中山大学数据科学与计算机学院本科生实验报告

（2017 年秋季学期）

课程名称：手机应用平台开发　　任课教师： 刘宁

年级	15 级	专业（方向）	计应
学号	15331421	姓名	郑佳锋
电话	13719325472	Email	1773643139@qq.com
开始日期	2017.9.22	完成日期	2017.9.23

一、 实验题目

Lab2: Git 的学习

二、 实现内容

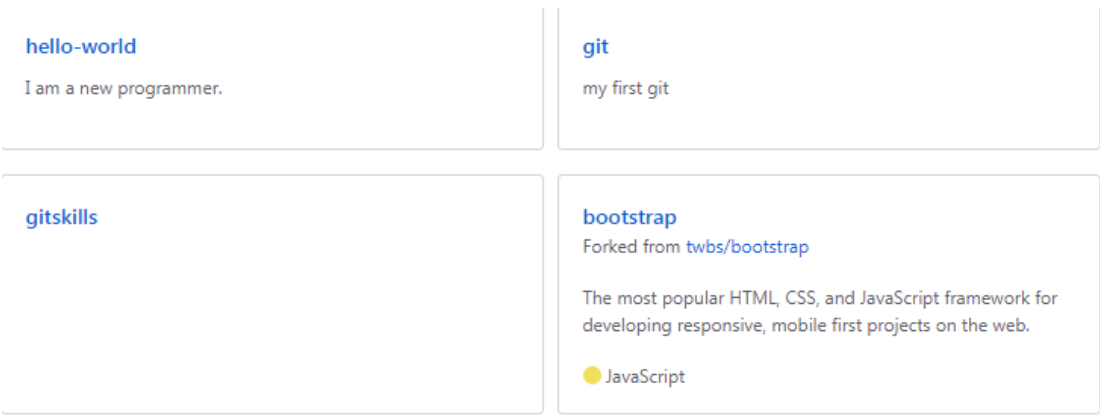
1.学习使用 Git 并创建自己的 github 账号。

2. 学习使用 Android Studio 的 git 工具。

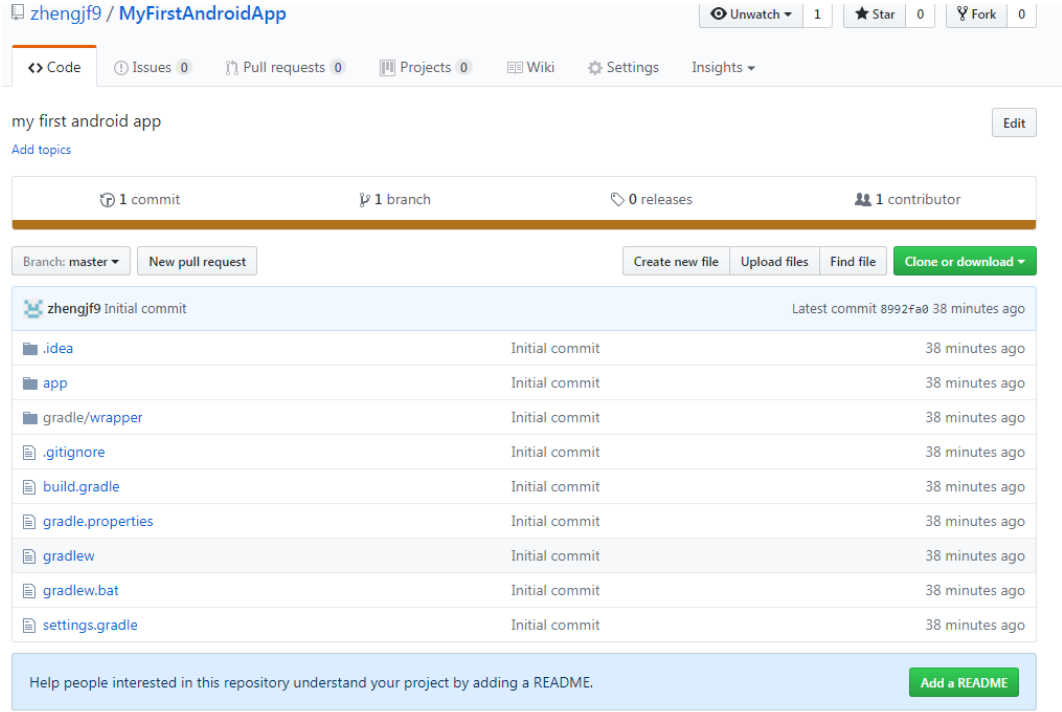
三、实验结果

1.实验截图

(1)学习使用 Git 工具后，上传本地项目到 github 账号，并尝试进行增删改操作。再从其他 github 账户下“Fork”bootstrap 项目到自己账户，克隆到本地。截图如下：



