Reinvent the Wheel

Zachary Ross

January 16, 2023

1 Motivation

Hello all. This document lays out the specifications for a new framework which attempts to completely reinvent web-frameworks, using a style similar to SwiftUI, except in Rust. The reason for this being that I am incredibly lazy and dislike the concept of splitting all web programs between front-end, back-end development along with all of the languages which compete in these spaces to accomplish what should be exteremly simple tasks. Take for example a regular web application. This generally requires an HTML/CSS/JS combination with various linkage patterns, design patterns, and in general knowledge of all three languages. SwiftUI is much simpler on the other hand because everything is unified and the tedious parts (e.g. routing) are all automated. I also in general just don't like the three languages associated with web dev but love Rust.

The main goals that need to be accomplished from the beginning:

- Automated routing based on Views
- Declarative view generations
- State management

We will probably just have to build this code as a transpiler to HTML/CSS/JS, but if WASM is available, maybe we can do dynamic state management using this.

2 Declaractive View Generation

The goal with this task is to build view generators which operate declaratively, almost as if they were in something like Haskell. The difficulty comes from wrestling with ownership in the Rust language, as having stack-like structures such as HStack, VStack, ZStack, etc., all require a different syntax.

In general, we model the problem as taking some data $x \in \mathcal{X}$ and a View builder $f : \mathcal{X} \to \mathcal{V}$ where $\mathcal{V}$ is the set of all possible views constructed using the nodes given by either HTML, SwiftUI, or any other framework, and constructing a view via $f(x)$. What's more difficult is that this needs to be done Monadically so that a tree of ViewBuilders is itself a ViewBuilder.