

Inform 7

Adam Thornton
Lambda Lounge
3 March 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

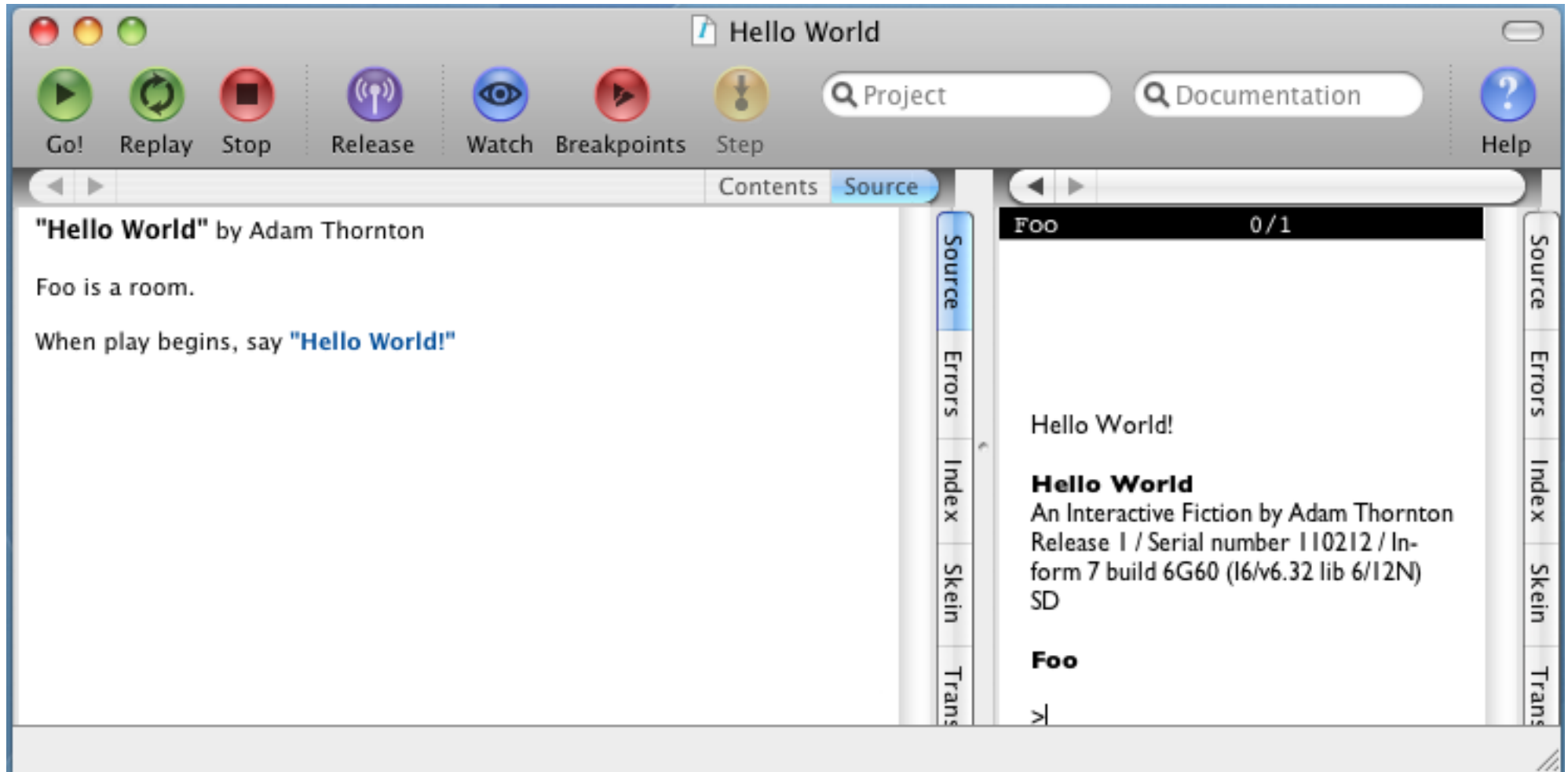
Audience

- Most functional or dynamic languages we study are intended for experienced programmers
 - Is Haskell anyone's first language? Fifth?
- Inform 7 is designed for nonprogrammers
 - In the proud tradition of BASIC, Applescript, and, yes, COBOL
- Basic conceit: writing a text adventure (modern, pretentious term: “Interactive Fiction”; henceforth, “IF”) should feel like playing one

Audience

- So why are we looking at it?
 - It's fun to look at a language designed for someone besides computer language geeks for a change
 - Syntax is delightfully weird
 - It's really fun and fast to write in, at least if you like Interactive Fiction
 - If you *don't*, you probably want to go take a smoke break for the next half hour. Fair warning.

