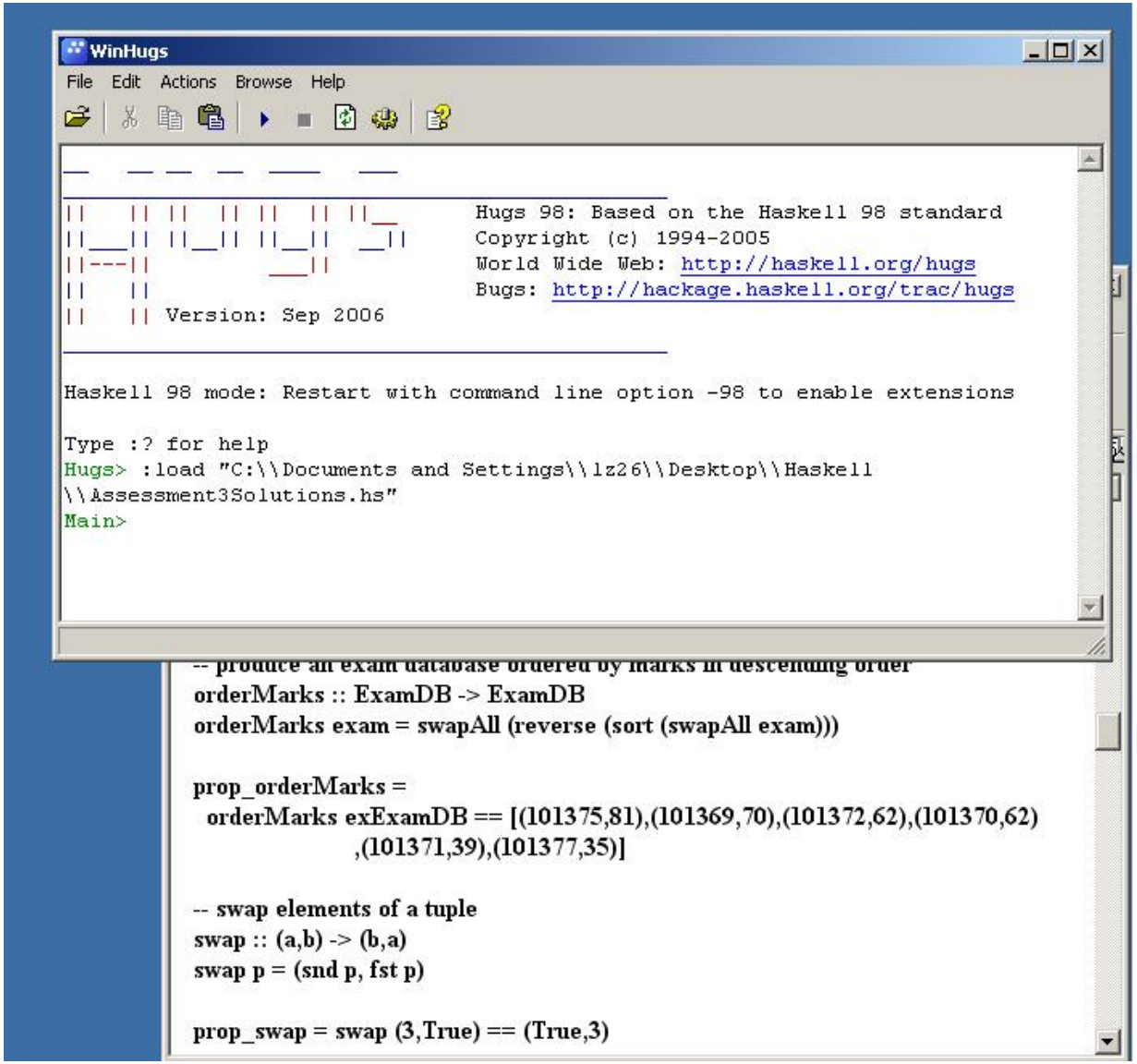
Heat — An Interactive Development Environment for Learning & Teaching Haskell

