

目录

在写这篇教程之前，我想说的话.....	1
1. 了解 Git 和 Github.....	2
1.1 先了解几个基本概念	2
1.2 github 几个主页面介绍.....	2
1.2.1 如何进入各个主页.....	2
1.2.2 Github 主页.....	3
1.2.3 仓库主页	4
1.2.4 个人主页	4
1.3 fork 和 pull request 两个重要操作.....	4
1.3.1 复制克隆项目（Fork）	4
1.3.2 发起请求（Pull Request）	5
1.3.3 开源项目贡献流程.....	5
2. Git 的使用.....	6
2.1 首先学着创建一个新仓库.....	6
2.2 第一种：拖拽法	7
2.3 第二种：http 上传（Git）（懒人推荐）	8
2.4 第三种：ssh 上传（Git）（rsa 加密）	9
3. 最后	15

在写这篇教程之前，我想说的话

作为一名老队员，我深知技术传承的困难和重要。

对于标准平台组来说，时间还会浪费在前人已经弄懂的 API 甚至是有关代码格式的问题上，这对我们去深入开发是非常不利的。

对于服务机器人组来说，更需要有一个代码托管，版本控制的系统来保证我们比赛所使用的工作空间可以继续被借鉴使用，从而越做越好。

但往往由于这些问题，我们无法安心地站在学长学姐的肩膀上继续前进，于是我推荐大家尽快入坑 Github。

在描述具体操作之前，我看过了很多 b 站 up 主的视频，以及网上的各种 github 博客，并针对我们活动队的情况写下了此教程，如果各位同学耐心仔细阅读，一定会有收获的。

1. 了解 Git 和 Github

Github 是全球最大的社交编程和代码托管网站 <https://github.com/>;

Git 是一个免费、开源的版本控制软件;

Github 这个代码托管平台借助 Git 来管理项目代码;

版本控制系统是一种记录一个或若干个文件内容变化,以便将来查阅特定版本修订情况的系统。

1.1 先了解几个基本概念

仓库 (Repository)

仓库用来存放项目代码,每个项目对应一个仓库,多个开源项目则有多个仓库

收藏 (Star)

收藏项目方便下次查看(对于 github 上的开源仓库,有一百个 star 已经非常非常厉害了)

复制克隆项目 (Fork)

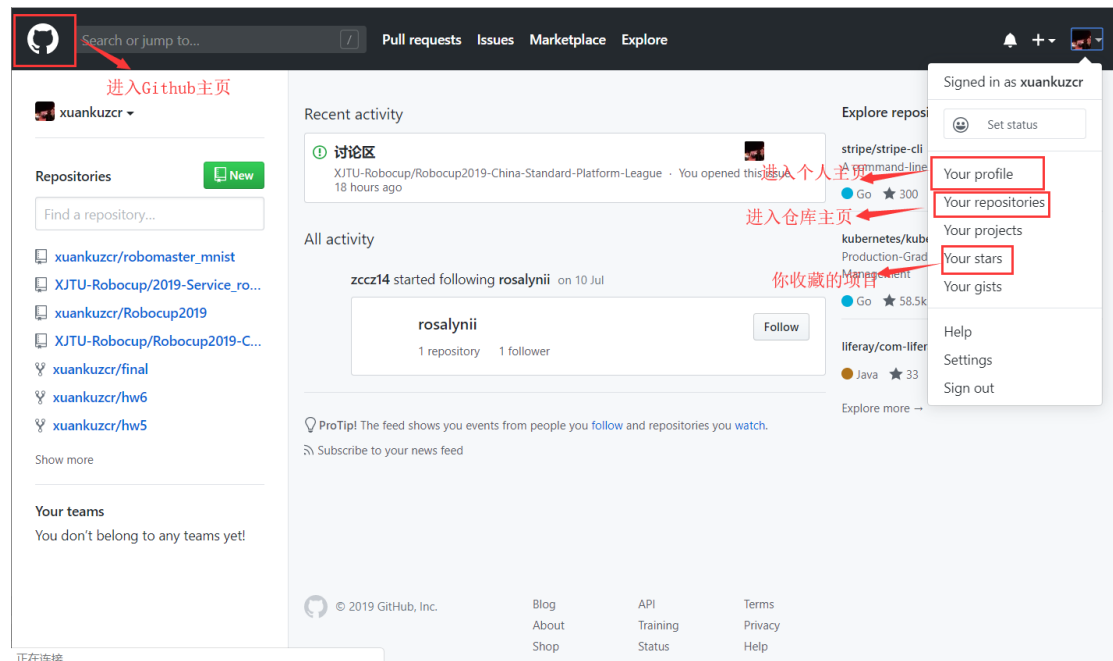
发起请求 (Pull Request)

关注 (Watch)

事务卡片 (Issue)

