VimConf 2024

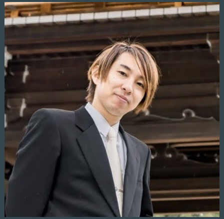
Can't Help **Fall**ing in Vim

Wise men say only fools reinvent the wheel,
but I can't help building yet another fuzzy
finder: **Fall**

https://bit.ly/3YrU5rU

This is Fuzzy Finder

Short story



tomoya

So, what exactly is a Fuzzy Finder in the first place?

Λlisue

Well, it’s something that finds things in a fuzzy way.

from “エンジニアの楽園 vim-jp ラジオ #8”

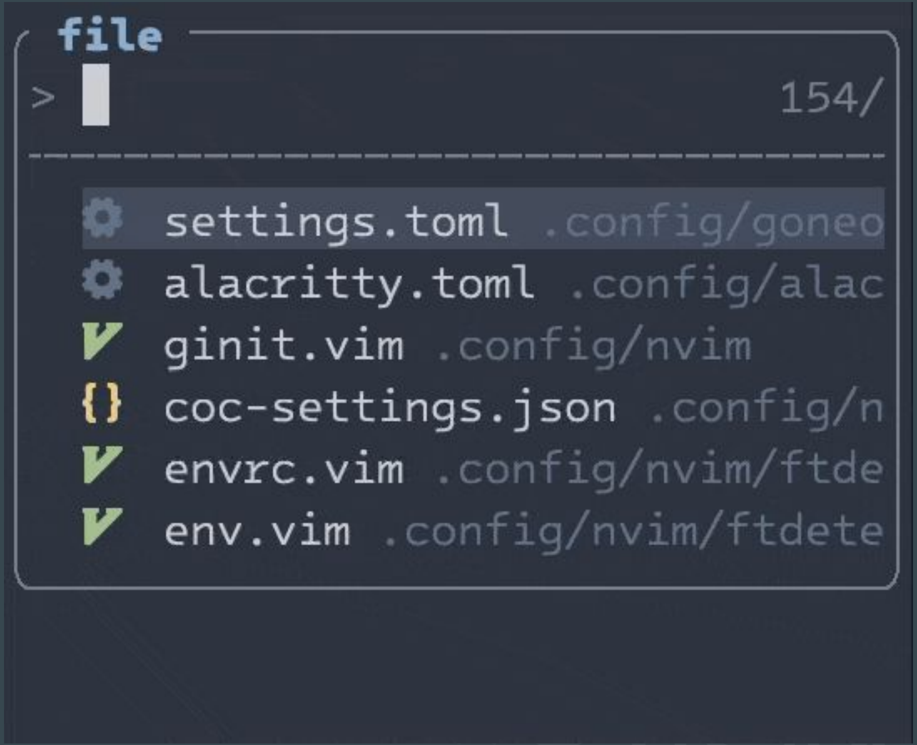
Fuzzy Finder

Purpose

Find items quickly from a large collection.

Method

Filter items in real-time by matching partial strings in their names.



```
file
>                                          154/
------------------------------------------------
   settings.toml .config/goneo
   alacritty.toml .config/alac
   ginit.vim .config/nvim
   coc-settings.json .config/n
   envrc.vim .config/nvim/ftde
   env.vim .config/nvim/ftdete
```

Difference

🔍

Fuzzy finder

Fast filtering

By using partial matches.

Partial knowledge

Unorganized data can be handled with partial knowledge.

🌲

Tree viewer

Quick overview

By visually grasping structures.

Structured data

No prior knowledge is required if the data is organized.

Me & …

Λlisue (Ali su e・ありすえ)

Denops

Vim / Neovim plugin ecosystem

Fern

Pure Vim script tree viewer

Gin

Git manipulation plugin

Fall

TODAY'S TOPIC!

My First Secret

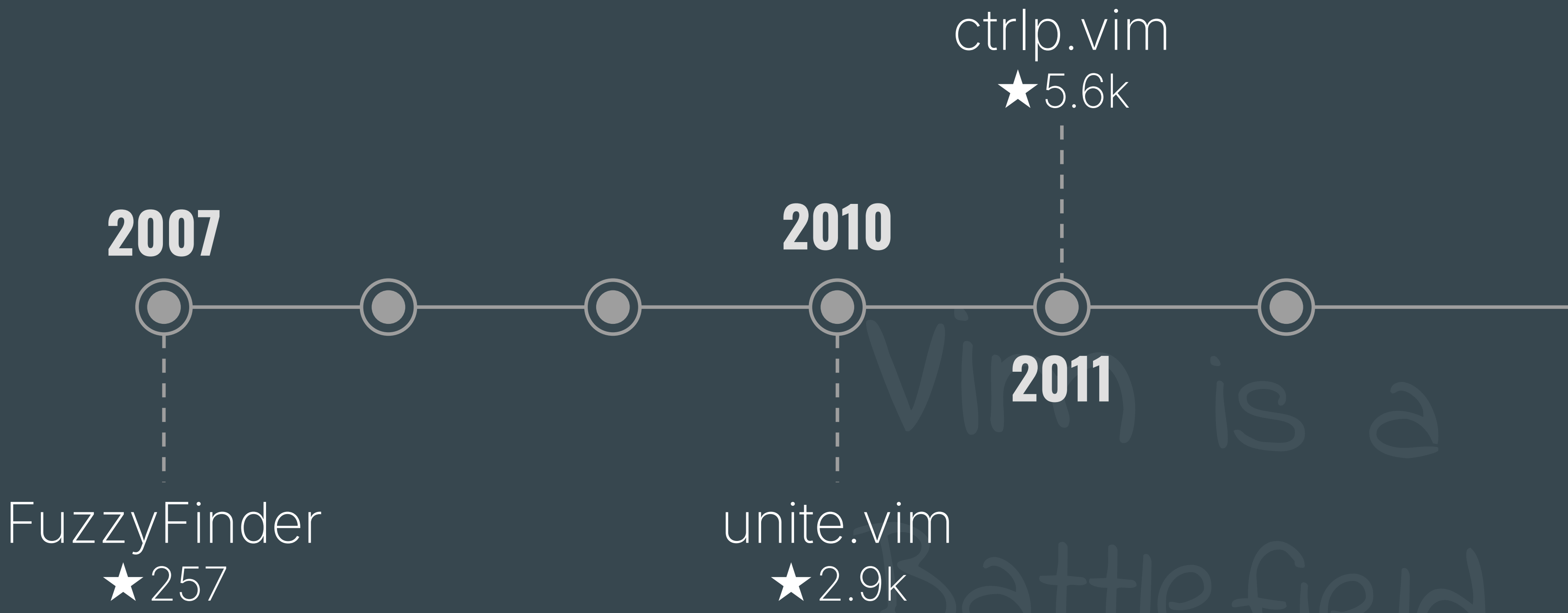
Catch A History

Fuzzy Finder I’ve known

ATTENTION

I have only reviewed the documentation of each plugin, so there may be some misunderstandings about their features.