Hello World



World Model

- A text adventure is usually concerned with simulating a world, and then reasoning about objects and events within the simulated world.
- World state modeled with hierarchy of built-in and user-extended types.
- Changes in state modeled with actions invoked by rules applying to game state.

Standard Rules

- Every game implicitly comes with “The Standard Rules”; this includes the object type hierarchy, 26 “activities” (e.g. “writing a paragraph about something”, “listing contents of something”), a bunch of standard text-adventure actions (“taking”, “telling it about”, “waving hands”), and sequence-of-play rulebooks.
- All of these things can be overridden, replaced, or delisted as desired by the author.

Built-in type hierarchy

- Object
 - Direction
 - Supporter
 - Room
 - Backdrop
 - Region
 - Device
 - Thing
 - Person
 - Man
 - Woman
 - Animal
 - Door
 - Container
 - Vehicle
 - Player's holdall
- next column
- That's all!

Before we get rolling: heads-up

- Identifiers can have spaces in them.
 - “Sasquatch's Den is a room.”
- What Inform calls a “kind”, most of us call a “class”:
 - “A coin is a kind of thing.”
- We can define adjectives which are then usable in play and in world-construction:
 - “Shininess is a kind of value. The shininesses are shiny and dull. A coin has a shininess. A coin is usually dull. The penny is a shiny coin in the Bank.”

Building the world model

- Done declaratively:
 - Muddy Field is a room. Scotland is a region. Edinburgh, Glasgow and Aberdeen are rooms in Scotland.
- Spatial relations (by default transitive)
 - Aberdeen is northeast of Glasgow. Edinburgh is east of Glasgow.
 - Glasgow is a room. “Gray and grim.” [This sets the “initial appearance” property.]

Populating the world

- Items are placed with declarations too:
 - An ugly American is a man in Europe.
 - The wooden table is a supporter in the kitchen. “A wobbly wooden table rests unsteadily on the floor.”
The description is “The table looks unsteady.”
Some butter is on the wooden table.
- And properties:
 - The butter can be edible. It is edible.
- Synonyms:
 - Understand “wobbly” and “unsteady” as the table.

World model

- Certainly, it's a simulation
 - But it's fundamentally a dramatic or narrative simulation, not a physical one
 - The language pushes this view with its constructions: things not in any room are “off-stage”, you “remove X from play”
 - Default sense-modeling and object-player interaction scoping is primitive
 - Other IF languages approach this differently: TADS 3 does very detailed sense-transmission modeling and more physics built into standard library

Kinds

- Object, number, time, truth state, text, indexed text, snippet, Unicode character, stored action, scene, table name, equation name, use option, action name, figure name, sound name, external file
- “List of” for any kind K, and “List of lists of...”
- List of K, description of K, relation of K to L, K based rule producing L, K based rulebook producing L, activity on K, phrase $K \rightarrow L$, K valued property, K valued table column

Adding Kinds

- Exactly what you'd expect. Either enumeration:

A sphere of dorkiness is a kind of value. The spheres of dorkiness are Trekkie, Star Wars Fanboy, Tentacle Hentai Enthusiast, and Joss Whedon Groupie.

- Or subclassing. And often both:

A nerd is a kind of person.

A nerd has a sphere of dorkiness. A nerd is usually a Joss Whedon Groupie.

Adjectives

- Understood by parser and in world construction
 - A coin is a kind of thing. A coin can be shiny or dull. A coin is usually dull.
 - Understand the shiny property as referring to a coin.
 - “referring to” makes it an adjective
 - A metal is a kind of value. The metals are copper, silver, and gold. A coin has a metal. Understand the metal property as describing a coin.
 - “describing” means you can use it as either a noun or an adjective
 - The penny is a shiny copper coin in the bank.

Scoping

- Primitive. Everything is either a global, or a lexical local defined with “let”.
 - It's easy to have confusing errors because you defined “the wooden table” but referred to it as “the table”, and the compiler thinks you're talking about “the table” you defined 20,000 words ago
 - Still, works pretty well in practice, once you're done being surprised the first couple times.
 - There's not much variable-passing, because typically things happen based on world-model state changes...so using globals actually kinda makes sense.

