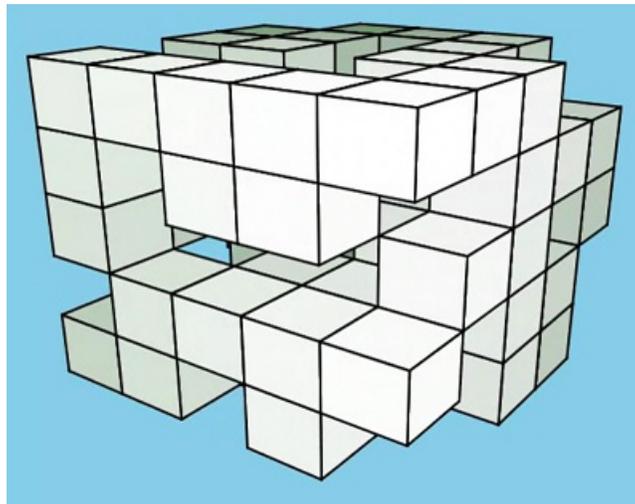


## 0.1 diagrams

Report by:	Brent Yorgey
Participants:	many
Status:	active development

The diagrams framework provides an embedded domain-specific language for declarative drawing. The overall vision is for diagrams to become a viable alternative to DSLs like MetaPost or Asymptote, but with the advantages of being *declarative*—describing what to draw, not how to draw it—and *embedded*—putting the entire power of Haskell (and Hackage) at the service of diagram creation. There is always more to be done, but diagrams is already quite fully-featured, with a comprehensive user manual and a growing set of tutorials, a large collection of primitive shapes and attributes, many different modes of composition, paths, cubic splines, images, text, arbitrary monoidal annotations, named subdiagrams, and more.

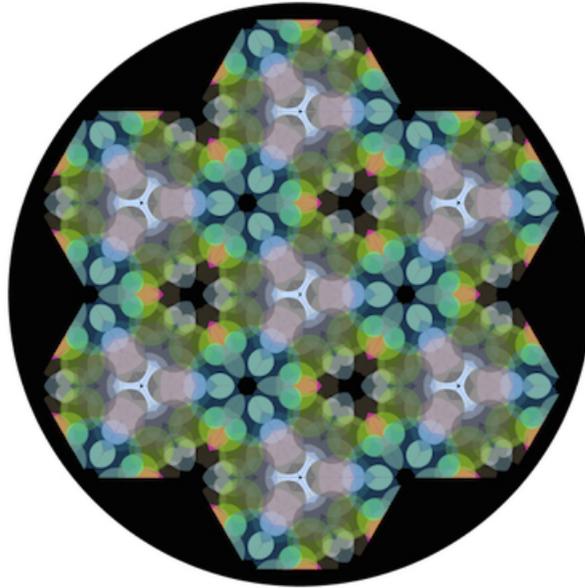


### What's new

Work on diagrams has slowed considerably since the release of diagrams 1.4 in October 2016, due to time constraints of the main developers. However, work on diagrams 2.0 is slowly but steadily progressing, targeting a release during the summer of 2018. Updates will include:

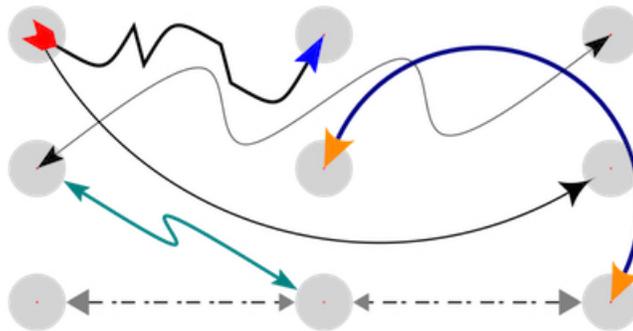
- Completely rewritten support for animations, with much better semantics and updated examples and tutorials.
- Death to the type-level “backend token”, which will allow much easier creation of diagrams that simultaneously work with multiple backends
- A complete rewrite of the library internals, resulting in better performance and enabling cool new features like diagram traversals.

- Lots of small updates and improvements.



### Contributing

There is plenty of exciting work to be done; new contributors are welcome! Diagrams has developed an encouraging, responsive, and fun developer community, and makes for a great opportunity to learn and hack on some “real-world” Haskell code. Because of its size, generality, and enthusiastic embrace of advanced type system features, diagrams can be intimidating to would-be users and contributors; however, we are actively working on new documentation and resources to help combat this. For more information on ways to contribute and how to get started, see the Contributing page on the diagrams wiki: <http://haskell.org/haskellwiki/Diagrams/Contributing>, or come hang out in the #diagrams IRC channel on freenode.



### **Further reading**

- <http://projects.haskell.org/diagrams>
- <http://projects.haskell.org/diagrams/gallery.html>
- <http://haskell.org/haskellwiki/Diagrams>
- <http://github.com/diagrams>
- <http://ozark.hendrix.edu/~yorgey/pub/monoid-pearl.pdf>
- <http://www.youtube.com/watch?v=X-8NCkD2vOw>