History (2007-2012)



2007

FuzzyFinder
★257

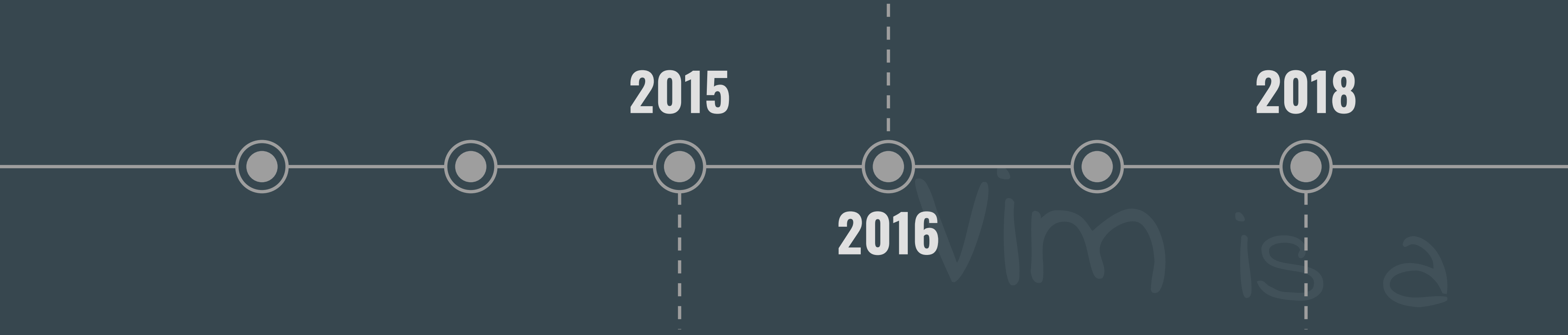
2010

unite.vim
★2.9k

2011

ctrlp.vim
★5.6k

History (2013-2018)



denite.nvim
★2.1k

2015

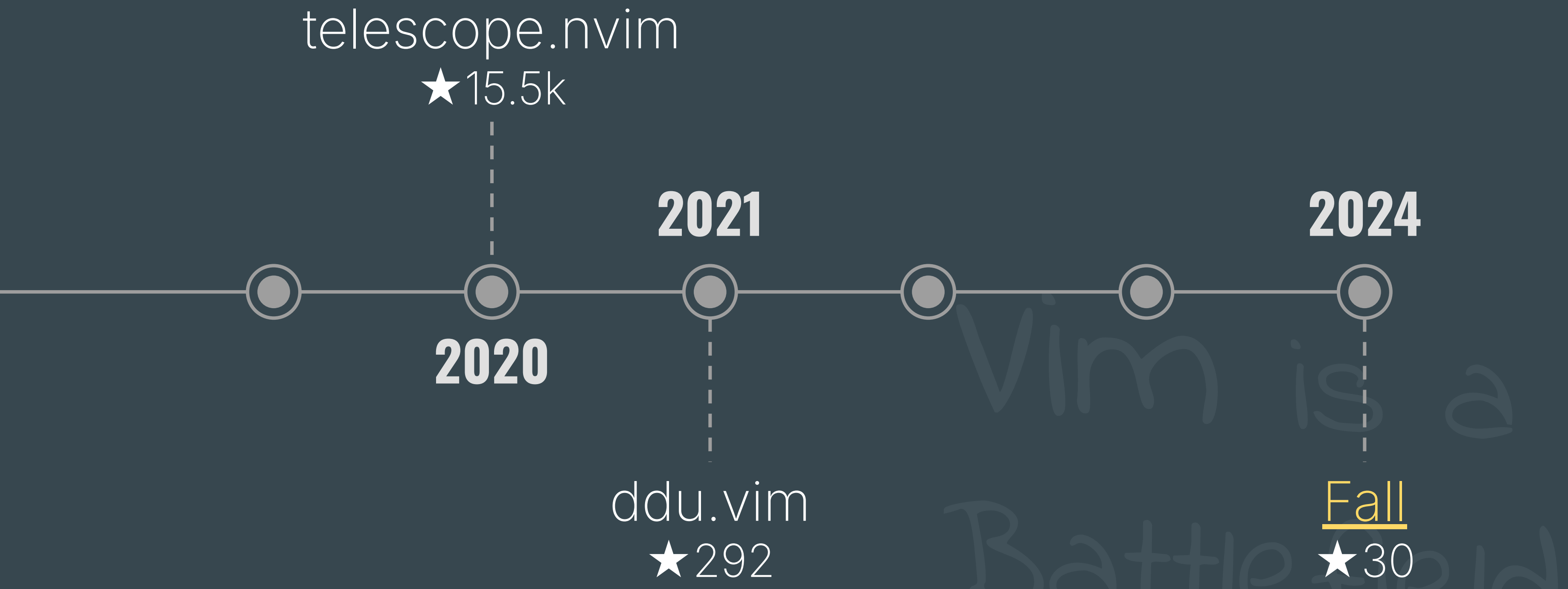
2016

2018

fzf.vim
★9.6k

fzf-preview.vim
★909

History (2019-2024)



telescope.nvim
★15.5k

2020

2021

ddu.vim
★292

2024

Fall
★30

Fall

Fall

Support Vim & Neovim

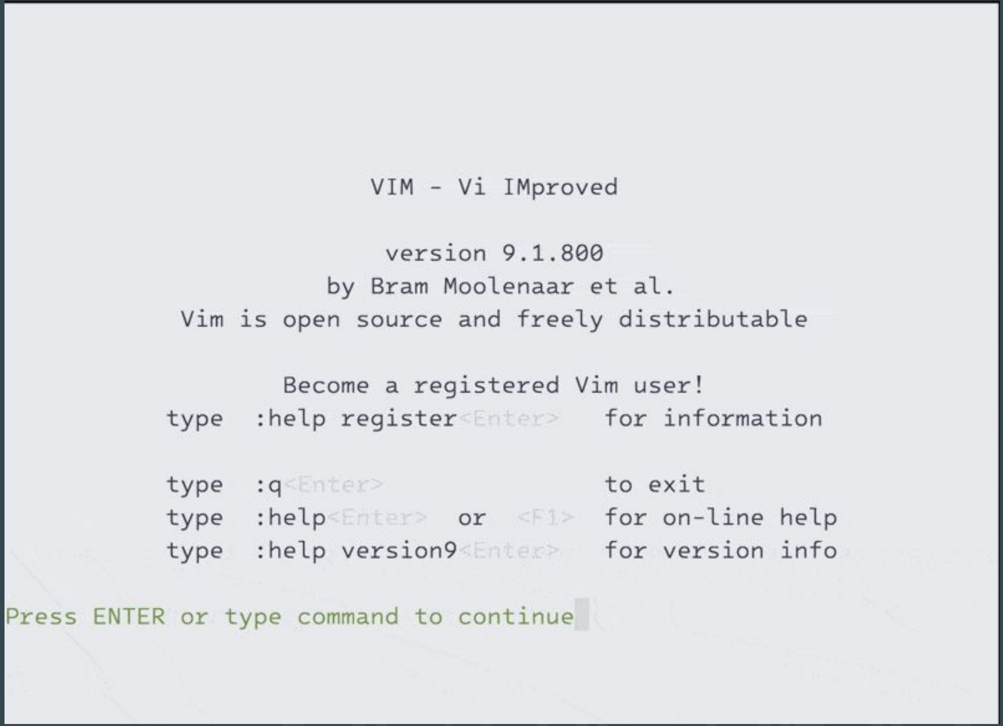
Floating window-like UI is also supported in Vim.

Focused on purpose

Only provide essential features for Fuzzy Finding.

Programmer-friendly

TypeScript-first design makes developers’ lives easier.



```
                    VIM - Vi IMproved

                    version 9.1.800
                by Bram Moolenaar et al.
      Vim is open source and freely distributable

              Become a registered Vim user!
     type  :help register<Enter>   for information

     type  :q<Enter>               to exit
     type  :help<Enter>  or  <F1>  for on-line help
     type  :help version9<Enter>   for version info

Press ENTER or type command to continue
```

Supports Vim & Neovim

Supports both Vim and Neovim

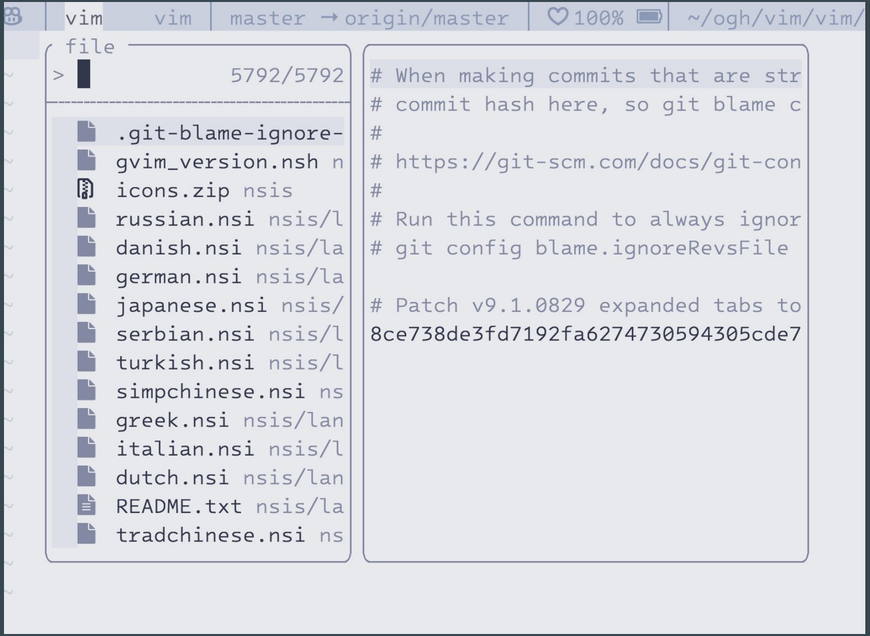
Few plugins support both

Vim and Neovim are evolving in different directions, making compatibility challenging.

