

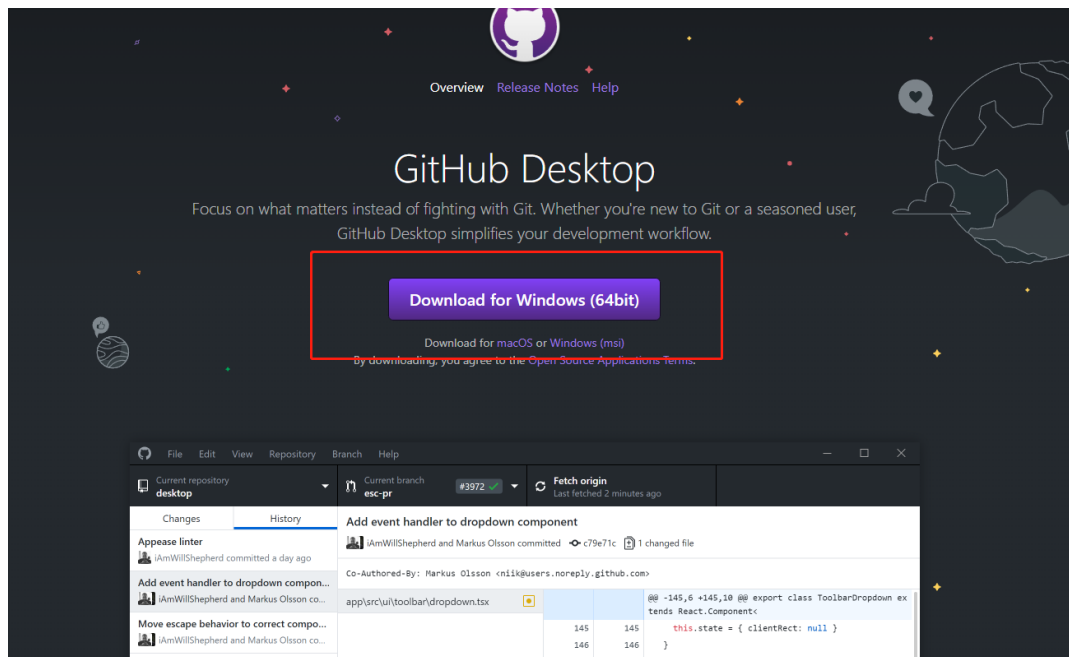
如何快乐地使用Github进行协同工作？

啥是Git/Github？

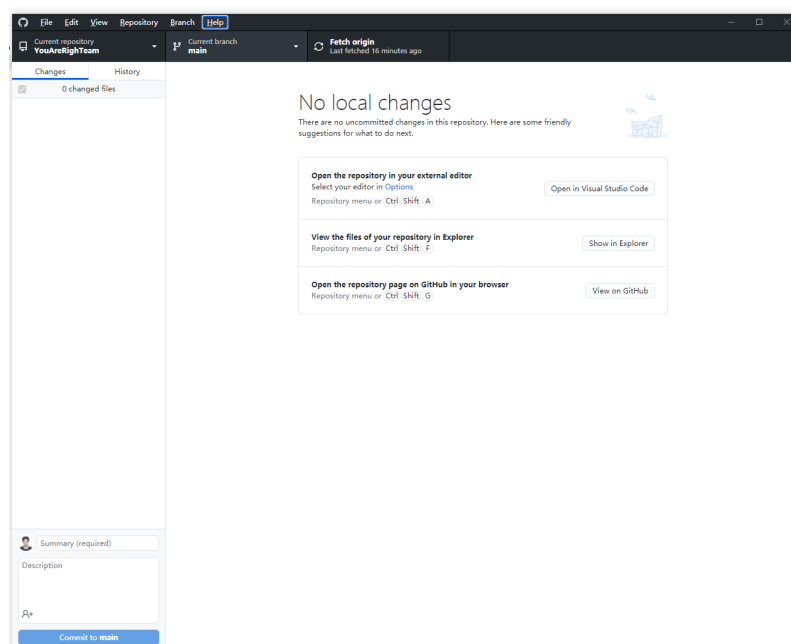
Git 是常用的版本控制系统 (Version Control System)，用于项目(Git 的基本单位，aka Repository)的版本控制。所谓版本控制，简单来说就是指我们的项目如果做到一定程度感觉做的不好了，我们可以回滚我们的项目重新做。同时版本控制软件就如同Icloud一样可以将我们任何人的工作成果同步到别人那里，这回让我们的协作非常高效。如果又觉得不好了甚至可以回到第一次我们绝的项目不好的点继续做。而Github就是基于这种版本控制系统的一个全球开源共享平台。你可以理解为程序员的Artstation或者程序员的SoundCloud。

