

## HOLLOW VOICE

Whatever happened to the old Infocom implementors? How long does it take to write a text adventure? What's involved in testing these games anyway?

These are some of the questions that show up in my *XYZZYnews* mail repeatedly, and I'm happy to report that we have some partial answers in this issue.

First up is a where-is-he-now interview with Dave Lebling, co-author of the original mainframe *Zork* and creator of numerous other Infocom titles. I'm happy we were able to track him down, and even happier that he was good-spirited enough to share his thoughts on IF gaming today. This issue also includes Stephen Granade's blow-by-blow description of how his game *Waystation* came to be written over the course of five years. Whew! Next, C.E. Forman offers some good guidelines for aspiring beta-testers, with a sidebar from Neil deMause about the making of his game *Lost New York*.

I should point out that some of the other more frequently asked questions that come my way are answered in the *XYZZYnews* FAQ at <http://www.interport.net/~eileen/design/xyzzyfaq.html>. If you have a question and don't see it answered there, please continue to send your questions my way via e-mail. :)

On a highly unrelated note: I recently began working on a book (it's about Adobe Photoshop, that wonderful mainstay of graphic designers everywhere) for Prima Publishing. If I have time and am truly inspired, I hope to extend my renewed interest in the graphic arts to revamping the print version of the 'zine.

Until next issue, happy gaming!

**Eileen Mullin**  
*eileen@interport.net*

### Contents:

*Top 10 Picks for IF  
on the Web*.....2

*Letters*.....3

*Interview with  
Dave Lebling*.....5

*Tales From  
the Code Front:  
Interactive Fiction  
in Five Easy Years!*.....7

*Oh No...Beta!  
Tips and Techniques  
for Beta-Testing  
Games-in-Progress*....12

*Beta-Testers I  
Have Loved*.....18

*Game Review:  
The Light: Shelby's  
Addendum*.....19

*What's on the Disk*....20

XYZZYnews is published bimonthly by Bran Muffin Communications, 160 West 24th Street, # 7C, New York, NY 10011, USA. Email: eileen@interport.net. Send all inquiries, letters, and submissions to the address above.

Contents © 1996 XYZZYnews. All rights reserved. Published in the United States of America.

Electronic versions: There are currently three versions of XYZZYnews made available online. One is in ASCII and can be viewed with any text reader. You can also download a .PDF file that mirrors the layout of the print version. Use the Adobe Acrobat Reader (available for Windows, Mac, DOS and UNIX) to view the .PDF file; no special fonts or linked graphics are needed. You can obtain Acrobat Reader from ftp.adobe.com in the pub/adobe/applications/Acrobat folder, or <http://www.adobe.com/Software.html>. Thirdly, you can also read this issue of XYZZYnews on the World Wide Web at <http://www.interport.net/~eileen/xyzy.8.html>

Subscriptions: Both electronic versions are available at no cost. You can obtain either one by FTPing to the ftp.gmd.de/if-archive/magazines/XYZZYnews directory. To be added to the mailing list, please write to eileen@interport.net and specify text-only or .PDF version. The print version includes a 3.5" Mac or PC disk and is \$21 (U.S.) for one year (6 issues) or \$3.50 for a sample issue. For print subscriptions outside the U.S. or Canada, please email or write for rates.

All products, names, and services are trademarks or registered trademarks of their respective companies.

Editorial deadline for Issue #9 is April 30, 1996.

---

**Editor:**

Eileen Mullin

**Contributors to this issue:**

Stuart Beach  
Neil deMause  
Stephen Granade  
C.E. Forman

## March/April Top 10 Picks for IF on the World Wide Web

---

**Jeff Mallett's Games Page**

<http://www.cruzio.com/~tao/games.html>

**Revenge of the Z-Machine**

<http://wanda.pond.com/~russotto/zplet/ifol.html>

**Neil Bowers' Interactive Fiction page**

<http://www.khoros.unm.edu:80/staff/neilb/intfiction/index.html>

**The Economist Review / Multimedia feature: Interactive Fiction**

<http://www.economist.com/review/1rt14/review.htm>

**BrymmStone's Dark Tower**

<http://w3.one.net/%7Ebrymstne/homepage.htm>

**Veritas: A Harvard Game of Interactive Fiction**

<http://www-leland.stanford.edu/~jreese/veritas.html>

**Games For The Non-Illiterate**

<http://ourworld.compuserve.com/homepages/dae/int-fict.htm>

**Faye's Shrine of Zork**

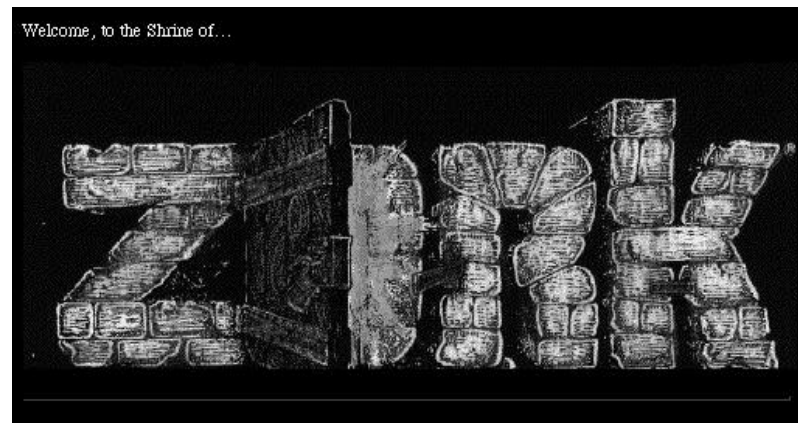
[http://www.bf.rmit.edu.au/~s9205250/Zork\\_Shrine.html](http://www.bf.rmit.edu.au/~s9205250/Zork_Shrine.html)

**Welcome to the Kingdom of Zork!**

<http://ampere.scale.uiuc.edu/~boraski/zork/>

**Infocom Pictures Index**

<http://yallara.cs.rmit.edu.au/~s9406702/if/infocompics.html>



Faye's Shrine of Zork.

Dear Eileen:

I first encountered bits and pieces of your magazine after a friend sent me some excerpts downloaded from Peter Scheyen's Infocom home page. I found the complete version of *XYZZYnews* while wandering around GMD's IF archive, and downloaded the ASCII versions of the first 3 issues. I was a bit frustrated with the double-spacing, but I was fascinated by the content. Then I downloaded a .PDF version of Issue #4 and printed it out.

(Insert lengthy pause here to indicate prolonged speechlessness.)

