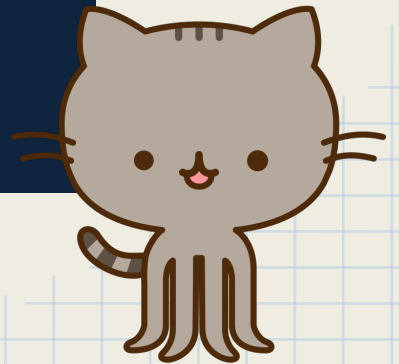


Branching and Merging

by Shannon Lee, Jonathan Miedel, and Alvin Wang



Last Time on Git

- git pull
- git checkout
- git reset
- git blame
- rebase interactive

Pop Quiz!

1. What is git pull used for?
2. What is the index?
3. Explain the difference between git reset --soft, --mixed, and --hard?
4. What are HEAD and master?
5. What is the difference between git reset <file> and git reset <commit>?
6. Name a use and danger for git rebase interactive.
7. What is git checkout for?
8. How are commits identified in git?
9. How do you reconcile a detached HEAD?

What is a branch?

- A branch is essentially a pointer to a commit in your git history
- master is the default branch
- Creating a branch is as simple as creating a new pointer to a commit in Git

An Example from Pro Git

```
$ git branch testing
```

This creates a new pointer at the same commit you're currently on (see Figure 3-4).

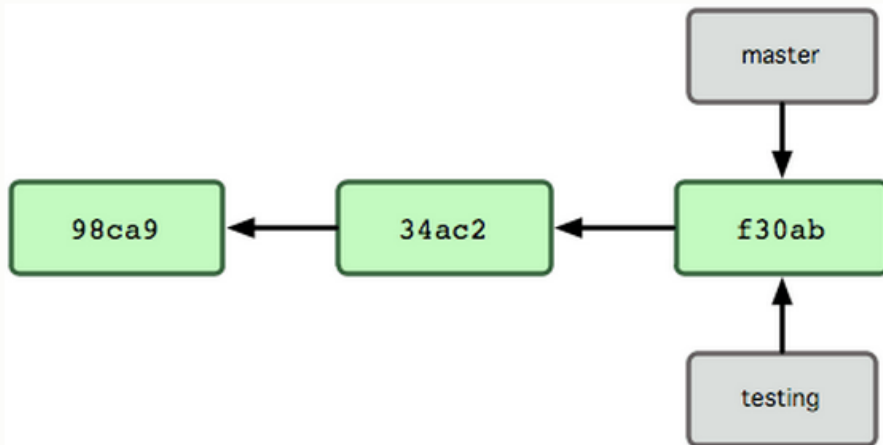


Figure 3-4. Multiple branches pointing into the commit's data history.

Git Branch

`git branch` - shows local branches; `-a` will show all
`git branch <branch>` - creates a branch off of HEAD
`git status` - this will tell you what branch you are currently on
`git remote show origin` - a lot of useful information about which branches are tracked and which need updates

Branching is Good

Pros:

- Allows for more isolation in workflow
- Allows for feature-based development

Cons:

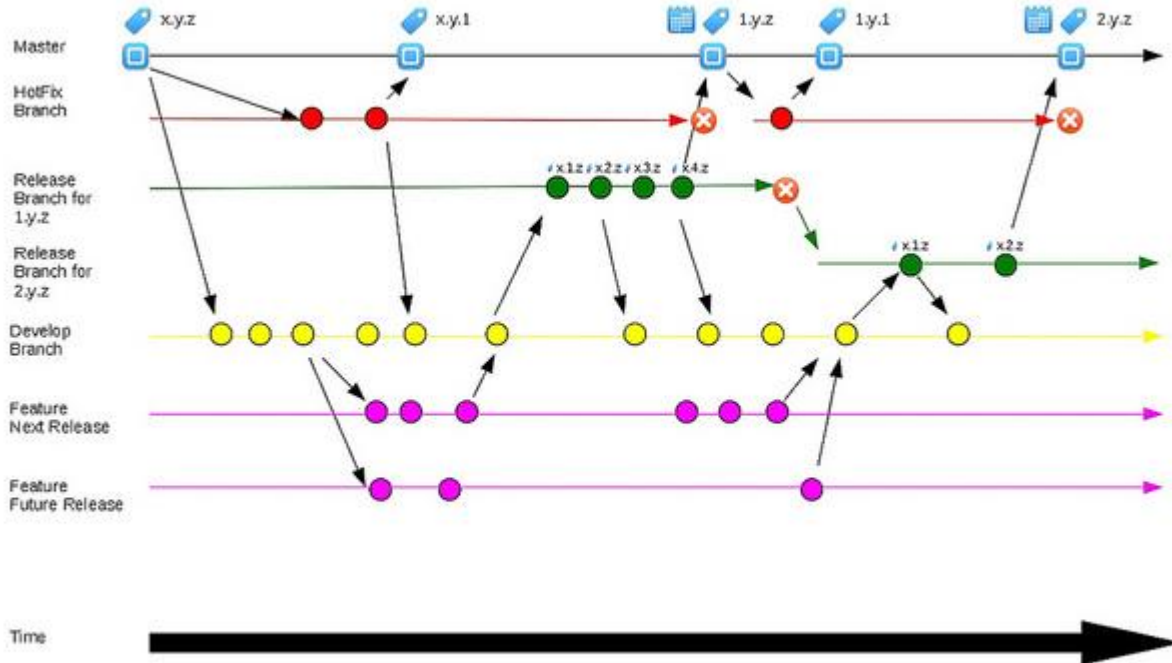
- More confusing
- commit history get more convoluted
- hotfixes / security leaks are not applied immediately

Branch Naming Conventions

- Short
- Descriptive
- Separated by “/”s
- Examples
 - tooltip/positioningFix
 - parallax/responsiveFix

Example Branch Workflow

GitFlow with Releasing Number



Deleting Branches

- `git branch -d <branchname>`

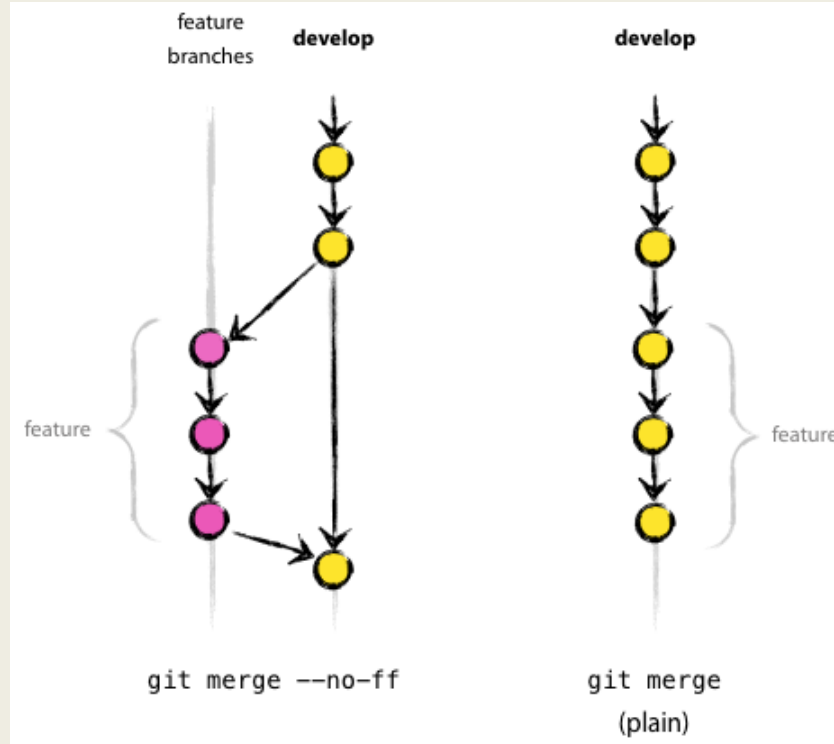
Git Checkout (again)

- git checkout
 - git checkout <branch>
 - checks out a branch
 - git checkout -b <branch>
 - creates a new branch and moves you to it

Merging

- It serves to integrate a branched tree back into a single tree
- Two main algorithms in Git
 - Fast-Forward
 - used by default if the branches have not diverged
 - 3-Way merge
 - uses diff3 to create the merge-commits we saw last time. 3-way refers to local, remote, common-ancestor