1. 如何使用Github？

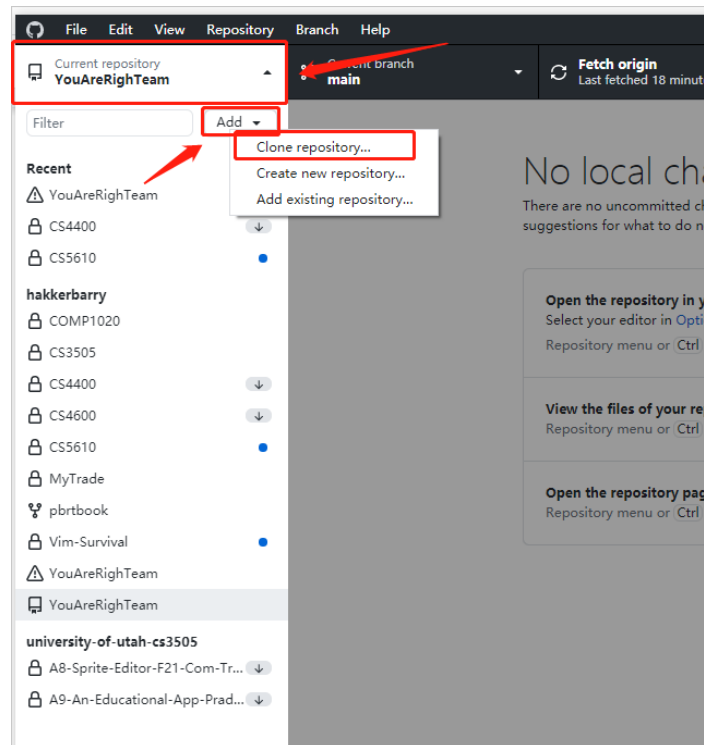
- 如果您是从事美术相关专业的，那么您可能并不喜欢命令行工具进行工作。没有关系, github拥有图形化的工具供您使用：<https://desktop.github.com/>



