



开篇

如今，Git 大行其道，颇有一统天下之势。

如果你的技能树上 Git 和 Github 的图标还没有点亮的话，你都不好意思说你是程序员。

别说互联网企业，我接触到的许多传统企业都在从 SVN，Clear Case 等迁移到 Git 上，甚至大厂还会有一个团队去定制适合自己企业的 Git 服务器。

很多人简历上写的「精通 Git 与 Github」，但如果你问他熟悉到什么程度的话，回答通常是「就是会用常用的 `add`，`commit`，`push` 操作」。

但工作中我们会遇到一些更加复杂的场景：

- * 忽略某些文件
- * 创建，删除分支
- * 找到最后修改某行代码的人
- * `commit` 后发现里边包含一个不应该提交的文件
- * `commit` 后发现少提交了一些文件
- * 一个文件中的多次有改动，怎么做到只提交其中的一部分？
- * 怎样整理提交记录使其更加整洁？
- * 快速定位是哪一次提交引入了 `bug`
- * ...

作为一线程序员，我们要如何快速学习 Git 以发挥其最大威力呢？

今天我就要向大家介绍一个工具，准确说应该是「游戏」，名字叫「githug」

(<https://github.com/Gazler/githug>)，它把平常可能遇到的一些场景都实例化，变成一个一个的关卡，一共有 55 个关卡：

#1: init
#2: config
#3: add
#4: commit
#5: clone
#6: clone_to_folder
#7: ignore
#8: include
#9: status
#10: number_of_files_committed
#11: rm
#12: rm_cached
#13: stash
#14: rename
#15: restructure
#16: log
#17: tag
#18: push_tags
#19: commit_amend
#20: commit_in_future
#21: reset
#22: reset_soft
#23: checkout_file
#24: remote
#25: remote_url
#26: pull
#27: remote_add
#28: push
#29: diff
#30: blame
#31: branch
#32: checkout
#33: checkout_tag
#34: checkout_tag_over_branch
#35: branch_at
#36: delete_branch
#37: push_branch
#38: merge
#39: fetch
#40: rebase
#41: repack
#42: cherry-pick
#43: grep
#44: rename_commit
#45: squash
#46: merge_squash
#47: reorder
#48: bisect
#49: stage_lines
#50: find_old_branch
#51: revert
#52: restore
#53: conflict
#54: submodule
#55: contribute

首先我们需要来安装这个游戏，githubg 是用 Ruby 编写的，可通过如下命令安装：

```
gem install githubg
```

如果遇到权限问题，请加上 sudo：

```
sudo gem install githubg
```

安装成功后，在 Terminal 里进入你常用的目录，输入 githubg，会提示游戏目录不存在，是否要创建一个，输入 y 然后回车：

```
x> ~/Documents gem install githubg
Successfully installed githubg-0.4.4
Parsing documentation for githubg-0.4.4
Done installing documentation for githubg after 0 seconds
1 gem installed
x> ~/Documents githubg
*****
*                               Githubg                               *
*****
No githubg directory found, do you wish to create one? [yn]  y
Welcome to Githubg!

Name: init
Level: 1
Difficulty: *

A new directory, `git_hug`, has been created; initialize an empty repository in it.
```

根据提示 cd git_hug 进入游戏目录，准备开始游戏。

基本命令

在开始前我们还需要了解游戏的一些基本操作：

- play - 默认命令，检查是否过关
- hint - 显示过关提示
- reset - 重启本关，或者重启到指定的某关
- levels - 显示关卡列表

来试一下，github reset：

```
x> ~/D/git_hug github reset
*****
*                               Github                               *
*****
resetting level

Name: init
Level: 1
Difficulty: *

A new directory, `git_hug`, has been created; initialize an empty repository in it.
```

示例

我以第一关为例子给大家演示一下玩法。

第一关的名称是：init，提示是：「一个新目录 git_hug 被创建了，请把它初始化为一个空仓库」。
假设现在我不知道该怎么过关，我可以查看过关提示：

```
x> ~/D/git_hug github hint
*****
*                               Github                               *
*****
You can type `git` in your shell to get a list of available git commands.
```

指示是：「你可以输入 git 命令来查看 git 命令列表」。

```
x> ~/D/git_hug git
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
    clone      Clone a repository into a new directory
    init       Create an empty Git repository or reinitialize an existing one
```

看最后一行，原来用 `git init` 就可以初始化一个空仓库，接着输入 `githubg` 进行过关检测：

```
x> ~/D/git_hug git init
Initialized empty Git repository in /Users/seabornlee/Documents/git_hug/.git/
x> ~/D/git_hug on master ◦ githubg
*****
*                                     Github                                     *
*****
Congratulations, you have solved the level!

Name: config
Level: 2
Difficulty: *

Set up your git name and email, this is important so that your commits can be identified.
```

太棒了！顺利进入第二关！

怎么样？明白了吗？后面的 54 关就靠你自己了哦！

友情提示

在闯关过程中，尽量使用 `githubg hint`，`git help subcommand` 获取帮助，也可以根据下面的两个教程来系统学习。

猴子都能懂的 git 入门 (<http://backlogtool.com/git-guide/cn/>)

廖雪峰的 git 教程

(<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>)

为保大家都能顺利通关，学到所有的知识点，接下来我会写过关攻略，详细介绍每一关的玩法。并且我不会直接给答案，而是演示整个过关的过程。

祝你好运！我们明天见。

「软件匠艺社区」(<http://codingstyle.cn>)旨在传播匠艺精神，通过分享好的「工作方式」和「习惯」以帮助程序员更加快乐高效地编程。