

# Inform 7

Lightning Talk

Adam Thornton  
Lambda Lounge  
6 January 2011

# Inform 7

- Domain-specific language for development of text adventures

Yawn.

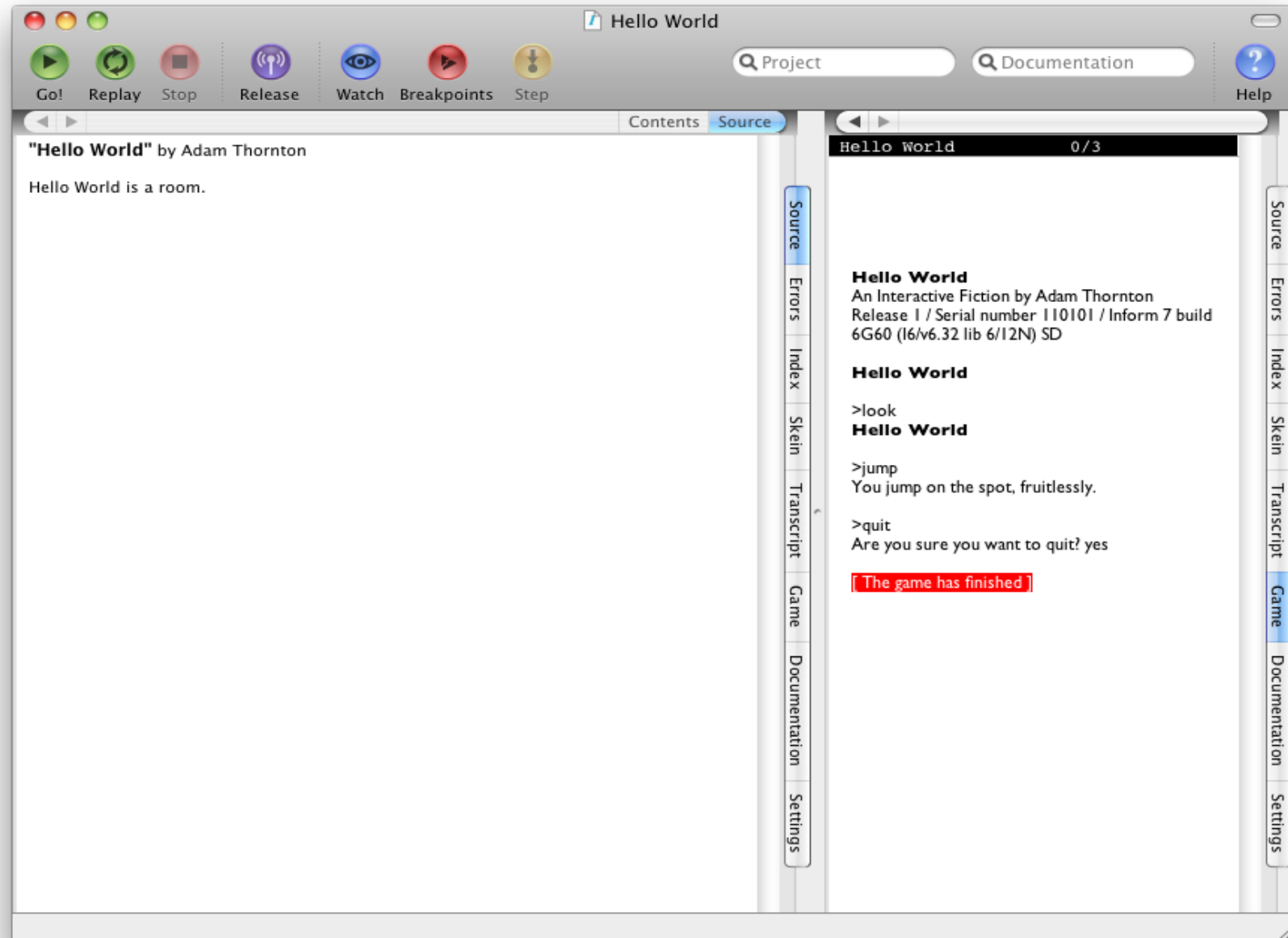
# Inform 7

- Declarative programming language
- With natural-language syntactic sugar
- Which is why it seemed that it might be interesting for Lambda Lounge
- Fun to try to wrap your head around; requires a different problem-solving approach than procedural, object-oriented, or functional languages

# Language Audience

- Designed to lower barrier to entry for writing text adventures, or, to be more modern and pretentious about it, “interactive fiction”
  - Writing ~~a text adventure~~ IF should feel like playing it
  - By the time you realize you're programming it's too late—you're committed
  - In short: probably not you
- I think it's fun to look at a language not designed for computer language geeks

# “Hello World”



# Where it gets interesting

- A text adventure is largely concerned with reasoning about items in the simulated world.
  - Build up a type system:
    - A coin is a kind of thing. Shininess is a kind of value. A coin has a shininess. The shininesses are shiny and dull. A coin is usually dull. The penny is a shiny coin in the Bank Vault.
    - Now you can refer to the penny with “penny”, “coin”, or “shiny”.
    - Adding synonyms is also trivial:
      - Understand “cent” as the penny.