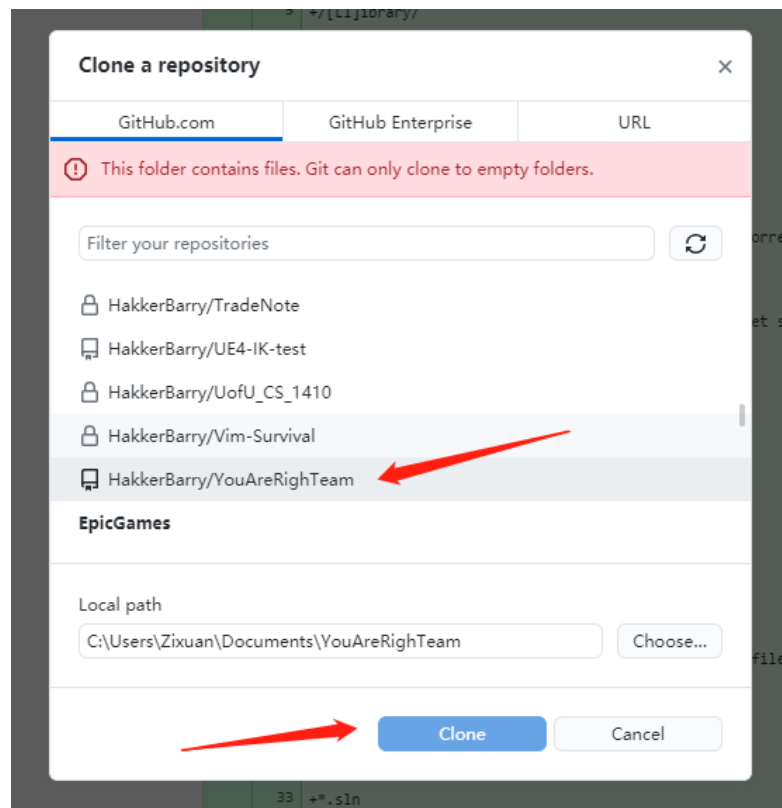
- 点击下载并安装该图形化工具并打开。您会看到这样的界面：



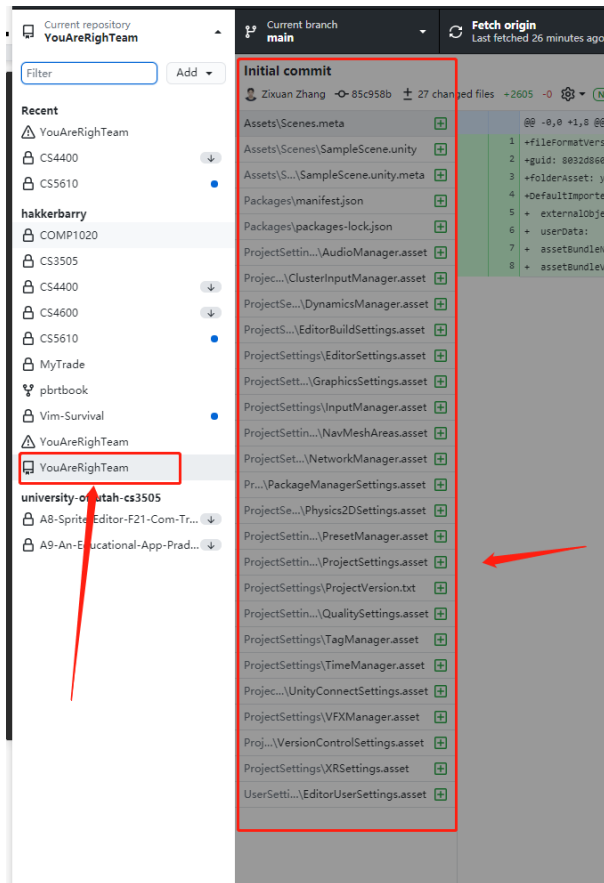
- c. 登录您的账号，您会获得您账号对应仓库的权限。
- d. 点击左上角最大的按钮您会打开仓库菜单，您的这里可能还是空的但不要紧，我们选择Add-Clone repository（添加，克隆仓库）



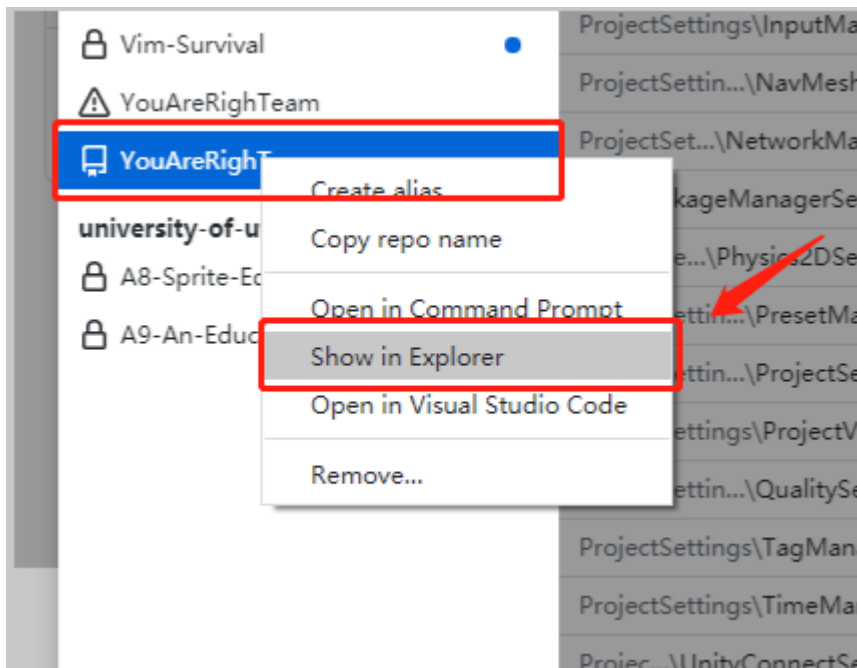
- e. 在弹出的窗口中，选择我们的项目仓库并点击克隆，我这里因为已经克隆过了所以文件夹重名冲突。