1.2 github 几个主页面介绍

1.2.1 如何进入各个主页

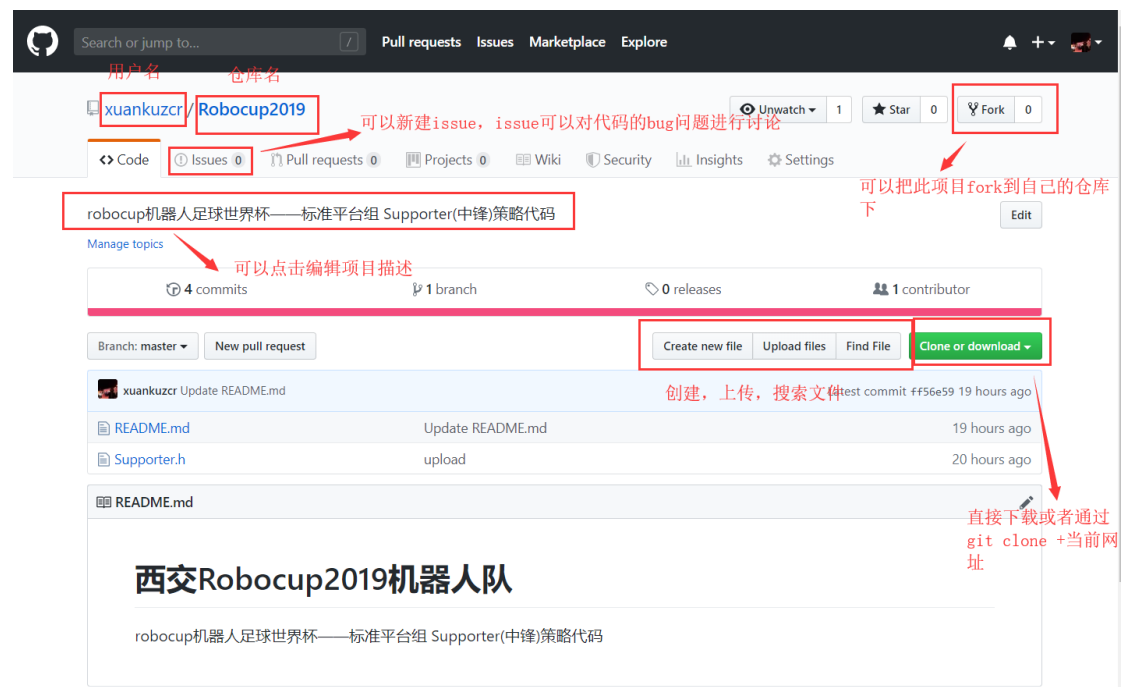


1.2.2 Github 主页

The screenshot shows the GitHub homepage for user **xuankuzcr**. Red boxes and arrows highlight specific features:

- Left sidebar:** A red box encloses the user's repository list and team information. An arrow points to the search bar with the text "自己所有的仓库信息" (Information about all your own repositories).
- Recent activity:** A red box encloses the "Recent activity" section, which shows a discussion in the **XJTU-Robocup/Robocup2019-China-Standard-Platform-League** repository. An arrow points to the "讨论区" (Discussion) icon with the text "github根据你的兴趣推荐给你的高star仓库" (GitHub recommends high-star repositories to you based on your interests).
- Explore repositories:** A red box encloses the "Explore repositories" section, which lists popular repositories like **alfio-event/alf.io** and **lhy0403/Qv2ray**. An arrow points to the "Explore repositories" title with the text "github根据你的兴趣推荐给你的高star仓库" (GitHub recommends high-star repositories to you based on your interests).
- Footer:** A red box encloses the footer area, which includes links to the Blog, About, Shop, API, Training, Status, Terms, Privacy, and Help. An arrow points to the footer with the text "自己关注的人的动态以及自己的动态" (Activity of people you follow and your own activity).

1.2.3 仓库主页



1.2.4 个人主页

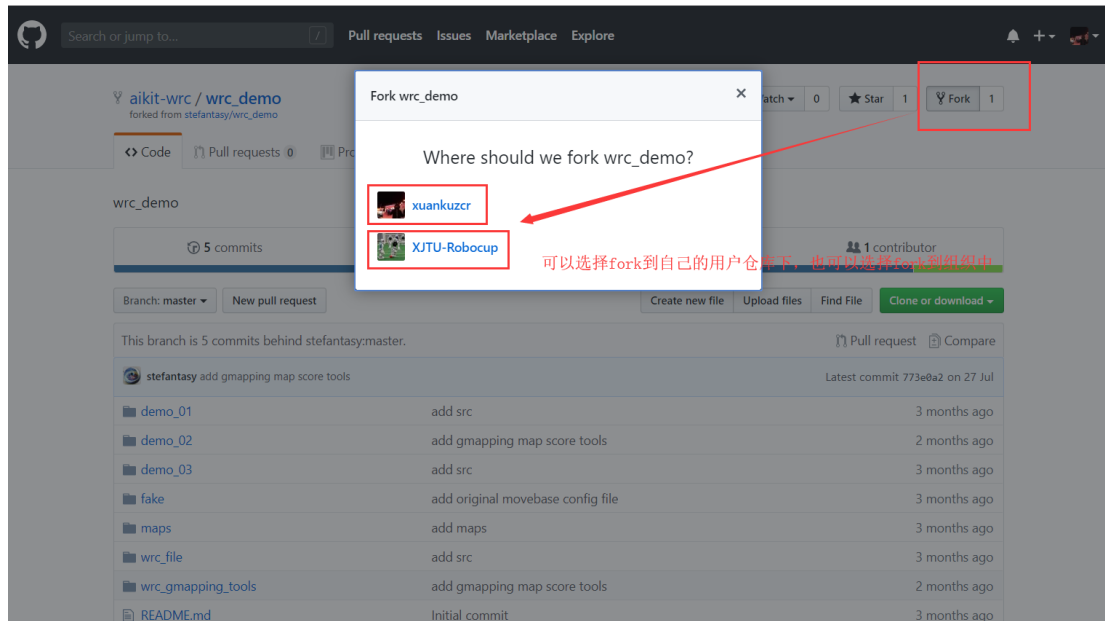
包含个人信息: 头像, 个人简介, 关注我的人, 我关注的人, 我关注的 git 库, 我的开源项目, 我贡献的开源项目等信息