Floating UI was a key

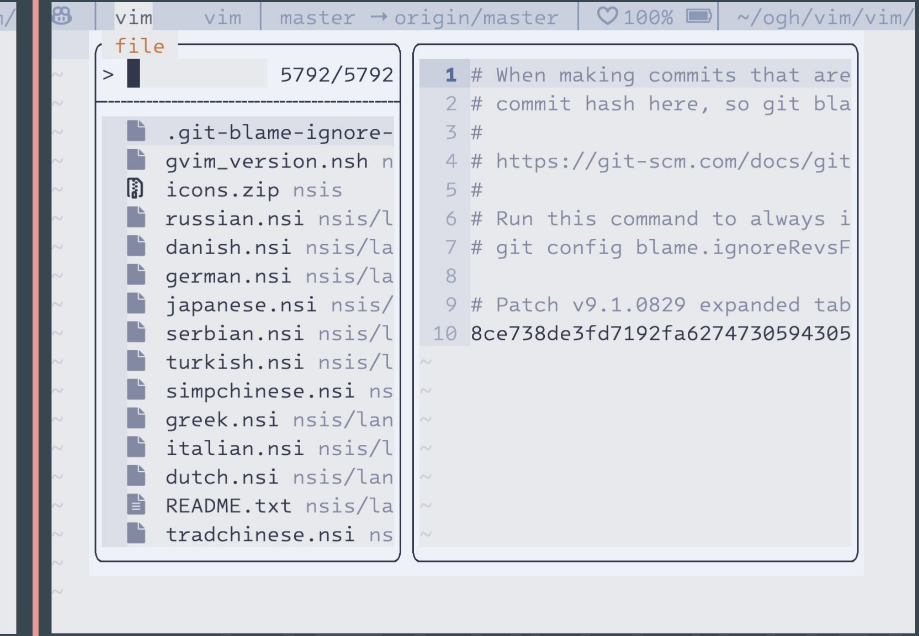
Fall was inspired by the idea of creating a Fuzzy Finder with a floating UI in Vim.

Quick quiz: which one is Vim?



```
 vim      vim  master → origin/master  ♡100%   ~/ogh/vim/vim/
 file
 >                  5792/5792   # When making commits that are str
                                # commit hash here, so git blame c
    .git-blame-ignore-          #
    gvim_version.nsh n          # https://git-scm.com/docs/git-con
    icons.zip nsis              #
    russian.nsi nsis/l          # Run this command to always ignor
    danish.nsi nsis/la          # git config blame.ignoreRevsFile
    german.nsi nsis/la
    japanese.nsi nsis/          # Patch v9.1.0829 expanded tabs to
    serbian.nsi nsis/l          8ce738de3fd7192fa6274730594305cde7
    turkish.nsi nsis/l
    simpchinese.nsi ns
    greek.nsi nsis/lan
    italian.nsi nsis/l
    dutch.nsi nsis/lan
    README.txt nsis/la
    tradchinese.nsi ns
```

Vim



```
 vim      vim  master → origin/master  ♡100%   ~/ogh/vim/vim/
 file
 >                  5792/5792   # When making commits that are
                                # commit hash here, so git bla
    .git-blame-ignore-          #
    gvim_version.nsh n          # https://git-scm.com/docs/git
    icons.zip nsis              #
    russian.nsi nsis/l          # Run this command to always i
    danish.nsi nsis/la          # git config blame.ignoreRevsF
    german.nsi nsis/la
    japanese.nsi nsis/          # Patch v9.1.0829 expanded tab
    serbian.nsi nsis/l          8ce738de3fd7192fa6274730594305
    turkish.nsi nsis/l         ~
    simpchinese.nsi ns         ~
    greek.nsi nsis/lan         ~
    italian.nsi nsis/l         ~
    dutch.nsi nsis/lan         ~
    README.txt nsis/la         ~
    tradchinese.nsi ns         ~
```

Neovim

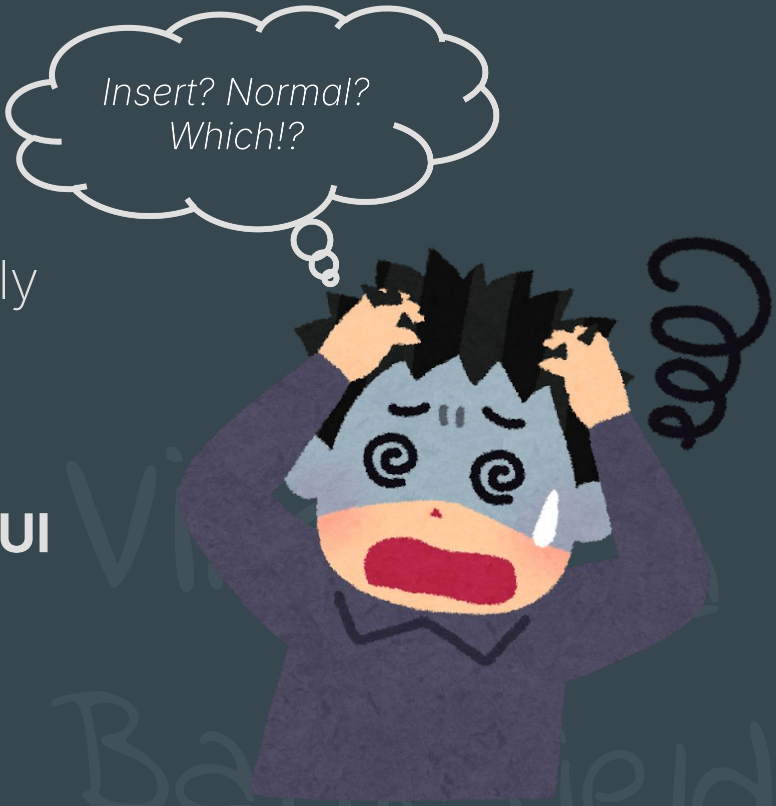
Focused on purpose

Focused on purpose

Do we really need modal...?

A fuzzy finder's purpose is to quickly narrow down options, not scroll endlessly through candidates (better for a tree viewer).

Considering this, I realized a **modeless UI** might be more suitable than a modal UI.