The Two Algorithms



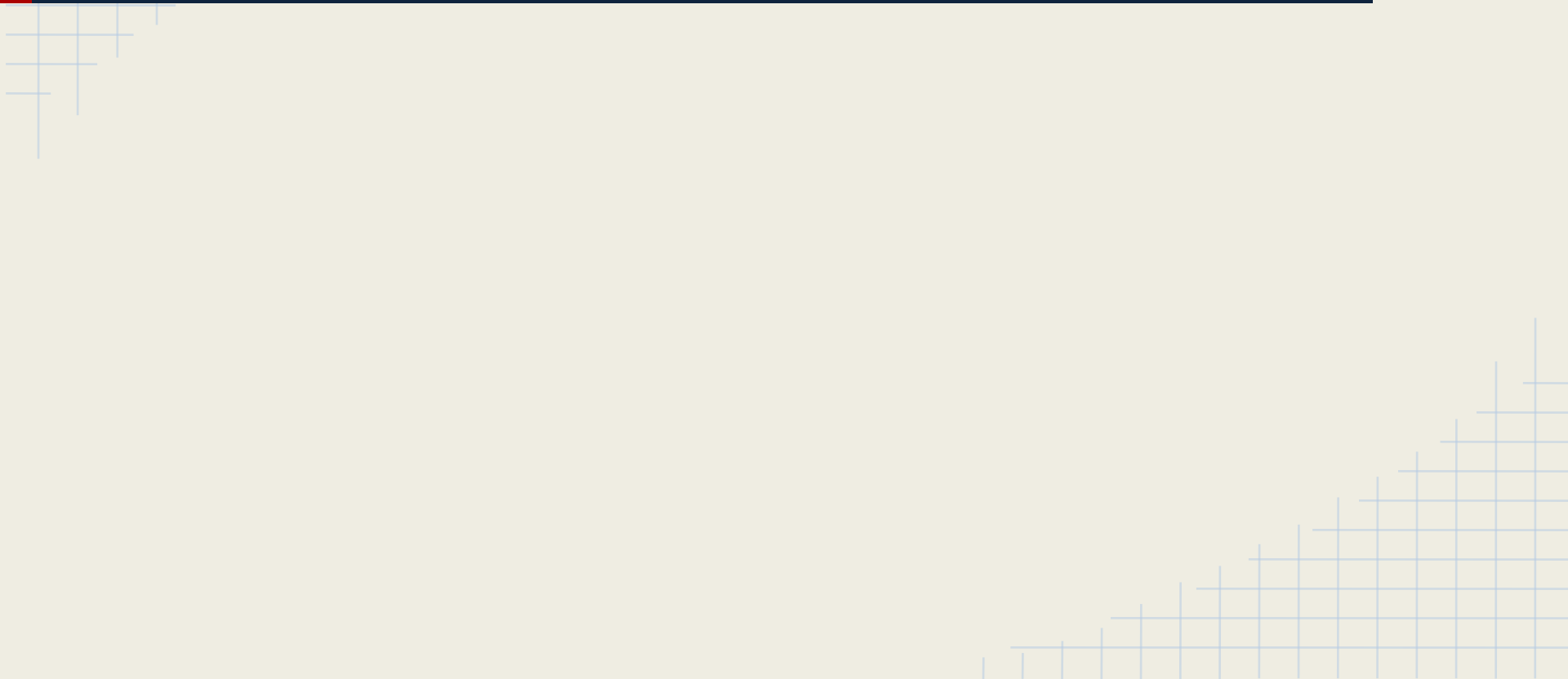
Git Merge

- `git merge <branch>`
- `--no-ff`
- `--no-commit`

Merge Conflicts

- Merge conflicts occur when the process of merging does not go smoothly
 - Often happens if two people edit the same line
- Git will mark the file where the conflict occurred
 - You can resolve these manually
 - A wide variety of tools can be used to assist (next week)

