Part 2

Moving HEAD

Moving HEAD

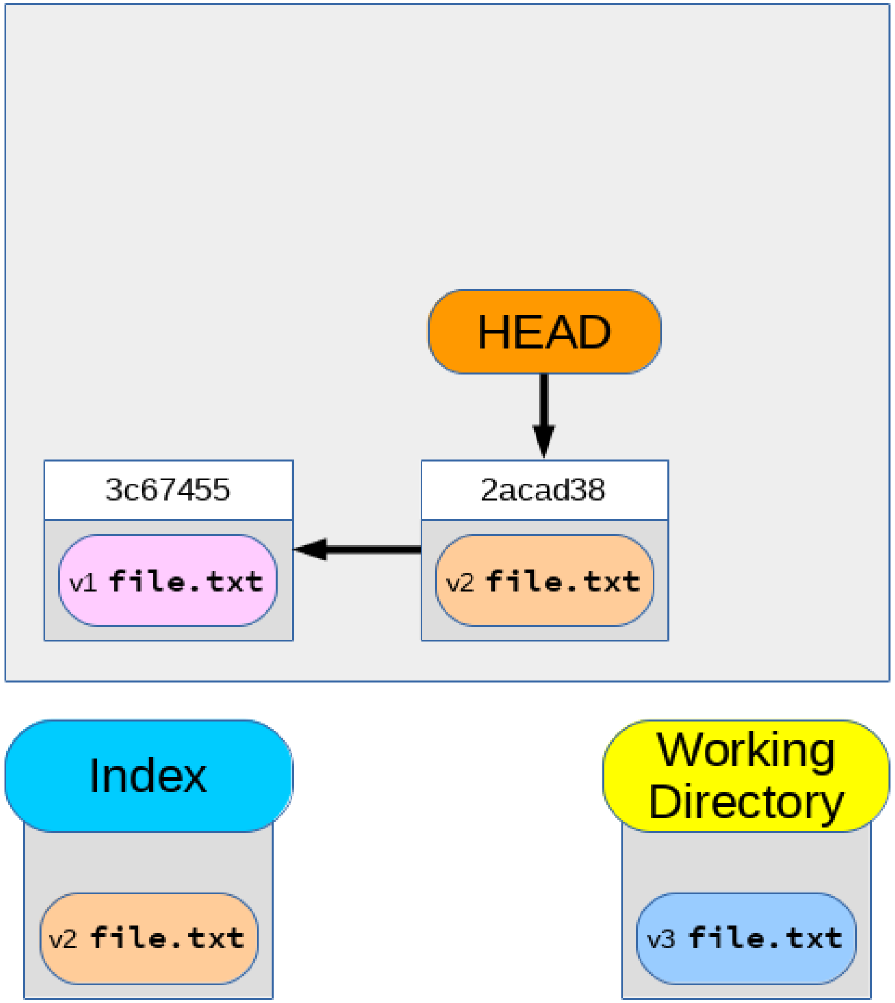
`git reset`

- Moves `HEAD` (mostly)
- Changes Index and working directory (mostly)