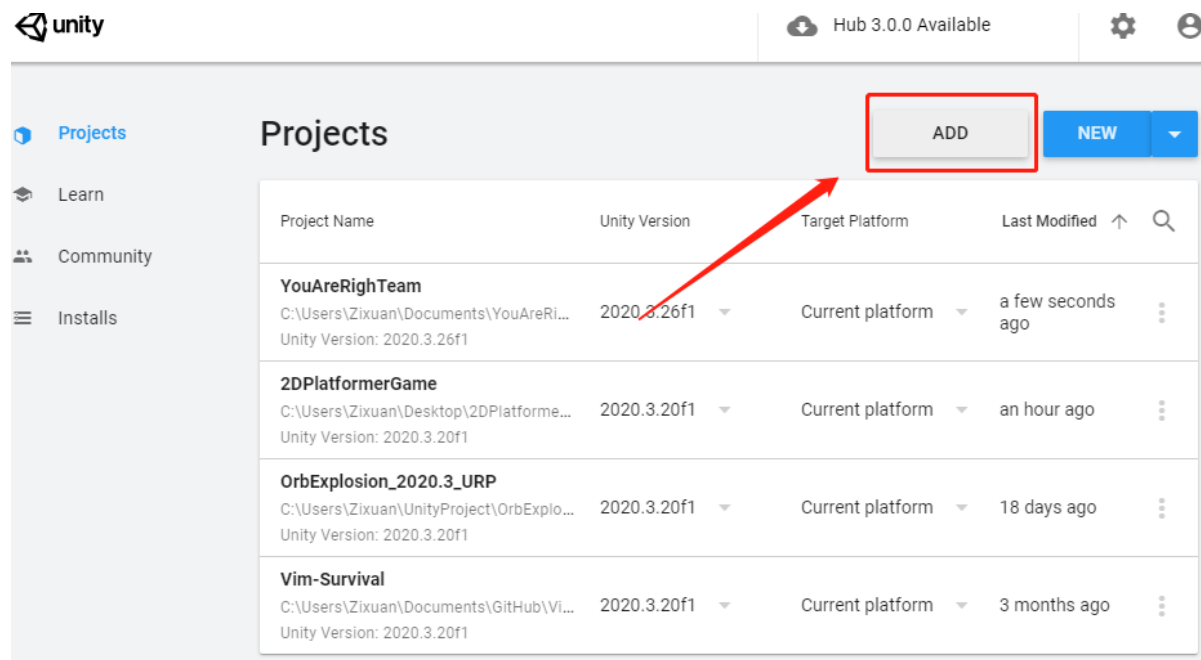
- f. 等待克隆完毕你应该已经有了我们目前的所有工作：



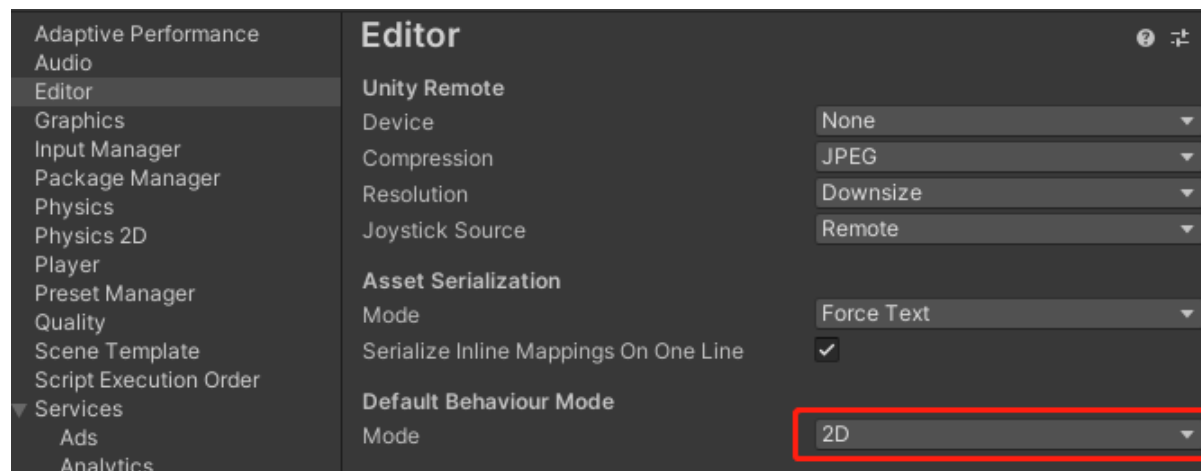
g. 右击我们的仓库，打开对应的文件夹，您应当可以浏览到我们当前的所有内容



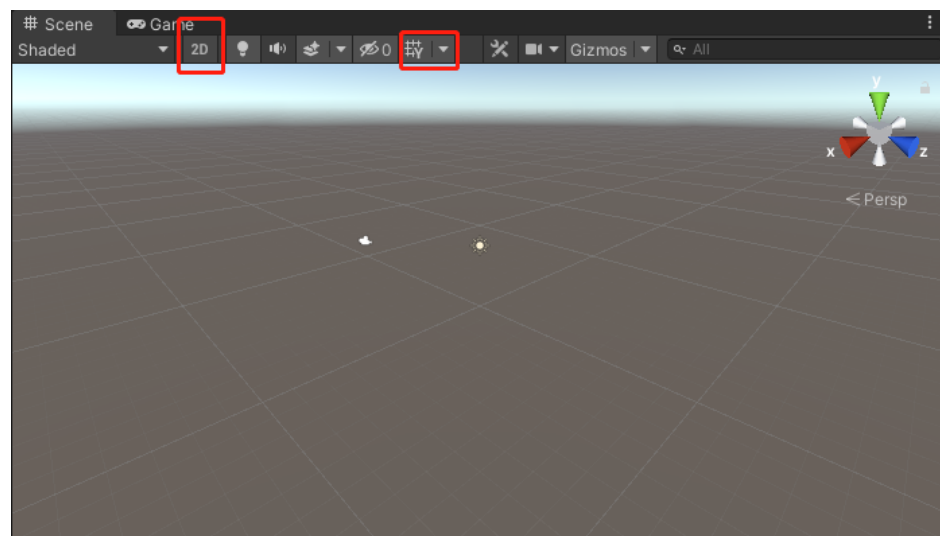
