Using Git and Github

Using our tools especially version control

Assignment

- After these slides you have an assignment:
  - Create a free github account
  - And send me the account name via email before the next class
  - This will go in your quiz grade

Caveat

- I took these screenshots before the semester began on my linux desktop
  - There may be some minor differences between what you see here and what you see on your own screen (I’ve noticed that Windows and Mac sometimes move the buttons to different locations, and linux will sometimes put it in the mac location and sometimes in the windows location)
  - But what you see should be close to my screen shots

Version control in this class

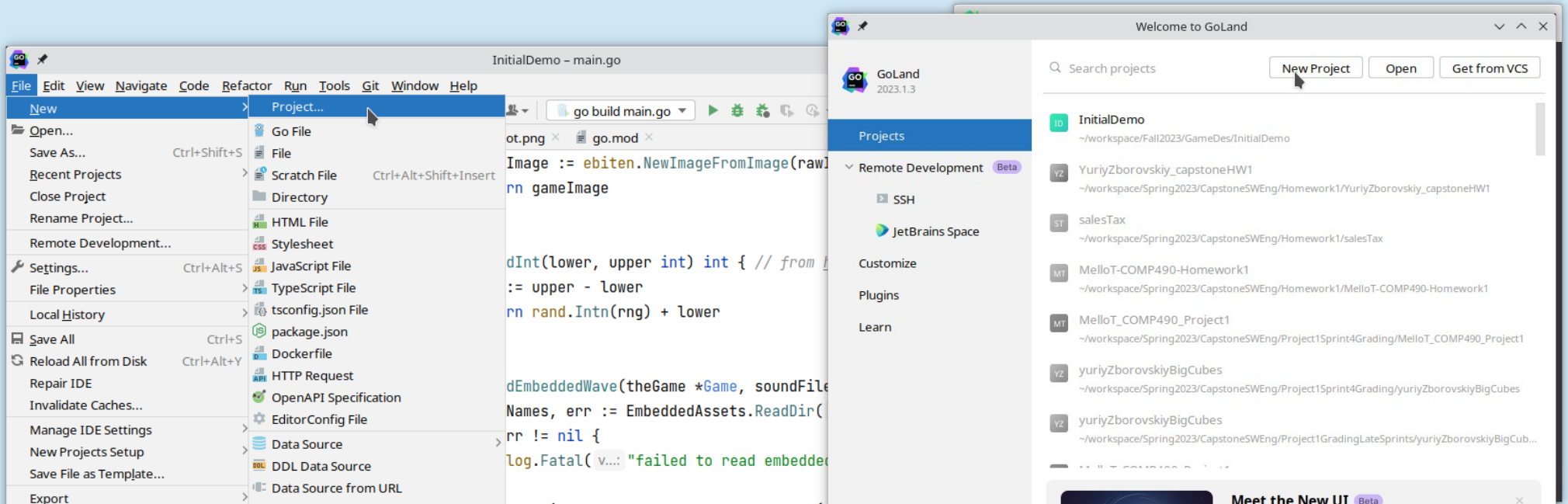
- Git as our version control
  - Installed on linux when you start
  - On mac either use homebrew or let goland install the one from the xcode command line tools (more than just xcode – and it takes a while to install)
  - On windows download from git website and install
    - Or use WSL (windows subsystem for linux) 2
  - Github as our cloud based version control
    - Good jetbrains integration

Let’s Start

- We’ll assume that your tools are installed. This set of instructions is how you should create each new lab, and then add it to the version control systems.

