

Using Git



CS/IT 490 WD, Fall 2013

Last update 2013-08-26

Written by Rachel J. Morris
Creative Commons Attribution-NonCommercial-ShareAlike
3.0 Unported License

Breakdown

- What is Source/Version/Revision Control?
- Types of Source Control solutions
- GitHub and BitBucket
- How does Git work?
- Basic Git Commands

What is Source/Version/Revision Control?

- Source Control is a method for:
 - Saving code (and changes to code) to a server periodically
 - Easily allowing merges between multiple peoples' changes to one file
 - Able to “Diff” to see changes between file “Commits”
 - Able to see “Blame” log to see who changed what in a file

What is Source/Version/Revision Control?

- Almost all companies that work with software development will be using Source Control.
 - Small businesses that aren't software/web oriented may not (i.e., a business with a webpage but not about web; may not realize Source Control exists.)
 - Source Control is a must-have tool for developers! It is an important tool!

What is Source/Version/Revision Control?

- The Joel Test: 12 Steps to Better Code

<http://www.joelonsoftware.com/articles/fog0000000043.html>

1. **Do you use source control?**
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?

And so on...

Types of Source Control solutions

- Some common source control solutions are...
 - TFS (Team Foundation Server) – Microsoft's version control
 - SVN (Subversion)
 - Hg (Mercurial)
 - Git
- Mercurial & Git are **Distributed Version Control Systems**

Types of Source Control solutions

- We're not going to get into why you should be using Mercurial or Git instead of SVN.
 - (Unless you really really want to)
- Each solution has its own nuances and methods for doing things.
- If you learn to use Git, using Mercurial and SVN and TFS should be pretty straightforward.

GitHub and BitBucket

- GitHub and BitBucket are websites that will host your Source Control repository for you. Code hosting!
- You might also be familiar with SourceForge if you download a lot of Open Source software.
- CodePlex is Microsoft's source hosting system for Open Source projects.
- Google Code is another.

GitHub and BitBucket

- GitHub gives you **Free Repo Hosting** if:
 - You're a university student OR
 - Your repository is public (Open Source)
- If you're going to host private repos without a student email address, you would have to pay GitHub.
- GitHub student accounts are only free for a limited amount of time (2 years)

GitHub and BitBucket

- GitHub is *very active* when it comes to Open Source community.
- They also have a job posting page.
 - Even though I'm not on the job posting page, sometimes recruiters find me via my GitHub projects.

GitHub and BitBucket

- BitBucket has free private hosting for teams of up to 5 people – Open Source or not!
- You can also host public projects on BitBucket.
- BitBucket lets you host Mercurial OR Git projects. GitHub only lets you do Git.

GitHub and BitBucket

- Other source hosting websites may let you use something besides Git & Mercurial
 - Google Code
 - CodePlex
 - etc.

GitHub and BitBucket

- GitHub and BitBucket hosting have basically the same features in their Web Interface
 - Browse the directory from the web
 - View code “Diffs” (changes)
 - View commit log
 - Setup issue log tracking
 - Wiki page

How does Git work?

- We are not going to cover all of the Git commands, we just need the basics:
 - Clone/Init
 - Add
 - Remove
 - Commit
 - Status
 - Push
 - Pull

How does Git work?

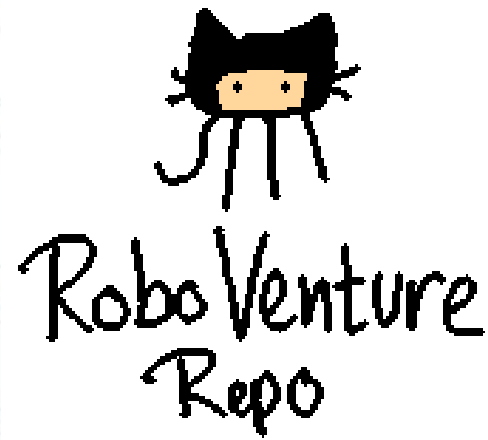
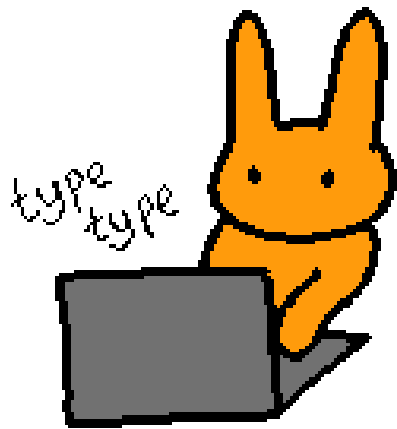
- First, let's assume you're working alone on a project. Source Control is handy whether you're working solo or in a group.