h. 使用UnityHub打开所对应的文件夹，您就应当使用Unity打开了我们的项目。



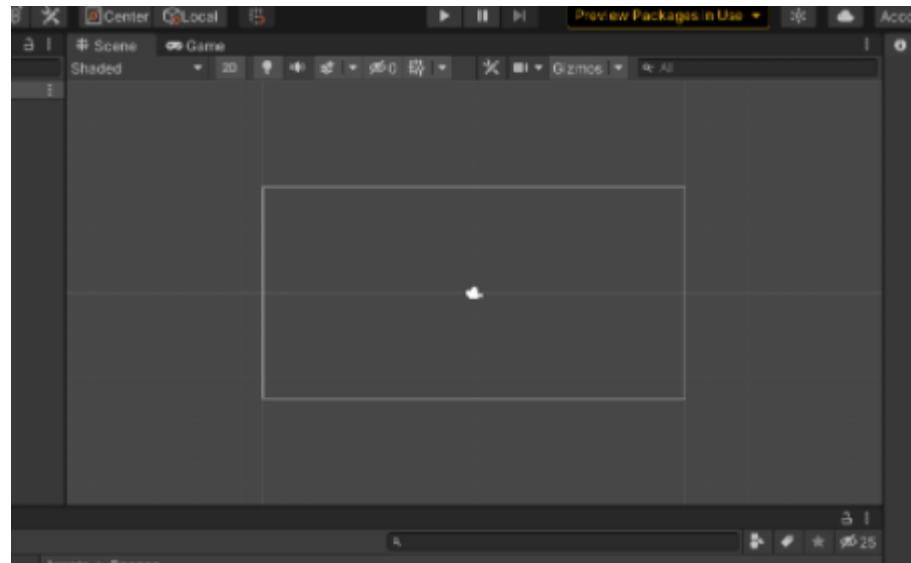
- i. 打开Unity 并 打开Edit-> Project Settings -> editor 将默认行为模式设置为 2D