Merge Demo



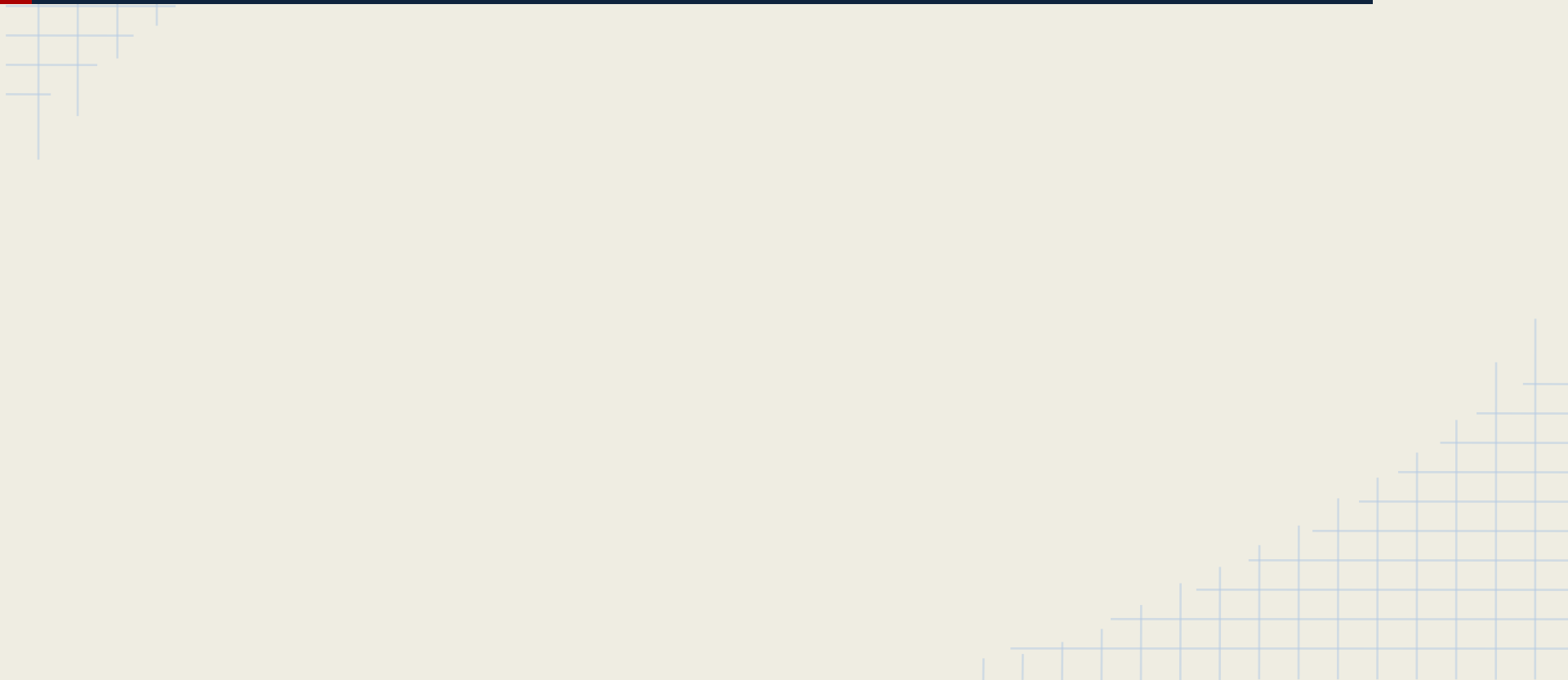
Pushing Branches to Remote

- simple (default in git 2.0)
 - pushes only the current branch
- matching
 - pushes all local branches if they already are tracked on remote
- `git config --global push.default matching`