Project:
RoboVenture!

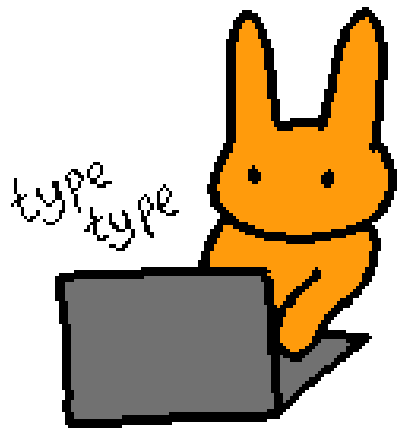
How does Git work?

- You create a repository, which you can host on your own server or on a service like GitHub.



How does Git work?

- Any time you add new or change existing source files, you **add** them with the `git add` command. However, this won't automatically make them show up on the server.

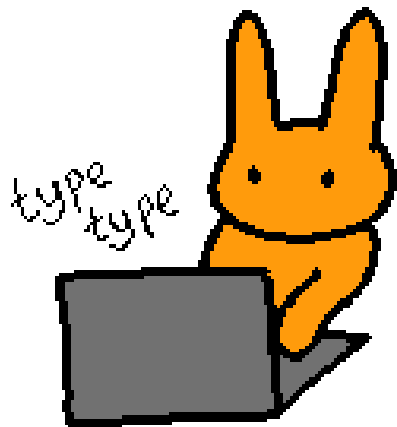


main.cpp
robot.cpp
robot.h



How does Git work?

- Instead, adding them is like adding these files to a package. This package will later on be pushed to the server all at once.

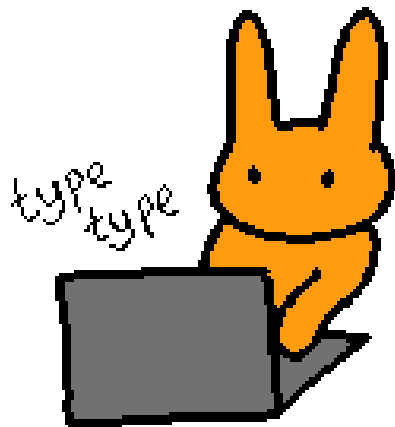


main.cpp
robot.cpp
robot.h



How does Git work?

- When you're done adding all the new or changed files, you will use the `git commit` command to finalize this package. You will be required to give it a label.

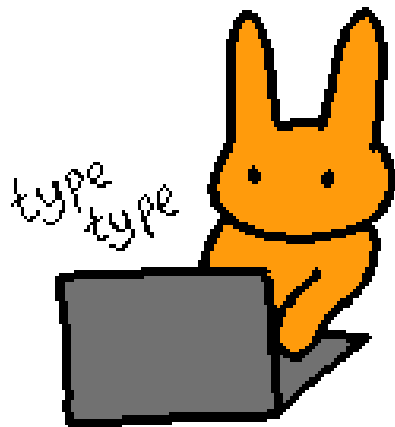


main.cpp
robot.cpp
robot.h



How does Git work?

- You might commit when you finish working on a small feature or task; “Added robot explosion effect” could be a sample commit message.
- The changes made to the files added will be logged.

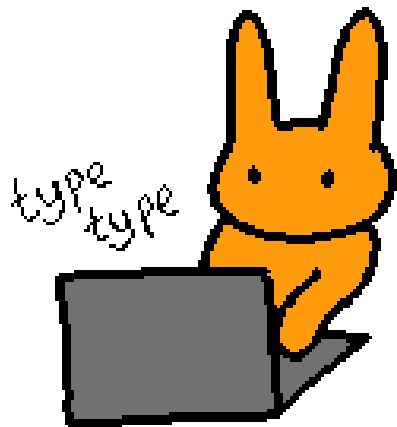


main.cpp
robot.cpp
robot.h



How does Git work?