- j. 开启场景中的2D选项，开启网格，您的场景最终应是如2D的默认项目场景一致。



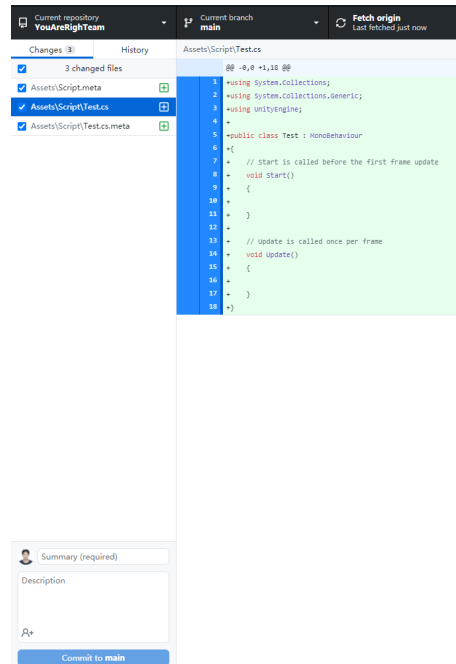
k. 最终效果如下



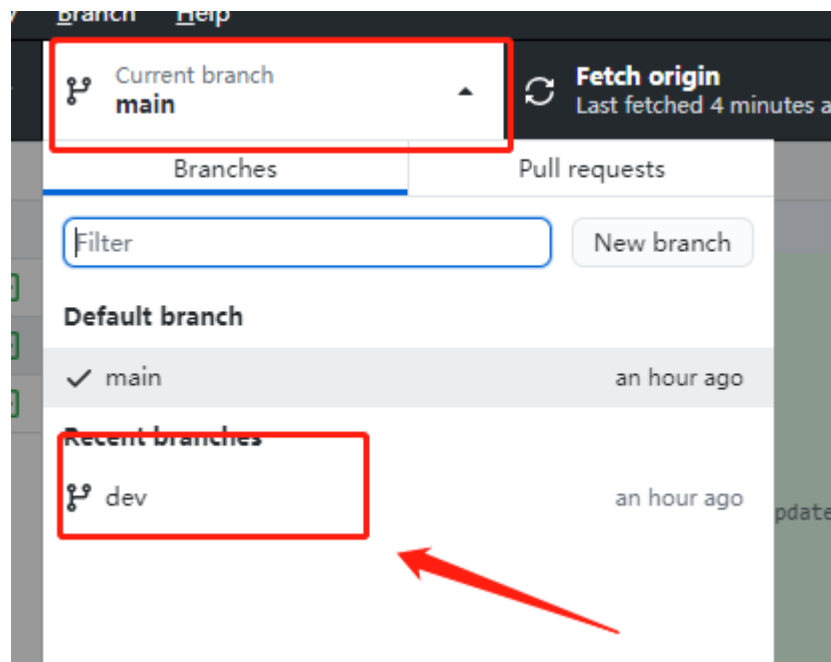
2. 如何使用Github Desktop？

- a. 这里我们必须介绍几个基础的Github概念以便您理解版本控制器的工作方式
- b. 分支 Branch, 当我们需要不同的功能时，多人编辑同一个文件可能产生冲突，这会让管理项目的人做大量无用的检查。于是我们有了分支，不同的功能在不同的分支中进行开发，在需要合并的时候合并之后再开发。您可以理解为不同的文件夹，一个存代码，一个存图片，我们需要图片的时候将图片拖拽到我们的主项目文件夹去。
- c. 提交 Commit, 这个单词看着有些复杂，但事实上您可以理解为保存。将您手上的文件保存起来以便于您做错了可以打开您保存的文件重新做。每一次Commit相当于您另存为了一个新的工作文件。比如现在您存了Work1.ps 那么当您Commit您就约等于保存了Work2.ps这样无论您什么时候需要您之前的文件都可以简单拿出来用。需要注意的是Commit是基于分支存在的，好比您有好多不同的文件夹每一次都是不同保存的。您只在您的分支中工作，好比您仅仅保存文件到指定的文件夹。
- d. 推送 Push, 就是同步，将您本地的更改推送到远端，好比您将整个项目拷贝到百度网盘。让我们所有人都能下载。
- e. 合并 Merge, 这个功能如果您并非程序那么您不用操作，但理解这个概念有助于您使用Github, 这好比我们保存不同的文件到不同的文件夹，最终统统复制粘贴到我们的Unity项目中去一样，这样主项目不会因为过多的临时文件而混乱同时也更加利于您方便地独立工作。

