

This repository


Search


Pull requests


Issues

Marketplace

Gist







splinterofchaos / Pure

Watch

13

Star

34

Fork

2

<> Code

Issues 1

Pull requests 0

Projects 0

Wiki

Insights

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

353 commits

10 branches

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

splinterofchaos

Update README

Latest commit 523845e on 9 Oct 2014

euler	Update Euler to put Monoid in correct namespace.	5 years ago
samples	tpl::applyTuple -> tpl::apply.	5 years ago
Applicative.h	Learn tpl::repeat.	5 years ago
Arrow.h	tpl::call -> apply.	5 years ago
Category.h	Objectify category.	5 years ago
Common.h	Improve ap.	5 years ago
Data.h	Maybe: ConstructPtr.	5 years ago
Fold.h	Move Monoid def from Pure.h to Monoid.h.	5 years ago
Free.h	Add Free.h.	5 years ago
Functional.h	Improve tpl.h.	5 years ago
IO.h	Add Monad, IO, Category, and Data.h	5 years ago
List.h	Move Get to tpl.h.	5 years ago
Monad.h	Is: Print directories, then executables, then files.	5 years ago
Monoid.h	Add Monoid.h (late).	5 years ago
Pure.h	Move operator^ (fmap) to Monad.h.	5 years ago
README	Update README	3 years ago
Set.h	Improve euler with pure::set. (Minorly)	5 years ago
State.h	ConstructT -> MakeT	5 years ago
examples.cpp	Shortened examples.cpp.	5 years ago
makefile	Add makefile. (How did I forget for so long!?)	5 years ago
tpl.h	Shortened examples.cpp.	5 years ago

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 preferred. 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.

