

Pan and Fran on Yampa

木戸崇裕 aka shelarcy

~~ とは

- Pan
 - DSL for Functional Image
- Fran
 - DSL for Functional Reactive Animation
- Yampa
 - DSL for Arrowised Reactive Programming

Pan

- Pan 自体は Haskell で書かれているが、実際の処理は出力した別の言語のプログラムで行われる
- Windows 環境のみで動作
- (Original) Pan -> C
- Pan# -> C#

Panic (Pan Interactive Client)

- wxHaskell 上に Pan を再現
 - 当然、Windows 以外の各種 Unix 環境上でも動く
- 100% Pure Haskell で動く
- しかしながら、Animation の機能が非常に貧弱

Fran

- Windows のみで動作
- FranTk
 - Fran の仕組みを利用して作られた GUI Library
 - Tcl/Tk を使っているためどの環境でも動くが、Fran と同等のことができる訳ではない

Yampa

- Fran や GUI Library の Fudgets などの DSL で使われている Reactive Programming という概念をライブラリ化したもの
- Monad よりも小回りが利くという風に考えられて出てきた Arrow を使用して実装されている

Yampa in GUI Library

- Fruit
 - Functional Reactive User Interface Toolkit
 - 低レベル部分では Java の GUI を利用
- wxFruit
 - wxHaskell 上に Fruit を実装しようという実験的プロジェクト

Pan を Yampa を使って

再実装する理由

- Windows 以外でも動く Fran が欲しい
- Panic は OpenGL を使っているため、うまくやれば Panic 上に Fran が実装できそう
- でも、IORef を存分に使っているため、Hack しにくい

実際に Yampa を使ってみて

- すっきりとした形にコードを書き直せる
- 変更が非常に容易
- でも、難しい
 - 学習曲線が高い
 - ドキュメント不足
 - 使っている人が少ない

Pan on Yampa の現状

- Graphics に対する Reactive Programming ができる
- でもまだまだ Panic にはほど遠い
 - 反応できる Event の種類はまだまだ少ない
 - ファイルから画像を読み取ることはできるけど、保存はできない