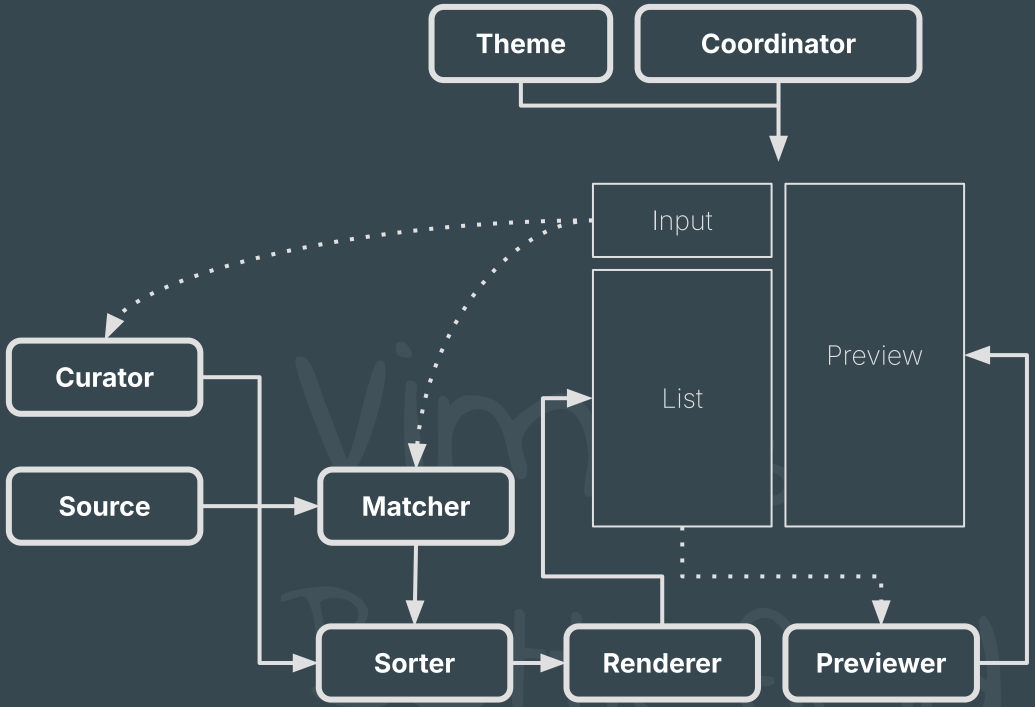


Focused on purpose

Modular design

Fall’s modular design lets you mix and match features freely.

Built-in modules are also provided as extensions, so third-party developers have flexibility in design.

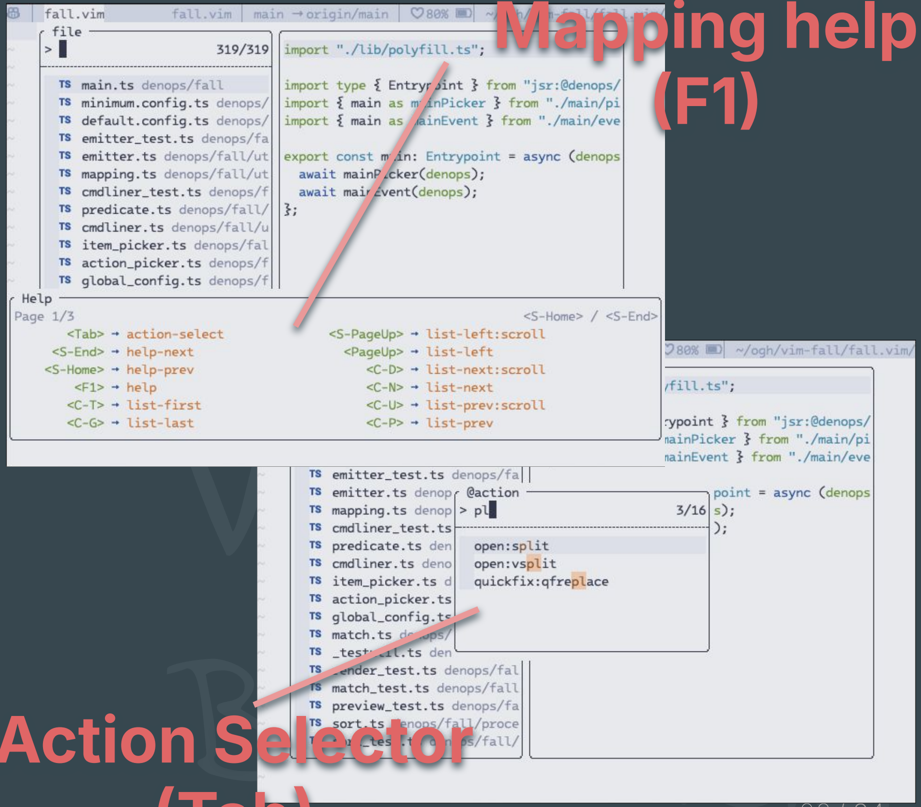


Focused on purpose

Born to Forget

A Fuzzy Finder’s purpose is filtering, not testing your memory of shortcuts. That’s why we’ve made it easy to check mappings whenever you need.

For actions without mappings, we provide an Action Selection UI, so you can access all actions without worrying about running out of keys.



```
 fall.vim                  fall.vim │ main →origin/main │ ♡80%
 file
 > ▌                          319/319 │ import "./lib/polyfill.ts";
                                      │
 TS main.ts denops/fall               │ import type { Entrypoint } from "jsr:@denops/
 TS minimum.config.ts denops/         │ import { main as mainPicker } from "./main/pi
 TS default.config.ts denops/         │ import { main as mainEvent } from "./main/eve
 TS emitter_test.ts denops/fa         │
 TS emitter.ts denops/fall/ut         │ export const main: Entrypoint = async (denops
 TS mapping.ts denops/fall/ut         │   await mainPicker(denops);
 TS cmdliner_test.ts denops/f         │   await mainEvent(denops);
 TS predicate.ts denops/fall/         │ };
 TS cmdliner.ts denops/fall/u         │
 TS item_picker.ts denops/fal         │
 TS action_picker.ts denops/f         │
 TS global_config.ts denops/f         │
 Help
 Page 1/3                                                   <S-Home> / <S-End>
      <Tab> → action-select                 <S-PageUp> → list-left:scroll
    <S-End> → help-next                       <PageUp> → list-left
   <S-Home> → help-prev                          <C-D> → list-next:scroll
       <F1> → help                               <C-N> → list-next
      <C-T> → list-first                         <C-U> → list-prev:scroll
      <C-G> → list-last                          <C-P> → list-prev
```

```
                                       ♡80%  ~/ogh/vim-fall/fall.vim/
                                       /fill.ts";

                                       rypoint } from "jsr:@denops/
                                       nainPicker } from "./main/pi
                                       nainEvent } from "./main/eve
 TS emitter_test.ts denops/fa
 TS emitter.ts denop  @action            point = async (denops
 TS mapping.ts denop  > pl▌        3/16  s);
 TS cmdliner_test.ts                     );
 TS predicate.ts den    open:split
 TS cmdliner.ts deno    open:vsplit
 TS item_picker.ts d    quickfix:qfreplace
 TS action_picker.ts
 TS global_config.ts
 TS match.ts denops/
 TS _testutil.ts den
 TS render_test.ts denops/fal
 TS match_test.ts denops/fall
 TS preview_test.ts denops/fa
 TS sort.ts denops/fall/proce
 TS _test.ts denops/fall/
```

Mapping help (F1)

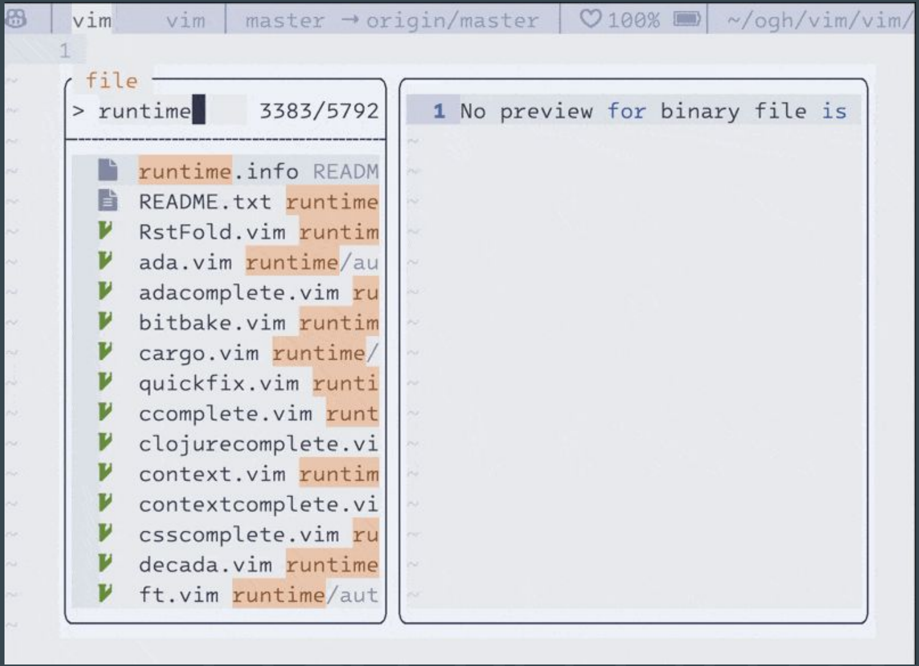
Action Selector (Tab)

Focused on purpose

Switch behaviors

Most people launch a Fuzzy Finder to quickly narrow things down.