... I am in AWE. I had no idea there was *anything* like this available. Your magazine's layout is breathtaking, but the relevance of the contents far outstrips mere superficial attractiveness. Granted, the download time for the .PDF format is about 3 times greater than the text version, but I'll just have to ration my online time next month to get the other back issues.

Please send me *XYZZYnews* in the .PDF format. I'm hopelessly hooked, even though I have to endure the relative eternity required to print it out. ;)

Jay A. Goemmer  
jgoemmer@magiclinc.com

---

To *XYZZYnews*:

In Issue #7 of *XYZZYnews* ["Interface Changes: A Brief Look at the Evolution of the Adventure Game Engine," by C.E. Forman], you talk about Mystery House and parenthetically state "I'd recommend that IF history buffs pick up a copy if you can locate one."

Mystery House was released into the public domain by Roberta Williams on Sierra Online's 10th Anniversary. (I seem to remember it being '87, but you state it was released in '79.. Whatever, it's public domain now.)

You can find it at  
ftp://ftp.gmd.de/if-archive/  
games/appleII/mystery.dsk.

Other stuff: Tass Times in ToneTown is available for many platforms, including the GS. It can still be bought from Bill Heineman who did the GS version.

The ICOM series also was done for the GS, and may still be available.

—Matt Ackeret  
unknown@apple.com

---

To *XYZZYnews*:

Just found out about *XYZZYnews* and have been reading back issues to catch up. I read in issue #4 about your difficulties with IF, commuting, and balancing a coffee cup at the same time.

This is something in which I have some experience. I'm an engineer, and do some traveling. I carry a laptop most places, and usually end up using up my battery on one IF adventure or another.

For me, half the fun of adventures is mapping everything out. I maintain a master copy of my current adventure map in Autosketch, and work off the latest printout. Any new areas I discover, I draw by hand and add on the computer later on. I keep a file in my briefcase with my latest notes and scribblings. All this I can fit on the surface of my briefcase or on an airline tray.

Sorry, the cup of coffee doesn't fit with the above scenario. I found that out the hard way.

Great job, keep it up, and I'm looking forward to the next issue!

—Don Vande Polder  
DVandep@aol.com

---

To *XYZZYnews*:

It was only recently that I discovered the sizeable coterie of people who continue in their devotion to text adventures. I remember playing Adventure back when those who had 512K RAM were thought of as show-offs. What a joy to find that people still play (and, better still, write) interactive fiction.

I just read issue #5 of *XYZZYnews* and loved it. Nothing gives me more pleasure than the knowledge that there are other eccentrics out there, who, though not unimpressed by DOOM, nonetheless persist in their attachment to the GUE and its cousins. Imagine! *Other* people out there who will not finish their PhDs because they insist on procrastinating with text-games!

Thanks for your devotion.  
—Stephen Ramsay  
sjr3a@virginia.edu

---

To *XYZZYnews*:

I just discovered your *XYZZY* homepage and let me tell you.... IT'S AWWWESOME. They should definitely have more pages like this one. I have been a fan of Infocom games since the early '80s and anything else I could get my hands on. I even would settle for the graphics/text games

like Amazon. I have just began playing around with computers again (sparked by interest when I wandered through the software department and saw The Zork Anthology and the Infocom compilations) and couldn't believe some of the cool stuff I found when I entered in a search for Infocom on Yahoo.

Also do you have any suggestions on text-based games I might want to try out since I've been away for so long? I used to love Enchanter, Sorcerer, all the Zorks, and Hitchhiker's guide to name a few.

—crash@sun.tir.com

*If you haven't yet discovered it, let me suggest you check out Carl Muckenhoupt's Baf's Guide to the Interactive Fiction Archive (at <http://www.tiac.net/users/baf/if-guide.html>).—EM*

---

### Infocom Bugs List Update

To XYZZYnews:

Saw that you created/maintain the Infocom bug list, and I have two contributions.

Bug #1: You can steal the thief's stiletto. When the thief is in the room, TAKE STILETTO. He'll swing it out of your reach. Type AGAIN or G. Keep doing this until you're dead or he leaves the room. If he leaves, do it again and you'll see the message

Taken.

If you check your inventory, you'll find no stiletto. Type LOOK AT STILETTO to check out your ghost stiletto, which you can drop, pick up and otherwise use, though it will never show up on room descriptions or inventory lists.

If the thief meets you in the dungeon after you do this, you'll see a message something like:

You feel a light-fingered touch, and turn around to see the thief smiling at you.

He has stolen back his stiletto. However, if you go to the Treasure Room while you have the ghost stiletto, he won't be able to steal it back. He will still attack you with a stiletto (which can kill you), but if you attack him (with any weapon) while holding the ghost stiletto you'll see the message

The unarmed thief cannot defend himself: he dies.

Entertainingly enough, if you give the thief one of the game's other weapons, he will be able to dodge and parry your attacks, though the game will still insist that the thief is attacking with a stiletto. After giving the thief a weapon, though (except the stiletto), you can always simply TAKE it back—while he's in the room!—and kill him as though he is unarmed again.

AGAIN may be buggy elsewhere throughout Zork; it doesn't make the same checks full commands do. I know for a fact that I can't use it to take anything from the unicorn or Wizard, though both attempts resulted in the game behaving as if they were still in the room when they weren't.

Bug #2: in Beyond Zork, you can get a grue in a lighted room.

Kill the Ur-grue, but leave the other grues alive. Don't take the coconut. Wander around the Underground and wait until a grue enters the room. Then LIGHT LANTERN. You will be

able to examine, fiddle with, and otherwise harass a helpless "lurking presence" (which does nothing entertaining, but oh well). If you attack the grue in a lit room, it will register damage but will not counterattack and will not run away when slain. Turning off the lamp or leaving the room will allow the grue to flee if it has received its death-blow. Continuing to attack a grue after dealing it a decisive blow generates *no* response.

Both of these work with LTOI.

I always hated grues and I always hated the stupid thief. Killing that lean-and-hungry not-so-gentleman with his own stolen stiletto is one of the more satisfying things I've ever done with my computer.

—David Gildemeister  
dgildeme@magnus.acs.  
ohio-state.edu

---

To XYZZYnews:

I noticed a very minor bug in Enchanter, version: Release 29 / Serial number 860820

When the adventurer gives you something, no check is made to see whether you are carrying too much already, so you can end up carrying more than you would be able to normally.