Rules

- The world model type hierarchy lets us create the nouns and adjectives. Now we need to act on those things with verbs.
- Application of rules to world model.
- Rule conditions map onto sorta-kind-of-like-a-regex-on-the-whole-world-state
- Perhaps an example will make this more clear. From Ron Newcomb's *Inform 7 for Programmers*:

Example rule

- *Instead of a suspicious person (called the suspect) burning something which is evidence against the suspect when the number of people in the location is at least two, try the suspect going a random valid direction.*
- Highlights: “Instead” is a rulebook name. “(called the suspect)” names a variable for reference later in the same rule. “Try” kicks off a new action and all its rulebooks. “Valid” is an adjective on class “Direction.”

Rules

- Sequence of play rules
 - “when play begins”
 - “every turn”
- Rules governing actions
 - Before, instead, check, carry out, report, and after are the big ones
- Preamble defines when rule is to be applied; code is what rule does. Preamble + code = rule

Example rules

- Check going up in Peak of Everest: instead say “There's no more 'up' from here.”
- Every turn when the nymph is in the location:
 - Let p be a random item enclosed by the player;
 - Say “The nymph adroitly relieves you of $[p]!$ ”;
 - Now p is held by the nymph.
- This is the stating the license rule:
 - say “This game is licensed under the Creative Commons 3.0 Attribution-NonCommercial-ShareAlike license.”

Adding new actions

- Define its arity, and come up with rules for its behavior. Really, the first part of *this* rule should be a “check” rule, not rolled into “report”

Digging is an action applying to one thing.

Report digging: if the noun is a person

begin;

say "I don't suppose [the noun] would care for that.";

otherwise;

if the noun is enclosed by the player

begin;

say "Since you are holding [the noun], you find this topologically confusing, and give up.";

otherwise;

say "You find [the noun] too hard to dig in.";

end if;

end if.

Understand "dig [something]" as digging. Understand "dig in [something]" as digging.

Rulebooks

- Every action gets a new Check, Report, and Carry out rulebook.
- Making a function call is a “to decide” rule (or “for deciding”. The thing that is decided on is the return value.
- “The fancy announce the score rule is listed instead of the announce the score rule in the carry out requesting the score rulebook.”

Rulebooks

This is the forcibly set the personal pronoun from multiple object lists rule:

```
let CI be the current item from the multiple object list;
if CI is a person
begin;
    if CI is a woman, forcibly set the female pronoun from CI;
    if CI is a man, forcibly set the male pronoun from CI;
    if CI is neuter, forcibly set the neuter pronoun from CI;
end if;
continue the activity.
```

The forcibly set the personal pronoun from multiple object lists rule is listed after the set pronouns from items from multiple object lists rule in the action-processing rules.

To forcibly set the female pronoun from (O – an object):

```
(- (SetPronoun('her', {O} ) ); -).
```

To forcibly set the male pronoun from (O – an object):

```
(- (SetPronoun('him', {O} ) ); -).
```

To forcibly set the neuter pronoun from (O – an object):

```
(- (SetPronoun('it', {O} ) ); -).
```

|

Relations

- Symmetric/asymmetric, one-to-one, one-to-various, various-to-one, various-to-various.

A role is a kind of thing. Characterization relates various roles to various people. The verb to be played by (he was played by) implies the characterization relation. Jennifer Aniston is a woman. Rachel is a role. Rachel was played by Jennifer Aniston.

- You can do most of this with tables too...which I find a lot easier.

Tables

- What they sound like. They have rows and columns; the things in a column are of the same type.

Choose a row in the Table of Real Cast Members with a Character entry of "Rachel". If the met entry is true....

| Character | Castmember | Met |
|--------------------------|---------------------------|-------|
| "Stiffy Makane" | "Ron Jeremy" | true |
| "Rachel" | "Jennifer Anniston" | false |
| "Librarian" | "Nancy Pearl" | false |
| "Cleopatra" | "Elizabeth Taylor" | false |
| "Madame Sosostriis" | "Prunella Scales" | false |
| "Pirate Captain" | "Johnny Depp" | false |
| "Persephone" | "Dolly Parton" | false |
| "Cerberus" | "Vinnie, Golem, and Ursa" | false |
| "Tantalus" | "Stephen Granade" | false |
| "The mysterious goddess" | "Rosario Dawson" | false |

Syntax and Sugar

- Natural language; some people find it inherently offensive. I don't mind it.
- Pythonic or Algolic block constructs
- Looping: usually done with “repeat with x running through...” rather than an explicit loop variable.
- Lists: apply (==map), filter, reduce. But it's cheesy because there's no lazy evaluation and there's no such thing as an infinite list.