Seeing the log

`gitk` shows it graphically

- `$ git log`
- `$ git log --oneline -n 2`



Mixed reset

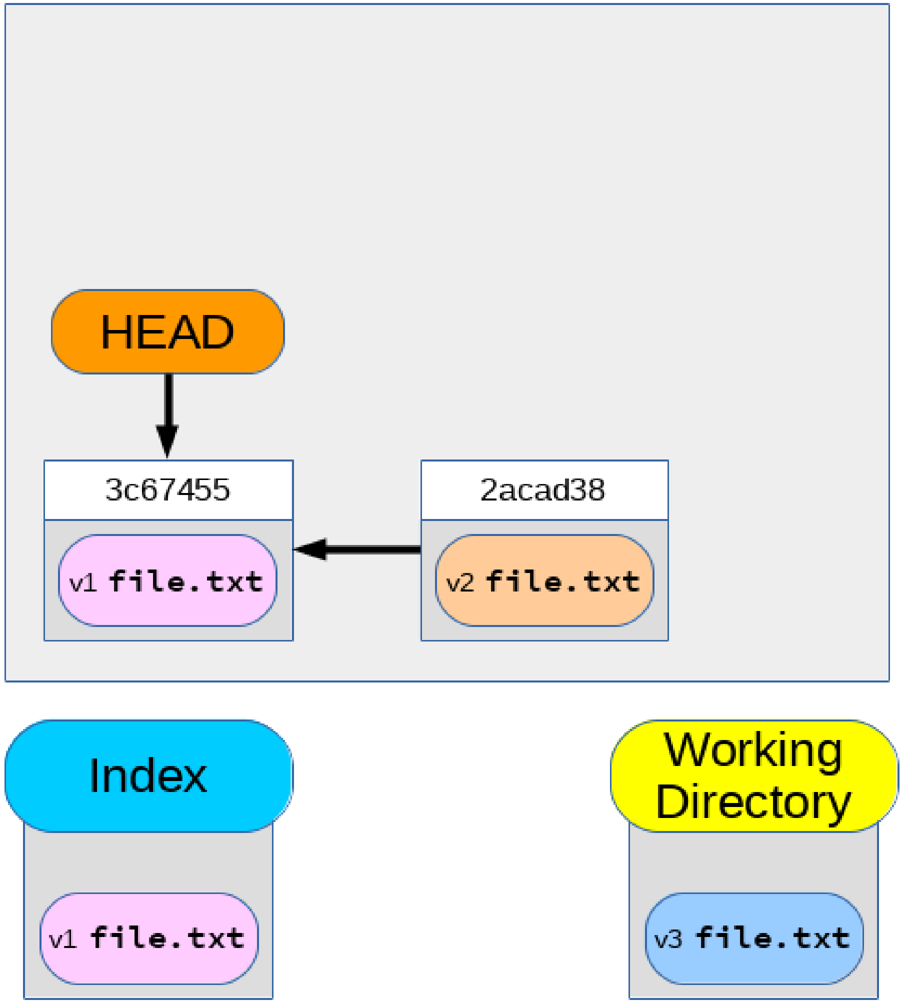
```
$ git reset HEAD~ --mixed
```

- Default behaviour

1. Moves **`HEAD`** to specified commit
2. Changes Index
3. Preserves working directory

Back to the position where you'd just committed **`HEAD~`**

- Undoes **`commit`**
- Undoes **`add`**



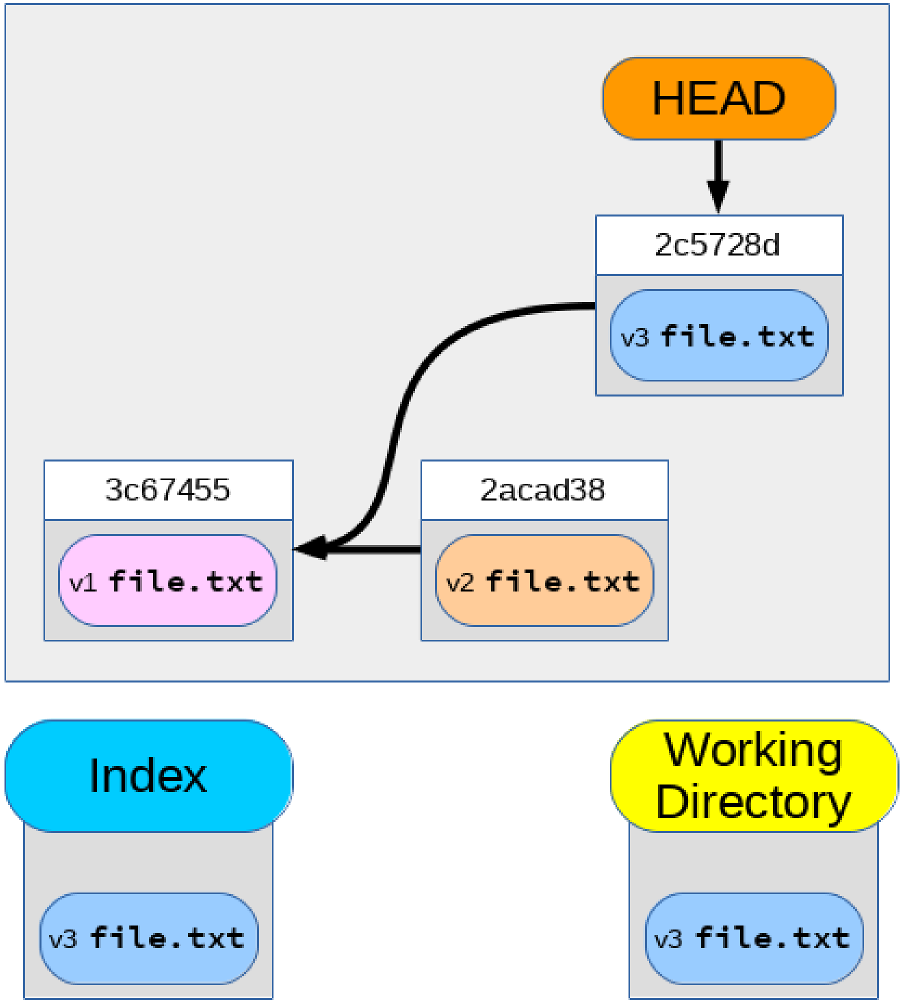
Mixed reset: why

```
$ git reset HEAD~
```

Get rid of a blind alley.

