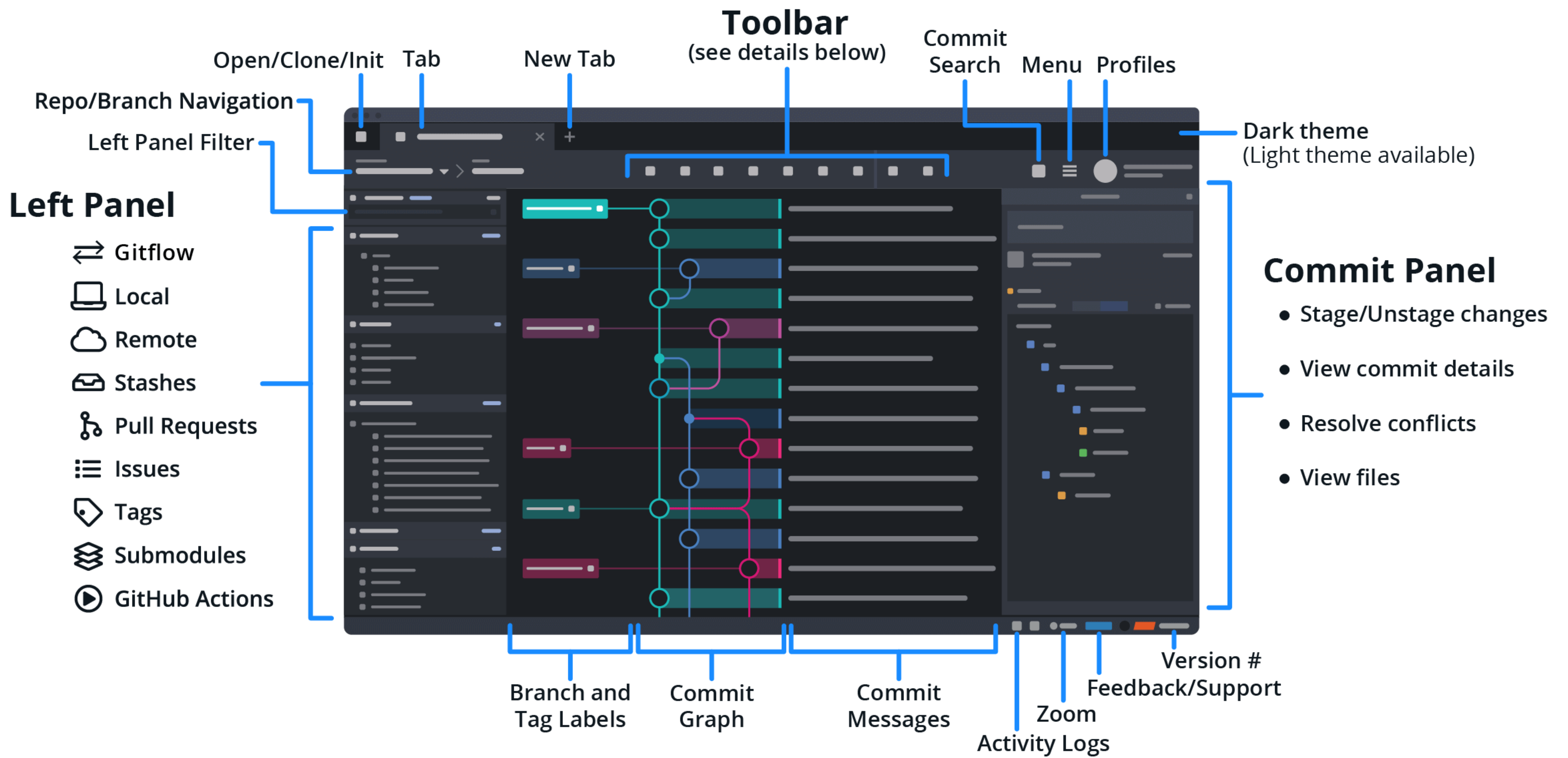


GitKraken Git GUI Cheat Sheet

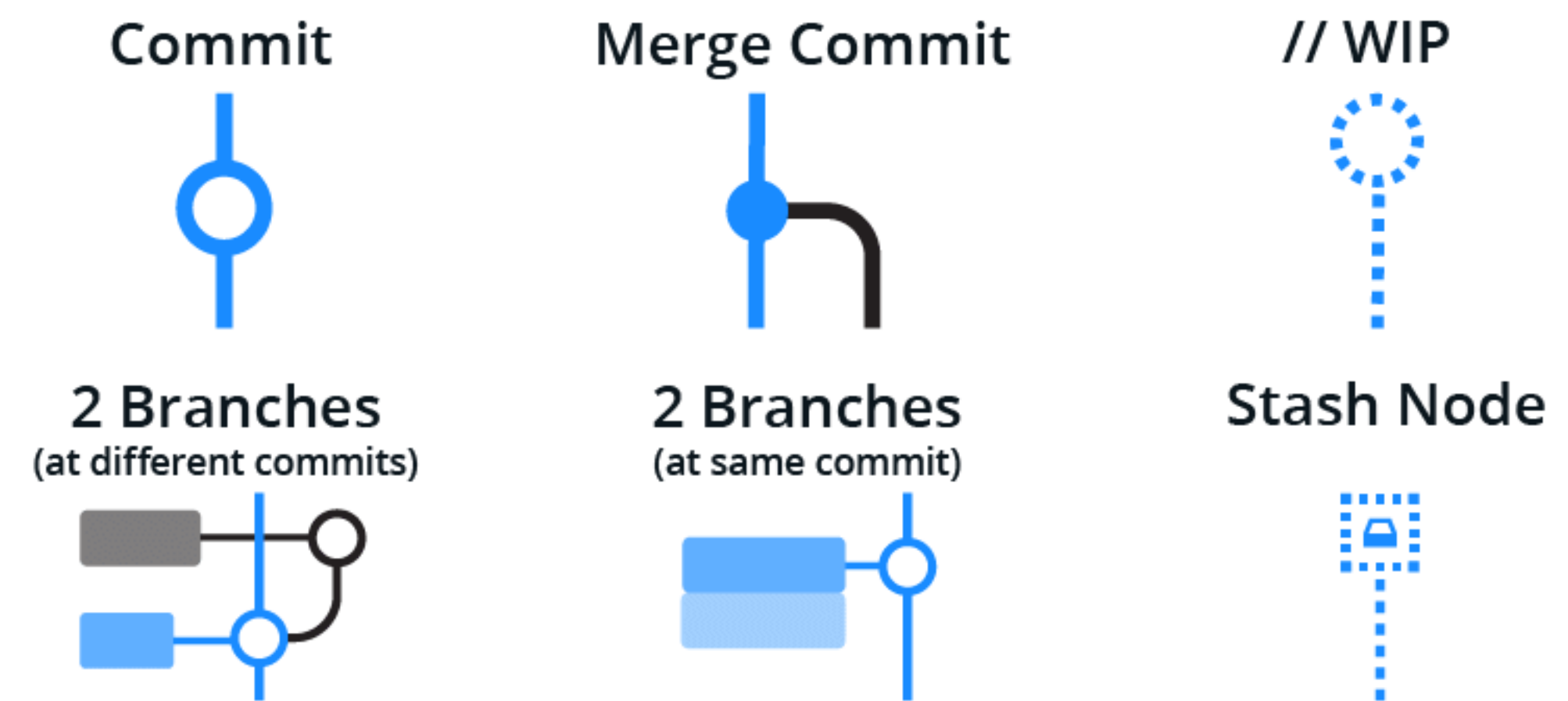
User Interface



Toolbar Details

- Undo
- Stash
- GitKraken Boards
- Redo
- Pop Stash
- GitKraken Timelines
- Pull/Fetch
- Branch
- LFS Large File Storage (initialized repos only)
- Push

