

Speedcubing in Emacs

Vasilij Schneidermann

November 2023

Outline

- 1 Intro
- 2 The wca-prep package
- 3 Outro

Section 1

Intro

About

- Vasilij Schneidermann, 31
- Cyber security consultant at CODE WHITE GmbH
- mail@vasilij.de
- <https://depp.brause.cc/>
- <https://emacsninja.com/>

Motivation and challenges involved

- I picked up speedcubing again
- Cubing hardware has improved a lot
- Scrambling and timing essential for getting better
- Seemingly all good options are web/mobile (cstimer, twisty-timer)
- No decent options inside Emacs
- Case study how to do better
- Making use of new Emacs features (svg.el, transient, sqlite)
- Most importantly: Having fun

Prior art

- <https://github.com/rafoo/stimer> (timer, shows stats)
- <https://github.com/param108/rubiks.el> (solver)
- <https://github.com/deadblackclover/scramble-el> (text-only scrambler)
- <https://github.com/queyenth/rubik.el> (scrambler, unfinished timer)
- <https://github.com/Kurvivor19/rubik-mode> (cube simulator)
- <https://gitlab.com/funkenblatt/elisp/-/blob/master/rubik.el> (group theory)
- <https://github.com/lukasjuhrich/cubing> (algorithm notation)
- <https://codeberg.org/akib/emacs-cube> (3D cube simulator)
- None of those does it all :(

Section 2

The wca-prep package

WCA what?

- Coming up with a good name was difficult
- World Cube Association holds events for twisty puzzles
- They standardized the puzzles, rules and notation
- Goal of my package: Helping people prepare for a WCA competition
- Therefore the name wca-prep was chosen
- This requires: Scrambler, scramble visualization, timer, stats

Non-goals

- To get somewhere, scope needs to be kept small
- Focus on 3x3x3 cube for now, expand later
- Out of scope:
 - Solving the cube (the speedcuber does that)
 - Simulating a cube (simulated cubes are awkward to use)
 - Non-WCA puzzles (there are a ton, many 3x3x3 variations, some are challenging to visualize)
 - Exotic WCA events (fewest moves challenge, blind solves)
 - Specialized scrambles (helping with algorithm practice)

Demo time

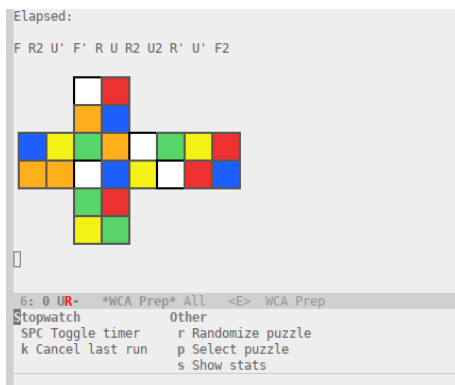


Figure: User interface

Challenge: Puzzle representation

- Many possible representations for cube state
- There is no obvious best representation
- Since programmatic solving is out of scope, I picked the easiest possible one (array of numbered facelets)
- Awkward to do scrambles on, but otherwise good enough
- Future: Learn group theory, pick better representation

Challenge: Scrambling

- After deciding on a representation, this should be easy
- A turn ends up swapping several facelets with other facelets
- Determining which facelets was very confusing
- One helpful suggestion: Make a paper cube with numbers on it
- `c1-rotatef` was highly useful for in-place swapping
- Future: Make use of group theory and better representation

Challenge: Scrambling



Figure: Two very different cubes

Challenge: Scramble algorithm

- WCA: 3x3x3 cube is scrambled by generating a random cube, solving it and emitting the sequence of solve moves
- WCA: Solve moves are canonicalized (folding subsequent moves when possible)
- Example: Rotate 90° CW, rotate 90° CW -> rotate 180° CW
- Seemingly all other scramblers: Generate random moves, some canonicalize
- This scrambler: Generates random move, canonicalizes with previous move, repeats until desired amount of moves has been generated

Challenge: Scramble visualization

- Mapping between every facet and unfolded puzzle view was generated
- Out of scope: Facet orientation (non-WCA puzzles would require this)
- Initial prototype used colored text
- `svg.el` turned out to be easy enough to use
- Future: Eliminate hardcoded puzzle face/color mapping for theming

Challenge: Intuitive UI with transient

- Idea: Self-documenting, magit-style UI
- Transient has become part of Emacs, so why not
- Issue: Documentation is difficult to understand, unclear what best practices are or how things should be done
- transient-showcase helps with examples, they feel half-done though
- Issue: Unhelpful error messages when using transient wrong
- Future: Find more packages using it and try to make better use of it

Challenge: SQLite book-keeping

- SQLite support landed in Emacs recently (29.1)
- It's still early days (found a bug in transaction macro)
- API is very basic, convenience helpers are missing
- The biggest challenge was figuring out a decent schema
- Finally no more persistence issues

Section 3

Outro

Lessons learned

- There's still plenty packages for Emacs to be written
- The 3x3x3 cube does not require advanced mathematics
- However: It proves there's finally a real-world use for group theory
- The new Emacs features are good enough, but need more testing
- Call for action: Please make use of SQLite, report bugs, ...

What next?

- More testing and UI improvements
- Better stats
- More puzzles
- ...

Questions?