For example:

```
>get pencil
Your load is too heavy.
```

```
>ask adventurer for pencil
The adventurer, not seeing any use in keeping the badly worn pencil anyway, hands it to you gladly.
```

—Neil B.  
neilb@khoral.com

# Interview with Dave Lebling



---

*Delve into the annals of adventure game history and you'll soon run across the works of Dave Lebling. This prolific Implementor co-authored mainframe Zork, Zork I, Zork II, Zork III, and Enchanter, and also wrote Starcross, Suspect, Spellbreaker, and The Lurking Horror. In our interview, we asked for an update on his post-Infocom activities and his thoughts on the current state of the gaming industry.*

---

**Q.** *How and when did you get involved in programming, and what led you to coding adventure games?*

I got involved in programming due to a fortuitous hole in my schedule in the first semester of my freshman year. My advisor said, "Why don't you take a programming course? It might come in handy." Well, I was more-or-less hooked from the start, and always had an interest in games and other fun stuff. I remember coding a version of the old Spacewar game, for example. I also was co-author of the first—that I know of—multiplayer MazeWars-style game; we just called it Maze. I helped Marc Blank and Tim Anderson [later fellow Zork implementors] write a Trivia game that maintained a user-contributed Trivia database of over a thousand questions. I had also written for my own use some Dungeons and Dragons "Dungeon Master Assistant" code that was still in a fragmentary state when Adventure hit MIT and changed everything.

**Q.** *Could you please summarize your post-Infocom activities, and what projects you're currently working on? What kind of work do you do at Avid?*

Well, I first worked on a cross-platform GUI spreadsheet. A very boring thing compared to Zork, but I learned a lot about C, Windows, X, Motif, and other nasty topics. It was also a great development group, so professionally it was a Good Thing, even if it wasn't gaming. At Avid, well... I could tell you, but I'd have to kill you afterward, if you catch my drift. Avid is a leader in software for producing and editing video, audio, and multimedia for the professional and broadcast marketplace. For example, "Home Improvement" is edited using Avid systems.

**Q.** *Do you still frequently hear from Infocom game fans? What kinds of things do they still ask you about most often?*

I wouldn't say "frequently," compared to how often I used to hear from them. In the old days I'd get strange phone calls a lot. When I started posting occasionally to rec.arts.int-fiction I began to get a reasonable amount of email, but it's nothing excessive.

**Q.** *Do you still keep in contact with any of the other Implementors?*

I see Tim Anderson, Steve Meretzky, and Stu Galley with some frequency—we all still live in the Boston area. I haven't actually seen any of the others in quite a few years, although I get Christmas cards or email every now and again.

**Q.** *How would you describe your involvement with the world of text adventure games these days? Do you still do any game development as a hobby?*

My involvement these days consists of reading r.a.i-f and r.games.i-f, browsing the Web, reading the magazines (such as Computer Gaming World), etc. I haven't actually done any

game development in quite a while—I've got a fairly large number of game ideas, but not a great deal of time!

**Q.** *Do you play the current games-under-discussion from GMD?*

I keep up with the newsgroups, but I haven't played any of the games. I feel bad about that, because some of them sound really fun. Perhaps I'll make time, but then, that's what I always said at Infocom, too—I eventually played all our games, but it often took a long time to get to them. I've now gone so far as to download a bunch of the new games. This is progress!

**Q.** *Do you have a favorite Infocom game? Of the ones you worked on, which one did you have the most fun developing?*

There are things I like about every Infocom game, even Shogun (which was, I think, both my worst and our worst). I've always been fond of Spellbreaker and Enchanter from my own list. I think I had the most fun writing The Lurking Horror, although I wish (we all wished) that it had been an "Ezip" (meaning larger-size) game. A lot of lovely shivers had to be cut out of the design, and some stuff out of the almost-finished product. It was a labor of love, set as it was at a thinly-disguised MIT, with lots of real places and a somewhat-accurate geography. Aside from the actual monsters, the course of the game duplicates "Institute Exploring" adventures I went on when I was a freshman.

**Q.** *In retrospect, are there any design changes you would have liked to have been able to make to your games, for example, different solutions for overly difficult puzzles?*

You know, one of the nice things about a game is that you usually do one version of it, and don't have to think too much about changing anything later. Still, the baseball puzzle in Zork 2 has always annoyed me. It arose from my hatred of mazes—I was always looking for ways to write mazes that weren't mazes if you guessed the "trick," but that one was pretty lame.

**Q.** *What are your thoughts about the current state of the computer gaming industry and the proliferation of blockbuster multimedia CD-ROM games?*

I think the biggest change since the Infocom days is the rise of expensive-to-develop titles. The sheer size of the budgets and development teams for today's adventure games (I won't call them "interactive fiction") is astonishing. It has made the industry more like the movie industry—a few big hits, a few mid-range successes, and a lot of dreck. Today, a lot of the titles that are fondest-remembered from the Infocom days, such as Trinity, would never be made—too esoteric, non-commercial, unlikely to make money.

**Q.** *What advice do you have for IF fans who say they wish they could make money writing and producing adventure games?*

I think it's unlikely anyone is going to make big bucks anymore writing text adventures. On the other hand, there's a CD-ROM adventure industry out there that could use an infusion of the enthusiasm for plot, character, internal consistency and so on that I see on the Web. There's no reason why graphic adventures have to be hoary old logic puzzles connected by video sequences. There's also plenty of opportunity to make money in that industry. The video and audio state of the art has advanced enormously in the last ten years, but the basic storytelling skills are in dire need of some new ideas. Maybe some of the IF fans out there will provide those ideas. ☼

# Interactive Fiction in Five Easy Years!

---

by Stephen Granade (sgranade@phy.duke.edu)

---

Write Your Own Interactive Fiction! v1.0 by Stephen Granade  
Developed with TADS, the Text Adventure Development System.  
Beginning game time: November 1989

---

## Enthusiasm, Part I

---

November 1989. The idea of writing an adventure game came to me while I was a high-school senior. It began life as an idea for a Sierra Online game. I had solved “Hitchhiker’s Guide to the Galaxy” and played “Ballyhoo” and “Infidel,” but didn’t know much about Infocom’s other games. On the other hand, I had solved almost every game Sierra produced from King’s Quest II through Space Quest III. I had grand visions of creating games for Sierra, a consequence of my surfeit of free time and lack of experience.

Despite these grand visions, I didn’t give serious thought to writing a game until I fell sick during Christmas break. Chained to my bed by illness, unable to sit in front of my computer, I was reduced to doodling on a pad. At one point I sketched a gauntlet which, when worn, would teleport the wearer to different locations. I began thinking about the consequences of such a device, how it could be used and abused. I suddenly realized that I had the concept for my game.

After that breakthrough, ideas began pouring forth. I decided to spread the game out over three worlds to give the game a feeling of space. The worlds would be connected by Waystations—teleportation booths resembling Dr. Who’s TARDIS. Naturally, the hero would have to save the universe, but from whom?