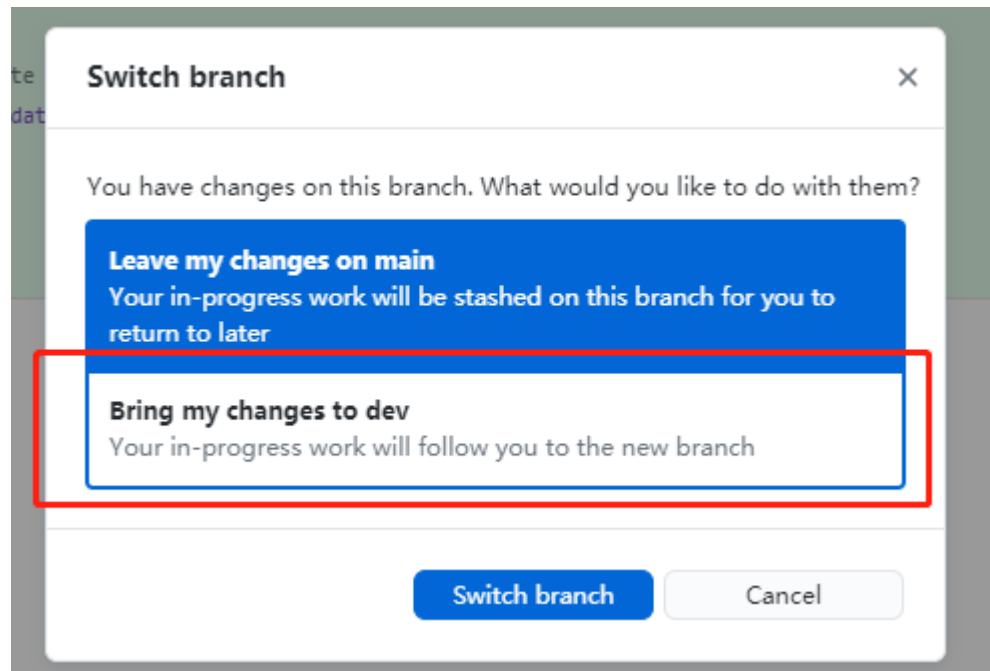
f. 使用，当您对项目做出了更改后，您的Github Desktop应如下所示。



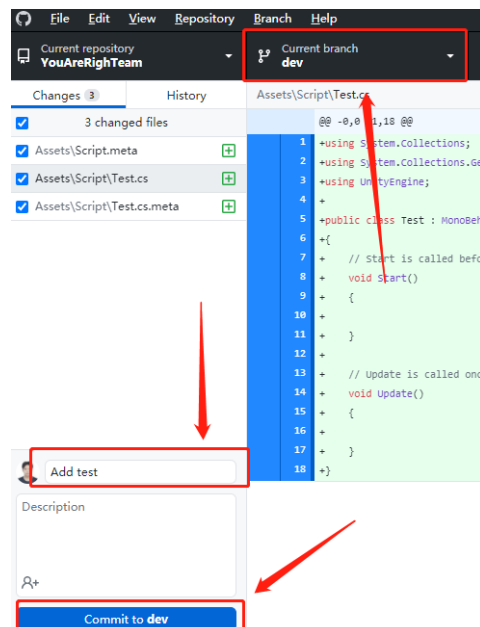
- g. 我们接下来就是要保存（应当是提交Commit）您刚刚完成的工作。
- h. 点开分支菜单，对勾地地方表示了当前所在的分支，我们目前处在main分支里即主分支。（以前叫master分支傻逼们说这个词歧视黑人搞得现在很多仓库主分支混乱）



- i. 我是程序，这里我选择dev (development开发) 分支，如果您是美术那么您的分支应是 assets (资产)。点击后，如果您之前做了更改您会有一个对话框询问是否切带着您的更改到新分支。



- j. 我们选择是的，携带我们的更改到新的分支，原则上我们不允许更改别的组别的分支，或是随意改动主分支。
- k. 确认分支是您当前工作的分支之后我们就可以Commit提交了。提交的时候需要写注释您本次提交做了些什么，比如我这次仅仅添加了一个演示用的测试文件，便写Add test 添加测试。您完全可以使用中文。



- l. 点击提交后，您右上角同步按钮会更新，点击这个publish branch 您的工作便上传到Github上了！这样我们都可以很快地下载下来。