Graph Elements & Actions



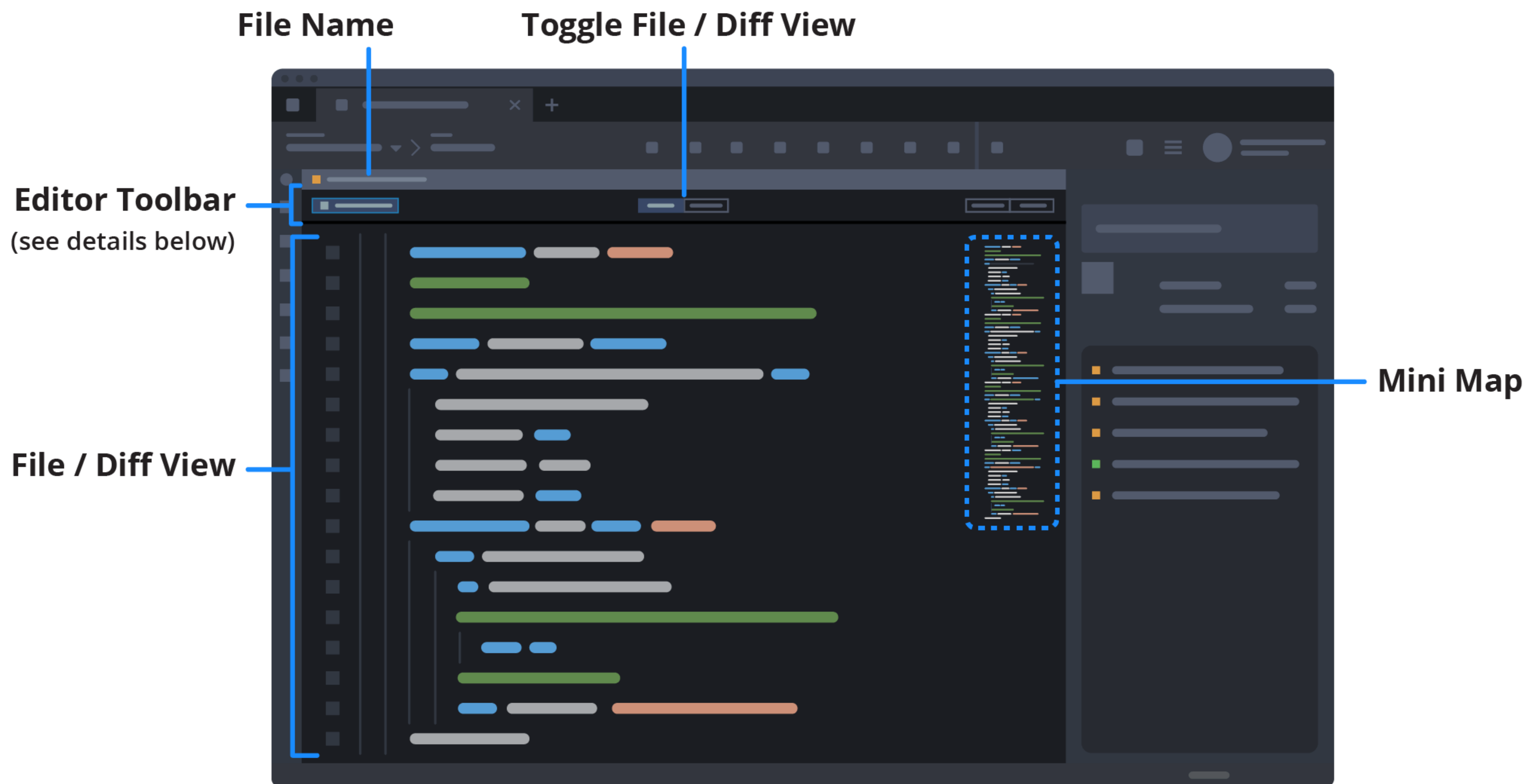
Keyboard Shortcuts

⌘ | Ctrl | /

REPO ACTIONS	
Create Branch	⌘ Ctrl B
Fetch All	⌘ Ctrl L
Commit	⌘ Ctrl Return Enter (when in message input)
Stage Current File	S
Unstage Current File	U
Stage All Files	⌘ Ctrl Shift S
Unstage All Files	⌘ Ctrl Shift U
Stage All + Commit	⌘ Ctrl Shift Return Enter (when in message input)
NAVIGATION	
Select Next Item	↓ or J
Select Previous Item	↑ or K
Select Next Item in Branch	Shift ↓ or Shift J
Select Previous Item in Branch	Shift ↑ or Shift K
Select First Item	⌘ Ctrl ↑ Home
Select Last Item	⌘ Ctrl ↓ End
Undo	⌘ Ctrl Z
Redo	⌘ Ctrl Y or ⌘ Ctrl Shift Z