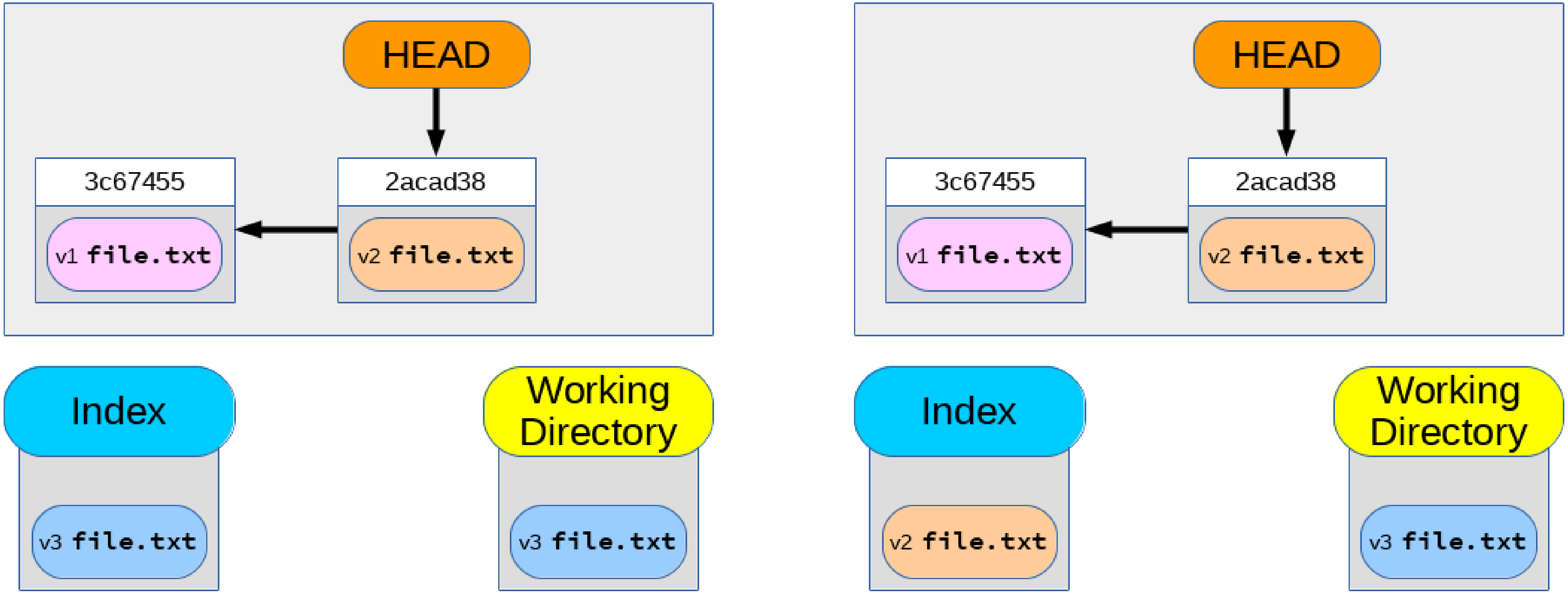
Rewind history

- Can rewind several commits
- Orphaned commit 2acad38 will eventually be tidied away by Git