# Verbs and clauses

- Standard Rules contain common text adventure verbs (e.g. “get”, “push”, “eat”, movement); also trivial to extend.

Understand the command **"earn"** as something new. Understand **"earn [something]"** as earning.

Earning is an action applying to one thing.

Report earning:

if the noun is not a coin, instead say **"You will have to be more specific about how you want to earn [the noun]."**

Carry out earning a coin:

if the noun is not shiny

begin;

instead say **"But that lucre is filthy!"**;

otherwise;

instead try taking the noun;

end if.



# Relations and Rules

- Now that you have kinds and values....
  - “Pet-ownership relates various animals to one person (called the owner).”
  - And you can test them with things like:
    - “If the owner of Golem is a man in the location....”
  - And you can have reciprocal relations:
    - “Meeting relates people to each other.”
- And rules are, in essence, functions
  - Can be named or anonymous; called with “consider” or “abide by”

# Other nifty data types

- Lists
  - Stacks, rings, queues
- “Indexed text” – basically text-with-regexp support
- Tables
  - Kind of an alternate way to do relations
- Scenes, stored actions, action names....
- See Documentation 21.1

# For instance, lists

- The Magpie verb: counts all visible shiny coins.

Carry out magpieing:  
consider the magpie rule.

This is the magpie rule:  
Let L be the list of all visible shiny coins;  
If L is empty:  
say "There are no shiny coins here. Bummer.";  
otherwise:  
let N be the number of entries in L;  
say "There [if N is 1]is[otherwise]are[end if] [N] shiny thing[if N is  
not 1]s[end if] here: [L with indefinite articles]. Er, 'Ha ha ha!'".

- Note that almost all the code is about pretty-printing the output. All the logic is in “list of all visible shiny coins” and “number of entries”.

This is the M2 rule:  
say "[Number of entries in list of all visible shiny coins] shiny coins: [List of all  
visible shiny coins]. 'Ha ha ha!'".

# What does Inform compile *to*?

- Inform (== “Natural Inform” == “ni” == “Inform 7”) compiles to Inform 6 ( == “i6” ) (sort of C-like) which compiles to zcode or glulx
  - Yuck. But consider OpenCOBOL, or C-INTERCAL, or first-gen C++
  - Zcode is the Infocom-story-format virtual machine; I7 cannot compile to z3 ( == *Zork* ), but can (for tiny games) to z5 ( == *Trinity* ) or, more usually, z8 (post-Infocom development, up to 512K (z5 is 256k) )
  - Glulx is a 32-bit analog to zcode (also separates presentation layer); 4GB is a very large text game