Humans, that’s who. I had just finished reading several dystopian novels, so the idea had a certain appeal. I pulled out my well-thumbed copies of “Writing Basic Adventure Programs for the TRS-80” [1] and “Creating Adventure Games on Your Computer” [2] and began work in earnest. Within a week I had the basic outline for Waystation completed.

Then I got well.

I belatedly discovered that being bedridden is one of the best ways to get work done. Waystation remained nothing more than fevered scratchings on paper for the next several months.

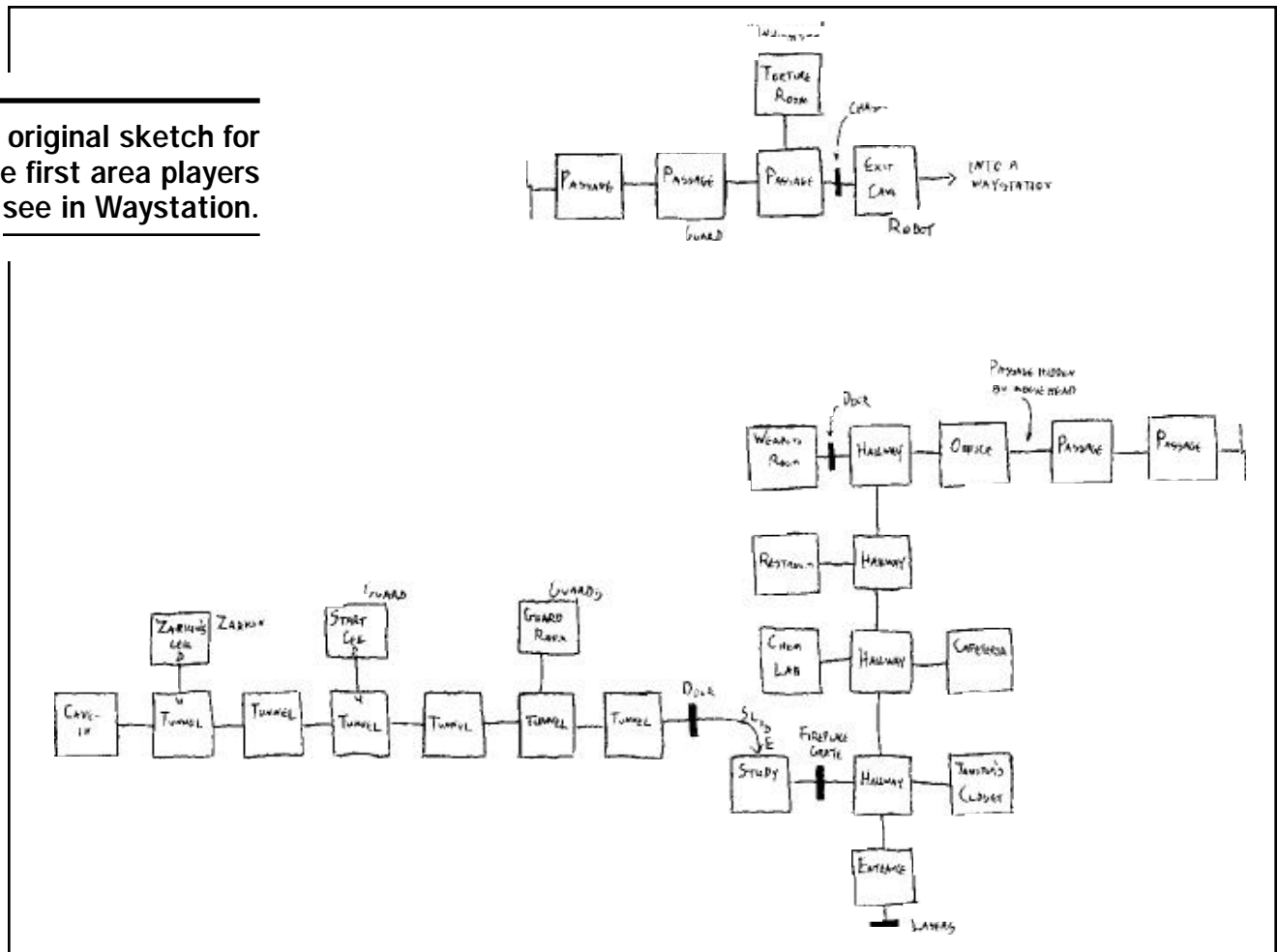
## Enthusiasm, Part 2

April 1990. With the coming of summer, I decided to finish Waystation and send it to Ken Williams, president of Sierra Online. I began by deciding what each world would be like, and deciding how the player would save the universe. Next came the map. I started with blank squares, wrote a two or three-word name for each room, then began designing puzzles for each room. After two months, I had the game designed. The final step was to write general descriptions for all of the rooms, a task which proved more daunting than I expected. One month and many thousands of words later, it was finished and sent via registered mail.

By August I had my reply. An anonymous Sierra secretary sent me a generic form letter. It contained about six paragraphs, ranging from "We loved your letter!" to "I'm afraid that Sierra Online does not give hints through the mail...." Each paragraph had a small blank in front of it; the paragraph which best answered your letter was checked. The checked paragraph in my letter said, "We are sorry to inform you that we cannot use your game ideas. Although we are constantly amazed by our customer's inventiveness...."

With a loud thud, all of my Waystation notes were tossed in the back of a closet.

An original sketch for the first area players see in Waystation.





---

## **AGT Rears its Head**

---

November 1991. A college friend showed me the copy of AGT he had downloaded from a local bulletin board. "You can write adventure games with this!" he told me excitedly.

Waystation is reborn. I hauled out my old notes and began work programming the game in AGT. I was able to take the room layout from the maps I had made. Writing the long descriptions of the rooms took some time, since my original descriptions were never intended to be actual printed descriptions. By January, the rooms were ready.

The puzzles were the stumbling block. AGT's metacommand language was quite sufficient for the simpler puzzles, but when I reached Waystation's kitchen area I was stumped. The next several months were spent trying to program and debug that one room. After a while, my enthusiasm dwindled away once more, leaving me with reams of AGT code and little else.

In my infinite wisdom, I decided to write my own interactive fiction language. It was modeled after AGT, the only language I was familiar with. I worked on it beginning in September of 1992, forgetting all about Waystation in the process.

---

## **TADS to the Rescue**

---

February 1993. The friend who introduced me to AGT handed me a copy of TADS. After perusing the documentation for one hour, I tossed the notes for my programming language in the closet and pulled out my notes for Waystation. For two weeks I played with TADS, getting a feel for the language.