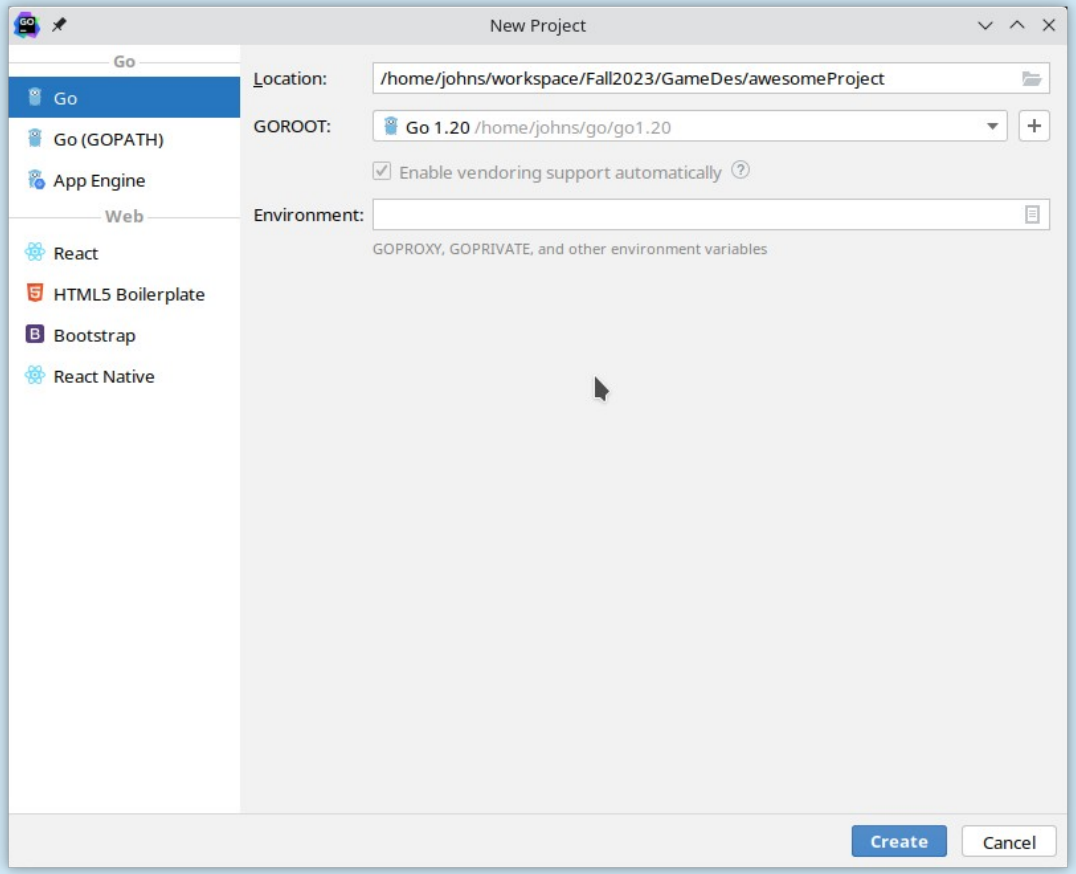
Create a new Project

- Two easy ways
- With prev project open
  use menu
- with no project open
  use new project button



Name your new project

- Erase “AwesomeProject” and replace it with your project name in the top textbox
-



Version control

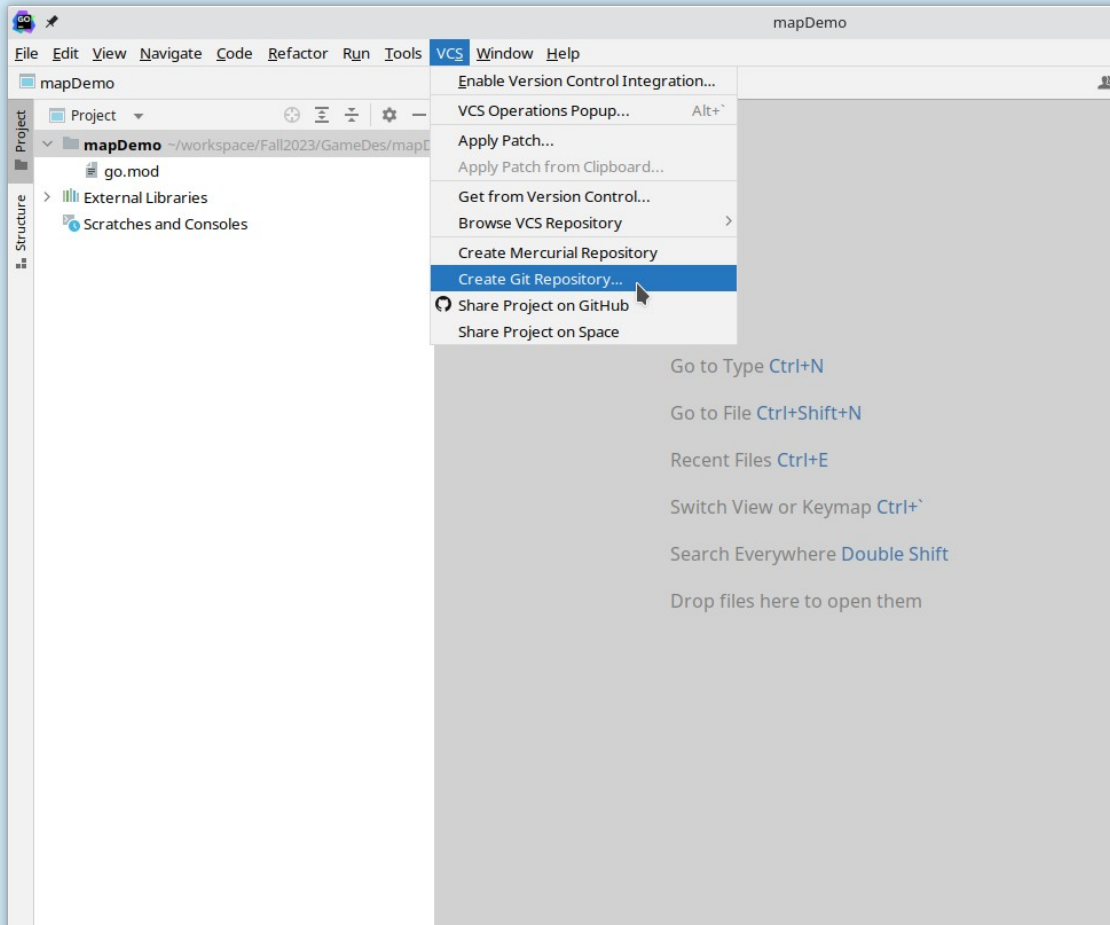
- Version control software
  - You are all upper division students, What is version control and version control software?
-

