



Letter from the Editor

While it has been a fairly slow month in terms of news specific to AIF, the IF world as a whole has been taken by storm. The release of Inform 7 and subsequent discussion have pretty much dominated the IF world. Please see A. Bomire's excellent review in this edition of the newsletter for a detailed discussion of Inform 7, what it is and what it can do. The question of how this new paradigm for IF creation will change the medium has been subject to a lot of lively debate. Even the AIF community is beginning to take note; Shadow Wolf, a long-time AIF community member, has taken a keen interest in the new system and has begun creating AIF libraries for it.

I spent a little time tinkering with Inform 7 when it first appeared, though I haven't been able to give it a significant or thorough evaluation. In fact, I haven't done more than create a few objects based on the first chapters of the tutorial. Still, I understand the excitement it has created. To be able to write simple sentences, like *My bedroom is a room. In my room is a container called my dresser. The description of the dresser is "It's the same dresser you've had since you were a kid."* and have the system understand this and build this into your IF world is very cool. I also think the development environment

Continued on page 3

This Month in AIF by BBBen

Without any real controversy or major events taking place this month, my news column may be a little light. However, there was plenty of discussion and a new game release, so it was a good month.

There's been a fair bit of interest around about the new Inform 7 system. Apparently (and I confess to not having actually researched this myself) it is a very accessible language even for those authors that prefer to use ADRIFT and avoid languages like TADS. That said, I do not believe that it is as straightforward as ADRIFT, but if authors out there are

Continued on page 3

Interview with Johnny Freebase by A. Ninny

Johnny Freebase is a long-time AIF community member and author. His one game *Stowaway* is overshadowed by his excellent AIF 101, a tutorial for AIF authorship, which will be re-published here beginning this month. We're grateful that he was willing to sit down and answer some questions about himself.

AN: Please tell us about yourself, and tell us what brings you to AIF.

Continued on page 2

Contents

<u>Page 1</u>
Letter from the Editor
This Month in AIF
Johnny Freebase Interview
<u>Page 4</u>
Inform 7 Review
<u>Page 8</u>
Seven Seas of Thëah
<u>Page 9</u>
AIF 101: Brainstorm
<u>Page 10</u>
Review of <i>Paradise Hotel</i>
<u>Page 11</u>
Review of <i>Gifts of Phallius I: The Baron's Plot</i>
<u>Page 13</u>
O Erin!
Info about <i>Inside Erin</i>

Mission Statement

Inside Erin is written and published by people who enjoy AIF. It is done for fun, but we also have some goals that we seek to achieve through the newsletter:

1. To encourage the production of more quality AIF games by providing advice from game developers, and by offering constructive criticism that is specifically relevant to AIF.
2. To encourage activity and growth in the AIF community. We aim to generate a constant level of activity so that there aren't long periods in which people can lose interest in AIF.
3. To help document and organize the AIF community. This is done through reporting on games and events, as well as by helping to organize community-wide activities such as competitions and the yearly Erin Awards.

JF: Well, I'm a programmer/artist/writer. Both IF and AIF are hobbies of mine. I began playing AIF shortly after *Moist* was released.

AN: Your "AIF 101" is an incredibly detailed instruction booklet on AIF creation. What led you to write it?

JF: Honestly, it was the dearth of quality work coming out at the time. At that point in AIF's history, we were getting a large number of poorly written AIF. It seemed most authors were only interested in publishing "vanity games"... creating games to be creating games, without thought to their playability. Of course, with every game getting such a great reception... even Vachon's unplayable incest fests... there was no real reason for most authors to dig any deeper. If I recall correctly, at the time games were being released without any sort of beta-testing.

AN: What results were you hoping to achieve by publishing it?

JF: I was hoping to give a little help to those who wanted to craft quality work, but really didn't have the skill or know-how to do so. I wanted to see more quality AIF. I thought that the genre of AIF had a lot more potential than most of what I was seeing, and hoped for a bit more "professionalism" - as much as you can get in a hobby. Beta-testing, consistent methodology, longer games...

AN: Do you think it was successful in achieving these results?

JF: I honestly don't know. The games that are coming out now are... in general... the kinds of things I wanted to promote, but I cannot say I had anything to do with that. AIF has matured, and continues to mature. Recently, an author posted a work that was untested, full of errors, and just wasn't ready. He pulled it almost immediately, got some beta-testers, and re-released a game that was buggy, full of errors, and just wasn't very good. But he tried. And I think, that if the author has an intent to become better, he can fulfill that desire. The standards of the AIF community have gotten higher, for whatever reason.

AN: Despite being able to sustain the attention necessary to write "AIF 101", you've only actually released one short AIF. Why is that?

JF: Well, it's more or less the fact that I have a lot of projects up in the air at once. I work on AIF, regular AF, various other games in C++ and Java, try to get a writing career kickstarted, and maintain a real life, all using different pseudonyms. Except in real life. For the most part, I use my real name in real life.

I have eight or nine unfinished AIF projects on my hard drive at any one time. My first project... I began it in 1998, and like most first projects it was much too ambitious. It had a huge map, 14 or 15 fuckable NPCs, many story driven events... I haven't worked on it in the last few years, but I'll eventually go back to it.

Eventually, I'll get back to the projects I'm working on. Many of them are very close to completion, but I cannot say for certain, and I don't want to announce anything until it's ready to be released.

