# GitUI

Fast terminal git UI

**START HERE**

- **1-5** Switch tab
- **f** Pull
- **Enter** Stage / unstage file
- **b** Branch list
- **c** Commit staged changes
- **h** Help popup
- **p** Push
- **q** Quit




Status tab: files left, diff right

## CONCEPTS

Press h anywhere for a help popup that lists the keys valid in the current view. The bottom bar shows the most useful ones; . toggles it.

Capital letters mean Shift+letter (D is Shift+d). Keys can be remapped in key_bindings.ron.

## TABS AND NAVIGATION

- **1** Status: working tree and index
- **2** Log: commit history
- **3** Files: browse the tree at HEAD
- **4** Stashing: create a stash
- **5** Stashes: list of stashes
- **Tab** / **Shift** + **Tab** Next / previous tab
- **Up** / **Down** Move selection
- **Left** / **Right** Collapse or expand, or change focus
- **Home** / **End** Jump to first / last entry
- **PageUp** / **PageDown** Scroll by a page
- **Shift** + **Left** / **Shift** + **Right** Collapse or expand a whole subtree
- **Esc** Close popup
- **q** Quit
- **Ctrl** + **c** Quit from anywhere
- **h** Show help popup
- **.** Toggle the bottom command bar
- **o** Open options popup

*Toggles settings such as whitespace in diffs and the number of context lines.*

## STATUS AND STAGING

- **Enter** Stage or unstage selected file
- **a** Stage all changes
- **c** Open the commit popup
- **D** Discard changes of the selected file

*Asks for confirmation; discarded changes cannot be recovered.*

- **e** Open file in your editor
- **i** Add file to .gitignore
- **w** Switch between workdir and staged view
- **Tab** Move focus between file lists and diff
- **U** Undo last commit (soft reset)
- **B** Blame the selected file
- **H** Show history of the selected file

## DIFF VIEW

- **Right** Focus the diff pane
- **Enter** Stage or unstage the selected hunk
- **n** / **p** Next / previous hunk
- **Up** / **Down** Move line cursor
- **Shift** + **Up** / **Shift** + **Down** Extend line selection
- **s** Stage the selected lines
- **d** Discard the selected lines

## COMMIT

- **Enter** Commit with the entered message
- **Ctrl** + **a** Amend the previous commit
- **Ctrl** + **e** Write message in your editor
- **Esc** Cancel the commit

## LOG

- **Enter** Show files changed in the commit
- **Space** Mark commit
- **Shift** + **Down** Select several commits
- **y** Copy commit hash
- **t** Tag the commit
- **T** Show tags
- **r** Reword the commit message
- **S** Check out the commit
- **R** Reset branch to the commit
- **f** Search commits

## BRANCHES

- **b** Open branch list
- **Enter** Check out the selected branch
- **c** Create a branch
- **r** Rename the branch
- **D** Delete the branch
- **m** Merge the branch into HEAD
- **R** Rebase onto the branch
- **Left** / **Right** Switch local and remote lists

## REMOTE AND STASH

- **p** Push current branch
- **P** Force push

*Overwrites remote history; use deliberately.*

- **f** Pull from remote
- **F** Fetch from remotes
- **s** Stash changes (Stashing tab)
- **u** Include untracked files in stash
- **Enter** Pop selected stash (Stashes tab)

## REMOTE AND STASH (cont.)

- **a** Apply selected stash
- **D** Drop selected stash

## CONFIG

**Files in the gitui config dir**

| File | Purpose |
|---|---|
| `key_bindings.ron` | Remap keys |
| `key_symbols.ron` | Rename key labels shown |
| `theme.ron` | Colours (see THEMES.md) |

Config dir: ~/.config/gitui (Linux, macOS) or %APPDATA%/gitui (Windows). Linux uses $XDG_CONFIG_HOME if set. Run gitui -l to log to ~/.cache/gitui/gitui.log.

**key_bindings.ron: vim-style movement**

```
// ~/.config/gitui/
key_bindings.ron
(
  move_down: Some(( code:
Char('j'), modifiers: "")),
  move_up: Some(( code:
Char('k'), modifiers: "")),
)
```

## RECIPES

**Commit part of a file**

```
# Status tab: select the file
Right
# move to lines, Shift+Down to
extend
s
Esc
c
# type message, then Enter
```

**Fix the last commit**

```
# stage the forgotten change
Enter
c
Ctrl+a
# edit message if needed, Enter
```

**New branch and push**

```
b
c
# type the name, Enter
# commit your work, then
p
```

**Park work in a stash**

```
4
s
# name the stash, Enter
5
# later: select it
Enter
```