Version control

- Version control software
  - Something all (modern) software development uses
  - Keep track of changes in software
    - Not just ‘last save’ but basically every save (or commit as version control calls it) is saved and available at later dates to look back at.
  - Really valuable
    - I wish I could go back to the way it was an hour ago.
    - (as long as you committed an hour ago.)
- We will use git as version control in this class
  - Like almost every project started in the last 3+ years

Add to version control

- In goland select the VCS menu and choose “Create Git Repository“
- It should bring up a file dialog with your project folder selected, choose OK
- At this point your VSC menu turns into a Git menu

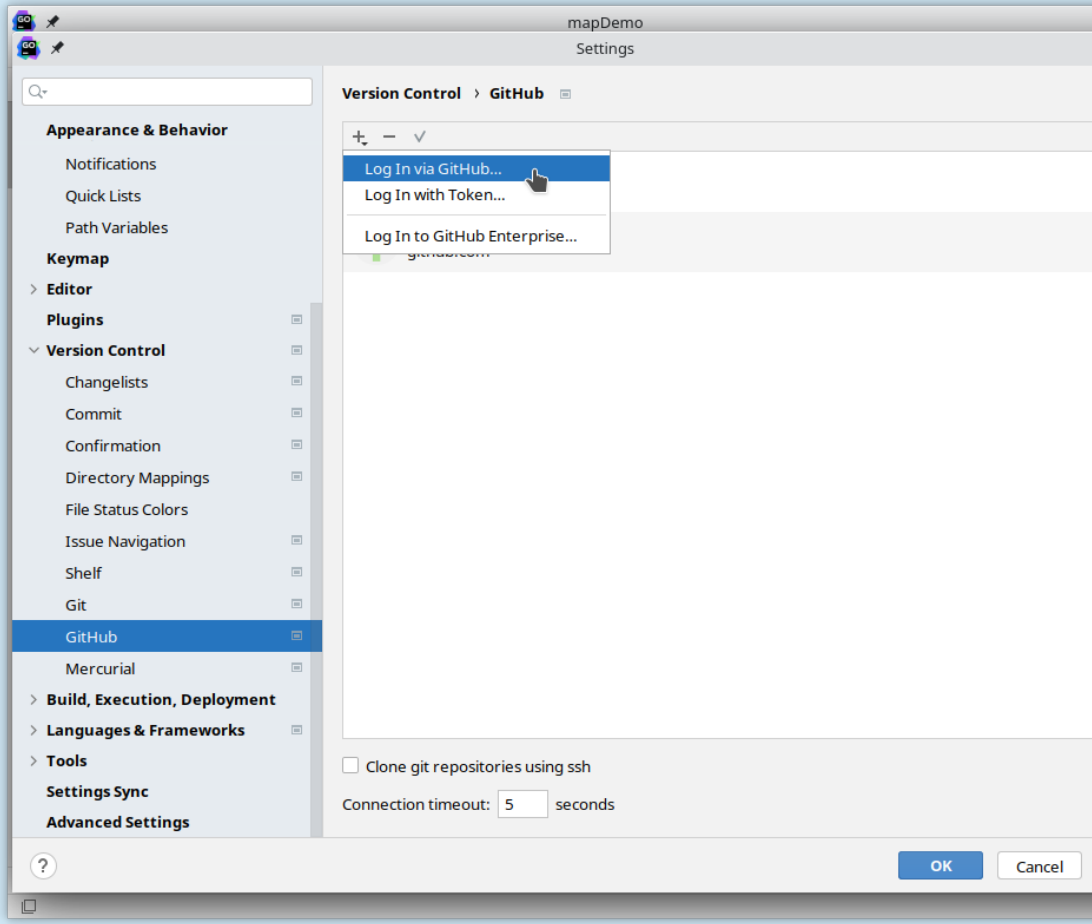