With Fall, you can register functions and switch behaviors instantly, perfect for broad starts and refining as needed.



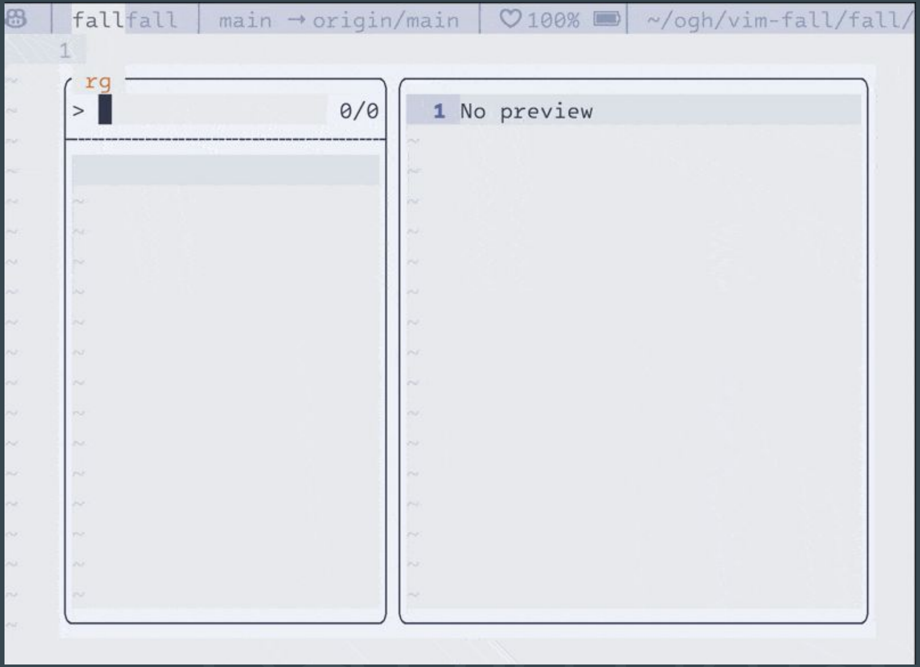
```
 vim      vim | master → origin/master | ♡ 100%  | ~/ogh/vim/vim/
 1
 ╭ file ─────────────────────╮ ╭──────────────────────────────────────╮
 │ > runtime      3383/5792  │ │ 1 No preview for binary file is      │
 │───────────────────────────│ │                                      │
 │   runtime.info READM      │ │                                      │
 │   README.txt runtime      │ │                                      │
 │   RstFold.vim runtim      │ │                                      │
 │   ada.vim runtime/au      │ │                                      │
 │   adacomplete.vim ru      │ │                                      │
 │   bitbake.vim runtim      │ │                                      │
 │   cargo.vim runtime/      │ │                                      │
 │   quickfix.vim runti      │ │                                      │
 │   ccomplete.vim runt      │ │                                      │
 │   clojurecomplete.vi      │ │                                      │
 │   context.vim runtim      │ │                                      │
 │   contextcomplete.vi      │ │                                      │
 │   csscomplete.vim ru      │ │                                      │
 │   decada.vim runtime      │ │                                      │
 │   ft.vim runtime/aut      │ │                                      │
 ╰───────────────────────────╯ ╰──────────────────────────────────────╯
```

Focused on purpose

Submatch to refine

Ever want to filter by multiple conditions, like finding TypeScript files with 'Deno'?

Fall's **submatch** action lets you refine results step-by-step and easily return to previous filters with ESC.



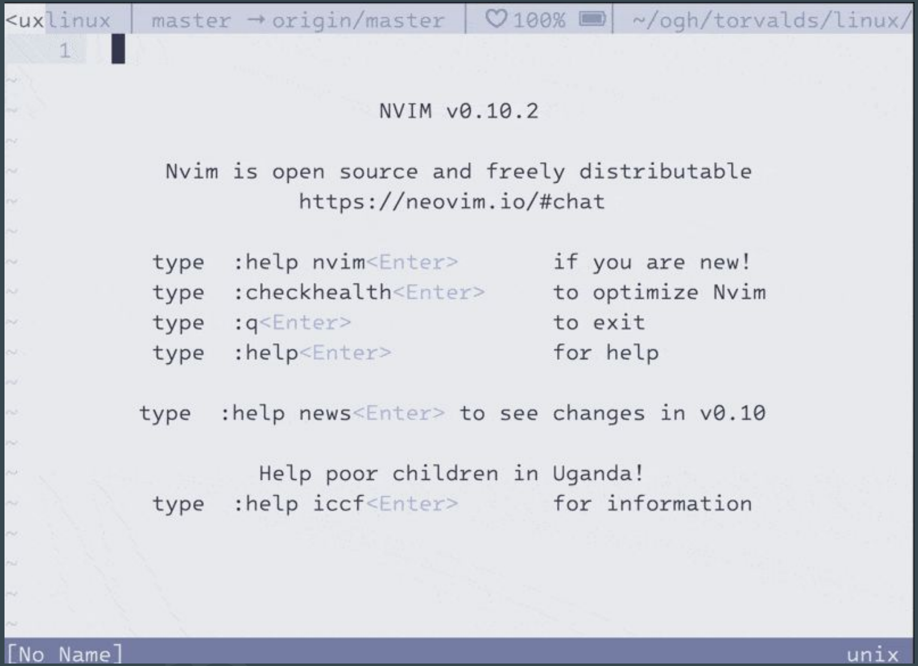
```
fallfall | main → origin/main | 100% | ~/ogh/vim-fall/fall/
rg
>                0/0        1 No preview
```

Focused on purpose

Latency over throughput

Fuzzy Finders require careful performance tuning for large option sets.

While throughput matters, Fall prioritizes latency to ensure a smoother experience.



```
<uxlinux │ master → origin/master │ ♡100%  │ ~/ogh/torvalds/linux/
  1  ▌

                          NVIM v0.10.2

            Nvim is open source and freely distributable
                      https://neovim.io/#chat

           type  :help nvim<Enter>       if you are new!
           type  :checkhealth<Enter>     to optimize Nvim
           type  :q<Enter>               to exit
           type  :help<Enter>            for help

          type  :help news<Enter> to see changes in v0.10

                    Help poor children in Uganda!
           type  :help iccf<Enter>       for information

[No Name]                                                      unix
```

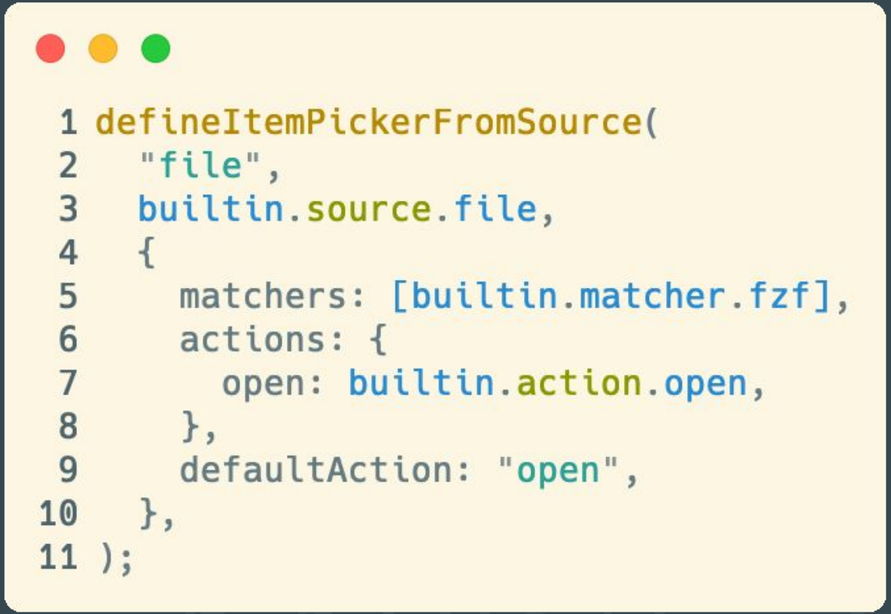
Linux repository has more than 90,000 files.