Deleting remote branches

- `git push origin --delete <branch>`

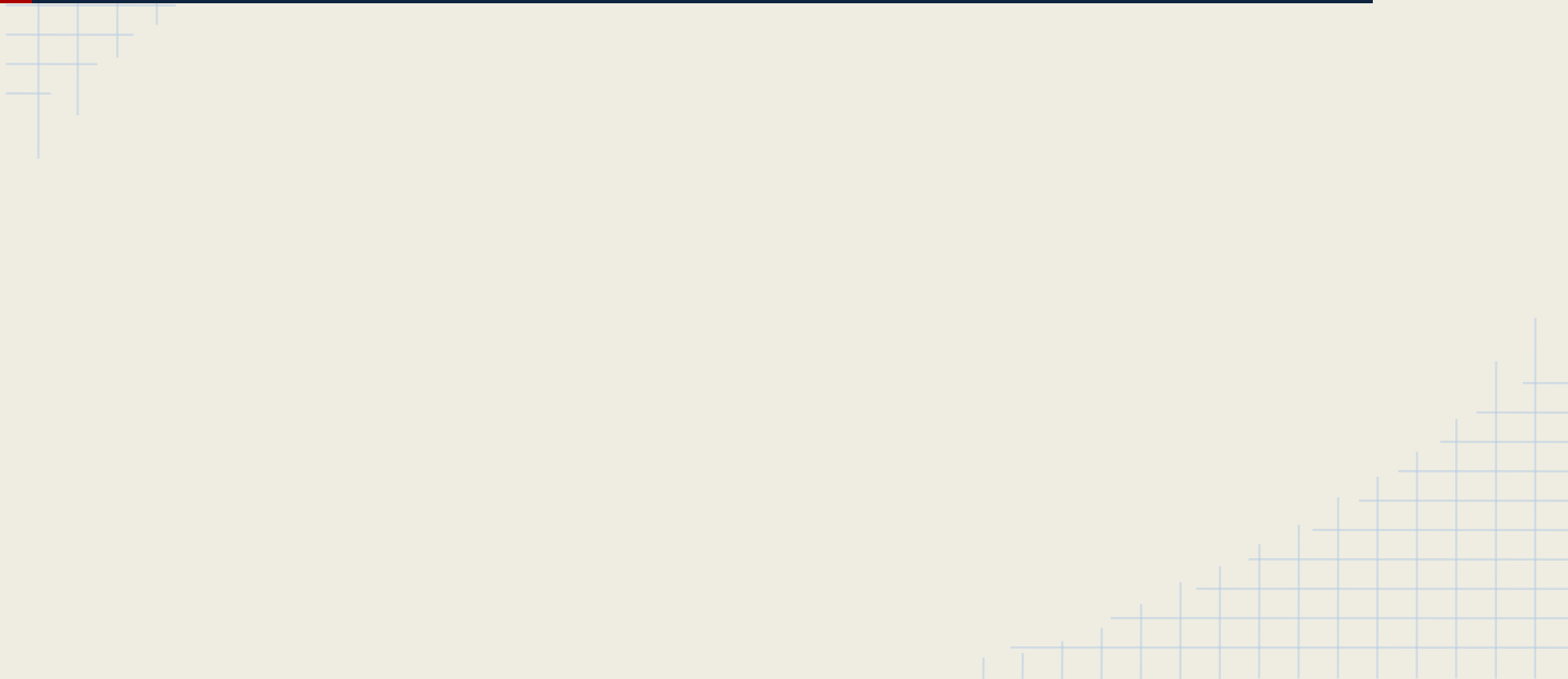
Deleting Remote Branch Demo



Fork

- GitHub feature
- Creates a copy of a repository in your own space
- Useful if you wish to contribute to repositories you do not have permissions for
- Also useful if using a repository as a base for a project

Forking Demo



Homework

- fork HW3 under your account
- clone it
- on master branch, reset back to the second commit so that the changes from the latest 3 commits are kept in your index.
- create a new branch called “HWBranch” and switch to it
- commit your index to the new branch with the message “readded changes”
- merge newFeature branch into the branch you just created successfully using `git merge origin/newFeature`; make sure you are on HWBranch when doing so
- You will get a conflict, resolve them manually by taking the changes from newFeature branch, commit after you resolve the conflicts with the message, “fixed merge conflicts”
- push only the HWBranch to your repo
- Once you are done, create an issue on Github on the Git-Stuco/HW3 repo with your andrewid as the title and a URL link to your forked repo as the body of the issue.

Complete HW git log

```
testing — bash — 96x30
bash
CMU-845210:testing Alvin$ gl
* 83ff28d - (2 minutes ago) fixed merge conflicts - Alvin Wang (HEAD, HWBranch)
| \
| * 0b1e67c - (57 minutes ago) localization fixes - Alvin Wang (origin/newFeature, newFeature)
* | bab2993 - (5 minutes ago) readded changes - Alvin Wang
| | * b477f09 - (57 minutes ago) globalization fixes - Alvin Wang (origin/master, origin/HEAD)
| | /
| * 543d4bb - (59 minutes ago) added cool features to DOM node insertion - Alvin Wang
| * d5122ac - (63 minutes ago) added random hi everywhere - Alvin Wang
| /
* b910bdd - (74 minutes ago) added JQuery - Alvin Wang (master)
* 717287a - (85 minutes ago) Initial commit - Alvin Wang
CMU-845210:testing Alvin$
```

Next Week in Git

- More advanced merge conflict resolution
- Tools to aid in the merge process

Extra Topics

SSH git
git gc