Git vs Github

- Git is the version control software
  - It keeps all of the committed versions of your program so you can go back to them
- Github is a cloud based service that will store git repositories in the cloud
  - Github owned by Microsoft is the biggest and most common, but others exist (gittea, gitlab etc)
  - Github will act as the backup and allow you to keep a copy off of your laptop
  - Also how you will share your code with instructor (that's me)

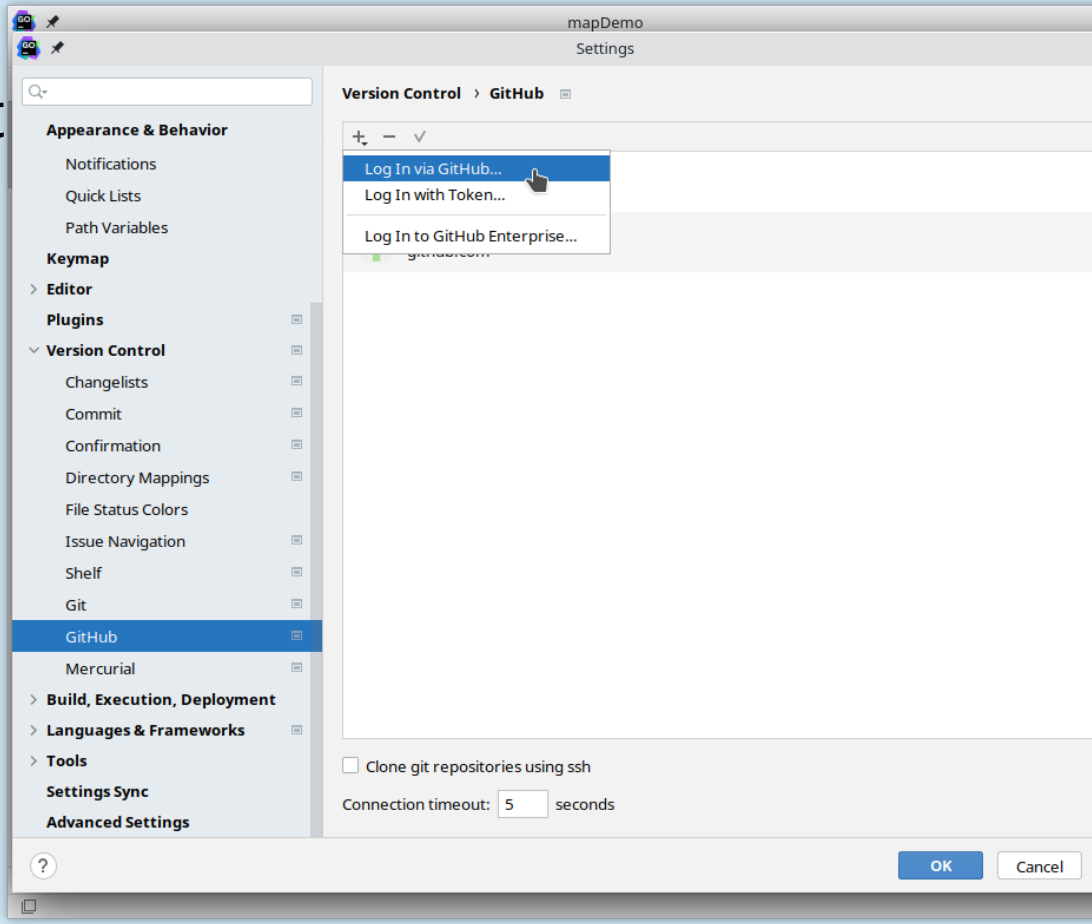
Setup Jetbrains for Github

- If you have never used goland with github, need to generate a token
  - linux/windows
    - <file><settings>
  - Mac
    - <pycharm><preferences>
  - Select GitHub from the version control section, press the'+' button and choose "login with GitHub"