- Even when you've made a commit, you've merely packaged your changes.
- Your files still aren't showing up on the server yet!

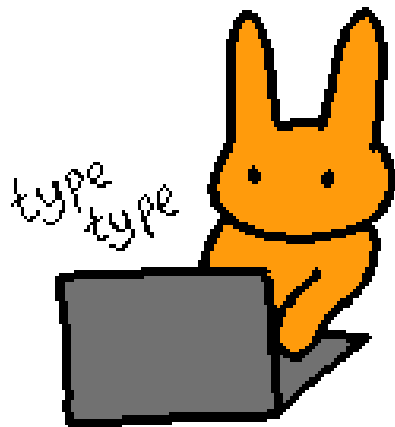


Commit
main.cpp
robot.cpp
robot.h



How does Git work?

- So, when you commit, you've packaged changes that are local to you; they're only on your machine.
- To push your changes to the server, you use the git push command.

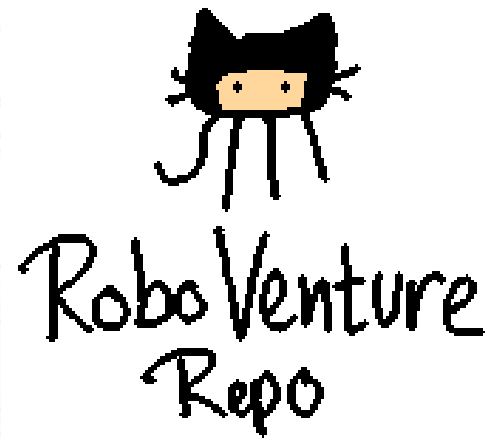
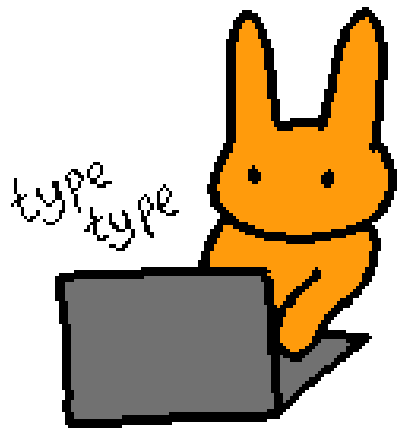


main.cpp
robot.cpp
robot.h



How does Git work?

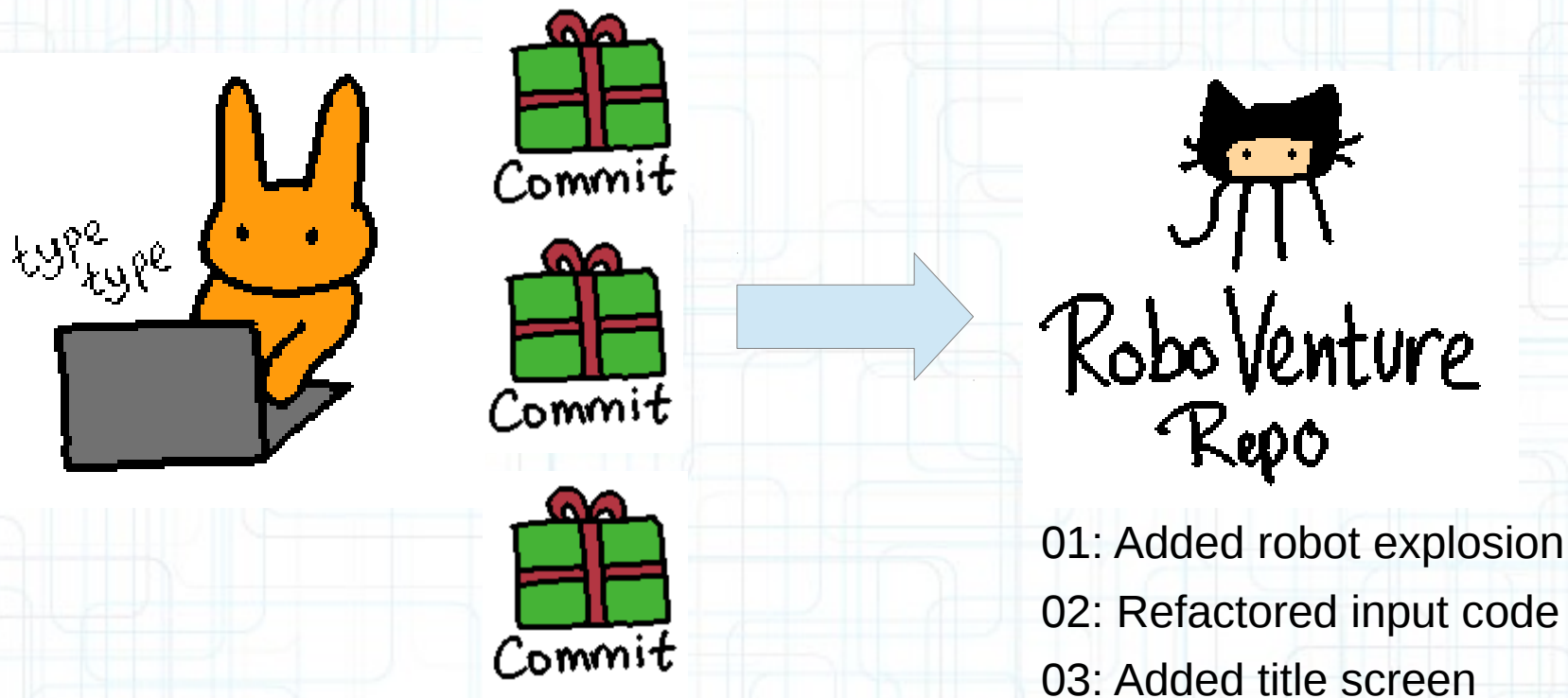
- Every commit will now be listed under the “commits” section of the webpage, you can look at changes and compare them between different commits.