1.3 fork 和 pull request 两个重要操作

这里不再赘述上几个图中的基本操作, 挑两个重要的来说。

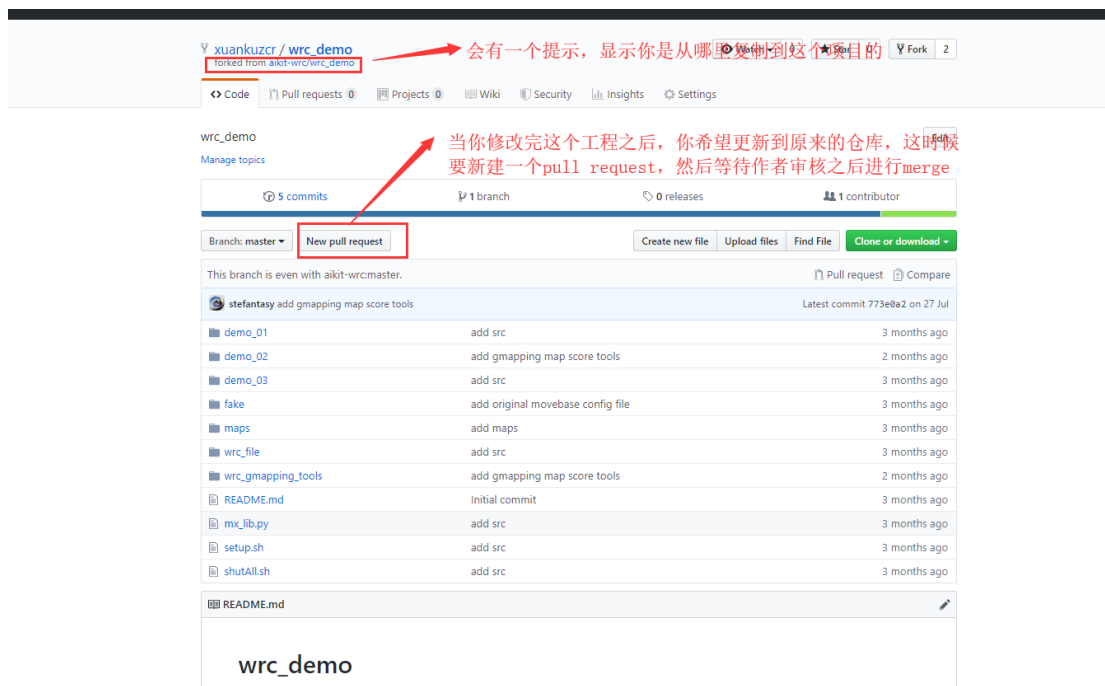
1.3.1 复制克隆项目 (Fork)

情景: 张三 fork 了李四的项目, 相当于张三复制了李四的项目, 所以自己也单独有了一个一样名称的仓库 (注: 该仓库会声明来自于李四, 但是独立存在)



1.3.2 发起请求（Pull Request）

情景: 张三修改了 fork 的项目中的文件, 希望更新到原来的仓库, 这时候他要新建一个 pull request



1.3.3 开源项目贡献流程

1) 新建 Issue

提交使用问题或者建议或者想法

2) Pull Request

步骤:

- 1、 fork 项目
- 2、 修改此时位于自己仓库的项目代码
- 3、 新建 pull request
- 4、 等待作者操作审核

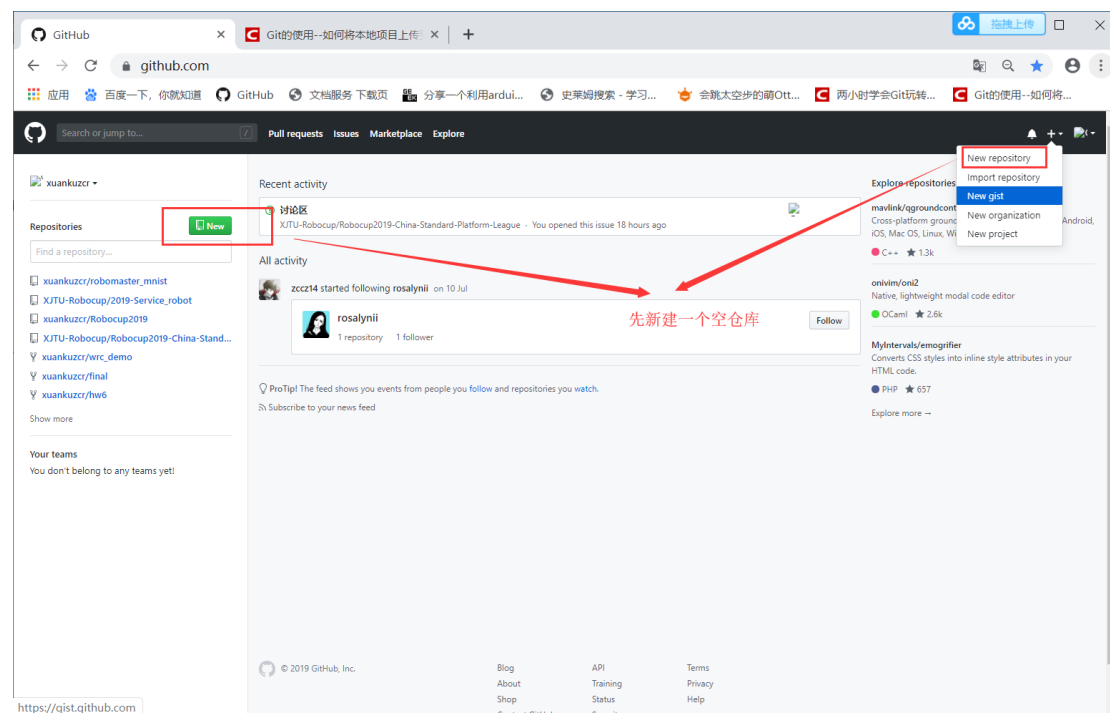
2. Git 的使用

2.1 首先学着创建一个新仓库

(我们用的 ubuntu16.04 或者 18.04 都自带 git, 不用安装)

Git 里面操作很多, 这里面主要讲如何提交自己写的工程文件到自己的仓库 Repository
此处介绍三种方法。

在介绍这三种方法之前, 要学着创建一个新仓库



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

仓库名

Owner: XJTU-Robocup

Repository name *: Robocup2018

Great repository names are short and memorable. Need inspiration? How about [glowing-succotash?](#)

Description (optional): 西交robocup2018标准平台组策略代码

简介

Public: Anyone can see this repository. You choose who can commit.