Olaf Chitil

11th September 2008

The Problem

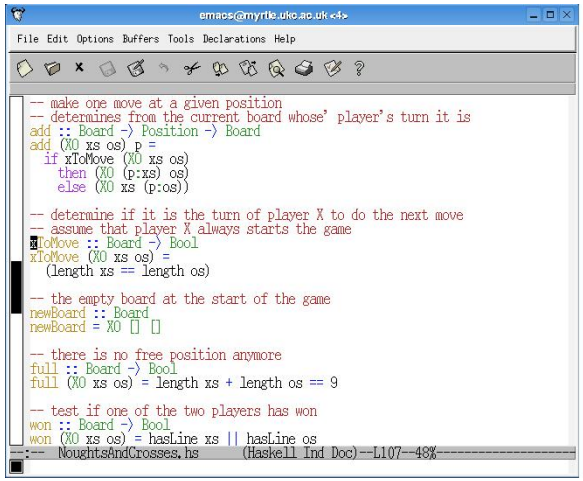


- two separate tools: editor + interpreter
- edit→save→switch →load - cycle
  - tiresome
  - easy to forget save or load
- command-line considered old-fashioned

Emacs and IDEs

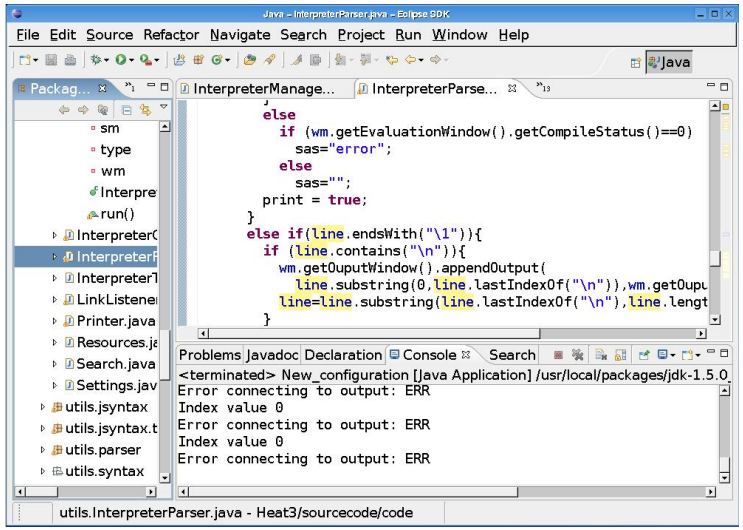
Emacs, gvim etc. with Haskell mode

- not fully integrated
  (evaluate wrt. old program)
- too many confusing features
- GUI not nice enough



Professional IDE: Eclipse, Visual Studio

- too complex
- Haskell modes not mature
- hard to install



Requirements

1. integrates editor and interpreter console within a single user interface

Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs

Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs
3. simple and fool-proof:
   only features that support students beginning to learn Haskell

Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs
3. simple and fool-proof:
   only features that support students beginning to learn Haskell
4. reliable

Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs
3. simple and fool-proof:
   only features that support students beginning to learn Haskell
4. reliable
5. runs on all major platforms used by our students and staff
   (Windows, OS X, Linux, Solaris)