More sugar

- Scenes – very useful for story purposes
 - Slowly drowning begins when the location is the Lake. Slowly drowning ends happily when the lifeguard is in the location. Slowly drowning ends hellishly when the location is Hades.
 - And then you can do things like
 - Every turn during slowly drowning....
- Regions: groups of locations
 - Can be nested, cannot overlap

WTF Language Features

- Dimensional Analysis:
 - “A length is a kind of value. 10 m specifies a length. An area is a kind of value. 10 sq m specifies an area. A length times a length specifies an area. 10 cu m specifies a volume. A length times an area specifies a volume.”
- And even equation support
- Equation – Volume of a parallelepiped

$$V=lwh$$

Where V is a volume, l is a length, w is a length, and h is a length

No, seriously, WTF ?

- Graham's a math professor; maybe there's some pedagogical aim here?

The screenshot shows the 'Equation Playground' application window. The title bar reads 'Equation Playground'. The top toolbar contains icons for 'Go!', 'Replay', 'Stop', 'Release', 'Watch', 'Breakpoints', 'Step', and 'Help'. Below the toolbar are search fields for 'Project' and 'Documentation'. The main content area is split into two panes. The left pane, titled 'Source', shows the source code for 'Equation Playground' by Adam Thornton. It includes sections for 'Part Zero - Definitions', 'Part e - Equations', and 'Part pi - Objects'. The right pane, titled 'Classroom', shows a scene with a 'Classroom' label and a '0/2' indicator. The scene contains text describing an 'infernial prism' and its volume and area.

"Equation Playground" by Adam Thornton

Part Zero - Definitions

Include Metric Units by Graham Nelson.
[Now we have to compile to Glulx, but we get lengths/areas/volumes and we also get fixed point math.]

Part e - Equations

Equation - Volume of a square parallelepiped
 $V = h \cdot l^2$
Where V is a volume, h is a length, and l is a length.

Equation - Area of a square
 $A = l^2$
Where A is an area and l is a length.

Part pi - Objects

Classroom is a room

The infernial prism is a thing in Classroom. It is fixed in place.

Carry out examining the infernial prism:
Let V be a random volume between 10 cu m and 1000 cu m;
Let A be a random area between 10 sq m and 100 sq m;
let l be given by the area of a square;
let h be given by the volume of a square parallelepiped;
say "The infernial prism shifts again. Now its height is [h].
Somehow you know its volume is [V], so the side of its
base must be [l] and the area of its base [A].";
stop the action.

Classroom 0/2

Equation Playground
An Interactive Fiction by Adam Thornton
Release 1 / Serial number 110213 / Inform 7
build 6G60 (16/v6.32 lib 6/12N) SD

Classroom
You can see an infernial prism here.

>x prism
The infernial prism shifts again. Now its height is 1.927m. Somehow you know its volume is 946.042489 cu m, so the side of its base must be 6.72m and the area of its base 45.1716 sq m.

IDE and programmer support

- OS X, Windows, and Linux GNOME ports: nice IDE.
 - Prettyprinting syntax and indentation
 - Headings for code management: Volume, Book, Part, Chapter
- Comprehensive documentation with (in IDE) Javascript pasting from Recipe Book or doc examples into source text
- Skein for playthrough-path management

IDE conveniences

- Automagic spatial index map (on Index tab) and as EPS
- Unit testing with the “test” command; also a way to fix-seed the PRNG for regression testing
- Comprehensive game index (Actions, Contents, Kinds, Phrasebook, Rules, Scenes, World) linked back into code
- Release along with....