Programmer-friendly

Programmer-friendly

Customization in TS

Compared to customization with Vim script, Fall's TypeScript-based customization offers type-checking, autocompletion, linting, and formatting with Deno LSP, making setup much easier and more reliable.



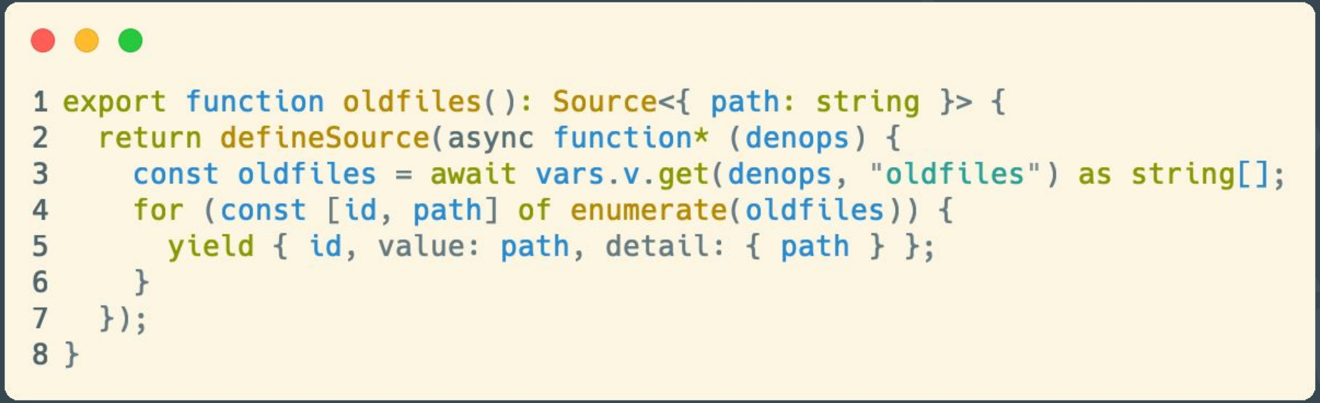
```
defineItemPickerFromSource(
  "file",
  builtin.source.file,
  {
    matchers: [builtin.matcher.fzf],
    actions: {
      open: builtin.action.open,
    },
    defaultAction: "open",
  },
);
```

Programmer-friendly

Custom extension development

Fall extensions are designed to be created with just a single function.

Since a Fuzzy Finder relies on a rich set of extensions, I've made it as easy as possible to write them.



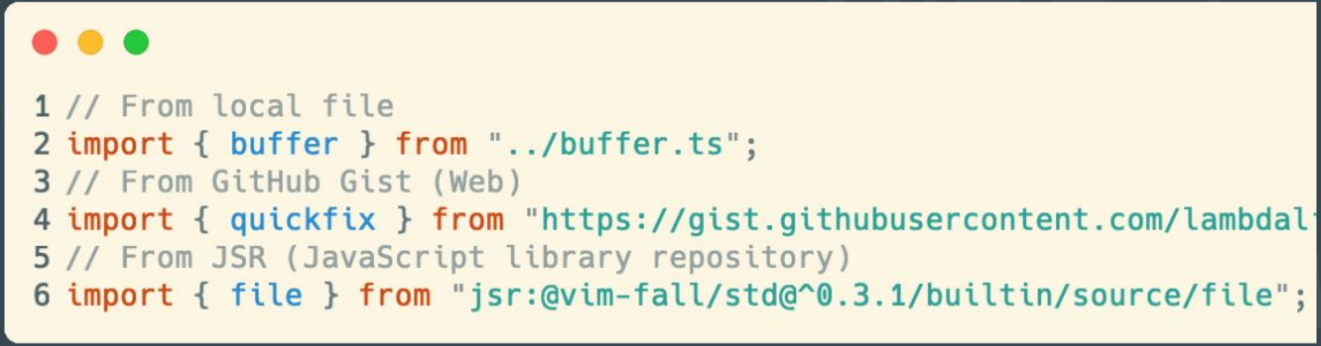
```
export function oldfiles(): Source<{ path: string }> {
  return defineSource(async function* (denops) {
    const oldfiles = await vars.v.get(denops, "oldfiles") as string[];
    for (const [id, path] of enumerate(oldfiles)) {
      yield { id, value: path, detail: { path } };
    }
  });
}
```

Programmer-friendly

Distribute as TS modules

Fall uses Deno internally, allowing extensions to be imported from local files, the web, JSR, or NPM registries.

This flexibility means Fall extensions can be located anywhere, though import maps may require some attention.



```
// From local file
import { buffer } from "../buffer.ts";
// From GitHub Gist (Web)
import { quickfix } from "https://gist.githubusercontent.com/lambdal
// From JSR (JavaScript library repository)
import { file } from "jsr:@vim-fall/std@^0.3.1/builtin/source/file";
```

Conclusion

Fall is **venry**

Venry = Very benri (useful in Japanese)

Announcements

Engineer’s Paradise Vim.jp Radio

- Fun chats with different guests.
- No set topics, it depends on the guest.
- Official podcast from AuDee (TOKYO FM).

https://vim-jp-radio.com/

🤑 Be a GitHub sponsor!

https://github.com/sponsors/lambdalisue

Fin

Agenda

Catch A History

Fuzzy Finders that gained more than 200 starts in GitHub.

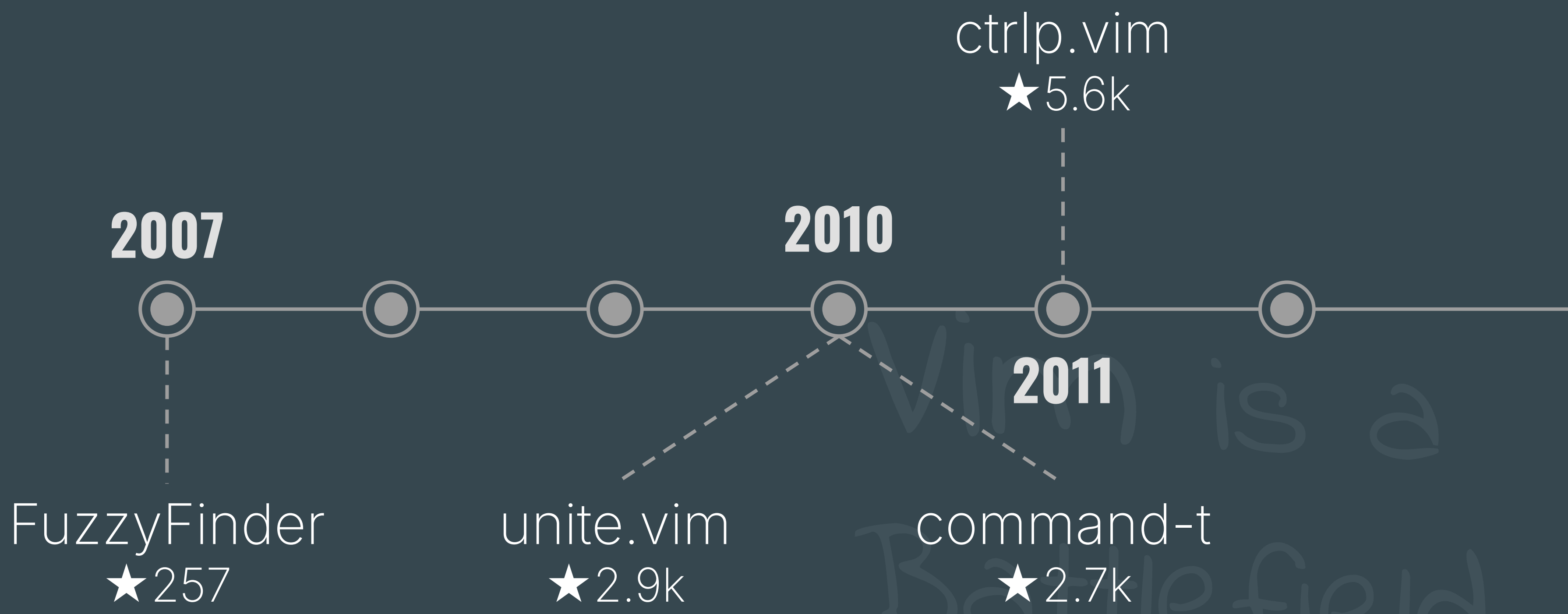
Classification

Classify existing Fuzzy Finders via its features.