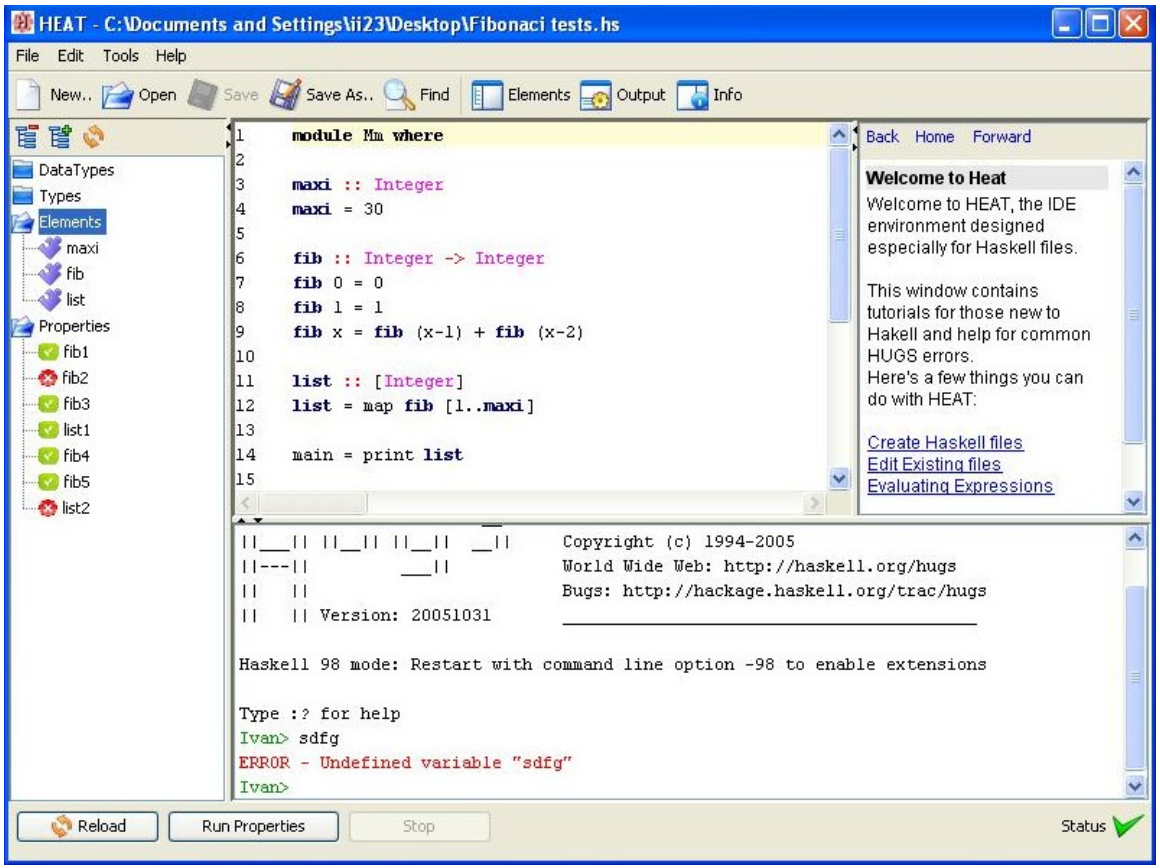
Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs
3. simple and fool-proof:
   only features that support students beginning to learn Haskell
4. reliable
5. runs on all major platforms used by our students and staff
   (Windows, OS X, Linux, Solaris)
6. easy to install

Requirements

1. integrates editor and interpreter console within a single user interface
2. provides a graphical user interface similar to professional IDEs
3. simple and fool-proof:
   only features that support students beginning to learn Haskell
4. reliable
5. runs on all major platforms used by our students and staff
   (Windows, OS X, Linux, Solaris)
6. easy to install
7. small source code and easy to maintain

The Solution: Heat



Editor

- for a single module
- syntax-highlighting
- matching brackets

Status: ? × √

Interpreter console

- highlight prompt & errors
- error: source line & explanation

Overview

- defined types, functions, . . .

Checking Properties

Design recipe

(*How To Design Programs*)

1. purpose in comment
2. type declaration
3. example properties
4. actual definition
5. test

```
-- yield square of given number
square :: Float -> Float
square x = x * x

prop_square1 = square 2 == 4
prop_square2 = square 0 == 0
prop_square3 = square (-4) == 16
```

Only Boolean unit tests, general QuickCheck later.

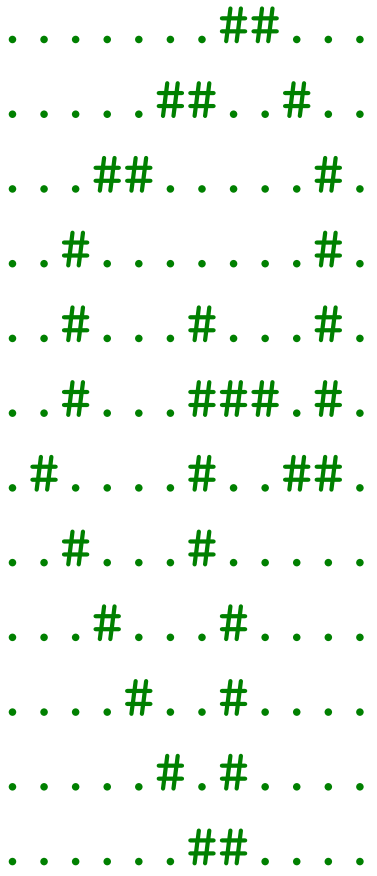
Properties separate in overview pane and automatic testing.