Release along with the source text, cover art, the library card, a ["Quixe"](#) interpreter, a file of ["Walkthrough"](#) called ["walkthrough.txt"](#) [autogenerated from the source by stiffload], a file of ["Map \(spoilers\) \(PDF\)"](#) called ["MMA-map.pdf"](#) [Created by hand from the EPS using Illustrator for now], a file of ["Map \(spoilers\) \(EPS\)"](#) called ["MMA-map.eps"](#), and a ["Stiffy"](#) website. [^ "Stiffy" just recommends appropriate terps for Glulx]

More IDE goodies

- Breakpoints and watchpoints, code stepper
- Transcript management, with “play to here” function, “bless” a transcript (great for regression testing), Skein integration
-or you can use the underlying ni (“natural Inform”, command-line tool) compiler directly and write your source text in the editor of your choice.
- Collaborative I7 at Guncho: www.guncho.com

A snippet from a larger WIP

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

The screenshot displays the MMA.inform software interface. The left pane shows a text editor with code for a game scene, including actions like 'say' and 'begin'. The right pane shows a world map with a grid and various colored tiles representing different locations. A legend at the bottom right explains the tile colors: Limbo (Lm), Scotland (St), Populated_Rome (Fr), and Rome (Rm).

Text Editor Content:

```
The vulture is part of the necropolis gate. The description is "A vulture, cowering above a soon-to-be corpse. Understand "cowering" as the vulture.
The death's head is a part of the necropolis gate. The description is "A huge iron skull is the centerpiece of the gate. It grins toothily." Understand "huge", "skull",
"toothy", "toothily", "tooth", and "death" as the death's head.
Brandon Van Every is a thing. (Go on. Demy it.)

Before touching, rubbing, pushing, pulling, or turning something:
if the noun is the guardian of the necropolis gate and the necropolis gate is locked
begin;
say "[The guardian of the necropolis gate] depresses slightly, with a click. In eerie silence, the gate swings wide.";
now the necropolis gate is open;
now the necropolis gate is unlocked;
really have the parser notice the necropolis gate;
stop the action;
end if.

Before closing the necropolis gate:
if the gate is open
begin;
say "The gate refuses to budge.";
stop the action;
end if.

Table of Guardians
tomb guard
raven
worm
crab
vulture

Book Four - "Tent"
Part One - "Tent"

The Fortune-teller's Tent is east of the Agora. The description is "Silks and gauze drape the surfaces in this tent. A small table, surmounted by a crystal ball, is the
focus of the room."

The small table is scenery in the fortune-teller's tent. The description is "A small, silk-draped table, surmounted by a crystal ball, adorned with a messy pile of face-
down tarot cards, is at the center of the room." Understand "silk" and "gauze" as the small table. Instead of doing anything other than examining with the small
table, say "Madame Sosostris freezes you with a withering glance."

Some tarot cards are scenery in the fortune-teller's tent. They are ambiguously plural. The description is "A bunch of cards, for divination, lie in a messy heap on the
table." Understand "heap", "messy", "divination", "pack", "wicked", "deck", and "bunch" as the tarot cards. Instead of doing anything other than examining with
the tarot cards, say "Madame Sosostris freezes you with a withering glance."

The crystal ball is scenery in the fortune-teller's tent. The description is "Strange shapes seem to swirl inside the crystal ball." Understand "strange" and "shapes" as
crystal ball. Instead of doing anything other than examining with the crystal ball, say "Madame Sosostris freezes you with a withering glance."

The Agora is outside from the Fortune-teller's Tent.

Part Two - "Madame Sosostris"

Madame Sosostris is a woman in the fortune-teller's tent. The description is "Madame Sosostris is a blowzy woman of a certain age. She regards you from behind
heavily-blued eyelids." Rule for deciding the concealed possessions of Madame Sosostris: no. Understand "clairvoyante", "famous", "medium", "famed",
"blowzy", "fortune-teller", "fortunetellers", "shaky", "drunk", "drunken", "Auntie", "Mame", "Aunt", "Sybil" and "Fawly" as Madame Sosostris. The conversation
of Madame Sosostris is the Table of Madame's Chatter. The default reply of Madame Sosostris is "Madame Sosostris merely gives you a portentous look, and
intones 'One fortune and one secret are all you get, dear. Don't be greedy.'" Instead of bugging, sucking, touching, tasting, wanking off, blowing, or rubbing
Madame Sosostris, try fucking Madame Sosostris. Madame Sosostris has a horniness. Madame Sosostris is unaroused. Madame Sosostris has a virginity. Madame
Sosostris is unboinked. (We will use this for something a bit different -- see Sweeney) Instead of fucking Madame Sosostris when the location is the Fortune-teller's tent, say
"Good heavens. What a repulsive idea.", instead of fucking Madame Sosostris, say "[default reply of Madame Sosostris][paragraph break]".

The eyelids are part of Madame Sosostris. Understand "blue", "blued", "heavily", and "heavily-blued" as the eyelids. The description is "Her eyelids are an unnerving
```