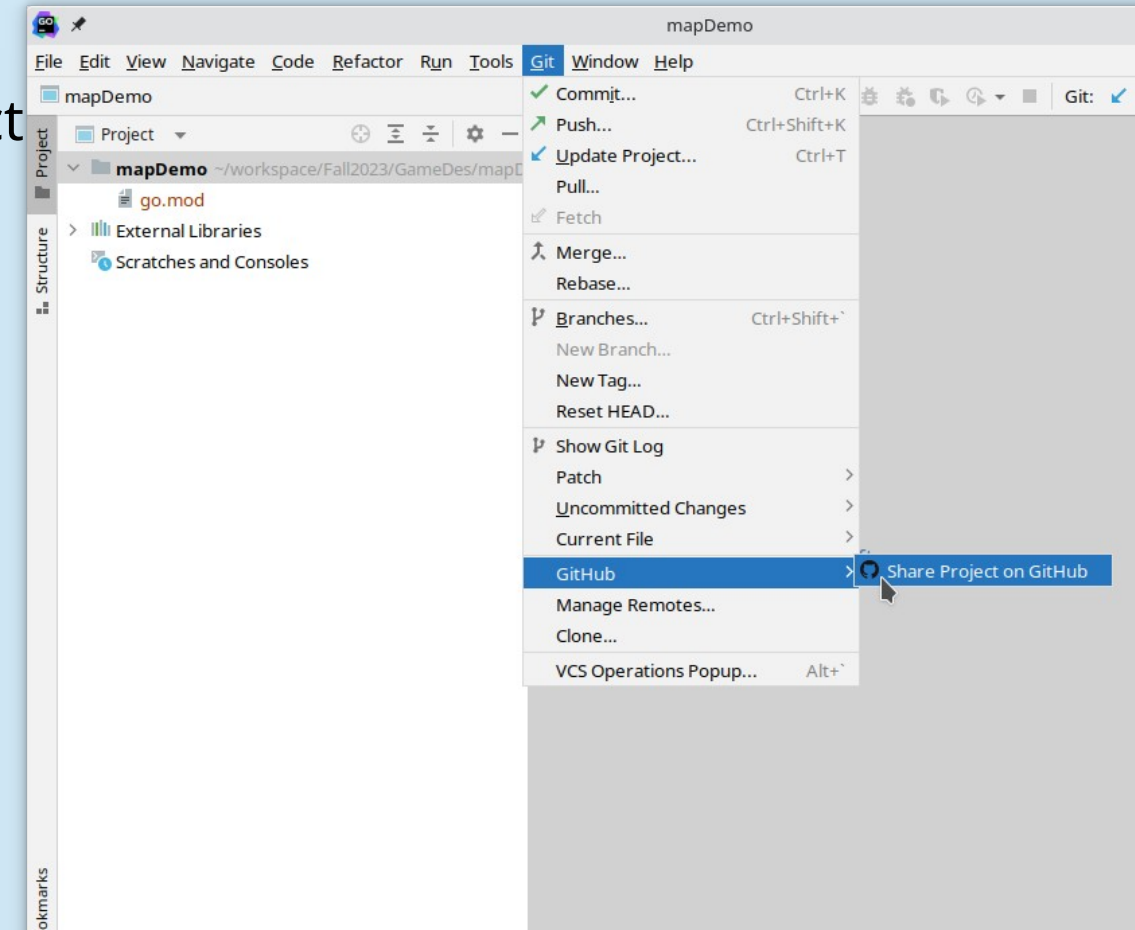
Setup Jetbrains for Github

- Login with github will bring up a jetbrains page in your default browser
- If you are logged into github you should be told to go back to goland and be all set
- Otherwise you'll need to login to github, then as above



