Flapjax

Introduction

Flapjax is built on top of JavaScript to exploit the availability the language (red. JavaScript) provides. It does differentiate itself from standard JavaScript libraries by being usable as a programming language, so it has been designed to accommodate both usages, though using it as a library does not work straight out of the box. Flapjax has two key features: event streams and that the language is reactive [1, p. 1].

Behaviors

Flapjax contains functions that they call behaviors. They are central to the language. If you end up building an expression as a behavior, all expressions become behaviors if their values depend on the first mentioned expression. This also helps the language by asserting that values of behaviors are automatically updated, so you can almost say that they act as variables that automatically gets updated over time. This might be the biggest plus there is to say about Flapjax. The validation example [3, validator] shows that the surrounding boxes are updated with a green color, when the text boxes are filled out correctly. Behaviors have been added to the outmost div and the text field inside it. When the behavior is updated for the text fields, the result propagates to the next element (the outmost div) [2]. Like they say in [1, p. 3], “the flapjax developer has left maintenance of consistency to the language”. The DOM (Document Object Model) already acts as a behavior according to [1, p. 4]. The authors [1] argue that is should not only be the bottommost layer that allows access to this kind of functionality.

Event Streams

Another important aspect (mentioned in the introduction) is the notion of event streams. Values like time, position of the mouse or contents of a field change continuously. If this is updated during observation the system have to make sure that all parts that depend on this value are updated as well. Flapjax tries to make sure these values are automatically updated [2]. This might sound like the same as behaviors, however, behaviors do not cover the term of a stream perfectly. Behaviors for keeping track of the most current value are a poor model, as they would pause the system for the duration of the update. Streams push the values in a discrete manner through the entire system (updating dependencies), which ensures a consistent system. These are called event streams [2]. An example of a stream is [3, Draft Saver] and [3, Letter Count]. Letter Count keeps updating the count of letters in a text field by updating the value in the DOM, so the letter count can happen live. This is one of the areas, where Flapjax actually seems useful, because it provides the developer some tools to avoid writing extra code for a simple counter update and extracting and inserting values in the DOM, however, jQuery contains similar advantages and is prefered by the masses. You win behavior compared to JavaScript

though. The Draft Saver is a bit different, as it updates the value of a text field in a persistent storage to be shared by more people that uses the site. We explored this by writing something on different machines and saw the value propagate to the other by storing it on the server.

Drawbacks

For the most part web developers are used to have a set of tools and add libraries, when their current library is not capable of the needed feature. Flapjax is meant to work as a library and a language. It does not work out of the box and require a lot of attention to function as a library, which is a major drawback. Adding the language and another compiler onto the development stack might not be acceptable either. Using the language without many other tools/libraries does not seem likely as a lot of other libraries provide better support and the same features (and more). Flapjax does not use callbacks, which is something that are widely used in many other libraries for event listeners.

Discussion

The fact that they have not even cared to update their website in years or check that their demos are still working [3, Flickr Demo] tells us that the language/library is not very popular. The paper and the tutorial show some nice features and the whole notion of giving the developer access to the bottommost layer is a good idea, but there are just so many other alternatives that received more attention and adapted to the current market. A lot of bugs occurred when testing their tutorial and running their code [2]. We do not expect to recommend our future employer to use Flapjax, even though it contains some good ideas.

References

[1] Meyerovich, L. A., Guha, A., Baskin, J., Cooper, G. H., Greenberg, M., Bromfield, A., & Krishnamurthi, S. (2009, October). Flapjax: a programming language for Ajax applications. In ACM SIGPLAN Notices (Vol. 44, No. 10, pp. 1-20). ACM.

[2] Flapjax tutorial, http://www.flapjax-lang.org/tutorial/, visited 24/11-2015.

[3] Flapjax demos, http://www.flapjax-lang.org/demos/, visited 24/11-2015.