World Map Legend:

- Metaphysics: Limbo
- Scotland: Battlement, Lake, Druidic Grove
- Populated_Rome: Forum, Via Sacra
- Rome: Curia Steps, Curia Interior, Temple of Eris, Altar of Eris, Temple of

Under the Hood

- Inform 7 compiles to Inform 6 (a much more traditional OO language), which in turn compiles to z-code or glulx virtual machine.
- Inform 6:
 - Sources: <http://inform7.com/sources/i6n/>
 - Manual: http://www.inform-fiction.org/manual/download_dm4.html
- Z-machine specs:
<http://www.inform-fiction.org/zmachine/standards/index.html>
- Glulx: <http://eblong.com/zarf/glulx/>

Inform 6

```
adam@filesaver: ~/IF/SMTUC — ssh — 80x24
File Edit Options Buffers Tools Inform Help
Message "[Including <smtuc-characters>]";

! The computer is ripped bleeding from DM4, Ex. 37 (p. 454)
! Yes, those are IBM-style error messages
Thingy computer "Computer"
  with name "computer" "dzogchen", ! Little joke there, folks
  grammar [;return 'dzogchen,'];
  ],
  speed 2,
  destination rigel,
  description "The ship's computer can hear you from anywhere in
    the ship (except when you're in a holodeck simulation).",
  before [;
    TalkTo: print "Unlike the actual characters, you address the
      computer with ";
      print (boldtext) "COMPUTER, ";
      print (italics) "order";
      ".";
    ],
  orders [x;
    if ((scene > 5) && (excretus has general) &&
-uu:---F1 smtuc-characters.inf Top L1 (Inform Fill)-----
Mark set
```

Inform 7 Source Code

- Implemented as a very large literate program
 - In “Inweb”, a superset of a subset of CWEB, implemented in Perl.
 - Actual code is ANSI C
 - Requires GNU make, GCC 3+, Perl 5
 - Interfaces have more dependencies
 - CLI requires Perl (or Python—XO port could be revived)
 - Not yet publicly available as source; will be someday. In the meantime, the Standard Rules and Inweb are available as Literate Programs:
<http://inform7.com/sources/webs/>

IF VMs

- Zcode: Infocom-format story file and virtual machine. 16-bit, 128K/256K/512K, depending on version.
- Glulx: 32-bit, a lot like zcode, but also separates presentation layer (into the glk API)
- Standalone zcode interpreters for damn near everything (including iPhone and Android) and glulx for most things modern (no mobile yet)
- Javascript terps; Android/iPhone doesn't quite work yet (displays, but no input). Fine in desktop browsers.

zcode/glulx Interpreters

- Standalone:
 - Gargoyle is my favorite: <http://ccxvii.net/gargoyle/>
 - Zoom's nice too:
<http://www.logicalshift.demon.co.uk/>
 - Or just poke around on www.ifarchive.org
- Javascript
 - Parchment (zcode):
<http://code.google.com/p/parchment/>
 - Quixe (glulx): <https://github.com/erkyrath/quixe>
 - Or just “release along with an interpreter”

Inform 7 Ports

- Full GUI on Mac OS X, Windows, Linux i386/amd64
- CLI only on Linux (i386,amd64, ppc, armv5tel, s390, s390x) and FreeBSD (i386, amd64)
- Used to be Solaris CLI ports too
 - if you want it and can make hardware available for me I'm happy to revive it. I stopped when I changed jobs and lost access to the Solaris boxes.
- And XO
 - If you have Python but not Perl. Could revive.

Resources

- Inform and Extensions: <http://www.inform7.com>
- Ron Newcomb, *Inform 7 for Programmers*:
<http://www.plover.net/~pscion/Inform%207%20for%20Programmers.pdf>
- Aaron Reed, *Creating Interactive Fiction with Inform 7* (ISBN 1435455061)
- Aaron Reed's *Blue Lacuna* source. 736 page PDF. Really.

<http://www.lacunastory.com/BlueLacunaSourceBook.pdf.zip>

Resources

- rec.arts.int-fiction and rec.games.int-fiction
- IF Comp (www.ifcomp.org) , Spring Thing (www.springthing.net)
- IF Archive: www.ifarchive.org
- IF Wiki: www.ifwiki.org
- *Get Lamp*: www.getlamp.com