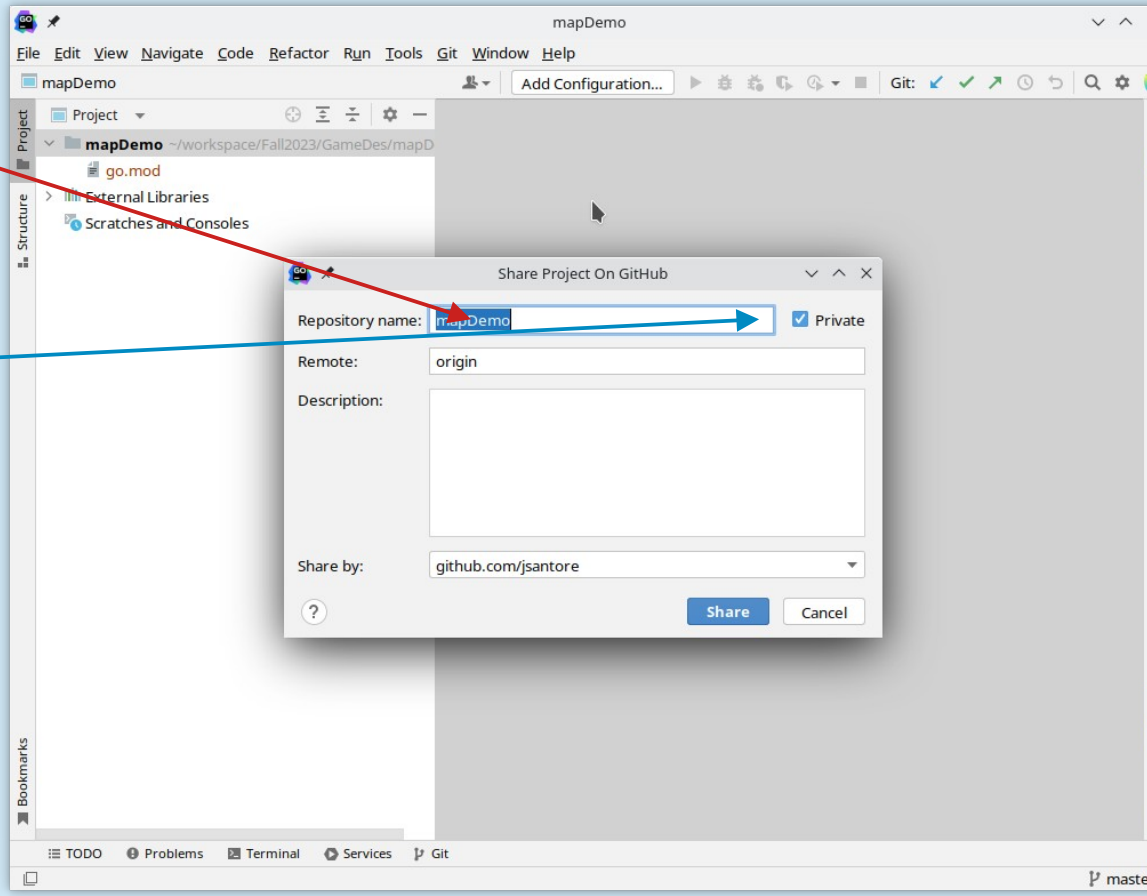
Share Project on github

- From your new git menu
  - Choose <github><share project on Github>
  - This will put the project on github (after naming – see next slide)
  - There may be some setup the first time