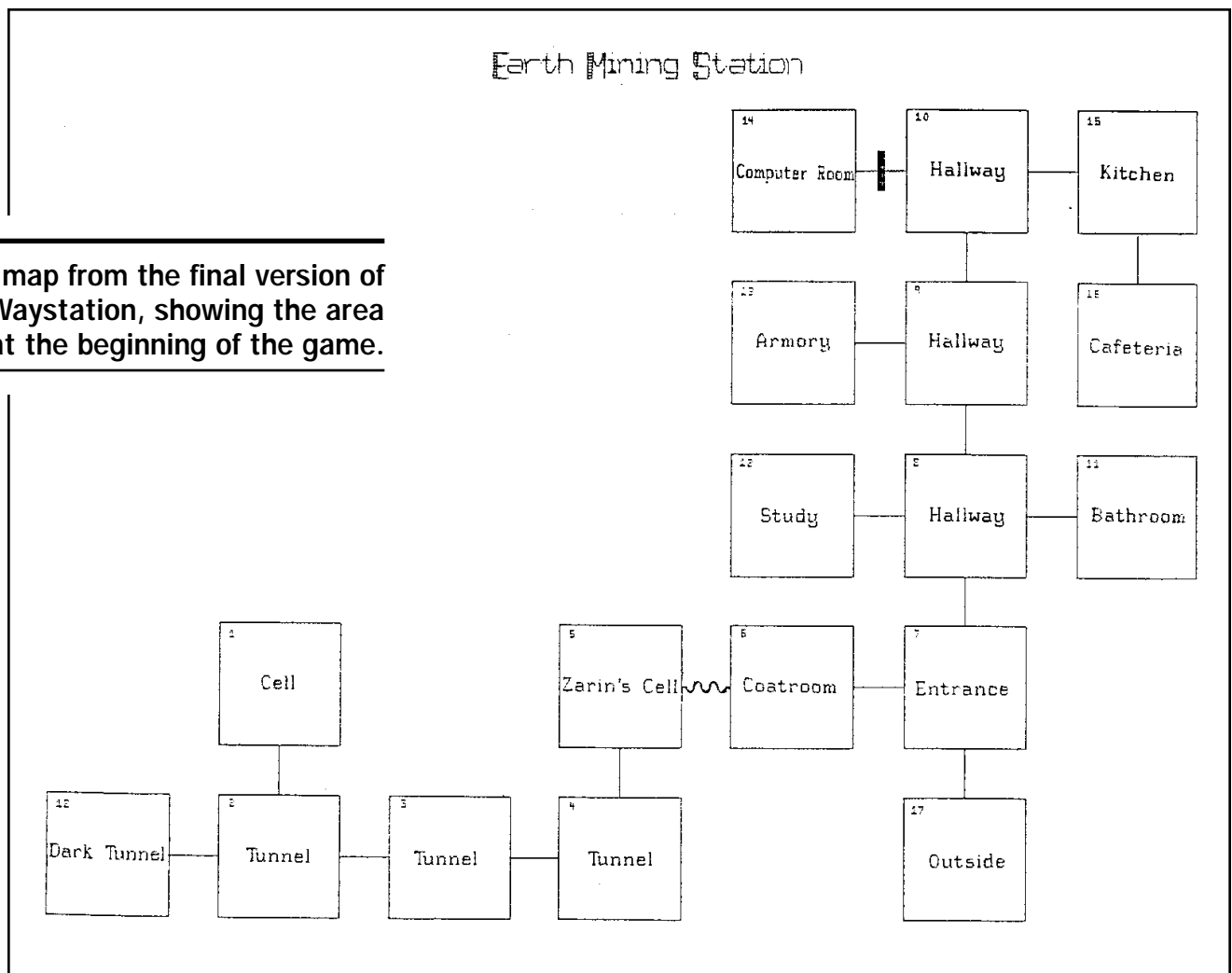
April 9, 1993. Waystation is reborn. Again. The first rooms went slowly: I was learning the language as I went while I waited for my manual to arrive. The kitchen and cafeteria took a week to code. My coding experience is best typified by what I went through in creating the kitchen and cafeteria.

- Write the room descriptions for the two connected rooms of the kitchen. 1 hour.
- Add the conveyor belt, rails, and other decorations to both of the kitchen rooms. 2 hours.
- Create the turnstile in the kitchen. Realize that a new class is required, one which will print out "Beyond the xxx you see..." 30 minutes.
- Go back and change all of the "==" to "="; TADS 2.1 doesn't support full C syntax yet. Repeat after every step. 2 minutes each repetition.

- Debug the “beyonder” class. Discover that the showcontcont() function (which displays the full contents of an object) must also be rewritten to handle the “beyonder” class. 1.5 hours.
- The second time the kitchen is visited, it is closed. Write all of the code to handle that case. 30 minutes.
- Move any item left in the kitchen so that it is not seen upon return visits. 20 minutes.

It took 7 hours of coding spread out over three days before I had a working kitchen. After that, though, I knew that TADS could handle any puzzle I could dream up. I finished the first half of Waystation by May of 1993. A summer research job squelched my momentum, and Waystation languished until the spring of 1994. At that point I *had* to finish the game. In a burst of frenetic energy, I wrapped up programming in two months and rounded up ten betatesters.

A map from the final version of Waystation, showing the area at the beginning of the game.



---

## Squashing the Bugs

---

August 1994. The betatesters began uncovering bugs two minutes after I mailed them their copies of Waystation. Within two days I had over twenty pages of bugs, typos, and suggestions, half of it due to Michael Kinyon alone. Debugging is a long, frustrating process. In Waystation's case, it took from late August until December before I had it in a form that I felt was ready for release. In early January, I uploaded Waystation to GMD.

---

## What I Learned, or How Not To Write IF

---

Five years of off-and-on work taught me a lot about the process of writing interactive fiction. Hopefully others will find the following advice helpful:

- Design everything you can on paper before you ever begin programming. It's much easier to toss out unworkable or silly sections if they're only written down. Once you've programmed a section of your game, you'll want to keep it no matter what.
- Choose the programming language which best suits your needs. TADS dovetailed nicely with my C programming experience; for others, Inform, AGT, Hugo, or any of a dozen other systems may be best.
- Work EACH day, even if you only spend five to ten minutes some days. Like water wearing away a stone, you can make the colossal task of writing IF manageable if you keep your momentum going. My constant starting and stopping delayed Waystation by over a year, just considering when I started working with TADS.

Thanks for playing Write Your Own Interactive Fiction!  
Elapsed game time: 5 years, one month. ☒

## Footnotes

[1] Dacosta, Frank. "Writing Basic Adventure Programs for the TRS-80." Tab Books: Blue Ridge Summit, PA, 1982.