- 01: Added robot explosion
- 02: Refactored input code
- 03: Added title screen

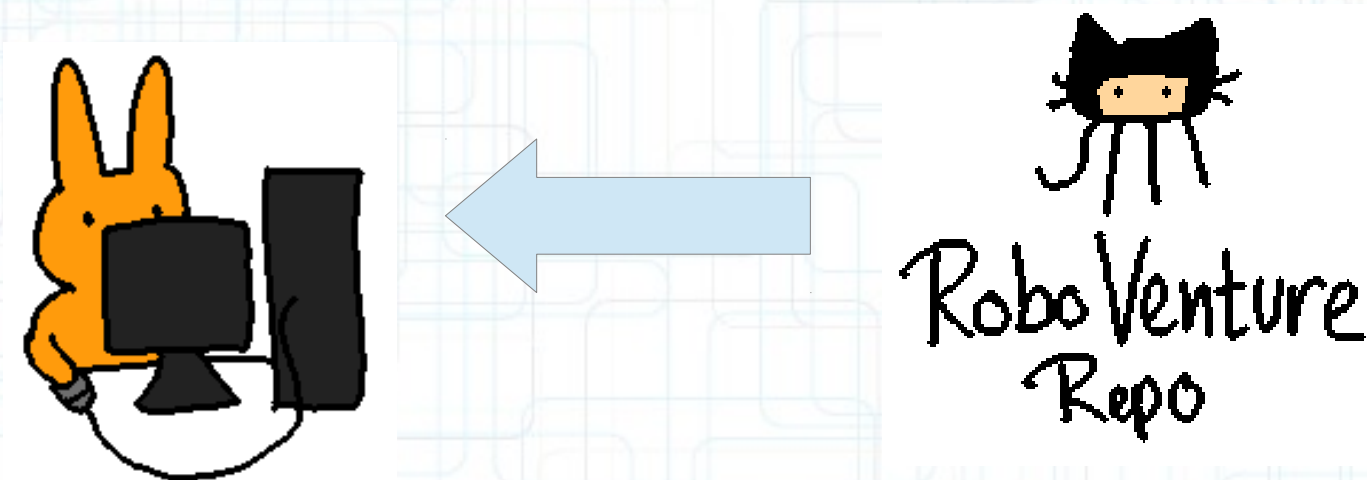
How does Git work?

- You don't have to push immediately after committing. You can create a bunch of commits and then push them all at once.



How does Git work?

- You can also go to another machine and pull down all those changes with `git pull`, enabling you to work anywhere and get the most updated code.



