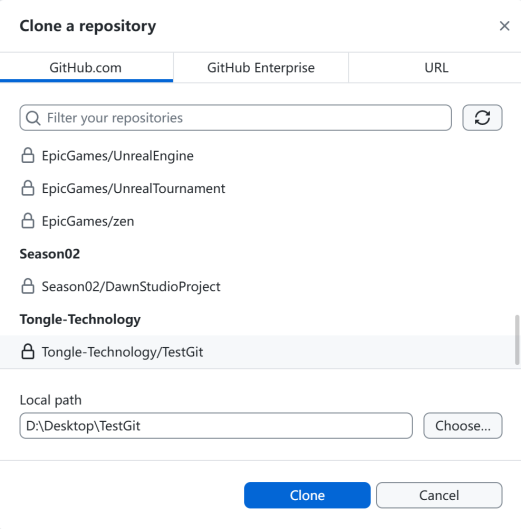
实际上，上文只说了Lizzy在Tongle的项目中的事情，并没有说Lizzy怎么加入到这个项目。

所以从Lizzy的角度来说，他需要先加入进去。

现在在Tongle的10875那边已经实现了上述所有内容，唯一差的就是让Lizzy真的进入项目。

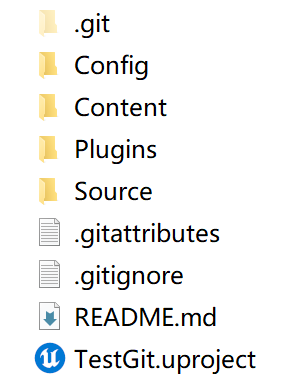
首先，将Lizzy拉入到组织中。项目工程仓库存放在组织里。因为项目是Private，所以需要修改一下设置

在Lizzy的12600中打开GithubDesktop，克隆这个组织的仓库

（GitBash应该也行）

克隆完成后会要求InitGitLFS，初始化就行

克隆完后如图



现在先不打开UE，先用GithubDesktop试一下Push，看能不能推送过去。因为是私有仓库，可能Lizzy没有写的权限。

如果能Push过去，则执行后面的操作。如果不能，则尝试解决且必须解决。可能的问题原因：private仓库的权限问题。

上一步验证完成后打开工程项目

现在还什么都没有，我们需要先编译

双击uproject就行，会让你Rebuild的

然后就会打开项目

打开之后先设置版本控制，选择Git LFS2

（这套设置比较吃网络）

理论上，很顺利

## 锁文件功能测试

当我在Lizzy这里打开一个蓝图时，即使没有改动，就一个Ctrl S，也会触发签出，然后锁文件。

这一瞬间，在Tongle那边是没有反映的。可以点击版本控制选择“查看变更”来刷新。但实际基本都是实时的。

此时在Tongle这里能打开这个蓝图，但是保存的时候有警告。

然后签入一下。

如果版本控制菜单中能选择“签出已修改文件”，就选它。否则的话，就右键选择这个签出的蓝图文件，选择“版本控制--检入”也是签入的意思。

然后在Lizzy这边蓝图没有标记了，但是在Tongle这边，这个蓝图会显示黄色的向下箭头，表示远程仓库有更新版本。只需要打开版本控制菜单，里面有Pull选项，就能下载最新版本消除警告，还能确保我们始终基于最新文件工作。

测试完美，Lizzy给蓝图中定义的新变量也在Tongle这边实现了

# 对于新项目的初始化

上述内容以讲解为主，这里我们集中关注对于一个新项目该如何配置这些git环境

我们直接在新项目中使用UEGitPlugin