Private: Your current plan does not support private repositories. [Upgrade to Team](#)

私有收费

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

自动初始化生成一个readme文件

Add .gitignore: None

Add a license: None

Create repository

2.2 第一种：拖拽法

优点操作极其简单，缺点速度极慢，一般都会卡住。推荐有 VPN 的同学使用。

XJTU-Robocup / Robocup2018

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

西交robocup2018标准平台组策略代码

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find File Clone or download -

Initial commit

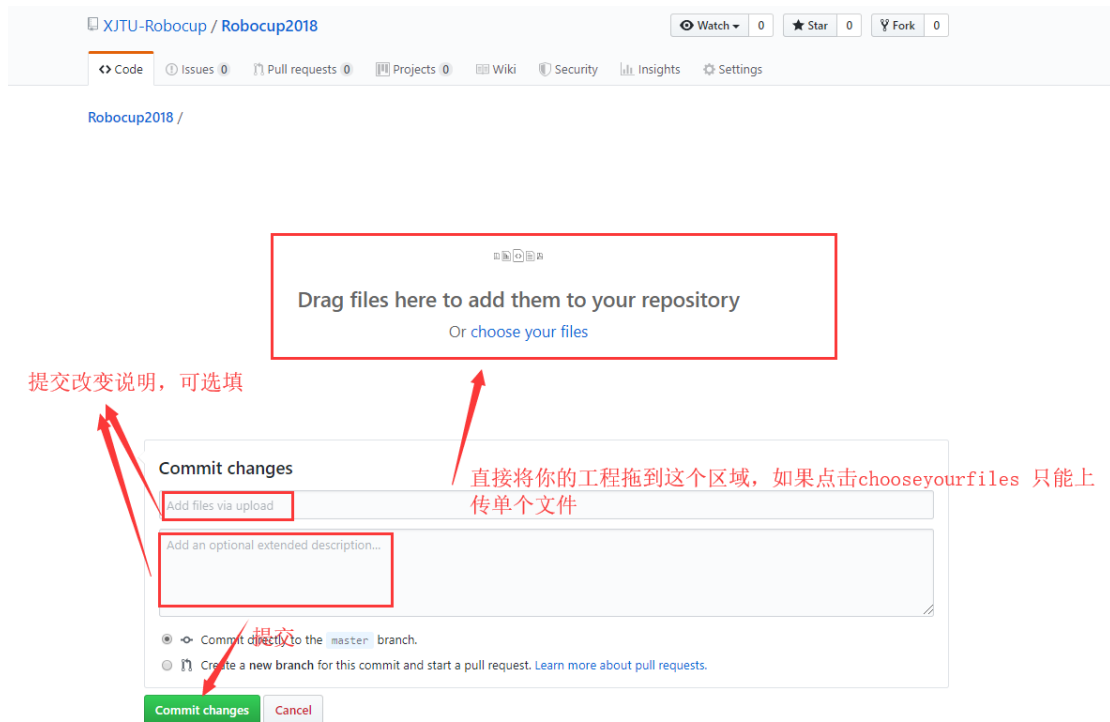
Latest commit 2028e36 31 seconds ago

31 seconds ago

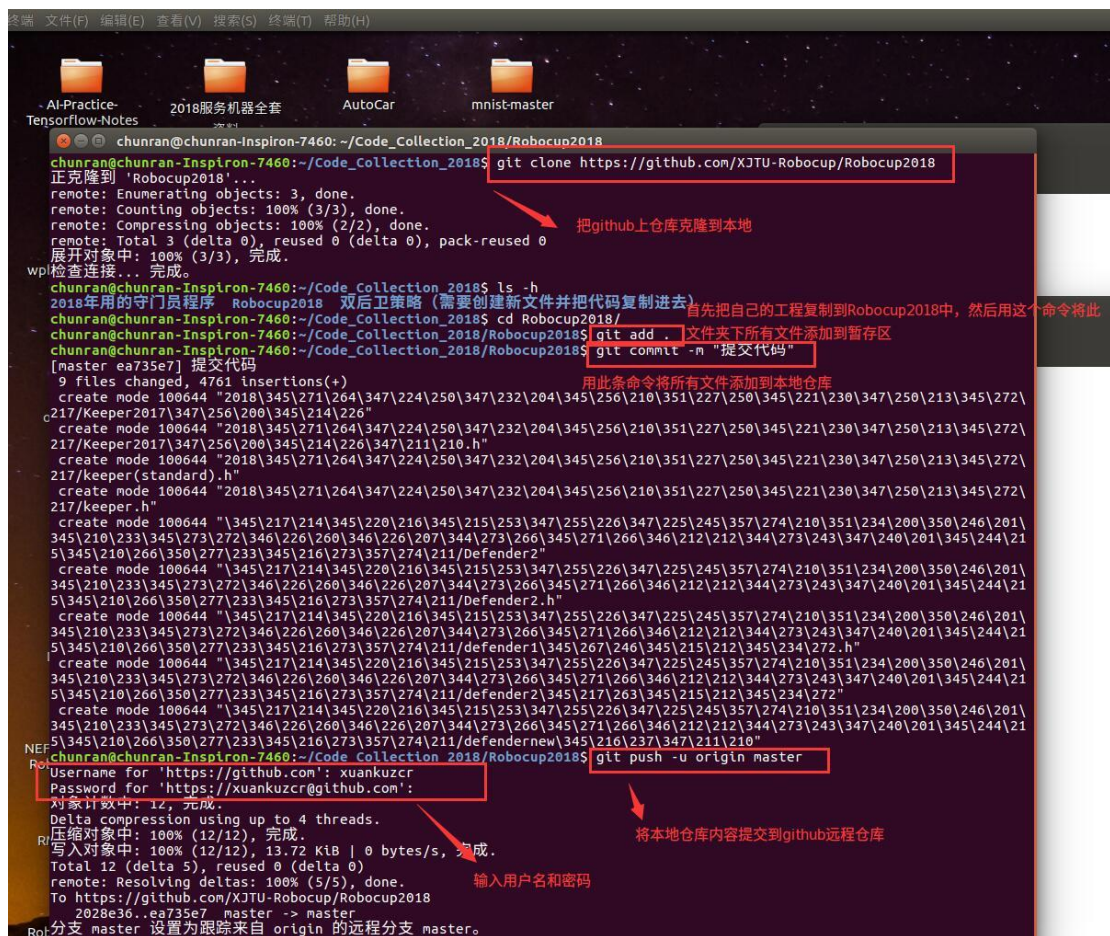
README.md

Robocup2018

西交robocup2018标准平台组策略代码

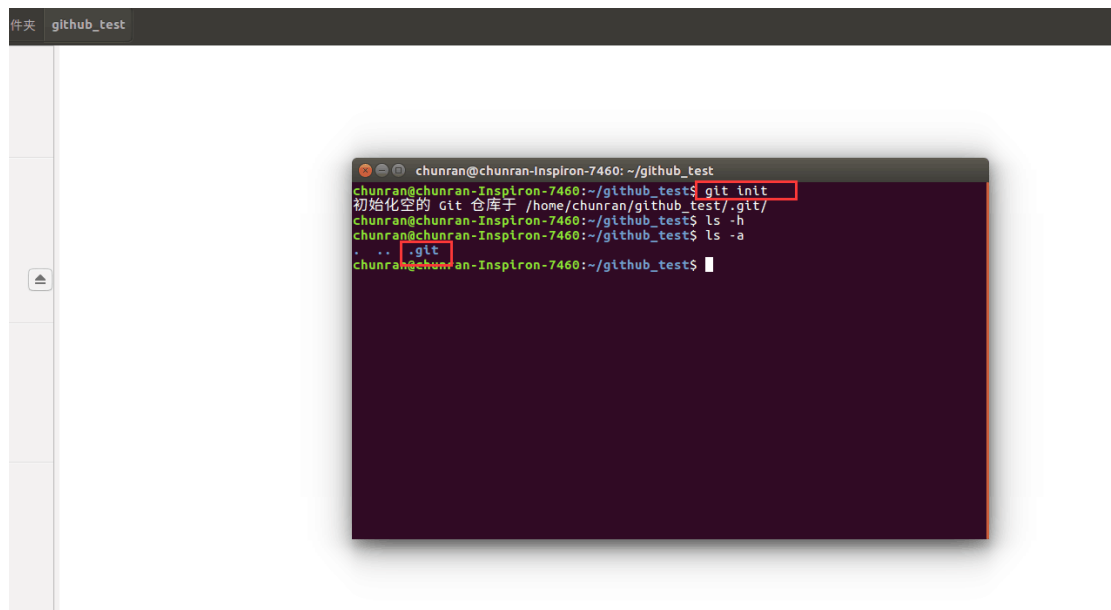


2.3 第二种: http 上传 (Git) (懒人推荐)

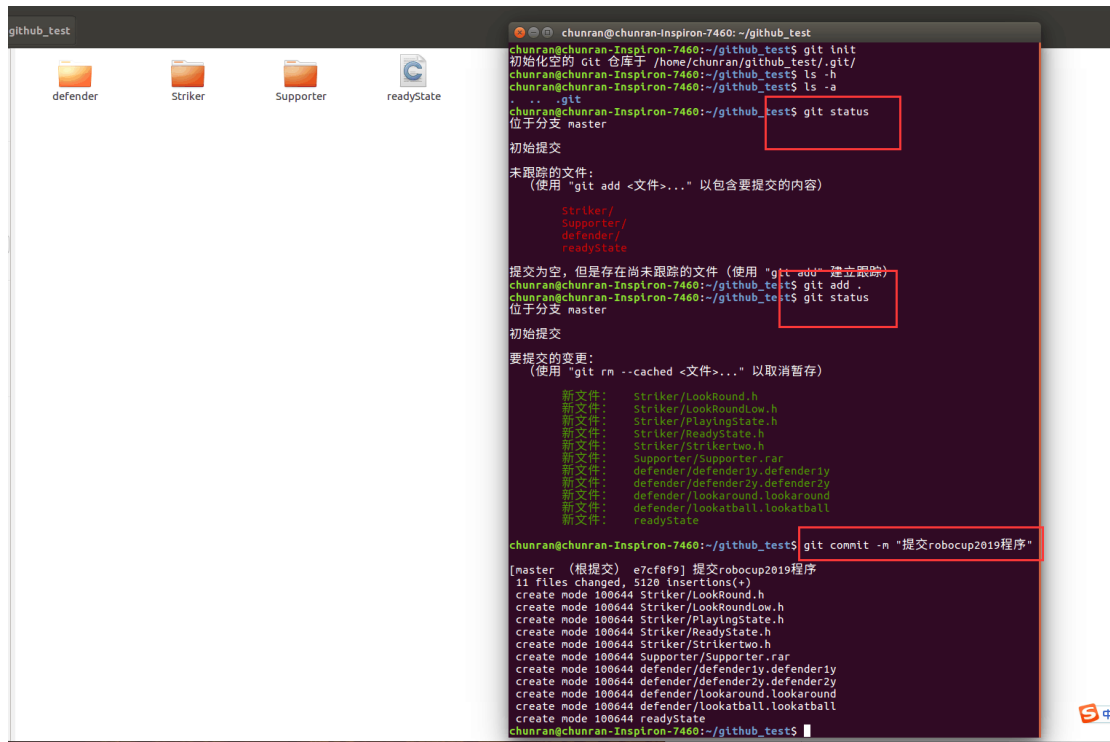


2.4 第三种：ssh 上传（Git）（rsa 加密）

1. 我们需要先创建一个本地的版本库（其实也就是一个文件夹）。现在我通过命令行新建一个 `github_test` 文件夹，并且进入这个文件夹
2. 通过命令 `git init` 把这个文件夹变成 Git 可管理的仓库。这时你会发现 `github_test` 里面多了个 `.git` 文件夹，它是 Git 用来跟踪和管理版本库的。（`ls -a` 可以查看隐藏文件）