Unstage a file: undo add

```
$ git reset HEAD file.txt
```



Soft reset

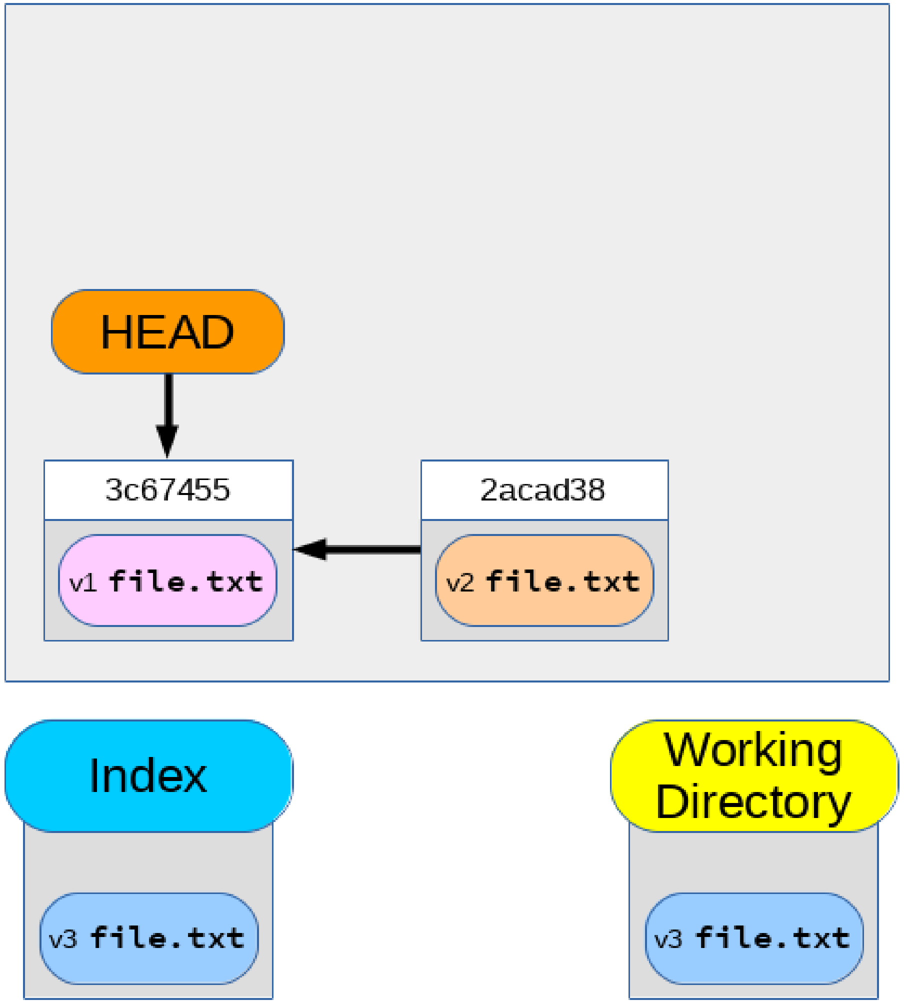
```
$ git reset HEAD~ --soft
```

1. Moves `HEAD` to specified commit
2. Preserves Index
3. Preserves working directory

Back to the position where you'd just committed `HEAD~`

- Undoes `commit`
- Keeps `add`

Useful for skipping intermediate commits



Hard reset

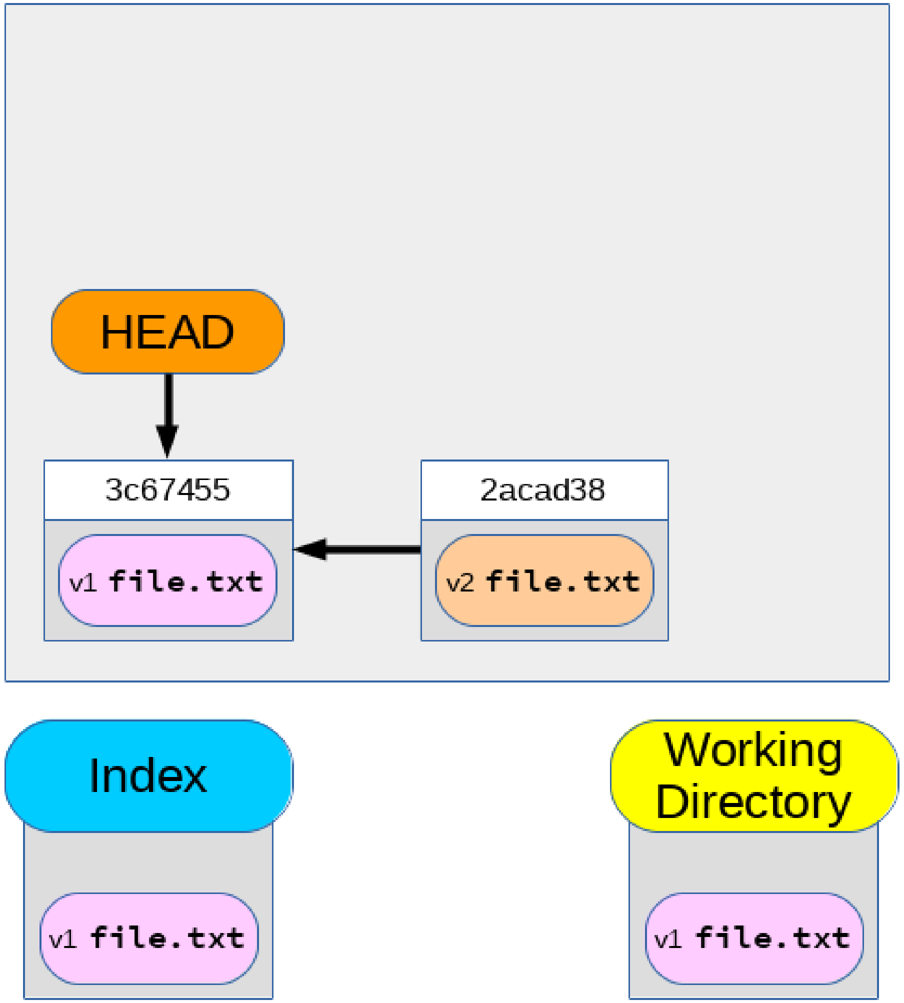
```
$ git reset HEAD~ --hard
```

1. Moves `HEAD` to specified commit
2. Changes Index
3. **Changes working directory**

Back to before you'd even *thought* about moving on from the new `HEAD`

Use case: clearing up disasters

```
$ git reset HEAD --hard
```



Tracked, untracked, ignore

- `git add` tells git to:

  1. Track a file
  2. Add it to index

  Tracked files are tracked forever

- `.gitignore` is a file describing things Git will ignore

  - Won't track them, won't moan
  - Minified files, images, etc.

End of part 2

- Moving `HEAD`
- Mixed, soft, hard `resets`

Hard reset to your third "Adding a section" commit

Take a break, get some coffee