Version Control using Git and Github

Erik Edrosa & Joseph Rivera

What is Version Control?

- Management of additions, deletions, and modifications to software or more generally any documents or files.
- Attempts quality documentation of changes throughout lifetime of software or document development.

- Version Control System
  - Usually a standalone application used to keep track of revisions.
  - Can be used on servers or with a website (like Github or Wikipedia)

Why is Version Control Necessary?

- Keeps older versions with a history of all changes made.
  - If you break something vital, and can't figure out what you did
    - "Roll back" to a working version!
  - Compare old implementations vs. new implementations
  - See when and why changes were made
- Managing multiple users
  - What if you want to develop a new functionality?
    - Create a "branch" that can be manipulated without affecting the original version!
    - When the branch is ready, you can merge it back in, handling any conflicts that may arise.
  - Keep track of who made what changes when, and why
    - You can comment about each "commit" you make, explaining your changes

Git ~ The Stupid Content Tracker

- A Distributed Version Control System
  - Each user has a complete copy of the repository
    - Protects against “central” repository failure
    - Work on your own without messing up other people’s stuff
    - Meritocracy without complete loss of control
- Developed by Linus Torvalds (Creator of Linux)
  - Initially created for use in Linux Kernel development
  - Open Source, so you can learn about version control (and make it better)!
  - Primary Goals
    - Speed
    - Non-linear development support
    - distributed

GitHub ~ Build software better, together

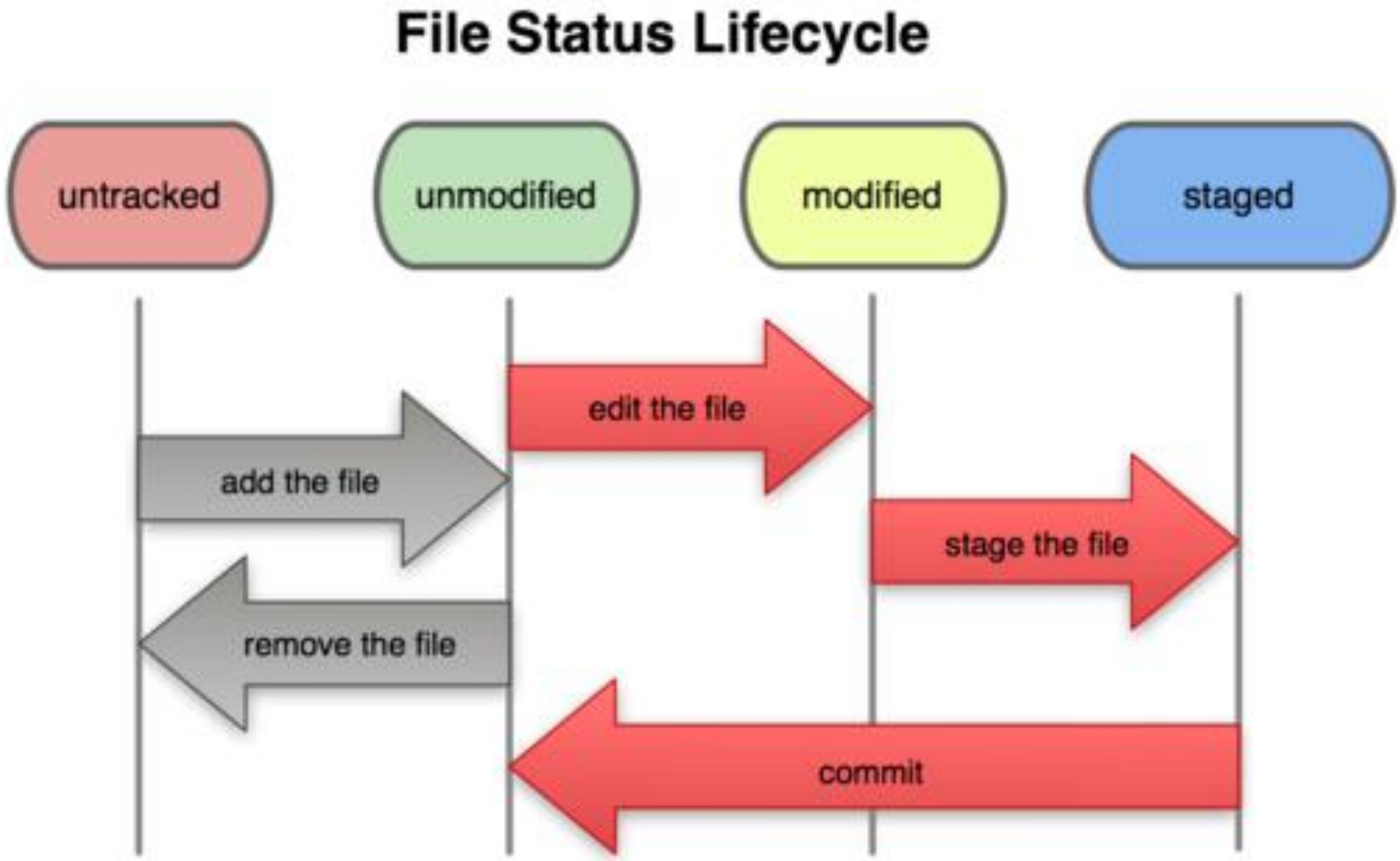
- Online project hosting for Git version control repositories
  - Great way to spread open source code.
  - Free hosting of public repositories, each up to ~1GB
  - Students can get 5 free private repos for 2 years
  - Pricing plans for more serious private repos