3. 这时候你就可以把你的项目粘贴到这个本地 Git 仓库里面（粘贴后你可以通过 `git status` 来查看你当前的状态），然后通过 `git add` 把项目添加到仓库（或 `git add .` 把该目录下的所有文件添加到暂存区，注意点是用空格隔开的）。在这个过程中你其实可以一直使用 `git status` 来查看你当前的状态。
4. 用 `git commit` 把项目提交到本地仓库，`-m` 后面引号里面是本次提交的注释内容



The screenshot shows a terminal window with a dark background. On the left, there's a file explorer showing folders named 'defender', 'Striker', 'Supporter', and 'readyState'. The terminal output is as follows:

```
chunran@chunran-Inspiron-7460: ~/github_test
chunran@chunran-Inspiron-7460:~/github_test$ git init
初始化空的 git 仓库于 /home/chunran/github_test/.git/
chunran@chunran-Inspiron-7460:~/github_test$ ls -h
...
chunran@chunran-Inspiron-7460:~/github_test$ git status
位于分支 master
初始提交
未跟踪的文件:
(使用 "git add <文件>..." 以包含要提交的内容)
Striker/
Supporter/
defender/
readyState

提交为空, 但是存在尚未跟踪的文件 (使用 "git add" 建立跟踪)
chunran@chunran-Inspiron-7460:~/github_test$ git add .
chunran@chunran-Inspiron-7460:~/github_test$ git status
位于分支 master
初始提交
要提交的变更:
(使用 "git rm --cached <文件>..." 以取消暂存)
新文件:
Striker/LookRound.h
Striker/LookRoundLow.h
Striker/PlayingState.h
Striker/ReadyState.h
Striker/Strikertwo.h
Supporter/Supporter.rar
defender/defender1y.defender1y
defender/defender2y.defender2y
defender/lookaround.lookaround
defender/lookatball.lookatball
readyState

chunran@chunran-Inspiron-7460:~/github_test$ git commit -m "提交robocup2019程序"
[master (根提交) e7cf8f9] 提交robocup2019程序
11 files changed, 5120 insertions(+)
create mode 100644 Striker/LookRound.h
create mode 100644 Striker/LookRoundLow.h
create mode 100644 Striker/PlayingState.h
create mode 100644 Striker/ReadyState.h
create mode 100644 Striker/Strikertwo.h
create mode 100644 Supporter/Supporter.rar
create mode 100644 defender/defender1y.defender1y
create mode 100644 defender/defender2y.defender2y
create mode 100644 defender/lookaround.lookaround
create mode 100644 defender/lookatball.lookatball
create mode 100644 readyState
chunran@chunran-Inspiron-7460:~/github_test$
```