[2] Hartnell, Tim. "Creating Adventure Games on Your Computer." Ballantine Books: New York, 1984.

# OH NO! BETA!

## Tips and Techniques for Beta-Testing Games-in-Progress

---

by C.E. Forman

---

**A**ny author who has written a work of IF of medium size or larger should be wholly familiar with the process of playtesting the game, discovering bugs and parser problems, making certain that the game is as clean as is reasonably possible before its official release to the IF archives. The key issue, though, is whether the game's testers themselves possess the same degree of familiarity.

The beta-tester's goal is not to play a game solely for leisure and escapism, but to willfully risk his or her own spoilage of the fantasy, with the intent of using it to prevent such disastrous results from reaching a broader audience. This is the price to be had in exchange for becoming a part of another author's world and exerting a certain degree of one's own influence upon the final outcome.

Testing, in its own way, is as much an art as the creating of the game world itself. To playtest effectively, one must come to understand the link between tester and game, and further, between game and author, in order to draw across the third and final gap between author and tester. Bridging the gap in the most mutually constructive manner is the goal of the IF beta-tester.

---

### Offering Your Service

---

Most game authors recruit the assistance of playtesters via a Usenet post, or by personal e-mail inquiries to testers whose names they are familiar with (possibly through recommendations from other authors). It is important at the outset to establish the amount of time you, as a tester, will be able to spend on the project. If you don't think you'll have much time (or any at all), either warn the author ahead of time so s/he can make arrangements with other testers, or don't sign up at all. If, once you've begun, it turns out that you have far less time than you'd anticipated, let the author know as soon as possible so that replacement testers can be found. Under no circumstances should you leave the author hanging as to whether or not you're still with him/her.

---

## Structure of the Playtester's Report

---

My typical beta-testing report is a standard plain-text file, divided into four sections, with a header at the very top identifying the game, the report number (numbering reports makes it easier for both authors and testers to identify and organize later), and the beta version the bugs came from. You may want to include your name here as well.

These four sections serve as a good method for breakdown and classification of bugs:

1. Programming bugs. These deal with significant errors found in a game, be they logical errors or outright crashes.
2. Parser problems. These consist of minor annoyances you've had when trying to select verbs, nouns, phrasing, et cetera.
3. Cosmetic bugs. Typographical errors, extra line-feeds and spaces, and the like. These are not true errors, but are important to the overall appearance of the game.
4. General feedback.

The first three sections should consist of appropriate segments of a game transcript (beta-testers should *\_always\_* keep transcripts) cut-and-pasted in, with the unnecessary steps edited out to reduce clutter. If the bug in question can be explained in only a few words, a transcript segment is not necessary. When using my own words rather than the game's, I prefer to surround comments with square [] brackets to make it easier for the author to identify where the text is coming from. In any case, include a line of dashes or asterisks to separate one bug from the next, to keep the report easy to read. Also, when dealing with typos, it's helpful to use a carat (^) symbol below the offending word to point out just where the error is.

Structuring reports in this format makes the author's job far easier, as it enables him/her to select the type of bugs to correct, based on the author's current mood. This stage of game development can be frustrating at times, and on some days an author simply may not feel up to tackling more complex bugs. It's more reassuring for the author to have an organized group of typos and parser quirks to work at during these times, rather than having to read through a jumbled report and pick out appropriate bugs on his/her own.

The fourth section, general feedback, should serve as a means of input from tester to author, and is used to cover anything that doesn't fit into the first three categories. This is the place for a tester's own personal evaluation, covering any plot holes, comments about the puzzles and overall layout, and general praise and (constructive) criticism. It is of the utmost importance that a tester provide a good deal of positive feedback here as well. Playtesting is the absolute last thing an author wants to do after spending months on writing and coding. By the time a game reaches the testing stage, the author's morale can be at its all-time low, and

it's likely to stay there if the only feedback received about the game is a long list of everything that's wrong with it.

Following these guidelines, here's a fictional sample report using some early bugs found in my own game, "The Path to Fortune" (but now fixed):

"The Path to Fortune" Playtesting Report #1

=====
1 - Programming Bugs =====
=====

[I can unlock the abandoned shed with any object in the game.]

>cash
You're broke.

>w

Borthur's Smithy

>cash
You have 12 gold Renkin, the standard unit of currency in the

[I haven't asked for the gold yet.]

=====
2 - Parser Problems =====
=====

>make a roll
You'll have to specify what you want to make.

[I've tried every verb I can think of, and can't get this to work. Are you sure the code is reachable?]

=====
3 - Cosmetic Bugs =====
=====

>x gold
Something about the unshaped nugget of gold unnerves you. When
you hold it up to the light, the nugget shines very brightly,

^^^^^

[Should be "shines. "]

=====  
 4 - General Feedback =====  
 =====

[...and so forth.]

Make certain that your messages are clear enough for the author to determine the problem, and don't send blatantly derogatory messages of any sort.

## Followups to Reports

It's quite likely that an author will elect to give feedback on your feedback, whether to explain his/her reasoning behind a choice of implementation for an aspect of the game, to request further details regarding a bug, or to assure you that a particularly nasty problem has indeed been conquered. In instances where an author needs to clarify a problem, it's important to leave enough of the previous message to allow the author to see what the original issue was. Don't edit the old parts out just to save space! (This is one of the few times on the net where you *want* repetition for reinforcement.)

Also on the subject of feedback, opinions differ, and authors may not agree with everything you say. They may have very good reasons for not wanting to implement some of your suggestions—some parser quirks may be more trouble to deal with than they're worth (particularly in the case of first-time authors); some solutions may contradict other parts of the game (which you may not have seen yet). Ask if you're not sure, but don't pressure authors too much—if they refuse, and offer at least a fairly reasonable explanation, let it go at that.

## Discovering Bugs

This is the heart of the playtester's job, and the most difficult aspect to learn to do effectively. It's also the most difficult to put into words, so here is a set of common methods I use to break a game:

- **Scanning text.** Personally, I can't stand reading my own text. As a result I don't look very closely for typos and extra spaces, instead relying on playtesters to hunt them down. Read all game text carefully.
- **Scenery.** A player should be able to examine almost everything the game mentions in the text.
- **"Reversing the solution."** It's quite common for an author to unintentionally implement a one-way puzzle. That is, one in which the solution could logically be reversed or recreated, but no code for it exists. An example of this would be breaking one of the mirrors in "Enchanter," then fixing it with the kreb spell (which works, by the way).