（2）利用 Android Studio 的 git 工具 lab1 的项目 push 到 github。截图如下：



2.实验步骤以及关键代码

Git 的主要操作如下：（摘自廖雪峰的官方网站-Git 教程，方便后面回顾）

初始化一个 Git 仓库，使用 `git init` 命令。

添加文件到 Git 仓库，分两步：

第一步，使用命令 `git add <file>`，注意，可反复多次使用，添加多个文件；

第二步，使用命令 `git commit`，完成。

要随时掌握工作区的状态，使用 `git status` 命令。

如果 `git status` 告诉你有文件被修改过，用 `git diff` 可以查看修改内容。

`HEAD` 指向的版本就是当前版本，因此，Git 允许我们在版本的历史之间穿梭，使用命令 `git reset --hard commit_id`。

穿梭前，用 `git log` 可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用 `git reflog` 查看命令历史，以便确定要回到未来的哪个版本。

场景 1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令 git checkout -- file。

场景 2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令 git reset HEAD file，就回到了场景 1，第二步按场景 1 操作。

场景 3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

命令 `git rm` 用于删除一个文件。如果一个文件已经被提交到版本库，那么你永远不用担心误删，但是要小心，你只能恢复文件到最新版本，你会丢失**最近一次提交后你修改的内容**。

要关联一个远程库，使用命令 `git remote add origin git@server-name:path/repo-name.git`；

关联后，使用命令 `git push -u origin master` 第一次推送 master 分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令 `git push origin master` 推送最新修改；

要克隆一个仓库，首先必须知道仓库的地址，然后使用 `git clone` 命令克隆。

Git 支持多种协议，包括 `https`，但通过 `ssh` 支持的原生 `git` 协议速度最快。

Git 鼓励大量使用分支：

查看分支：`git branch`

创建分支：`git branch <name>`

切换分支：`git checkout <name>`

创建+切换分支：`git checkout -b <name>`

合并某分支到当前分支：`git merge <name>`

删除分支：`git branch -d <name>`

当 Git 无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。

用 `git log --graph` 命令可以看到分支合并图。

Git 分支十分强大，在团队开发中应该充分应用。

合并分支时，加上 `--no-ff` 参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而 `fast forward` 合并就看不出来曾经做过合并。

修复 bug 时，我们会通过创建新的 bug 分支进行修复，然后合并，最后删除；

当手头工作没有完成时，先把工作现场 `git stash` 一下，然后去修复 bug，修复后，再 `git stash pop`，回到工作现场。

开发一个新 feature，最好新建一个分支；

如果要丢弃一个没有被合并过的分支，可以通过 `git branch -D <name>`强行删除。

查看远程库信息，使用 `git remote -v`；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用 `git push origin branch-name`，如果推送失败，先用 `git pull` 抓取远程的新提交；

在本地创建和远程分支对应的分支，使用 `git checkout -b branch-name origin/branch-name`，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用 `git branch --set-upstream branch-name origin/branch-name`；

从远程抓取分支，使用 `git pull`，如果有冲突，要先处理冲突。

命令 `git tag <name>`用于新建一个标签，默认为 `HEAD`，也可以指定一个 commit id；

`git tag -a <tagname> -m "blablabla..."`可以指定标签信息；

`git tag -s <tagname> -m "blablabla..."`可以用 PGP 签名标签；

命令 `git tag` 可以查看所有标签。

命令 `git push origin <tagname>`可以推送一个本地标签；

命令 `git push origin --tags` 可以推送全部未推送过的本地标签；

命令 `git tag -d <tagname>`可以删除一个本地标签；

命令 `git push origin :refs/tags/<tagname>`可以删除一个远程标签。

3.实验遇到困难以及解决思路

廖雪峰的学习网站-Git 教程主要以 Linux 下的操作为例，在 windows 下操作发现几处不同，收录如下：

(1)显示文件内容用 type 不用 cat。

(2)git reset --hard HEAD^命令由于^在 windows 命令行是特殊字符，所以应该加上双引号。

(3)删除文件用 del 而不用 rm。

四、实验思考及感想

很早就听过 Git 的大名，但是因为没用到所以一直没动力去学。这次实验课上老师要求去学，TA 提供的教程网站也非常的棒，所以就仔仔细细地跟着教程一步一步去做，前后大概花了好几个小时，总算明白了 git 的基本操作，感觉受益匪浅。当然，一下子那么多指令也记不牢固，只有在后面边用边查，不断地熟悉，才能真正变为自己的知识！