5. 由于本地 Git 仓库和 Github 仓库之间的传输是通过 SSH 加密的, 所以连接时需要设置一下: 创建 SSH KEY。先看一下/home 下有没有 .ssh 目录, 有的话看下里面有没有 id_rsa 和 id_rsa.pub 这两个文件, 有就跳到下一步, 没有就通过下面命令创建
- \$ssh-keygen -t rsa -C "youremail@example.com"** 然后一路回车。这时你就会有 /home 下的 .ssh 目录里找到 id_rsa 和 id_rsa.pub 这两个文件

```
chunran@chunran-Inspiron-7460: ~/github_test

初始提交

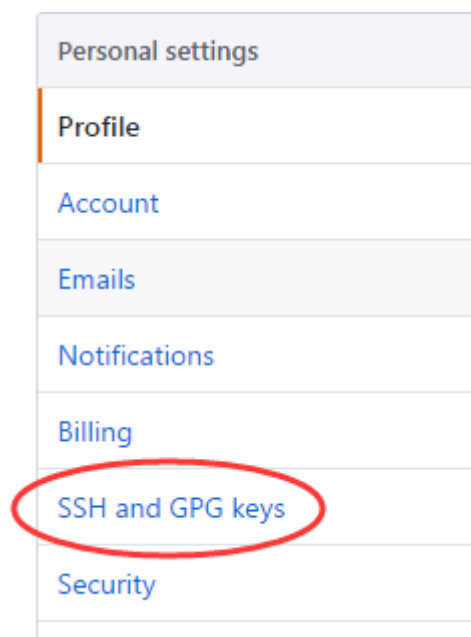
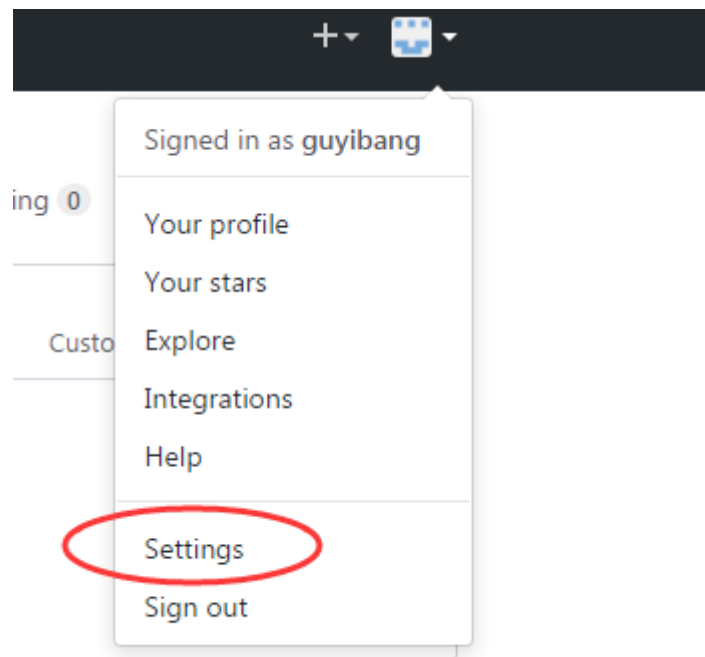
要提交的变更:
(使用 "git rm --cached <文件>..." 以取消暂存)

新文件:      Striker/LookRound.h
新文件:      Striker/LookRoundLow.h
新文件:      Striker/PlayingState.h
新文件:      Striker/ReadyState.h
新文件:      Striker/Strikertwo.h
新文件:      Supporter/Supporter.rar
新文件:      defender/defender1y.defender1y
新文件:      defender/defender2y.defender2y
新文件:      defender/lookaround.lookaround
新文件:      defender/lookatball.lookatball
新文件:      readyState

chunran@chunran-Inspiron-7460:~/github_test$ git commit -m "提交robocup2019程序"
[master (根提交) e7cf8f9] 提交robocup2019程序
11 files changed, 5120 insertions(+)
create mode 100644 Striker/LookRound.h
create mode 100644 Striker/LookRoundLow.h
create mode 100644 Striker/PlayingState.h
create mode 100644 Striker/ReadyState.h
create mode 100644 Striker/Strikertwo.h
create mode 100644 Supporter/Supporter.rar
create mode 100644 defender/defender1y.defender1y
create mode 100644 defender/defender2y.defender2y
create mode 100644 defender/lookaround.lookaround
create mode 100644 defender/lookatball.lookatball
create mode 100644 readyState
chunran@chunran-Inspiron-7460:~/github_test$ ssh-keygen -t rsa -C "18846640555@163.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/chunran/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/chunran/.ssh/id_rsa.
Your public key has been saved in /home/chunran/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xxvRVdKpReF8sPn/p+90shcArI+WNEb8HEft6YHpSnA 18846640555@163.com
The key's randomart image is:
+---[RSA 2048]----+
|      . . .o0+. |
|      + + *oB |
|      . * =oX o |
|      B E. = * |
|      S O o o o |
|      + + . O. |
|      . . . . = |
|      .   += |
|      O = |
+----[SHA256]-----+
```

```
chunran@chunran-Inspiron-7460: ~/ssh
chunran@chunran-Inspiron-7460:~$ ls -a
.          .pki
..         .presage
1.mp4      .profile
2018服务机器全套资料  Recognition-Algorithm
2018服务机器人比赛程序 RM19windmillDemo
2019机器人比赛程序.tar.gz RM_code.tar.gz
AI-Practice-Tensorflow-Notes RM-computer-vision
AutoCar    Robomaster2017
.bash_history Robomaster2018-SEU-OpenSource
.bash_logout .ros
.bashrc      .sdformat
.cache       ShowRune
Code_Collection_2018 SJTU-RM-CV-2019
Code_Collection_2019 snap
.config      segouinput
.config      .ssh
darknet      .ssr
DaShenFuCode .sudo_as_admin_successful
.dbus        Supporter.h
.deepinwine  Tencent Files
deepin-wine-ubuntu test
.dmr         TEST2
Downloads    .thunderbird
.gazebo      .viminfo
.gconf       .vnc
.git         .vscode
.gitconfig   .vscode-cpptools
github_test  .wget-hsts
.gnupg       .wine
.ICEauthority wpb_home_face_detect.cpp
.keras       wrc_demo
.local       .Xauthority
mnist-master .xinputrc
.mozilla     .xsession-errors
.mume        .xsession-errors.old
NEFU-ARES-Vison-Robomaster2018 zcr.md
.nv          模板
opencv-3.2.0 培训任务二.pdf
opencv_RM.tar.gz 培训任务一.pdf
opencv_test   视频
chunran@chunran-Inspiron-7460:~$ cd .ssh
chunran@chunran-Inspiron-7460:~/ssh$ ls
id_rsa id_rsa.pub known_hosts
chunran@chunran-Inspiron-7460:~/ssh$
```

6. 登录 Github, 找到右上角的图标, 打开点进里面的 Settings, 再选中里面的 SSH and GPG KEYS, 点击右上角的 New SSH key, 然后 Title 里面随便填, 再把刚才 id_rsa.pub 里面的内容复制到 Title 下面的 Key 内容框里面, 最后点击 Add SSH key, 这样就完成了 SSH Key 的加密。具体步骤也可看下面:



Marketplace Explore

SSH keys / Add new

Title

keys

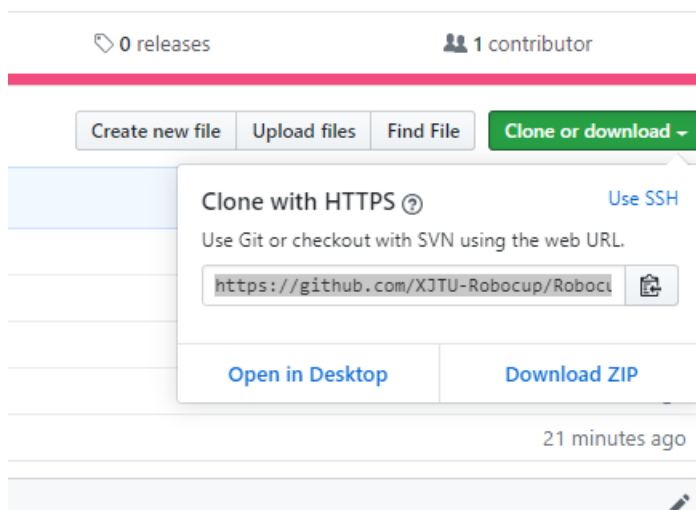
Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDQogbCuEFp5SY8ywTwUJ7nGT92fzASaACg6udUHZcM9z0M1FnrGt
JS9UZXMTW0wY7JcwN7Y/0+G/LclV4uiHQI9ybppi2gRFRdhp4cbGt/akTU1JamkbUcvq7aC6tOqVWnTRDDL9ofTp
MqSliGEvOWuIKBkchjKZ9y0JZ33CDfabKSBCYWWHmgouHpUNXNvaThJnZSRD7hZ0jW83KsXLAq4P4iWYsew
HtkZf3tP0MYU63WW4iqWP6PJqeOR/UjQhBUr/SL0+eLbV1ISwAsNf/StYL02oZFY8Eq/SoktvWgVQdhqn2yJm9CIP
kUq5fE2z8ciSV7pzE4N6/u2hgH 18846640555@163.com
```

Add SSH key

把id_rsa.pub里面的内容复制过来

7. 现在我们可以远程仓库和本地仓库进行关联了，可以在本地 `github_test` 仓库的命令输入：`$ git remote add origin https://github.com/XJTU-Robocup/Robocup2019-China-Standard-Platform-League.git`
- 注意 origin 后面加的是你 Github 上创建好的仓库的地址，如下图。



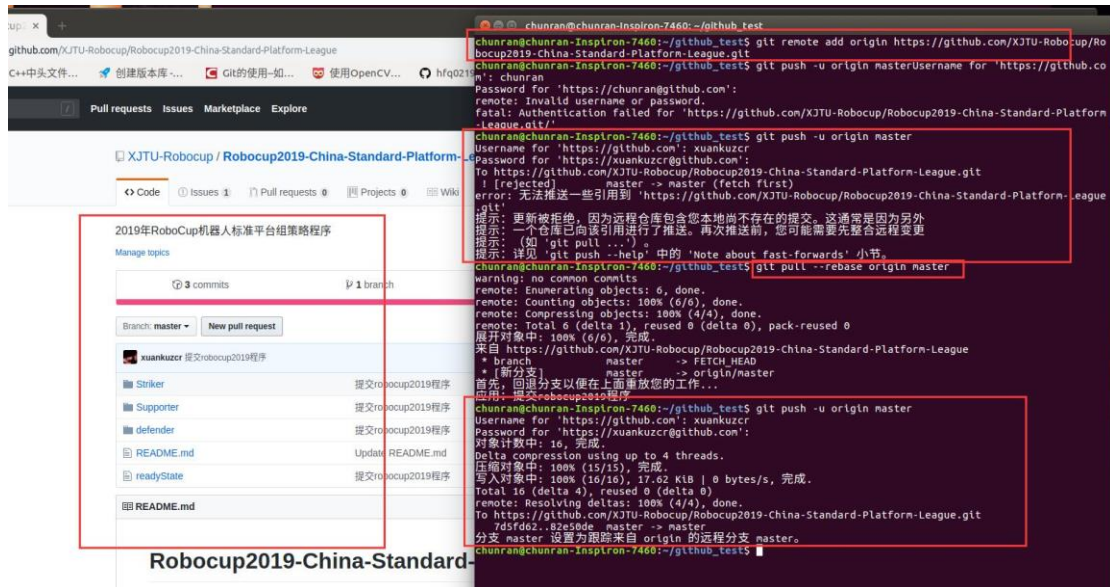
8. 关联好之后我们就可以把本地库的所有内容推送到远程仓库（也就是 Github）上了，通过：`$ git push -u origin master`
- 至此就完成了将本地项目上传到 Github 的整个过程

!!!

注意：另外，这里有个坑需要注意一下，就是在网页上创建新仓库的时候，如果你勾选了 Initialize this repository with a README（就是创建仓库的时候自动给你创建一个 README 文件），那么到了第八步你将本地仓库内容推送到远程仓库的时候就会报错。这是由于你新创建的那个仓库里面的 README 文件不在本地仓库目录中，这时我们可以通过以下命令先将内容合并以下：

`$ git pull --rebase origin master`

这时你再 push 就能成功了。



总结：其实只需要进行下面几步就能把本地项目加密上传到 Github

- 1、在本地创建一个版本库（即文件夹），通过 `git init` 把它变成 Git 仓库；
- 2、把项目复制到这个文件夹里面，再通过 `git add .` 把项目添加到仓库；
- 3、再通过 `git commit -m "注释内容"` 把项目提交到仓库；
- 4、在 Github 上设置好 SSH 密钥后，新建一个远程仓库，通过 `git remote add origin https://****.git` 将本地仓库和远程仓库进行关联；

这里注意：git 使用 https 协议，每次 pull, push 都要输入密码，很麻烦。使用 git 协议，然后使用 ssh 密钥。这样可以省去每次都输密码，只有第一次配置麻烦。

- 5、最后通过 `git push -u origin master` 把本地仓库的项目推送到远程仓库（也就是 Github）上。（若新建远程仓库的时候自动创建了 README 文件会报错，解决办法看上面）。

3. 最后

针对我们队情况的小白 github 快速上手指南就结束了，表扬大家的耐心学习！我们良好编程生态需要大家一起去创造维护，谢谢大家！



XJTU-Robocup

Repositories 4

Packages

People 8

Teams

Projects

Settings

Type: All ▾

Language: All ▾

Robocup2019-China-Standard-Platform-League

2019年RoboCup机器人标准平台组策略程序

C++ 0 0 1 0 Updated 1 hour ago

Robocup2018

西交robocup2018标准平台组策略代码

C++ 0 0 0 0 Updated 2 hours ago

2019-Service_robot

2019年中国机器人大赛——超市服务机器人赛项

0 0 0 0 Updated 21 hours ago

RobocupXJTU

Forked from RobocuProbe/RobocupXJTU

Treasure created by everyone in RobocupXJTU

1 0 0 0 Updated 14 days ago