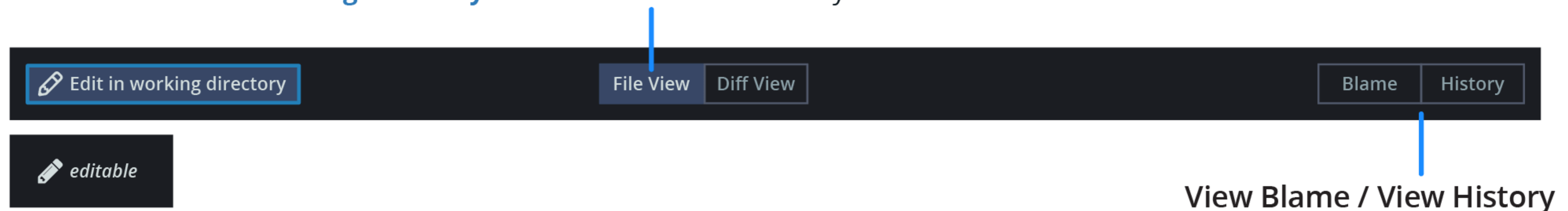
FUZZY FINDER	
Toggle Fuzzy Finder	⌘ Ctrl P
Search Commits	⌘ Ctrl F
Open Repo	⌘ Ctrl Shift O via Fuzzy Finder
Search for a File	⌘ Ctrl Shift H to view its History/Blame
Open Selected File	⌘ Ctrl D in External Diff/Merge Tool
UI	
Open a New Tab	⌘ Ctrl T
Jump to Next Open Tab	Ctrl Tab
Jump to Previous Open Tab	Ctrl Shift Tab
Jump to a Specific Tab	⌘ Ctrl 1 - 9
Close File/Tab	⌘ Ctrl W
Toggle Left Panel	⌘ Ctrl \
Increase Zoom	⌘ Ctrl +
Decrease Zoom	⌘ Ctrl -
Reset Zoom	⌘ Ctrl 0

Code Editor: Functionality Overview



Editor Toolbar Details: File View

Review a file's contents in the **File View** tab. Use the file mini map on the right to quickly scroll through the code or even hit the **Edit in working directory** button to edit the file directly.

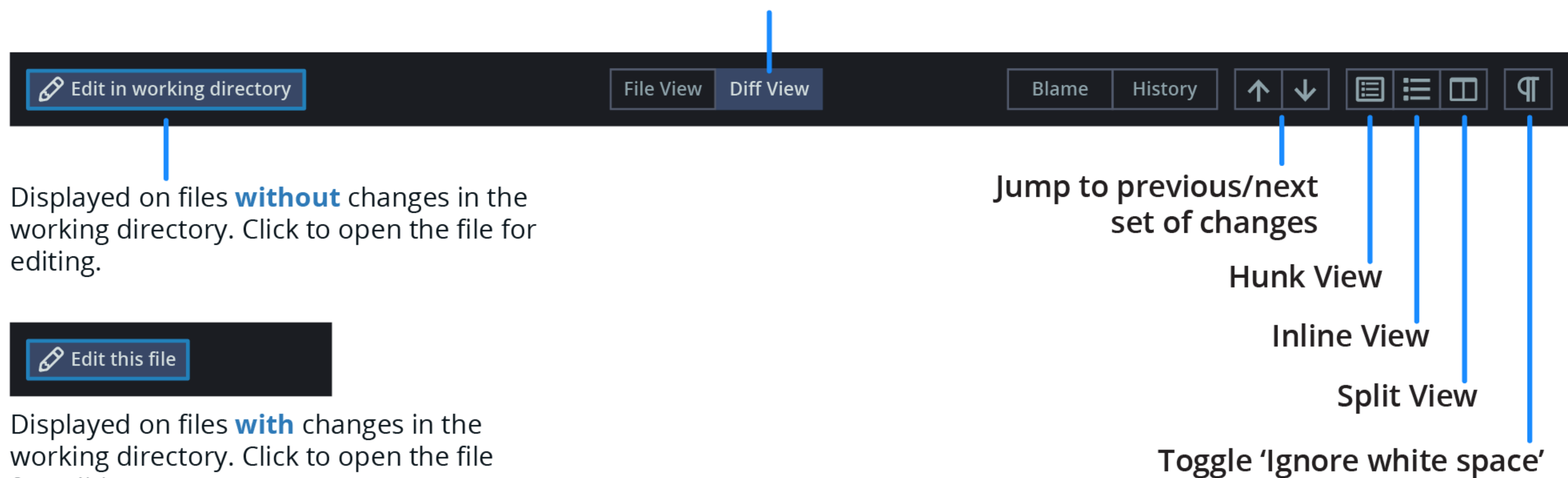


Indicates the file is in edit mode.