Fall

Introduce the philosophy and features of Fall.

History (2007-2012)



2007

2010

2011

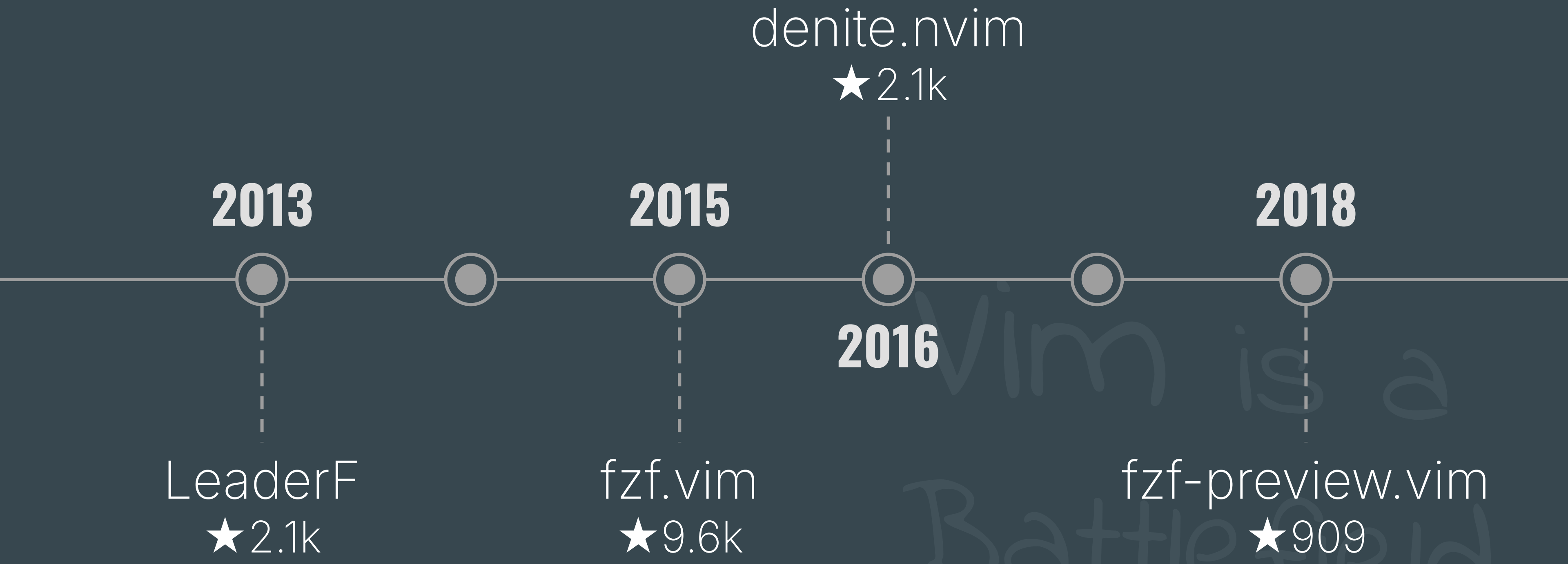
ctrlp.vim
★5.6k

FuzzyFinder
★257

unite.vim
★2.9k

command-t
★2.7k

History (2013-2018)



2013

LeaderF
★2.1k

2015

fzf.vim
★9.6k

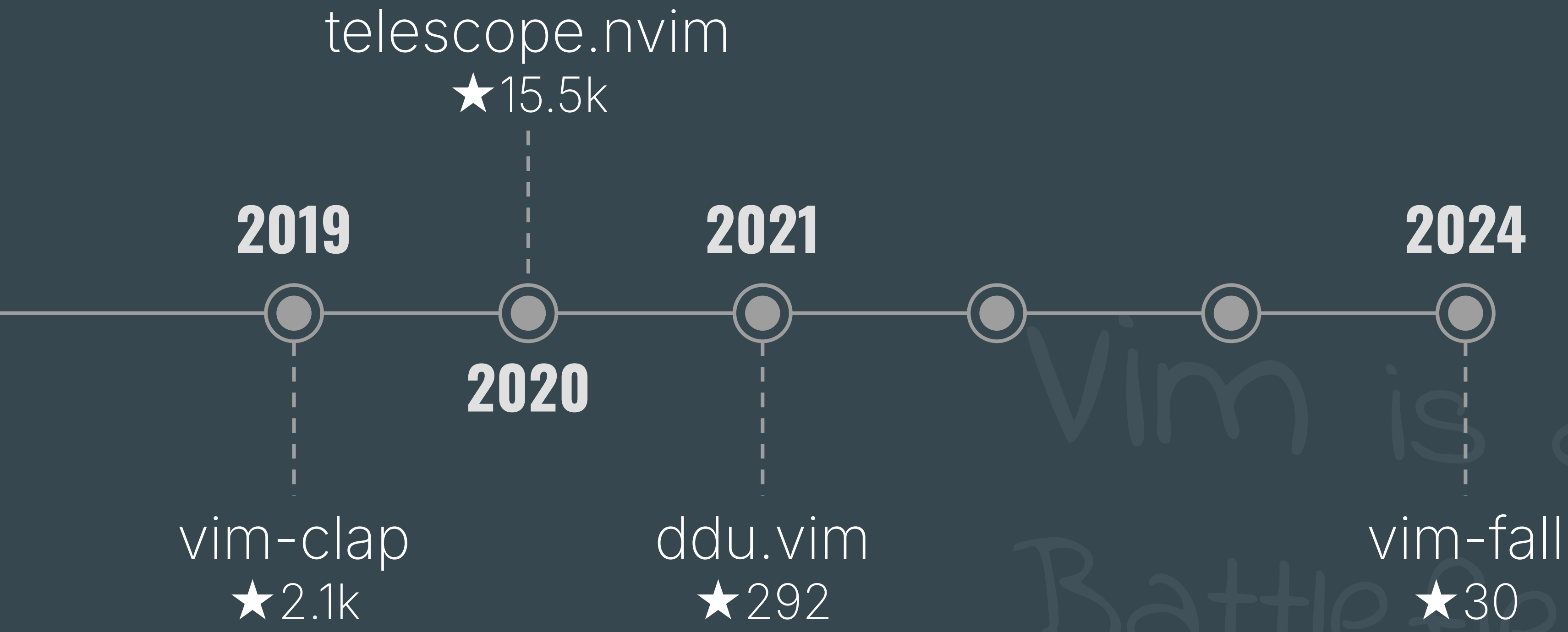
2016

denite.nvim
★2.1k

2018

fzf-preview.vim
★909

History (2019-2024)