Composing Pictures

Simon Thompson *The Craft of Functional Programming*

A small library for making pictures compositionally:

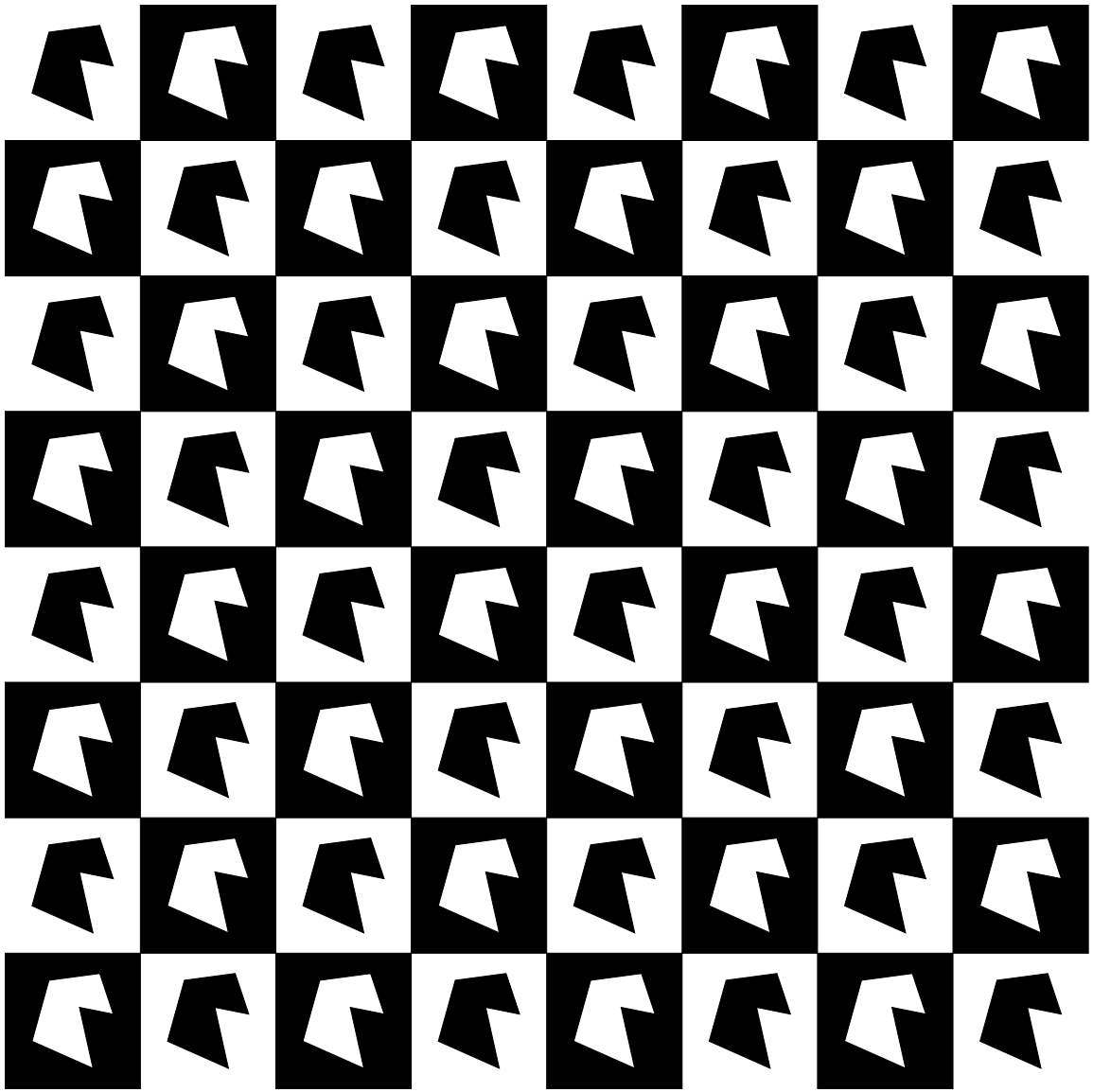
```
data Picture  -- abstract

horse :: Picture
rotate90, flipV, flipH, invertColour
  :: Picture -> Picture
above , sideBySide, superimpose
  :: Picture -> Picture -> Picture
```



ASCII art does not impress students!

