Scripting GIMP with Racket

Samuel A. Rebelsky
Grinnell College
(fifth RacketCon)
Context – Rethinking Intro CS
Context – Rethinking Intro CS

Scheme was a given
Context – Rethinking Intro CS

*Scheme was a given (HtDP should have been)*
Context – Rethinking Intro CS

Scheme was a given (HtDP should have been)

• Guzdial’s Media Computation
  – Students motivated by image making

• Eisenberg’s SchemePaint
  – Application scripting reveals the power of computing

• Walker and Stone’s Workshop CS
  – Learn by doing, not by listening
Let’s explore...
Some Outcomes

• Expected: Making and manipulating images motivates students
• Images let us do HOP early (cut and compose in week 2)
• Students tell us that images help them understand what’s going on with their code
• Naturally involves a lot of mathematical thinking
Architecture

- GIMP extension for Dbus procedure calls
  - Additional features include tile iterator and more compact color representation
- louDBus: Racket library (in C) for basic Dbus communication
  - “Inside Racket” model
- gigls: Racket library for novice scripting
  - Mostly Racket
  - Some parts in C (e.g., image-variant)
Selected Issues

- Racket already has a rich image library, do we really need other ways to make images?
  - Application scripting is a powerful model
- DBus is probably not the best platform (not even the second-best)
- GIMP doesn’t always immediately notice the changes – can lead to confusion
- Not everything in GIMP is scriptable
- Needs to be more Racket-like
Questions?

github.com/GlimmerLabs/louDBus
github.com/GlimmerLabs/gigls
github.com/GlimmerLabs/gimp-dbus
github.com/GlimmerLabs/virtual-mediascheme

rebelsky@grinnell.edu