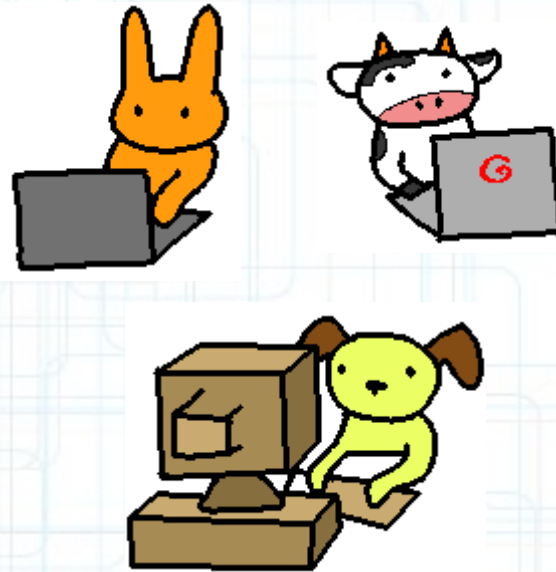
01: Added robot explosion
02: Refactored input code
03: Added title screen

How does Git work?

- **git add:** Adding a file to the commit. This can be a new file, or it can be a file that has changed since the last commit.
- **git commit:** Assign a “commit message” to this package of changes.
- **git push:** Push any commits you have set up to the repository (server).
- **git pull:** You or other teammates can pull the latest code from the server.

How does Git work?

- What about working on a team? Multiple people might be changing the same files simultaneously.



How does Git work?

- As everybody works, the files they add and commit are local only to their machines. Nothing is affected on anyone else's computers.
- Everyone's code is whatever is latest on the server. What is on the server ought to be stable (Commits are good for Work In Progress tasks!)



main.cpp
npc.cpp



main.cpp
level.cpp



logger.cpp
debug.cpp

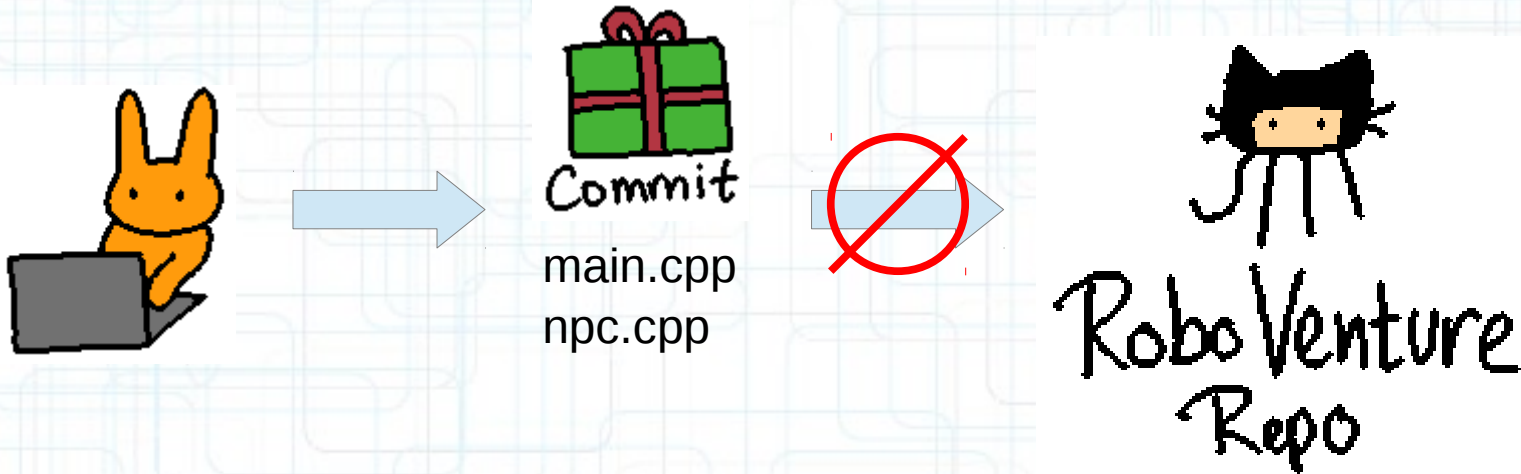
How does Git work?

- After a task is complete, commits are ready to be pushed to the server.



How does Git work?

- If someone pushed their updated main.cpp before you did, and you go to push your changes to the server, Git will stop you.



