fountain-parser v.0.1.0.0 README

Synopsis

`fountain-parser` is a small parser library for the Fountain screenplay format, fully supporting 1.1 version syntax and producing a simple, easy to grok AST.

`fountain-parser` is written in Haskell and it uses the Megaparsec library for parsing.

Disclaimer

Currently, this is *pre-alpha* software, not yet usable in productive form.

This software is distributed under the *BSD Three-Clause license*. See the LICENSE file for more details.

Motivation

The "*Developers*" section of the Fountain site provides a link to a parsing library in Objective C. This already presents a portability issue: there *are* projects that make it possible to bridge Objective C and Haskell, but they're platform- or framework-specific. It also employs a multi-pass stategy where every stage creates a modified version of the source, and it's heavily reliant on *Regular Expressions*.

Thus, to create a light-weight, performant and portable solution, it's necessary to start from scratch.

`fountain-parser` aims to power a series of command-line utilities for conversion from Fountain to a series of convenient formats, such as `.OTF`, `.TEX` or `.PDF`, without the intervention of thirds.

My software already supports Fountain

Of course. And the "*Apps*" section of the Fountain site lists a few that also import or export the format. **The caveat**: most are either cloud-based and/or proprietary. By favoring (mostly) open formats, *fountain-parse* allows integration into many FLOSS tools, helping the creation of compound documents (such as production bibles) and entirely non-proprietary workflows.

Implementation Specifics

- As per spec:
  - This library expects Fountain text to be encoded in UTF-8.
  - Tabs are converted into four spaces.
  - Your line-spacing is respected.
- All parsing functions expect `Text` inputs. File I/O is left to the application or framework.
- Formatting (boldface, underline) found in such entities as character names or scene headings is ignored.
- Vertical tabs an form feed characters are ignored. Use line returns and the Fountain form feed character sequence ("===") instead.
- The parser keeps everything, including notes, boneyards and synopses, in case the target format is capable of storing this metadata.

Building

GHC 9.6.7 and Cabal 3.0 (or greater) are required to compile and run the test suite (once implemented.)

The project uses the `GHC2021` language default. While it might be possible to compile it in earlier versions than 9.6.7, this default is only available since 9.2.1., so that constitutes a hard version limit for those who might wish to experiment.

Some of the included scripts require Linux or a Linux-like environment (e.g., MSYS2.)

Contact

Please create an issue if you find one.

I can be reached directly at *10951848+CübO̲fJúdãhsLîòn* ă(t) *users/noreply/gīthụb/cȯm* (without accents and replacing slashes by periods.)