- **“Do It twice.”** AGT games in particular award points for players repeating the same action more than once. Try solving puzzles twice to be sure.
- **“Parser probing.”** IF language tools sometimes use two verbs that mean the same thing, yet behave differently (“move” vs. “push” is a good example). Try performing actions using every possible method of phrasing you can think of.
- **Containers and surfaces.** Objects that can hold other objects can cause a lot of trouble (as evidenced by the famous (infamous?) axe-in-the-bucket bug in Unnkulian Underworld). Make sure that players can’t sneak objects into places they shouldn’t be able to through the use of a container or surface, and make sure that all such objects only hold the size and number of items they would in a “real world” setting. (Large containers used for inventory management, such as the rucksack in “Curses,” are exempt from this last rule.)
- **Animate characters.** Even simple NPCs tend to have potential for bugs, so try out all the major NPC interactions (“kiss,” “kill,” “show,” “give,” “tell,” “throw,” “ask about” and “ask for”). Make certain that you give each character an order or two as well.
- **Flammables.** Adding a torch or matchbook to an in-depth game can geometrically increase complexity. A default message is commonly put to use for most items, but many times special responses will be needed. Pay particular attention to effects such as lighting a match with a torch, vs. a torch with a match, and lighting one torch with a second.
- **Liquids.** IF liquids should be drinkable (or have a special message if you try), and able to be spilled out onto the ground, poured into other containers (assuming they’ll hold liquids), and so forth. Check to confirm that a liquid can not be carried in a player’s hands.
- **Multiple-step puzzles.** Recent games no longer resort to the “one object = one puzzle” format of early IF. It is often necessary to solve problems via a step-by-step process. As a tester, make certain that the objects’ states change when one part of the puzzle is solved, to avoid nonsensical repetition. An example of this type of bug might be a vending machine that drops a bag of chips into a bin after a player inserts a coin and pushes a button. Check to make sure that, if the player pushes the button again, the text for the chips does not appear a second time.
- **Daemons.** Puzzles involving timing are the most complex to implement. Test for redundant actions—doing the same thing twice to start the daemon twice. Also make sure that the daemon in question only prints text when its object is in scope to the player (i.e., a player shouldn’t be able to see or hear a car running after s/he’s left the room it is in).



As a final note, keep your old reports and re-check all the bugs later, after the author has sent an update. Make sure the bug fixes work without causing further bugs, and that they don't disrupt the overall logic of the game in any serious way.

---

## **Playtester Ethics**

---

Finding bugs is not the only concern of a beta-tester. Believe it or not, the job can also raise concerns about proper ethical behavior. Game authors put a lot of trust in their testers, and it's important that the testers not betray that trust. Keeping that in mind, here are some simple (if you think about them) guidelines that I adhere to, and that I advise other testers to follow as well:

- First, and most obviously, never distribute a game you're playtesting. If the author wanted this, s/he'd have released it already, without your help.
- Keep your playtesting experiences confidential. It's okay to confer with other people beta-testing the same game, but once the game has been released, don't tell players about bugs you found in the beta versions. Authors don't want to appear sloppy or stupid to the gaming public, and indiscriminate discussion of bugs that have already been fixed can make them feel that way. (Again, let the author initiate this sort of talk if s/he wants to.)
- Don't write zine reviews of games you've beta-tested. Like the authors themselves, you've probably been working too close to the project to give an effective, unbiased evaluation.
- Don't give out hints over Usenet to games you've tested...at least, not for awhile. Out of necessity, you've already seen the solutions, but other players have not. Give them time to figure things out on their own and to help each other out first, rather than spoiling the game and the author's hard work right at the outset. (By the same token, wait a good time before writing a walk-through or hints for a game. And even then, make sure you have the author's consent.)

---

## **Conclusion**

---

Playtesting can be a very painful process for an author...but it doesn't have to be. With a little extra effort on the part of the playtester, it can be an enjoyable and rewarding experience for both parties. ☹

# Beta-Testers I Have Loved

---

by Neil deMause (neild@echonyc.com)

---

“I found a bug in your game,” goes the kind of e-mail I’ve been getting lately. “I get a TADS error message whenever I try to take more than 201 napkins from the napkin dispenser.”

Messages like these pour into my mailbox with alarming frequency. But I don’t mind—I don’t tell their authors to quit nitpicking and get a life. Because these are my beta-testers. I count on them to be insane.

Yes, insane. C.E. Forman has laid out quite a nice list of guidelines for play-testing in his article (see page 12, “Oh No! Beta!”), but he fails to mention the most important requisite for a successful beta-tester: you must be absolutely, stark raving mad.

Look at it this way: Say you’re an interactive-fiction author. (I say it all the time, it shouldn’t be too difficult for you.) As soon as your game is ready for debugging, you painstakingly walk it through all its paces, taking every logical step and ensuring that it works properly. Once you’re done, you release the game to your battalion of beta-testers, secure in the knowledge that you’ve accounted for all but a few sensible actions.

What a fool you are! (This, too, I say to myself all the time.) You may have accounted for every sensible action, sure, but those who play your game are likely to be anything but sensible—they may try odd things because they’re stuck, because they’re contrary, or just because they’re plain lame-brained, but there’s no way they’re going to behave as expected. Which is why the best beta-testers are those who can routinely do the unexpected. Which is where the insanity comes in.

Take, for example, one of my favorite beta-testers. He has an odd compulsion, when he plays IF games, to close doors behind him. It’s a bizarre fastidiousness, not even remotely useful for an IF player, but I love him for it, because he has uncovered bugs in this way that I never would have found by my own, door-opening self.

And then there’s my napkin-thieving friend (whose initials, incidentally, are CEF). It’s a wonder he can ever finish a game, he’s so busy scouting out blind alleys and trying to interact with the scenery. Without his help, I never would have known how my game responded if you tried to push on an NPC, or talk to a brick.

But that’s what good beta-testers do: they command NPCs to walk through walls, flip coins that they aren’t holding, pour pepper on goats. Give them a fork in the road, and they will likely take neither path, but attempt to EAT SPAGHETTI WITH FORK.

I’ve tried to learn from my beta-testers. When I play-test my own games now, I flip, prod and manipulate every existing object (and even some that aren’t there) in an attempt to find illogical behavior—which is never a good thing in an IF game, even in response to an illogical action. There may be no good reason for a player to type GIVE PIANO TO ORANGUTAN, but if the game replies “The orangutan tosses the piano about in its hands for a bit, then gives it back to you,” it tends to break the spell of realism.

