Introduction to Git

Enabling collaboration and code sharing

What is git?

- It is a version control software
  - Stores multiple versions of your files/code
  - Each version is defined as the changes from the version before it

- Often a distributed system to prevent data loss

- Websites built around git to allow for easier code sharing and collaboration
  - i.e., GitHub, GitLab, and BitBucket

Some vocabulary used in git

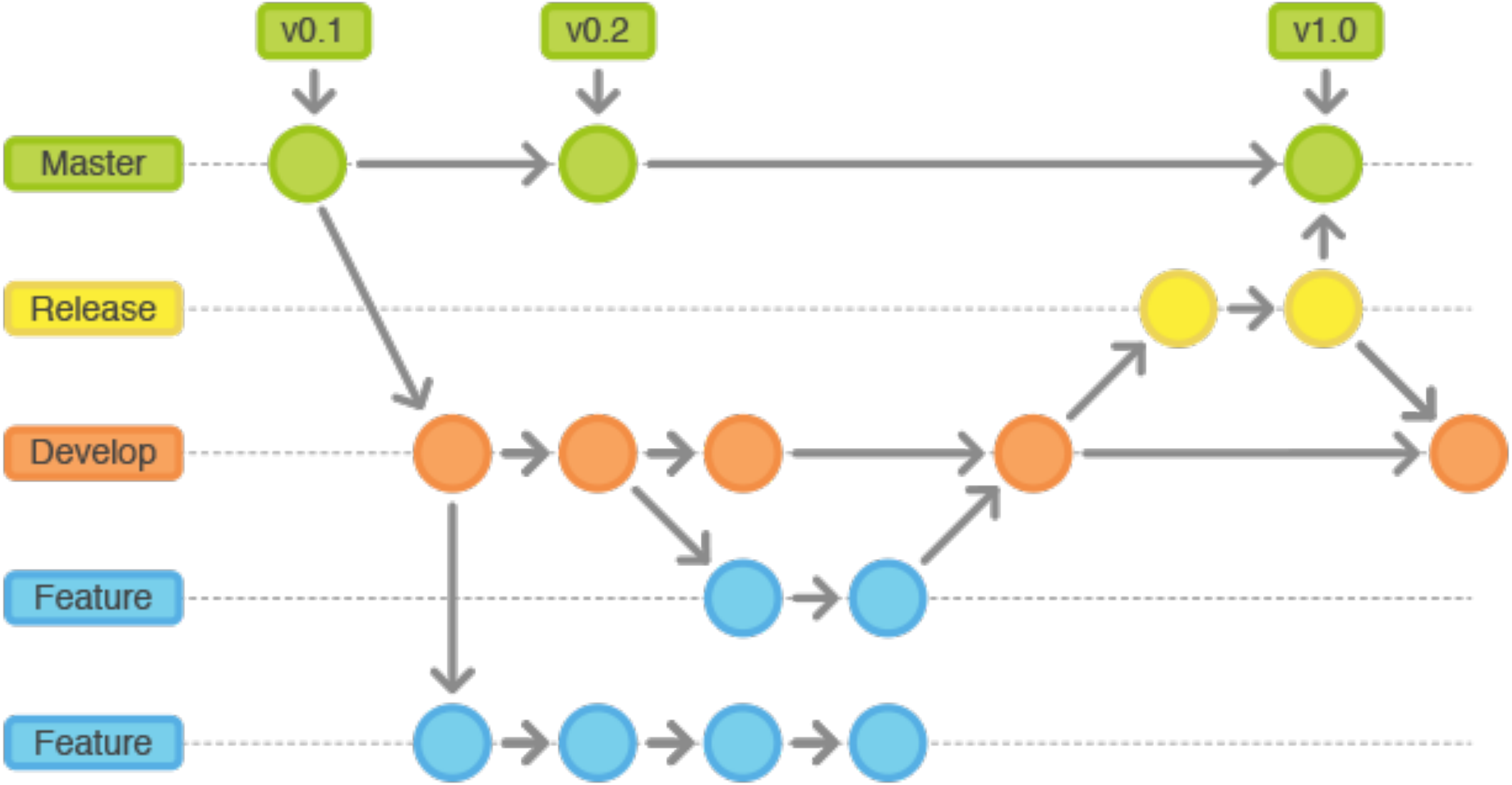
- Repository – A git project (often shortened to repo)
- Origin – The location where a repo is stored, for example a GitHub link
- Clone – Downloading a repository for the first time
- Fetch – Checks for any new changes in the repo
- Pull – Downloads new changes to your local repo
- Add – Stages (prepares) changes you want to upload
- Commit – Creates a new version with a title and description to indicate what was changed
- Push – Upload all the commits you have created to the origin (or elsewhere)

