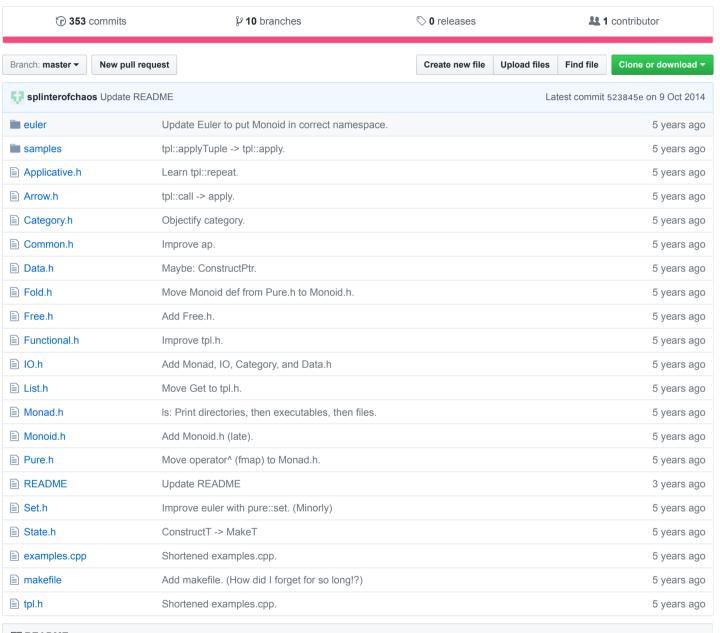


An almost-pure C++ library for writing functional code.



## **README**

PLEASE USE FTL INSTEAD (https://github.com/beark/ftl)

This project has not been maintained for quite some time and I don't have any plans to work on it. FTL has slightly more modern code, a more sane internal design, and even if it hasn't seen an update for quite a while, should probably be prefered. That being said, I do believe I've implemented a few things in Pure not in FTL, but they can be ported, and FTL has a few things missing from Pure such as sum types.

INTODUCTION

Pure is an experimental library to reimplement Haskell in C++. There are many reasons for wanting to do so but the main one is that it makes code easier to write in many ways. Explaining what the library does would be equivalent to explaining Haskell, so use and development of it requires a working knowledge of Haskell.

See the wiki for documentation.

Some of the techniques used to do this, however, are generally useful and I have been maintaining a blog about it:

http://yapb-soc.blogspot.com/

The basic functionality of the library comes from Functional.h and List.h. The first defines types and functions from which to build new functions. The most commonly used functions are closure, closet, compose, and ncompose. List.h defines almost all the functions from Haskell's Data.List. Monad.h defines Functors and Monads. Most of the other files define their Haskell counterparts with the same names.

- ./examples.cpp is a demonstration of each part of the library. When a new feature is added, an example is often added here to test it.
- ./euler contains solutions to problems 1-30, with a few missing, from the Euler Project. It uses mostly List.h functions.
- ./game contains a generic implementation of the AStar algorithm.

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About