COLBYN WADMAN

DEVELOPER

EMPLOYMENT

SPACE MONKEY

Decentralized, p2p cloud storage. 2012 | 16

Brief summer internship, under my mentor Tom Metge (when their office was located in Murray).

GALILEO

Prepaid credit card processor. 2014 - 2015 | 17 - 18

Data Center Tech.

UPLYNK/VERIZON DIGITAL MEDIA SERVICES

Multimedia streaming backend for primarily broadcast providers (such as ABC, and ESPN). 2015 - 2017 | 19 - 20

Jr. Developer.

SELF EMPLOYMENT

See the following section "Projects". 2017 - 2018 | 20 - 21 | Current

PROJECTS

THE HELM LANGUAGE (I.E. COMPILER)‡

https://github.com/SubSys/Compiler

A rather long term project of mine. The overall aspiration is to create a simple medium for CPU/(GP)GPU computing. Initially, intended to serve as the 'beginner friendly' language for a video game engine focused on streaming, or rather, more commonly known as 'cloud gaming'.

NOTABLE COMPONENTS:

- The Frontend Parser*
- Type Inference/Type Checking‡
- The rust code-gen⁂

For more information regarding the compiler, the readme page should go much more in-depth on various details of interest (via the given link to the repository).

OTHER PROJECTS TO ASK ABOUT‡

- Perceptual image hashing experiments

  https://github.com/colbyn/ContentID

  If this interests you, let me know ahead of time so I can clean up & recall the details (VIA reviewing a paper)....
- A CLI tool simular to 'cargo-web' for ghcjs.†
- A text editor compiled with ghcjs. So far all that works alright is my multi cursor implementation, since I just recently started this project.†
- A simple ghc/ghcjs macro that generates function stubs when compiled with ghc.†
  The elaborate, although this isn't ideal, the purpose of the aforementioned macro is to declare JavaScript/ECMAScript FFI declarations that will compile with normal GHC. I.e. You declare your FFI declarations using the provided macros that will become regular JavaScript FFI declarations when compiled with GHCJS, and function declaration stubs when compiled with normal GHC.
  All it really does is not brake my beloved ghc based development tools.
  E.g.

```
[Call.jsFFI|
    appendChild :: Instance.Node -> Instance.Node -> IO Instance.Node
    $r = Node.prototype.appendChild.call($1, $2);
|]
```

* https://github.com/SubSys/Compiler/tree/master/src/SLIR/HelmSyntax/Module/Core/Parser

‡ https://github.com/SubSys/Compiler/tree/master/src/SLIR/HelmSyntax/Module/Core/TypeCheck

⁂ https://github.com/SubSys/Compiler/blob/master/src/CGIR/Rust/Dev/DryRun.hs

Regarding ⁂: The dry-run/development compilation pipelines expect to be invoked in 'stack ghci'.

‡ If inclined to see, these are fairly presentable (something I quite enjoy doing). :)

† These projects will be open sourced as a whole when reasonably complete (or factored out). Likewise, if inclined to see, these are fairly presentable.

SKILLSET

FUNCTIONAL PROGRAMING

If it's not obvious already :)

- Haskell expert beginner
- Elm proficient
- PureScript beginner

WEB TECHNOLOGIES

To name a few.

- XML/HTML(5) proficient
- CSS(3) proficient
- SVG proficient
- JavaScript(ECMAScript) proficient
- Browser APIs
  - HTML5 (various interfaces)
  - DOM level 3, and 4 (various interfaces).
  - A various multitude of extensions to the aforementioned interfaces.
- WebIDL (sadly)
  - A rather contrived, ad-hock sudo language that had better typing information for FFI bindings from a typed language (with regards to Mozilla's own API references).

RUST BEGINNER

I'm comfortable with the language, likewise I quite enjoy it's type system. Not so with popular apis and the like, so i'm most certainly a beginner in this regard.

UNIX

General unix concepts; though mostly practical knowledge such as the command line/cli tools, and shell scripting.

ADAPTATION

Although it's been quite a while, I'm certainly adaptable with regards to relearning python and various other high level scripting languages.

SCHOOL

MATC

Basic IT certifications (special high school program). 2011 - 2012 | 16

WESTLAKE

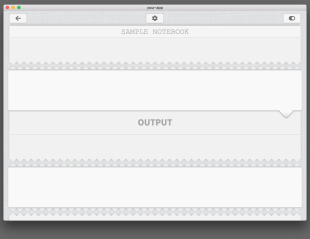
Graduated 2014 | 17

DESIGN BEGINNER

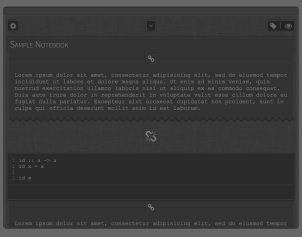
Experience, mostly limited to UI implementations.

EXAMPLE:

layout prototype I'm working on for a project



The above is based on an old idea, with renewed interest. With regards to the old notebook concept (pardon the syntax error):



PRINTER FRIENDLY VERSION

https://github.com/colbyn/resume/tree/master/assets/pre-built/forPrint.pdf

DIGITAL VERSION PREFERRED

(contains the "Design" section i.e. more information)

https://github.com/colbyn/resume/tree/master/assets/pre-built/default.pdf