# Playing the output

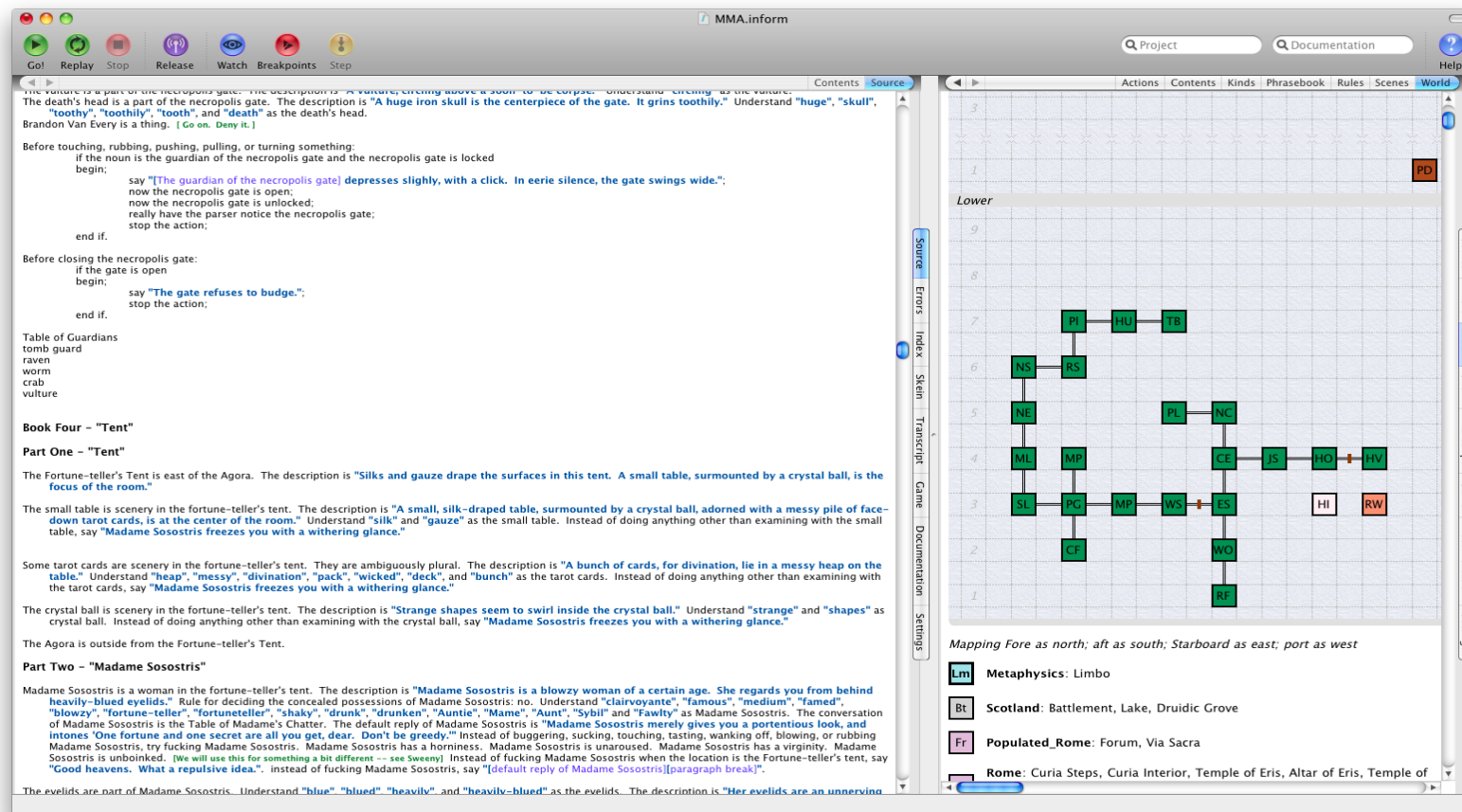
- Standalone interpreters (Frotz, Zoom, Gargoyle, Twisty, etc.)
  - Yes, there's iPhone Frotz, but playing non-bundled games is a PITA. Thanks, Apple.
- *Release along with an interpreter.*
  - Builds a couple of HTML pages and bundles game with a Javascript interpreter for glulx or zcode
  - For instance, <http://eblong.com/zarf/glulx/quixe/i7-release/play.html>

# Other nifty features

- Built-in unit testing with the “test” verb
  - *Test me with "/e/magpie/earn dime/earn quarter"*
  - In platforms with IDE, use this, plus the skein, plus fixing the RNG, to do automated regression testing
- Automapper/Index/Excellent documentation
- Implemented as a very large literate program
  - Source not yet available; will be someday
  - In “inweb”, a stripped-down perl implementation of CWEB

# A snippet from a larger WIP

- About 160,000 words right now; release in Spring Thing 2011 (early April)



# Available for:

- Full IDE: OS X, Windows, Linux i386/x86\_64
- CLI only:
  - Linux i386,x86\_64, ppc, armv5tel, s390, s390x
  - FreeBSD i386, amd64
- <http://www.inform7.com>



# Additional Inform 7 Resources

- Downloads/Extensions/Worked Examples
  - <http://www.inform7.com>
- Aaron Reed: Inform 7 for writers:
  - *Creating Interactive Fiction with Inform 7*, ISBN 1435455061
- Ron Newcomb: Inform 7 for programmers:
  - <http://www.plover.net/~pscion>

# Additional IF resources

- rec.arts.int-fiction and rec.games.int-fiction
- Jason Scott, *Get Lamp*
  - <http://www.getlamp.com>
- [www.ifarchive.org](http://www.ifarchive.org)
- [www.ifwiki.org](http://www.ifwiki.org)

# Contact

- Adam Thornton
  - [athornton@gmail.com](mailto:athornton@gmail.com)
- I'm happy to do a non-lightning talk about I7 or IF in general, if there's interest