- Why use it?
  - Great way to get experience and show off what you can do
  - Keep track of statistics about your projects
  - Easy access to your repos

Git ~ Basics

- Install Git using a package manager or build from source
- Creating a Git repository
  - Create a repository from an existing repository
  - “Clone” an existing repository (remote or local)
  - Creating a new repository
    - git init
    - Set default username and email
      - git config --global “John Doe”
      - git config --global “jDoe123@gmail.com”
  - Clone remote repository (using http)
    - git clone https://github.com/jrivera777/OpenGL-Asteroids.git

Git ~Basics (Cont)



Git ~Basics (Cont)

- Add file to be “tracked”
  - Tracked files will have changes stored during a commit (called staging), and eventual push.
    - git add README.md
  - Untracked files will begin being tracked and are staged.
- Stop tracking files
  - git rm stupidFile
- Commit changes
  - git commit –m “A commit message!”
  - If you want to just stage all tracked changes and commit them
    - git commit –a
    - Not using the –m flag will make a default editor show up asking for commit message. This is useful for more detailed messages.
    - git config --global core.editor lets you change your default editor (to emacs of course!)

Git ~Basics (Cont)

- Check tracked and staged files
  - git status
  - Shows all files that are going to be committed (staged files)
  - Shows files that have been removed and un-staged
  - Shows files that have been modified, but are not staged

- Seeing modifications not yet staged
  - git diff
  - --cached/--staged show staged changes instead

Git ~ Basics (Cont)

- Pushing your commits to a remote repository
  - Changes the remote repository to reflect all changes made during commits on your local copy
  - git push [remote-name] [branch-name]
  - git push origin master
    - Pushes changes (commits) made in your local master branch to the remote origin branch. Cloning sets up these names automatically.
    - If someone else has pushed before you, this command fails (as it should)! Why?
- Pulling and incorporating changes from a remote repository
  - git pull
    - Attempts to retrieve changes in the remote branch. If conflicts occur with commits you made locally, you can resolve them at this point.

Git ~Basics (Cont)

- Still not sure about a command?
  - git help
  - Shows a synopsis of the most commonly used commands
  - git help <command> gives details about a particular command
  - -a lists all commands
- Want more information or examples about git and it’s commands?
- Visit:
  - http://git-scm.com/book
  - http://www.ericsink.com/vcbe/
- Ask PLUG officers and members!

Links and Stuff

- For more info about GitHub and how to use it, visit:
  - https://github.com/
  - For info on free private repos for students: https://github.com/edu
- For Git GUI client (Boo!) downloads as well as tutorials on using Git visit:
  - http://git-scm.com/
- Fun tutorial
  - http://try.github.com