

# Haskell で GUI

shelarcy

# wxHaskell

- wxWidgets の Haskell バインディング
- IO を使用
  - `start :: IO a -> IO ()`
  - メニューやコントロールなどの作成
  - `set` 関数を使って状態を設定
    - 外見に反映されるもの
    - コールバック関数
    - など...
  - `get` 関数で状態を取得

# 良いライブラリは IO を使わせない

- ユーザーの書いたプログラムを、実際の IO 処理に変換してくれる関数を用意
- SOE (Haskell School of Expression)
  - FAL.lhs
    - `reactimate :: String -> Behavior Graphic -> IO ()`
    - ユーザーは Behavior a 型の関数を使ってプログラミング
  - Robots.lhs
    - `runRobot :: Robot () -> RobotState -> Grid -> IO ()`
    - ユーザーは Robot a 型の関数を使ってプログラミング

# wxSOE

- wxHaskell を使って SOE を実装
  - 未公開
    - 微妙に Windows や Mac ではうまく動かない
  - wxHaskell にマージ予定だったが……
  - その後の音沙汰なし

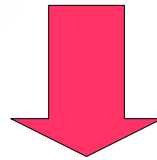
# wxFruit

- wxHaskell を使った high-level ライブラリ
- wxSOE より前に公開
- サポートしている機能が貧弱
  - ただし、SOE の FAL.lhs の paddle は動く
  - 逆に言えば、それに必要な機能しかない
  - なので、OpenGL Canvas 機能やダイアログボックスのサポートなどを入れてみましたが……まだまだこれから

現在のところ、OpenGL やダイアログボックスの部分では結局 IO に頼らざるを得ない……

# AutoForms

- データ型の定義を宣言すれば、その定義を見て関数を自動的に生成
  - Form の外観
  - Form 内でのデータのやり取り
- コントロールの作成や set 関数を使った設定などの煩わしい IO 処理を行わなくて良い



部分的に IO モナドから解放される