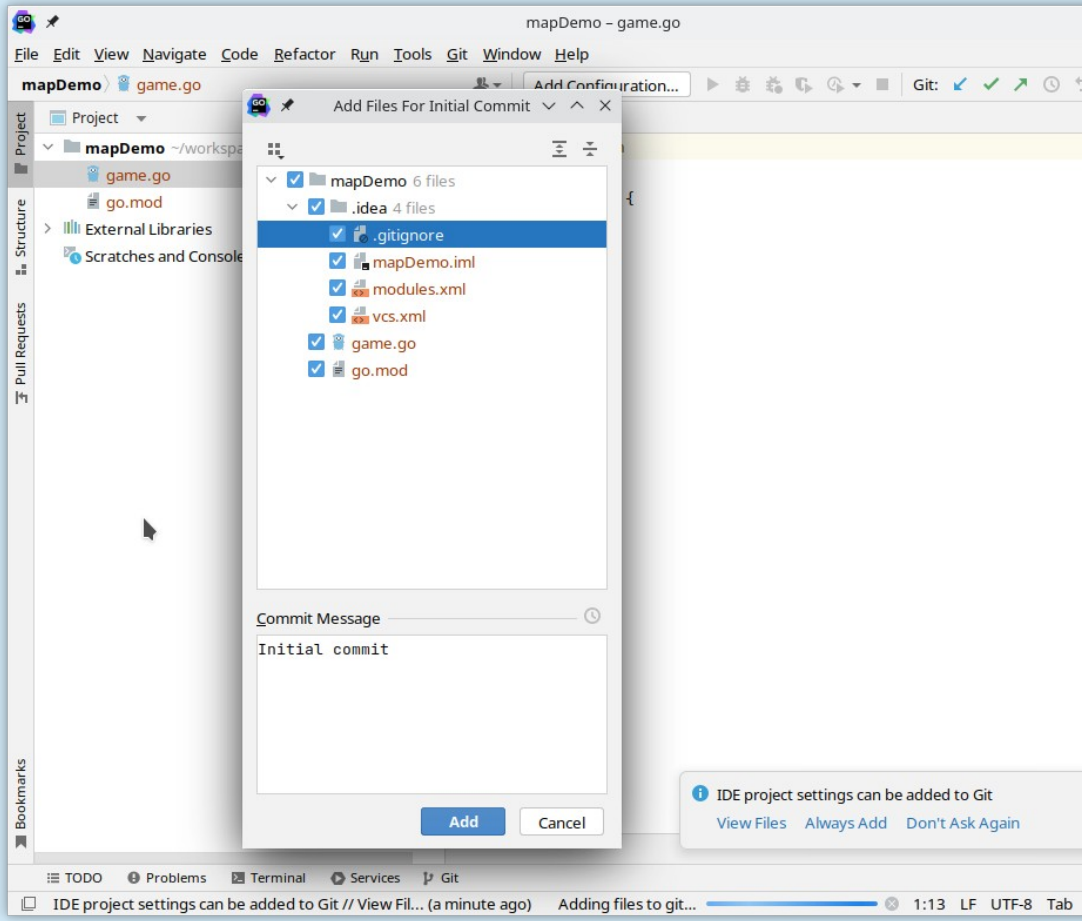
Naming Your project

- When you share on github
- Give project name not yet used on your github
- For your graded projects tick the “private” box till graded
- For non-graded work you can leave the private unchecked



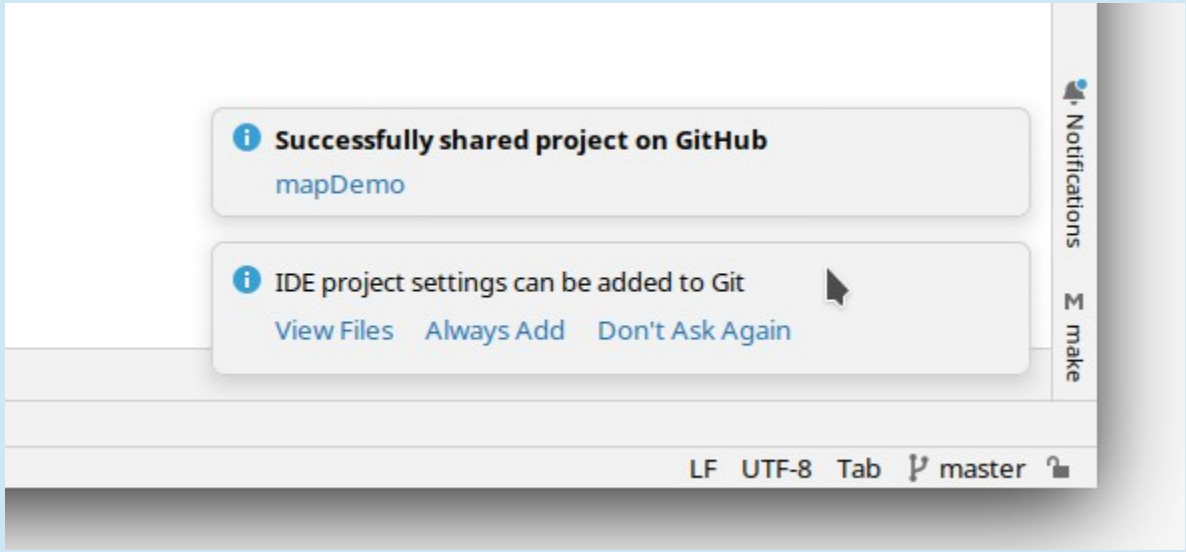
Choose Which files

- Next you much choose which files will be uploaded to github
- And press add
  - It is almost always best to just go with the defaults suggested by goland
  - By default files which shouldn't go up to another machine are highlighted in yellow and not suggested for upload



Success!