Graphics via PDF

- generate a PDF file
- automatically start PDF viewer
- separate from Heat



Implementation

Written by final year project students.

- 15.000 lines of Java
- jEdit syntax package
  - lexing for syntax-highlighting
- Hugs as separate process
  - communication only via textual input/output stream
  - no interrupt signal: restart Hugs process
- new basic Haskell parser
  - Hugs' `:browse` and `:info` only works for valid Haskell
- graphics in PDF
  - incremental graphics model of PDF poor fit

Student Feedback

It's a great idea and much easier than using Hugs itself. However it was inconsistent on the Mac and a total disaster on Vista which with the number of both on campus (CS specifically) is not very useful. Preferences saving between sessions didn't work very well and the pop-up help did it's own thing.It has issues with line breaks too.

Evaluation

- Reliability (spurious bug) main issue

  Syntax error in expression (unexpected ';', possibly due to bad layout) is all that needs to be said.

- Fool-proof?
  Enter full path of Hugs...
- Expect standard GUI features
  shortcut keys, OS X shortcuts, automatic indentation, ...
- Missing or awkward features not a problem
  no interrupt; expression input box

Why Not Use the Glasgow Haskell Compiler?

- GHC's error messages more detailed but
  - many messages, no stop after first
  - mention language extensions: `fst :: forall a b. (a,b) -> a`
  - use substantial indentation
  - error location confusing, if in input expression
- GHC is harder to install
- GHC does not have built-in `observe` (future debugging)

Still, Heat should support GHC in future.

Big Role Model: DrScheme

Unfortunately relies on close integration of IDE with compiler and runtime system.

Different language levels with different error messages highly desirable, especially language with limited classes and without partial application.

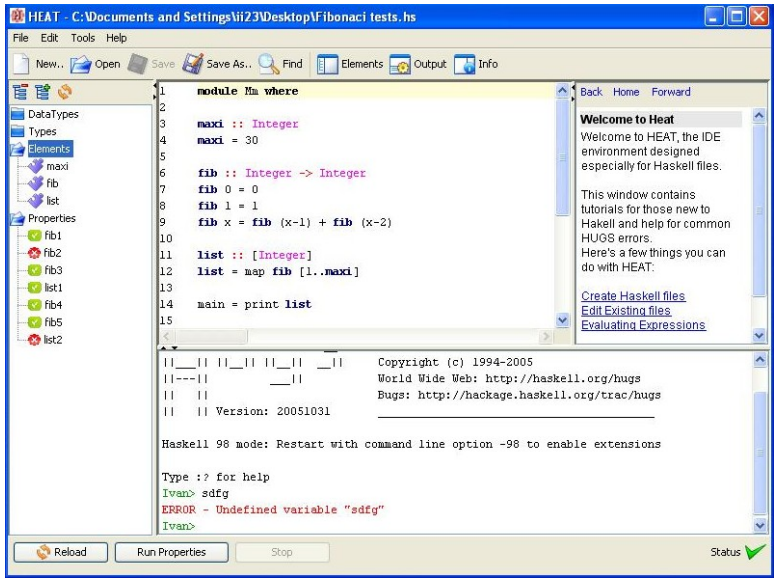
Future Extensions

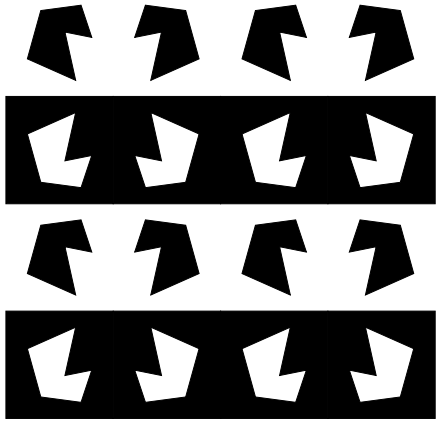
Not many!

- include library documentation
  - Haddock: extensible, standard
  - handwritten: Prelude for novices (several levels?)
- spelling help
- support full QuickCheck
- debugging via observing functions

Conclusions

- small but effective wrapper IDE possible
- reliability most important, not features





Availability

- Heat 1.1

  `http://www.cs.kent.ac.uk/teaching/resources/haskell/heat.html`
- Heat 3.0

  real soon ... watch Haskell mailing list