

Essay outline

Background

- Editors
 - The early editors (ED?)
 - First modeless editor
 - Emacs and vim (multimodal editors)
- Accessibility
 - Disabilities (e.g Carpal tunnel)
 - WCAG
 - Assistive technology
 - * Eye-tracker
 - * Text-to-speech
 - * Screen readers
- Voice programming
 - History
 - * Dragon naturally speaking
 - * Natlink
 - * dragonfly
 - * That one talk on youtube
 - Talon
 - Other solutions? (Caster, voicecoder, etc)

Structural editing

- Definition
- Examples
 - Scratch
 - Paredit (Cursive, fireplace.vim, etc)
- Language theory (mostly parsing)

Mouse free selection

- Tabbing around
- Vimium
- Better autocomplete menus

Architecture

- The elm architecture
 - Can this be a significantly relevant part, or is this only interesting for my implementation?
- Client/server
 - Why is keyword emulation not enough?
 - How can this improve accessibility? (intellij, jupyter)

Problem

Editing the structure of complex languages efficiently without the mouse.

Solution

- Searching through an abstract syntax tree
- Action based architecture
- Lenses? (formal editing semantics)

Literature

Really unsure about this point so far.

Evaluation

How would i evaluate the results? User tests are difficult due to the expected high learning curve. Also the target demographic is relatively small. If i were to do user tests, i would probably have to reach out to online communities (talon slack, caster chat).

Possibility for static analysis? It would be interesting if i could formally prove some accessibility features. This might work due to how elm works. Could be just using the type system, or peripheral tool. Not sure if this is feasible though.