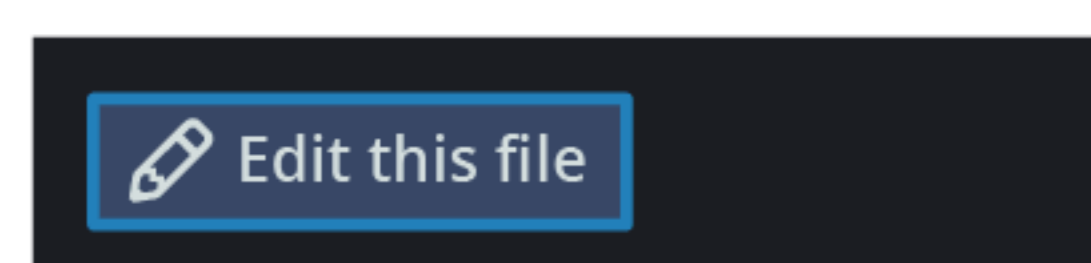
View Blame / View History

Editor Toolbar Details: Diff View

The **Diff View** shows what was added or removed from a file in one of three views: **Hunk, Inline, or Split**. A red background indicates lines where content was removed, whereas a green background indicates new lines added.



Displayed on files **without** changes in the working directory. Click to open the file for editing.



Displayed on files **with** changes in the working directory. Click to open the file for editing.

Jump to previous/next set of changes

Hunk View

Inline View

Split View

Toggle 'Ignore white space'

The Fuzzy Finder

Use **⌘ | Ctrl P** to toggle the Fuzzy Finder: a text input where you can type commands to open repositories, view the history of files in the current repository, and more.

For example, if you want to open a repository, type: "open repo" and then press `enter`. From there, start typing the name of the repo you're looking for, and you'll quickly see a list of relevant repos you can open.

You can also perform actions on your current repository. **For example, you can fetch all, create or pop a stash, undo or redo actions, etc. from the Fuzzy Finder.** As you type, the relevant commands will be displayed in a list. This allows you to perform many actions without ever having to leave the comfort of your keyboard.


Operating on Other Branches


One of the more subtle capabilities of GitKraken Git GUI is the ability to interact with branches (and tags) other than the one currently checked out. You can fetch, pull, and push a branch while being on a different branch, provided no merge conflicts occur.

For example, if you have the "development" branch checked out, you can right click on the "feature/123" branch and push it independently.

You've always been able to push that branch, but now you can save the extra steps of stashing, checking out that branch, pushing, checking out your previous branch again, and popping your stash.

Hiding and Soloing

 **Hiding branches is a useful way to reduce clutter in your commit graph. Any branches that are visible will have a green "eye"-con to the left. Clicking this icon will hide the branch in the graph, and it will update the icon to a gray eye with a slash through it.** This can also be done with remotes, which will hide all of a remote's branches.

 **To solo a branch, right-click a branch and select "Solo" from the drop-down menu. An orange icon will display to the left, which indicates the selected branch has been soloed.** You can solo as many or as few branches/tags as you like—even entire remotes. Click "Stop Soloing" at the top of the panel to leave the soloing state and restore branches and tags to their previous visible/hidden states.

Undo and Redo

Undoing Git operations manually is complex. Undoing a merge involves hard resetting branches or even deleting a branch, which means manually parsing through the reflog and dangling blobs hoping that you can find your changes and save them before they're cleaned up. With GitKraken, it's easy: just click the `Undo` or `Redo` buttons in the toolbar.



Undo

GitKraken Git GUI can undo a lot of these changes with a single click of the `Undo` button (or, you can do it through the Fuzzy Finder).



Redo

Have you done something, and then undid it, only to wish you had just left it alone? There is a `Redo` function for that!

Pull Requests

Switching between applications can be time-consuming and disruptive to your workflow—especially when it comes to pull requests. By integrating GitKraken Git GUI with services like GitHub, you can create new PRs and view pending PRs for multiple repos without ever leaving the loving tentacles of GitKraken.

If you are using the GitLab or GitHub integrations, you can also add a pull request assignee and label(s) to your pull request. GitKraken Git GUI will then pass these values onto GitLab or GitHub when the pull request is created.

Additionally for the GitLab integration, you can hover over an existing pull request to show any assignees or labels associated with the pull request. And for GitHub, this tooltip will show assignees, labels, reviewers, and build status.

Integrations



GitHub



Bitbucket



Trello



GitLab



Azure DevOps



Jira

GitKraken Git GUI can integrate with these hosting and issue tracking services to make working with your remote repositories and issue backlogs even easier.

