

Grimoire: Using Git for Brain Management

Andrew Egbert¹

¹Affiliation not available

April 28, 2020

I summarize briefly the ideas behind Grimoire / Grok for the purposes of academic reference. Grimoire / Grok is a memory state saving application that is aimed both at expanding the current working space a mind has available, and also keeping track of a larger number of projects / ideas / factoids, than a mind is capable of by itself. This could be useful either for student studying purposes, for researchers, or (hopefully - further research needed) for mental disabilities such as Alzheimer's / dementia, which limit the brains ability to keep track of and recall certain thoughts or memories. For example, one could couple the application with some sort of unobtrusive heads up display, of which several types exist on the market (current the software is rendered in a browser, so it should work on a hud).

Grimoire / Grok has two modes. The "Grimoire" mode is aimed at collecting / preserving / updating / keeping track of a large number of segmented thoughts. Thoughts are organized by topic / item, for example: "calculus" / "stokes theorem" is one possible topic / item pair. Users may navigate to different topic / item notes through a central index, a search bar, or by linking from page to page with links. Thoughts are written in markdown / html / latex / javascript, with the primary mode being Markdown. Thoughts are generally separated by Markdown header, which serves a dual purpose for Grok mode. For a folder structure, thoughts are stored by grimoire / topic / item / (files related to this thought). Typically different Grimoires should correspond to contexts, such as 'work', 'home', 'hobby', 'school', 'research' etc. Topics are split up into subareas of context. For instance, school might have topics such as 'geometry', 'language', 'history', 'art'. Items then deal with specifics. For example, one might store in the programming / C++ / some items corresponding to strings, math, and so on. Thoughts, as files and a folder structure are tracked through git (200, 2009) so that any accidental changes can be reverted, and so that a clear progression of thoughts can be maintained. The author imagines that stronger cryptographic guarantees could be given to memory and mental state through signed git messages, although recently the hash function used in git (sha1) has been shown to be non-collision resistant by Google (citation needed), so some changes to the software would likely be necessary for human lifespan length use. Further types of security guarantees are likely possible (and likely desired, if one is to rely on such software for the integrity of ones thoughts).

While the Grimoire mode is aimed at context specific long-term memory recall, Grok mode is aimed at short-term, working memory improvement. Grok mode works by first selecting a subset of topics, and then the program proceeds through each item in the topic. Each item is split up by markdown headers. Each header is asked as a question, while the body below the header corresponds to the answer. The user is expected to actively improve and prune each note throughout the process of using grok mode. The user then decides how well they know each topic, and if they know a certain topic, selecting 'good' will increase the time before it is asked about again, similar to the Pimsleur scheduled memory learning (Pimsleur, 1967). In fact, the application uses a simple progression based on the Fibonacci sequence: items are quizzed again after 1, 1, 2, 3, 5, 8, ... days, assuming they are answered successfully each time. There are other software applications which do spaced repetition learning as well, such as Anki (citation needed). Grimoire / topic

/ item structure is exactly the same as in Grimoire mode, so that no additional work needs to be done to create quizzes. In this way, a user can quickly refresh a given topic shortly before it is needed.

References

- Distributed Git. In *Pro Git*, pages 107–142. Springer Nature, 2009. doi: 10.1007/978-1-4302-1834-0_5. URL https://doi.org/10.1007%2F978-1-4302-1834-0_5.
- Paul Pimsleur. A Memory Schedule. *The Modern Language Journal*, 51(2):73–75, feb 1967. doi: 10.1111/j.1540-4781.1967.tb06700.x. URL <https://doi.org/10.1111%2Fj.1540-4781.1967.tb06700.x>.