2019

vim-clap
★2.1k

2020

telescope.nvim
★15.5k

2021

ddu.vim
★292

2024

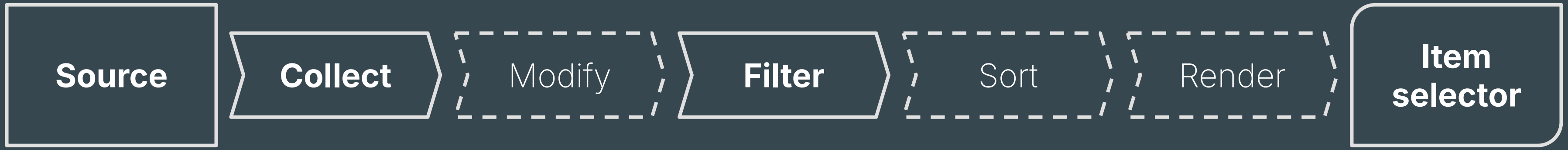
vim-fall
★30

Classification

ctrlp.vim / fzf.vim / telescope.nvim/ ddu.vim

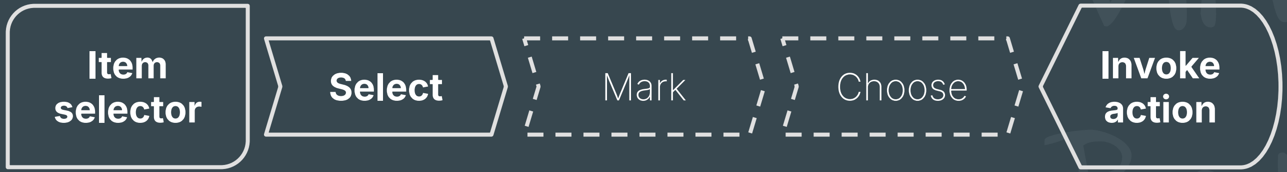
Flow classification

Filtration flow



Source **Collect** Modify **Filter** Sort Render **Item selector**

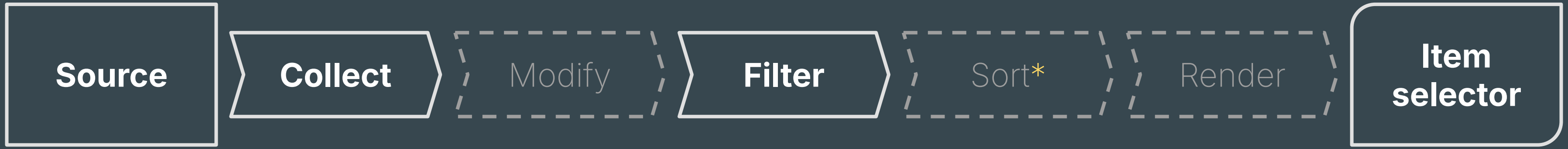
Action flow



Item selector **Select** Mark Choose **Invoke action**

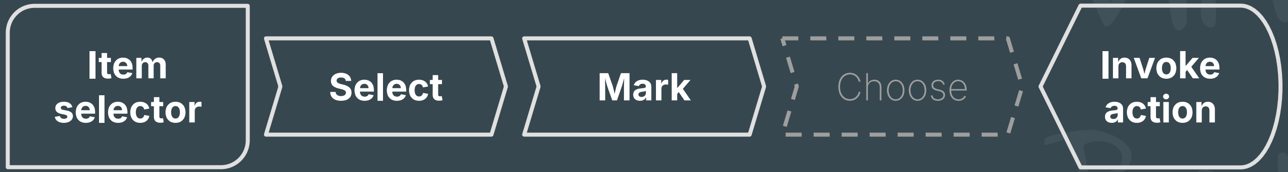
Flow: ctrlp.vim / fzf.vim

Filtration flow



Source → Collect → Modify → Filter → Sort* → Render → Item selector

Action flow

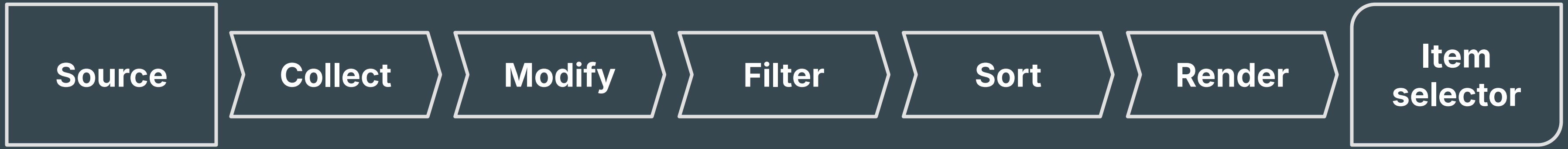


Item selector → Select → Mark → Choose → Invoke action

* Fixed sort so that users cannot change the algorithm

Flow - telescope.nvim

Filtration flow



Source

Collect

Modify

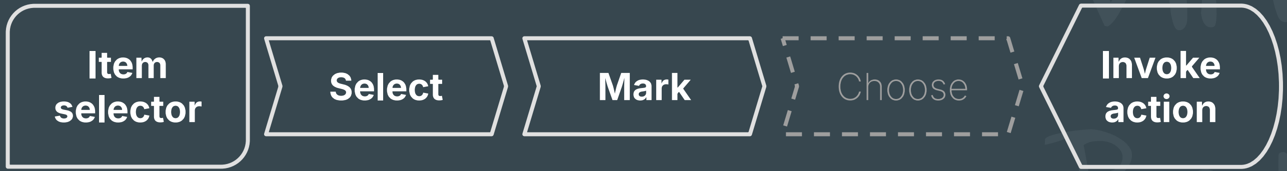
Filter

Sort

Render

Item selector

Action flow



Item selector

Select

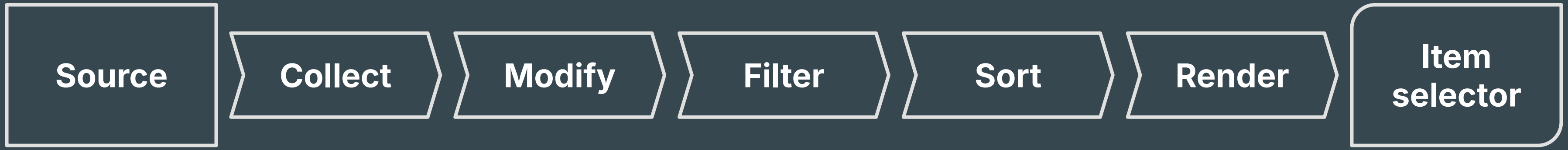
Mark

Choose

Invoke action

Flow - ddu.vim

Filtration flow



Source

Collect

Modify

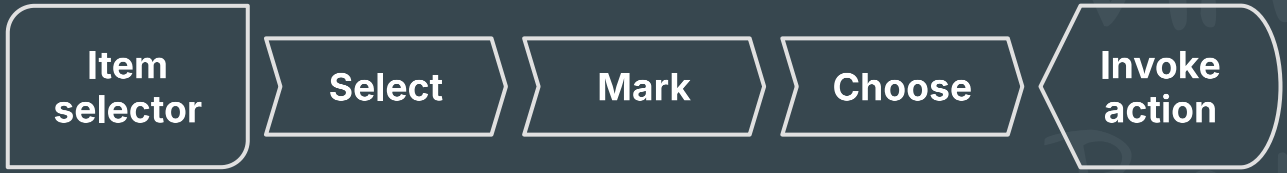
Filter

Sort

Render

Item selector

Action flow



Item selector

Select

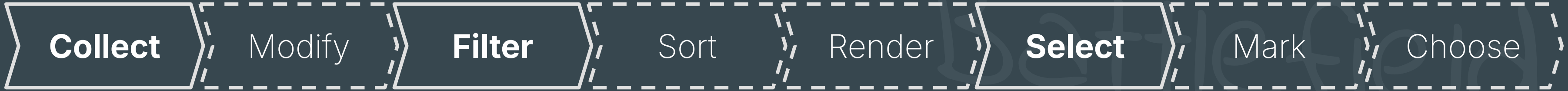
Mark

Choose

Invoke action

Flow classification

	C	M	**F**	S	R	**S**	M	C
ctrlp.vim	✓		✓			✓	✓	
fzf.vim	✓		✓			✓	✓	
telescope.nvim	✓	✓	✓	✓	✓	✓	✓	
ddu.vim	✓	✓	✓	✓	✓	✓	✓	✓
Fall	✓	✓	✓	✓	✓	✓	✓	✓



Feature classification

Floating window

Supports modern, flexible floating windows.

Modal or Modeless

Specifies if the interface uses modes or not.

Preview support

Allows live previews for quicker decisions.

Layouter support

Offers flexible component layout options.

Feature classification

	Floating	Modal	Preview	Layout
ctrlp.vim				
fzf.vim	✓		✓	
telescope.nvim	✓	✓	✓	✓
ddu.vim	✓	✓	✓	✓
Fall	✓		✓	✓