:turbovi

vi, in 3,000 lines of Python.

A rewrite of the parts of vi and vim you type every day, built on TurboDesk, an immediate-mode terminal UI library for Python. Small enough to read in an afternoon; solid enough to edit with every day.



```
    # `live` here is one frame behind the scroll and the size settled below, and that
    # is not a problem: setting an unequal value marks the frame dirty, so a session
    # that has just measured a new terminal is re-rendered before any key is delivered
    # and the handler that sees the key closes over the settled value.
    @ui.on_event
    def on_key(event: Event) -> bool:
        if isinstance(event, Paste):
            set_live(session.pasted(live, event.text))
            return True
        key = encode(event)
        if key is None:
            return False
        if key == "<C-c>":  # vi answers Ctrl-C with a nudge, it does not quit
            set_live(session.settled(live, editor.noted("type :q to exit")))
            return True
        set_live(session.apply(live, key))
        return True

    # Keep a copy of every buffer that differs from its file, and drop it the moment it
    # does not. `:w` clears `modified`, so writing takes the copy away by itself.
    ui.every(RECOVERY_SECONDS, lambda: _keep(live))

    if editor.quit:
        for buffer in live.buffers:
            if buffer.path:
                persist.drop_recovery(buffer.path)
        ui.exit()

    # Setting an equal value does not schedule a frame, so storing the scroll this frame
    # settled on is free on every frame that did not scroll.
    scrolled = session.followed(live, ui.size)
    set_live(scrolled)

    ui.set_cursor(
        screen.CURSOR, kind="bar-blinking" if editor.mode == INSERT else "block"
    )
    view = session.render(scrolled, ui.size, ui.theme)
    ui.set_wraps(tag for tag in view.tags if isinstance(tag, screen.Wrap))
    return view
NORMAL  src/turbovi/__main__.py                                  139:1  62%
```

Abilian · turbovi.lab.abilian.com · Built on TurboDesk: turbodesk.lab.abilian.com

THE IDEA

How much of vi fits in a small program?

The grammar

Counts, operators and motions, text objects, registers, marks, macros and . repeat.

The editor

Buffers and split windows, :s, find in project with a quickfix list, colouring and crash recovery.

Not the rest*

No plugins, no Vimscript, no :g, no folds. Searches are Python's re, not vim's dialect.

Nothing here is new to a vim user. The point is how little code it takes, and what it is built on.

* Some additional features may be added over time.

TURBODESK

An app is a function from state to a view

```
from turbodesk import View, run

def app(ui):
    return
View.text("Hello!").center(within=ui.size)

run(app)
```

Immediate mode

The function re-runs when a frame is dirty; the renderer diffs it by row.

Hooks for state

ui.state, ui.every and ui.task live on the handle, matched by call order.

Views are values

An immutable tree, built from combinators and never patched in place.

turbovi borrows two widgets, a file picker and a border. The text, gutter, status line and command line are drawn from views by hand: proof that the core carries an application on its own.

ONE KEYSTROKE

Each key turns one state into the next

keymap

`State, key → State`

The grammar. Pure, with no UI.

787 lines of code

→

session

`buffers, windows`

What the keymap asks for.

569 lines of code

→

screen

`State → View`

Text, gutter and status.

229 lines of code

→

TurboDesk

`View → terminal`

Diffs by row, writes escapes.

the library

Undo is a list

State is a frozen dataclass, so undo, the . repeat and macros are storage rather than machinery.

Requests, not side effects

:e, :sp and the clipboard are recorded on State and carried out by the session. That is what makes them testable.

BY THE NUMBERS

Small, and held to account

3,039
lines of code, plus 1,208 of comments and docstrings

601
tests in three tiers, the last driving the real program on a pty

2
TurboDesk widgets used; everything else is drawn from views

1
dependency: TurboDesk, which brings Rich and Pygments

No linter silenced

Ruff with every rule on, then ty, pyrefly, zuban and mypy.

Mutation-checked

A test is run against a deliberate bug before it is trusted.

In daily use

It is the editor its author works in, every day.

AVAILABLE NOW

Try it today

macOS · Linux · Python 3.12 or later

```
uvx turbovi file.py
uv tool install turbovi
```

The first line tries it without installing; the second installs turbovi and its short name, tvi. Inside, :help lists what is there and :q leaves.

Docs: turbovi.lab.abilian.com

TurboDesk: turbodesk.lab.abilian.com

Package: pypi.org/project/turbovi

Free software under the Apache 2.0 licence, made by Abilian in France. turbovi stands on TurboDesk, Rich and Pygments. Vim was created by Bram Moolenaar (RIP, he was a cool hacker); turbovi is not affiliated with it.