For example, when cloning a repository or adding remotes, GitKraken Git GUI will display the repos you have access to on your connected service(s), in addition to the ability to manually enter a repo URL.

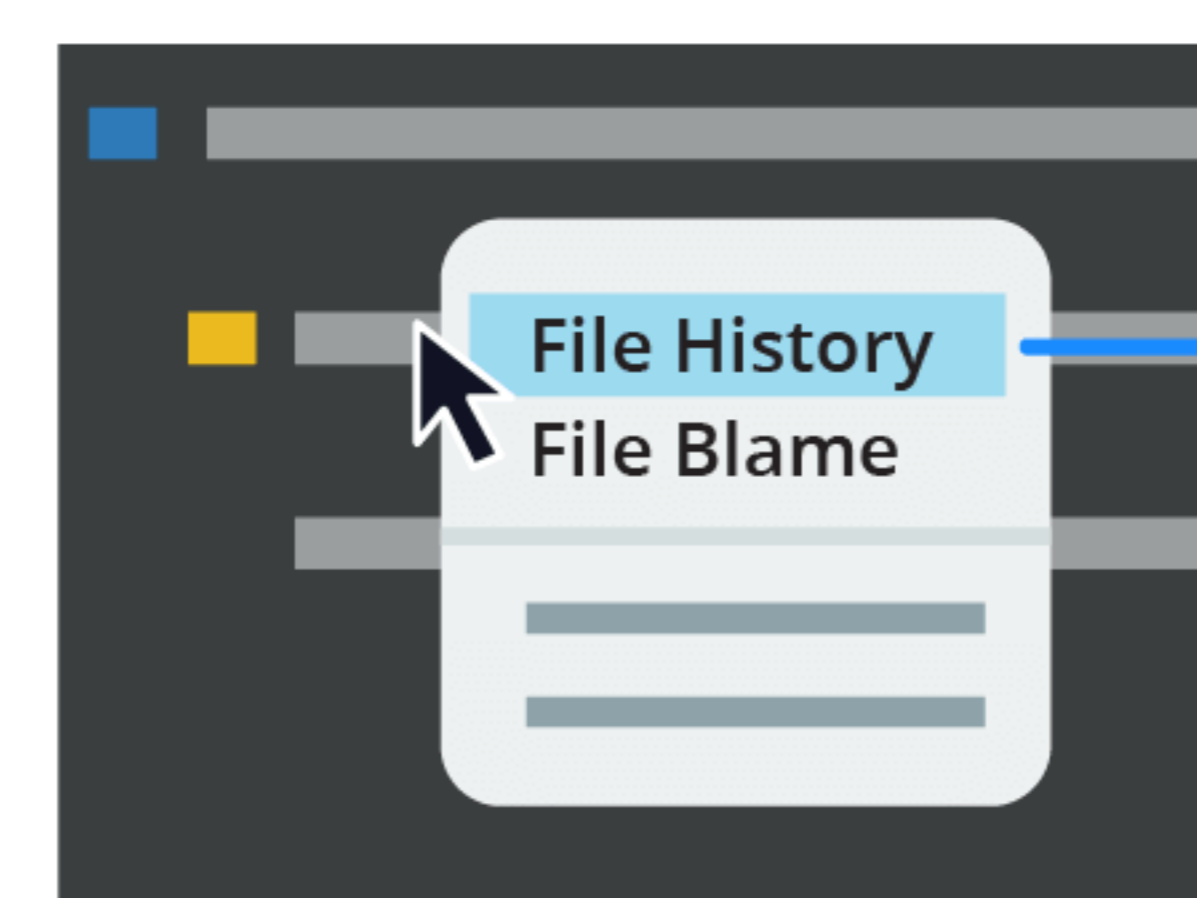
With the issue tracking integration, you'll be able to view issues from supported services inside GitKraken Git GUI and easily filter them to focus only on work items relevant to you, including the ability to create a new branch for an issue with the click of a button.

Secure Shell (SSH)

GitKraken Git GUI can generate an SSH Key for you automatically. It can even add it to your GitHub account, if you give it permission. Since GitKraken Git GUI uses its own bundled copy of an SSH library, nothing needs to be configured outside of the app.

File History and Blame

History and Blame information for a file show up in the same view of the commit panel:



Choose to view History or Blame by right clicking the file that appears when you select a commit.

You can also use the Fuzzy Finder **⌘ | Ctrl P** and then type "history" followed by a space and then the filename. Once you're in the History view, you can see the commits in that file's history on the lefthand side.

