Proposta organizacional do Cabeça

O conteúdo abaixo está em inglês porque ele originalmente era parte das minhas anotações no Obsidian. São uma reflexão dos sistemas do Bookstack e como encaixar coisas nele.

UGD's Bookstack

This note is mostly for scribbling down details about how I'm organizing UGD's Bookstack/Wiki. Nothing here is set stone, and my colleagues may object to my decisions.

The Layers

Bookstacks are divided into four main layers:

1. Shelves
2. Books
3. Chapters
4. Pages

The idea being that each page correspond to a traditional Wiki page, and therefore each Chapter corresponds to a *group of related pages* and a book is a *group of categories*. The whole thing quickly grows out of proportion and overflows the relationship we expect from things called "Chapters" and "Books". There are even the "Shelves" above all that, which are too expansive if we're using Bookstack like this.

Instead, it seems to me each Page should be equivalent to a section of a traditional Wiki's page, and each Chapter correspond to a traditional page, and each Book cover a category. Shelves are larger meta-categories, and it's worth noting *Books can belong to multiple Shelves*.

For instance, here's how I imagine our **resources and references** would be organized:

1. **Shelf:** "Resources"
2. **Book:** Area/Field

3. **Chapter**: resource type (Tool, Reference, Guide, etc)
4. **Page:** the resource itself

Sometimes it might be worth going the opposite direction when structuring the hierarchy. For instance, I imagine all our **meeting records** should be organized thus:

1. **Page**: the specific meeting's record
2. **Chapter:** either all meetings of a type in a specific interval (all short meetings in a specific year) or all meetings in a specific interval full-stop (all meetings in the first half of a specific year). The former is clunkier, but allows us to separate meetings of different types more easily (prospective meetings, etc). The latter is more chronologically faithful but harder to peruse
3. **Book**: a book called "Meeting Records" or some such, in all likelihood
4. **Shelf**: So, this whole structure doesn't really call for an entire shelf, so I imagine there would be a special kind of shelf, a "meta" shelf for UGD, where this kind of data could be archived.

So far so good, but we're also beating around the bush here. The most important thing in the Wiki at the moment is arguably all the **records on past projects**. Even if they are all abandoned and most disappeared without a trace, they are at the very least *the kind of thing we want the Wiki to be used for*; they're important because we want them to be, and the system is meant to aid that.

The new question, then, is "should old projects be kept alongside ongoing ones, or should they be kept in an archive?" I see plenty of pros and cons of both approaches, chief among them being that moving chapters/books around as projects get abandoned is kind of messy, while simply tagging projects as active/inactive would allow one to separate/filter them at a glance. On the other hand, that does depend on how well Bookstack deals with tags. I'm also getting ahead of myself here: I haven't even considered how projects fit in the Bookstack hierarchy. Each project being a **chapter** with pages dedicated to each system/thing sounds like a safe bet, especially considering how little actual meat there is in most of them (that isn't exclusive to abandoned projects; even large active ones just don't have enough to justify filling multiple chapters). Wanting each project to fill a whole book is wishful thinking, even if it's easy to compare documentation to real-life books. Bookstack books are simply too big for this kind of correspondence.

So here's what I think we're working with **projectwise**:

1. **Chapter:** Project, either active or inactive
   - **Page**: Project systems and templated data, like a standardized "participants" page
2. **Book:** Collection of projects. Possibly **two books**, one for active projects and one for abandoned ones. That's very little, but trying to separate projects by, say, what year it began or what engine it uses sounds very arbitrary. Maybe there's a **different worthwhile classification**, like "videogame projects," "tabletop projects," "tools," etc.
   - **Tag:** if we can settle on a **type system** for projects, where each type is a book, then tags will be the way to assign active/inactive status to each project.
3. **Shelf:** who even knows, to be honest. A shelf containing *two* books (active/inactive) is rather silly, but if there are enough different *kinds* of projects I could see a "**projects shelf**" work

Revision #1
Created 1 May 2023 14:44:06 by Cabeça
Updated 1 May 2023 14:48:46 by Cabeça