How does Git work?

- You will get a message back saying that the file has been updated, so you need to run `git pull` and merge your changes with the latest changes before you can push to the server.



main.cpp
npc.cpp

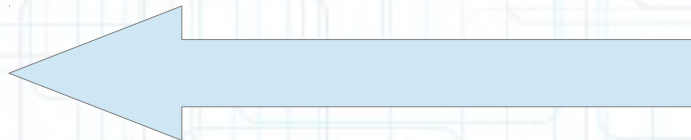


How does Git work?

- Generally, merges will happen automatically.
- Git might not be able to merge changes if, say, one person deletes a function but another person updates it.
- If there is a merge conflict, you have to fix it manually.



main.cpp
npc.cpp



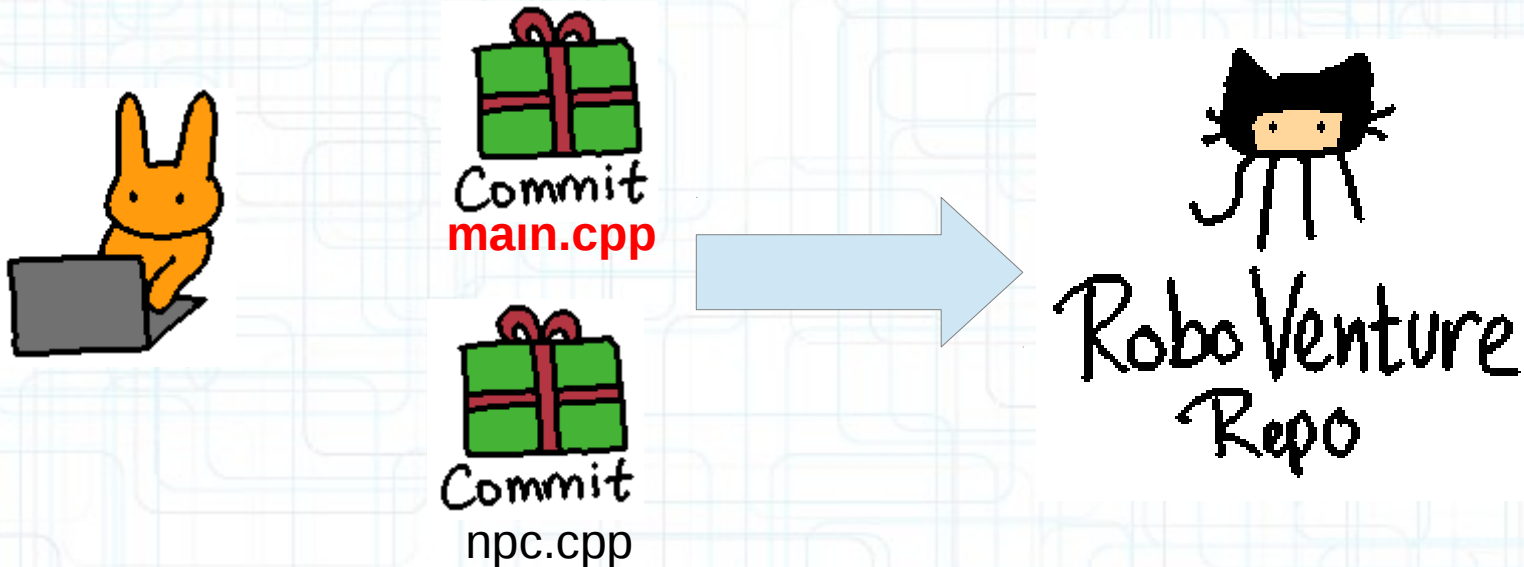
main.cpp



Robo Venture
Repo

How does Git work?

- Once everything is merged, Git will make you create another commit. This commit package is basically just a “these two files were merged”.



How does Git work?

- Afterward, the server will have the latest code and everybody can continue working.



Basic Git Commands

- You can also branch your code and merge branches together, but we're not going to get into this.
- If you know how to use it, you're free to do so (make sure your team knows how to as well!)

Basic Git Commands

- See the “Git Reference” document for a list of commands, and the 2nd practice assignment (which we will do in class).
- You can also watch this video walkthru of using Git:
<http://www.youtube.com/watch?v=IjgLXhzHeMA>

References

-