Branches

- A way to improve organization during collaboration
  - Can separate features into branches while working, and merge it into the main branch when done
  - Merging one branch into another means that all changes from the branch will be copied from one branch to another

- Each repo always has a default branch called main

- More branches can be created as needed

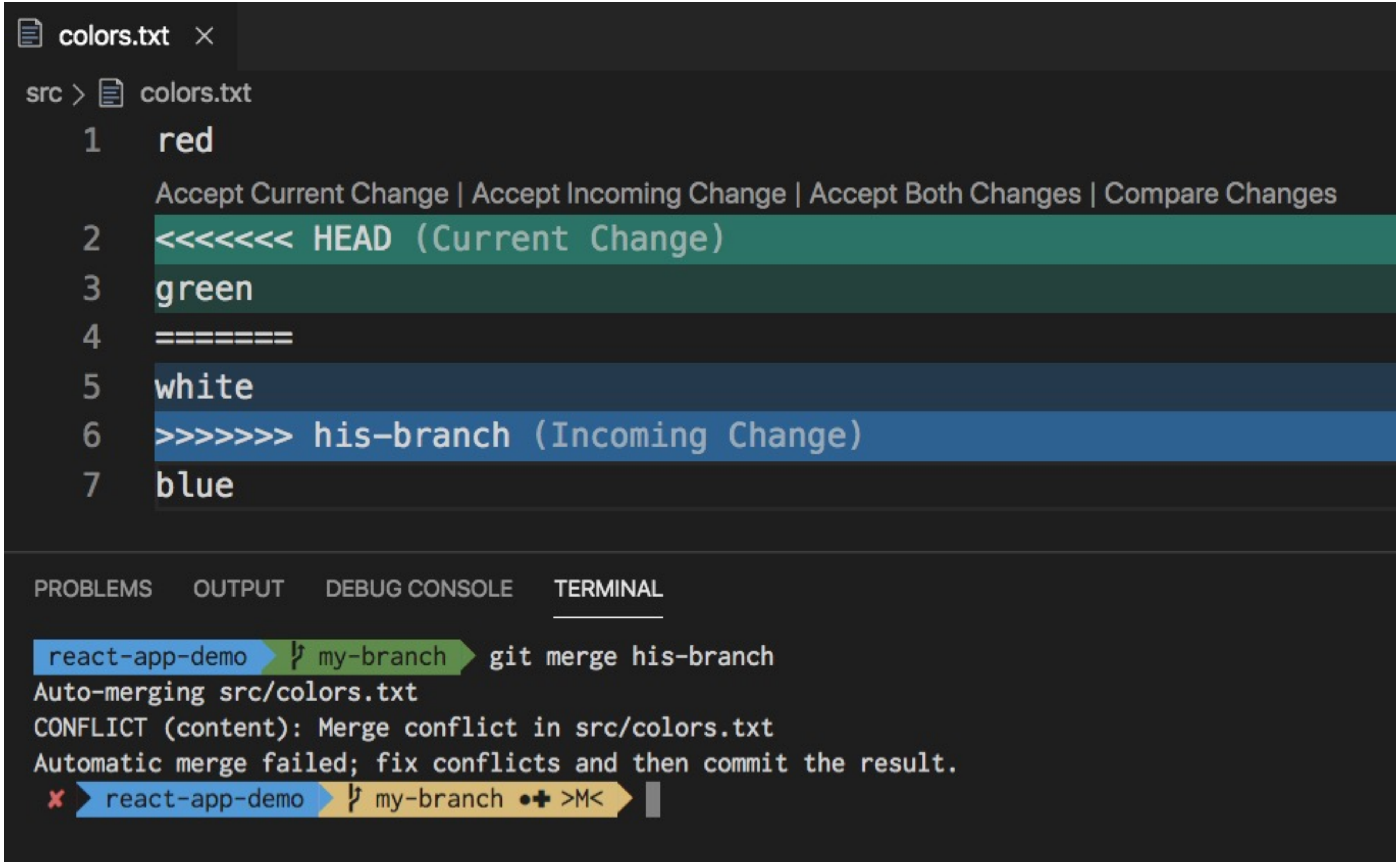

v0.1
v0.2
v1.0
Master
Release
Develop
Feature
Feature

Merge conflicts

- Sometimes you may encounter merge conflicts when merging branches, or pushing changes

- This happens because multiple commits change the same line(s)