But I know I’m doomed to failure in my attempts. I’m simply too close to my game to see all the possible actions, and I’m naturally more inclined to look for things that *do* work than those that don’t. And so I happily put up with bizarre midnight messages about napkins and one-way doors, give my beta-testers nice big thanks in the game credits—and all the while hope and pray that nothing ever happens to make them accidentally go sane. ☒

# The Light: Shelby's Addendum

release 1.1

Parser: TADS

Author: C.A. McCarthy (mlkuehl@students.wisc.edu)

Availability: ftp.gmd.de/if-archive/games/

Requires: TADS run-time interpreter; standalone versions are also available.

Response to the XYZZY command: "A hollow voice says, 'Wow! You must be, like, really old.'"

You're an apprentice to the keepers of the Lighthouse, who have a very important mission—protecting a phase modulator, the only preventative measure against a world-destroying dimensional shift—and who are conducting research for a compelling project called EUNICE. As you return to the lighthouse after a short journey, you discover that something terrible has happened: charred bodies are strewn over the road to the village, the sky seems to be on fire, and a creeping mist is slowly but surely completely enveloping the land.

What's going on here? Reading your employer Holcroft's diary will offer much background material: while working on the EUNICE project, your other supervisor, Barclay, appears to have made a discovery that caused him to lose his senses and take a series of actions that now jeopardize the planet's safety. Barclay has stolen the phase modulator—the Lighthouse's light—and gone to a distant retreat. You must learn as much as you can about the situation, prepare a submersible for your travels, then track down Barclay and stop him.

Players must solve a timed, potentially fatal puzzle early in the game, which consists of finding and wearing an anti-phase bracelet within the first 100 turns of play, or your body will turn transparent and you'll die. This was an interesting variation on the timed-starvation puzzle, but I wish the ill effects of the dimensional shift kicked in much later—say after 200 or 300 turns instead. Before I stumbled across that puzzle's solution quite by accident, I was frustrated by not being able to explore very much of the game's surroundings before that 100-turn limit kicked in.

I really like puzzles in adventure games that include what I think of as many interlocking parts—for example, in such a puzzle you'd need an object from one area of the game to access

another whole section, but you need yet another object first to make the first one functional...I mention this because I think The Light: Shelby's Addendum has a good variety of this kind of interlocking, scavenger hunt puzzles. Your mileage may vary, though, if you don't particularly enjoy traipsing back and forth throughout a game! :)

Although many of the parsing problems that were in the original version of this game have been fixed in version 1.1, I still noticed several bugs that significantly affected play. For example, I'm quite sure that the way I managed to obtain the anti-phase bracelet was a lucky break from a coding error; even after playing Shelby a good deal I'm not sure what the correct steps for the solution are. I was also confused about the logic of a couple of puzzles even after I solved them—for example, exiting from the secret chamber with the concentric circles.

The room and object descriptions are very well detailed, which contributes a lot to the atmosphere. Very often using the EXAMINE ALL or SEARCH ALL commands is key to finding a missing crucial object or discover how to solve a certain puzzle. There are also many nice touches that add an air of sinisterness, such as the blood-covered mophead or a lingering bad smell.

During the early part of the game, there are not too many interactions with NPCs, except for a scared workman and a chicken. I very much liked the character development of Holcroft and Barclay, especially as you discover the extent of Barclay's actions. The unfolding of this malevolent behavior reminded of Jack Nicholson's character in "The Shining"!

The game contains no online help, but hints are available from the author for registered users (US\$10). This is certainly the author's prerogative; I think I've just been spoiled by the number of other shareware and freeware games that do include some form of mild spoilers. I try not to resort to hints of course, but would have liked to have a peek while working out my own parser problems. Without online help—at least for the beginning timed puzzle—I think fledgling players are likely to get frustrated and abandon play fairly quickly. To the game's credit, I found it wholly absorbing once I was able to overcome the initial obstacles and get drawn into the drama of the adventure. ☒

—Stuart Beach

The companion disk for XYZZYnews #8 contains the following game files. It's a good deal for people who have slower modems—at 2400 bps, it'd take a heck of a long time to download the contents of the companion disk. It's also a good deal for people with limited or no access to FTP sites or online services as a source for new games. If you're reading an electronic version of this issue, you can obtain this games disk with a print copy of XYZZYnews #8 by enclosing \$3.50 for postage and handling with the coupon on the bottom of this page. If you play and enjoy these games, please pay the shareware fees as applicable.

**THE BROKEN STRING**—Being part of the punk scene and pursuing anarchy was the best time of your life. Can you start your own band and help bring back the good old days? This TADS game is e-mailware; that is, you should send the authors an e-mail if you enjoy playing it.

**LOST NEW YORK**—In this shareware TADS game, you're a visitor to the city with some time on your hands. Your trip to the Statue of Liberty begins ordinarily enough, but when you find yourself thrust in the past, you're forced to discover for yourself how to interact with a diverse cast of characters and find your way home.

**MAGIC TOYSHOP**—In search of a birthday present for your young niece, you decide to stop in at a particularly intriguing-looking toyshop to see what you can find. There you'll meet Catharine, the proprietor's daughter, who presents you with a number of unusual gadgets and brainteaser puzzles for you to solve. See overview, XYZZYnews #5.

**LETHE FLOW PHOENIX**—Out on a camping trip and awakened by strange noises at night, you venture into the night and accidentally pitch yourself over a cliff. When you awaken, you find yourself in a grassy field, left to discover both how to maneuver about this completely unfamiliar environment and your purpose here. See review, XYZZYnews #7.

## XYZZYnews Magazine/Disk Order Form

Checks and money orders preferred. Please send coupon with payment to:

Eileen Mullin  
XYZZYnews  
160 W. 24th Street, Ste. 7C,  
New York, NY 10011.

- Please send me a copy of the print version and companion games disk for XYZZYnews Issue #8. I have enclosed \$3.50 for postage & handling. (Check one: Mac disk \_\_\_ or PC disk \_\_\_)
- I need just the companion games disk for XYZZYnews. I have enclosed \$2.50 for postage & handling. (Check one: Mac disk \_\_\_ or PC disk \_\_\_)
- Sign me up for a 6-issue subscription (within the U.S.). I have enclosed US\$21 for postage and handling. (Check one: Mac disk \_\_\_ or PC disk \_\_\_)
- Please send me a copy of the upcoming XYZZYnews Issue #9. I have enclosed \$3.50 for postage & handling. (Check one: Mac disk \_\_\_ or PC disk \_\_\_)

Full Name (please print): \_\_\_\_\_

Street Address: \_\_\_\_\_

City: \_\_\_\_\_ State: \_\_\_\_\_ Zip/Postal Code: \_\_\_\_\_

Country: \_\_\_\_\_ Email Address: \_\_\_\_\_