Lecture 5
Remotes
Midterm Next Week

• No stressing allowed!
• 50 minutes in-class. You probably won’t need the whole time.
• You can use the Git man pages and the lecture slides, but no googling
• Questions very similar to the homework questions

Last year’s midterm (which didn’t include remotes):
Homework 4 Review
What We’ve Learned So Far

• Creating and cloning repositories
git init, git clone

• Linear commit histories and diffs
git log, git show, git diff

• Using the working directory and staging area and making commits
git add, git reset, git checkout, git commit

• Using branches
git branch, git checkout, git merge

• How Git’s model for commit histories works
Today

- Remotes
- `git remote`
- `git fetch`
- `git pull`
- `git push`
- Github
8b7d883: Bob: begin work on feature B
8fc42c6: Alice: begin work on feature A
6f96cf3: Alice: more work on feature A
8277e09: Bob: even more work on feature B
e167179: Bob: more work on feature B
8b7d883: Bob: begin work on feature B
b5f3729: Alice: even more work on feature A
b4e2c29: initial commit
db82ca7: Merge branch 'featureA' into master
db82ca7: Merge branch 'featureA' into master
master, HEAD
featureA
featureB
From the First Lecture: Git is a DVCS

Alice’s Repo

Charlie’s Repo

Bob’s Repo

push, pull, fetch
Flashback: Cloning a Repo

Downloads the repo, puts you on the master branch

$ git clone https://github.com/aperley/dino-story.git
Committing and Pushing Back to Github

```
git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
nothing to commit, working tree clean
```

```
git log --decorate --all --graph --abbrev-commit
* commit 93d2876 (HEAD -> master)
  Author: Aaron Perley <aaron.perley@gmail.com>
  Date:   Thu Feb 22 14:58:44 2018 -0500
    Change Tyler->Teresa

* commit aa2d5ac (origin/master, origin/HEAD)
  Author: Aaron Perley <aaron.perley@gmail.com>
  Date:   Thu Feb 22 14:50:19 2018 -0500
    Add chapter 1 of dino-story
```
Committing and Pushing Back to Github

93d2876
"Change Tyler -> Teresa"

aa2d5ac
"Add chapter 1 of dino story"

Aaron’s Computer

Github’s Server
Listing Remote Branches

```
$ git branch -a
*
  master
remotes/origin/HEAD -> origin/master
remotes/origin/master
```
Pushing

$ git push origin master

Pushes the local branch called master to the branch called master on the remote named origin

This is how we move where remote branches point to

$ git remote -v
origin  https://github.com/aperley/dino-story.git (fetch)
origin  https://github.com/aperley/dino-story.git (push)
Pushing Back to Github
Fetching

Aaron’s Computer

93d2876
"Change Tyler -> Teresa"

aa2d5ac
"Add chapter 1 of dino story"

origin/master
origin/HEAD
master
HEAD
origin/master
origin/HEAD

Github’s Server

3cd86b0
"Add chapter 2 of dino story"

93d2876
"Change Tyler -> Teresa"

3cd86b0
"Add chapter 2 of dino story"

master
HEAD

origin/master
origin/HEAD
master
HEAD
origin/master
origin/HEAD
Keeping Local Branches Up To Date

Aaron's Computer

3cd86b0
"Add chapter 2 of dino story"

93d2876
"Change Tyler -> Teresa"

aa2d5ac
"Add chapter 1 of dino story"

Github's Server

3cd86b0
"Add chapter 2 of dino story"

93d2876
"Change Tyler -> Teresa"

aa2d5ac
"Add chapter 1 of dino story"
Fetching and Merging

$ git fetch origin
    Fetch all updates from the remote named origin
    Downloads new commits, moves origin/<branch> pointers
    Cannot move local branches!

$ git merge origin/master
    Merges (fast-forwards, hopefully!) the local master to the
    commit pointed to by origin/master
How to get a local branch for another remote branch?

$ git checkout --track origin/experiment

Realizes that you don’t have a local branch called experiment, so creates it and switches to it.

--track will show information in git status about whether branches are “ahead” or “behind” each other
Summary

• Configuring remotes:
  • git remote [-v] – lists remotes [verbosely]
  • git remote add <remotename> <remoteurl> - configure a new remote
  • git branch –r or –a – lists branches including remote tracking

• Fetching:
  • git fetch <remotename> - downloads updates to all remote-tracking branches to match the remote
  • git pull <remotename> - runs `git fetch`, then merges in updates to the current branch

• Pushing:
  • git push <remotename> <branchname> - uploads changes in your branches to the remote
Activity / Homework!

1. Create a GitHub account (if you don’t have one)
3. Clone your fork
4. Add your name to the byline (“By: Aaron”) and push the change to the master branch on your fork on GitHub
5. Checkout the chapter-2 branch and create a commit adding your own chapter 2
6. Merge the byline change from the master branch into the chapter-2 branch
7. Push the chapter-2 branch to the chapter-2 branch on your fork on GitHub