- To resolve a merge conflict, you choose which version to keep, or combine the versions


colors.txt
src > colors.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

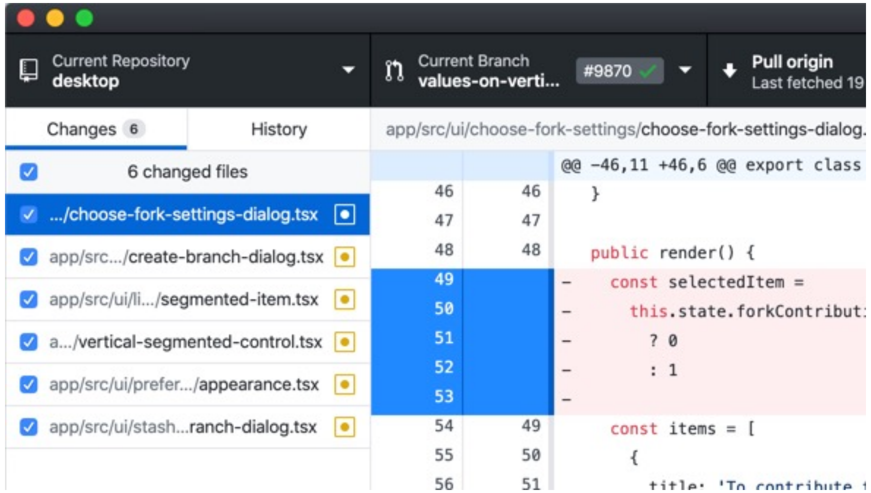
```
red
<<<<<<< HEAD (Current Change)
green
=======
white
>>>>>>> his-branch (Incoming Change)
blue
```

```
react-app-demo  my-branch  git merge his-branch
Auto-merging src/colors.txt
CONFLICT (content): Merge conflict in src/colors.txt
Automatic merge failed; fix conflicts and then commit the result.
 ✘  react-app-demo  my-branch ●✚ >M<
```

GitHub features for collaboration

- Issues – Create a post that describes a bug, or feature that should be added to the repo. Issues can be discussed and resolved by users
- Pull requests – Someone makes a request to the repository owner(s) for them to merge some code, often to fix an issue. These can be reviewed; resulting in acceptance, requesting changes, or denial. This often results in better code quality due to the code review
- Projects – Projects can be created to track tasks and issues for the repo. The collaborators can also indicate what they are currently working on

Using git



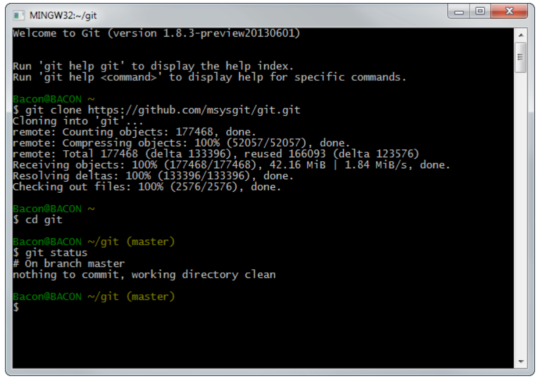
- ☑ .../choose-fork-settings-dialog.tsx
- ☑ app/src.../create-branch-dialog.tsx
- ☑ app/src/ui/li.../segmented-item.tsx
- ☑ a.../vertical-segmented-control.tsx
- ☑ app/src/ui/prefer.../appearance.tsx
- ☑ app/src/ui/stash...ranch-dialog.tsx

app/src/ui/choose-fork-settings/choose-fork-settings-dialog.

```
@@ -46,11 +46,6 @@ export class
   }

   public render() {
-    const selectedItem =
-      this.state.forkContribut
-        ? 0
-        : 1
-
     const items = [
       {
         title: 'To contribute
```

- Graphical client
  - Easy to use, but limited in features
  - Many different clients, simplest would be to use the one from GitHub



```
Welcome to Git (version 1.8.3-preview20130601)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

- Command line
  - Requires knowing the commands to use
  - Able to do more advanced actions compared to graphical
  - Can use dedicated git command line or install it for you preferred command line
  - Will look at this option more in-depth on day 3 of the course

Resources

- Missing semester (git) – A short course that goes into more detail on git by MIT https://missing.csail.mit.edu/2020/version-control/
- Learn git branching – An interactive website for learning more about branching and trying some of the commands https://learngitbranching.js.org/
- GitHub education pack – Many tools and credits you can get with GitHub using your university email https://education.github.com/pack
- Git cheat sheet – Many of the most useful git commands in two pages https://education.github.com/git-cheat-sheet-education.pdf

Bonus Task

- Use git to clone all the course repositories
- Alternatively, you can fork them to your own GitHub user in order to be able to push any changes