- If all went well you should see something like this in the lower right corner

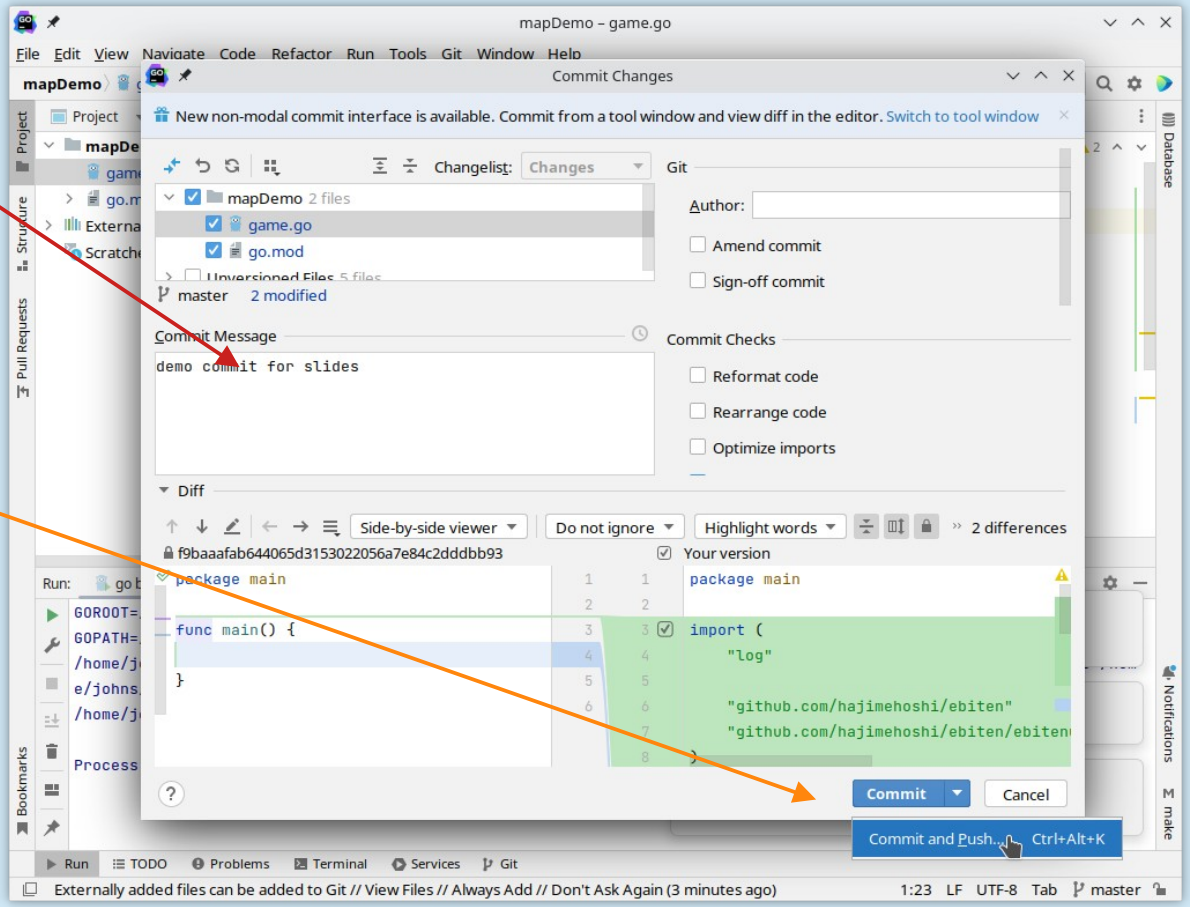


Making changes after you add

- In real software development,
  - you need to keep improving things
  - Often need to make changes after initial share
  - Called a commit (git’s analog to saving a file)
  - Choose commit from the git menu
    - Only available after some change in file
  - This will bring up commit screen
    - See next slide

Commit dialog

- Put in a commit message
  - To tell you (and me) what this save was all about
- Then pull the commit menu down and choose “Commit and Push”
  - This will save it both locally and to github
    - Push dialog comes up next
    - Just press push



REMINDER Assignment

- After these slides you have an assignment:
  - Create a free github account
  - And send me the account name via email before the next class
  - This will go in your quiz grade