AN: You ran the ill-fated AIF Challenge. Please recap for those who may not remember what the challenge was.

JF: The AIF challenge was a Yahoo! group I started up with the intention of running monthly contests, where authors could build games around different themes.

AN: What were you hoping would come of the Challenge?

JF: At the time we were getting one game every few months or so, and some of those games weren't any good. The challenge itself was an opportunity, or so I thought, for authors without ideas to get a gem, a tiny spark of an idea, that might inspire them to create a game.

AN: Even though nobody ever entered a game, do you think it accomplished anything useful? What sort of feedback did you receive from the community?

JF: I didn't get feedback other than "it seems like a good idea but I'm not entering" for whatever reason.

AN: What are your favorite AIF games – and what do you look for in the AIF that you particularly like?

JF: I've enjoyed Chris Cole's work, One-Eyed Jack's *Sam Shooter* series, and NewKid's games. I play almost everything that comes out at least a little bit, but if it doesn't grab my attention, or is littered with errors, or leaves me wondering what I'm supposed to be doing, I won't continue it. While actually playing, I'm engaged by steamy sex scenes, a tightly woven plot, and enjoyable characters to interact with. What I'd like to emphasize most, however is elegance in design. Every element of the game should serve to advance the story, provide for character development, or reinforce the game's themes. This includes sex. Do whatever you want, just do it well.

AN: What projects are you working on? Anything we can look forward to seeing any time soon?

JF: I've mentioned I don't like to announce the games I'm working on, because I don't like creating hype when I might not finish the game, but here are a few close-to-done games I've got in the works:

Summer's Last - Teen romance style game similar to a John Hughes film.

Snowstorm - Simple game set in a cabin used to test my custom AIF libraries.

Stowaway - the unfinished TADS3 version that I based my sole AIF release in Adrift 4.0 on.

Huh, lots of S's. Well, if you see any of these games, then I've actually bothered to finish something.

Johnny Freebase is the author of *Stowaway* and the "AIF 101" tutorial.

Letter from the Editor (Continued)

the Inform 7 creators have included is fantastic and would love to have something similar to work with in TADS.

Personally, I think that for the time being I will stick with TADS. I'm comfortable with the object-based system and I like the way it can be used to organize extremely complicated game interactions.

What I'd like to hear more of (and maybe I'm just not looking in the right places) is what ADRIFT writers think of Inform 7. It's very different from ADRIFT, certainly, but because it doesn't contain the gobbley-gook appearance of TADS code, I think it might be an attractive alternative for them. What is unclear is whether using Inform 7 to write games is actually easier than using an object-oriented language and whether it's also faster than ADRIFT.

This Month in AIF (Continued)

skeptical about ADRIFT and intimidated by TADS, it might be worth while checking out the discussion on the AIFGames.com forum.

There's also been some technical trouble with some of the games downloads at AIFGames.com. It seems that some (but not most) people are getting incomplete files when they download them from the site. This has been discussed quite a bit and still hasn't been resolved, so in the meantime if you're having this problem then you'll have to try getting the games from the Yahoo! Groups or Matrix Mole's aifcommunity.org archive.

More generally, I've noticed a lot of potential games being discussed lately (hey, I solicited some of the discussion myself) and it got me thinking about the number of games that disappear into the internet mists. I think that about 90% of projects that get talked about as being in production don't get made. Is this because people are talking about them? I think that actually is part of it. You see, when one is developing a game there is a certain pressure to release it, just so that people know about it. However, this pressure can be relieved somewhat by talking about the project in detail, showing the uncompleted version around the place, or releasing a first part consisting of the work you've already done. Speaking from my experience with the *Crossworlds* games (CW4 is still in the works, I promise! I'm cutting back on some of the bits that bore me and trying to put in more of the fan favourites), getting what you've already done out there really does diminish the weight

on your shoulders, which unfortunately means there's much less likelihood of you finishing the project. Eventually it will have sat there long enough that you'll just give it up as lost. So my advice is to keep those projects largely to yourself until they're finished - be disciplined, and if you've got writer's block, try to get away from that project and maybe later rediscover what you liked about it.

Anyway, it seems to me that the mini-comp did a good deal for the health of the community, so we'll have to keep it up as an annual tradition. I may even take another shot in the next one.

New Games

Gifts of Pallius 1: The Baron's Plot by Goblin Boy, released for TADS 2, May 10. You are a spy for King Goldik, working undercover as a guard for Baron Jalafar, in a fantasy/mediaeval setting.

A Review of Inform 7 by A. Bomire

My Bedroom is a room. In My Bedroom is a dresser, a desk and a bed. My wallet is on the dresser. On the desk are my keys.

Wouldn't it be great if writing Interactive Fiction were just that easy?

In 1993, Graham Nelson began writing the Inform language for creating works of interactive fiction. By 1996, he had worked through 6 major versions to create Inform 6. And other than minor updates, that is where Inform has been for 10 years. That's not criticism; that is a statement of accomplishment.

Three years ago, Graham Nelson looked at something similar to the description of "My Bedroom" above and said: "I can read this sentence and immediately understand the items listed and their relationship to each other. Why can't an authoring system do the same?" And so with a lot of help from people like Emily Short, Andrew Plotkin, and others too numerous to mention, he developed Inform 7. What's the big deal about Inform 7?

Let's start with the fact that the sentence with which I opened this article is actual working Inform 7 source code.

Have you got your breath back? Okay. Yes, Inform 7 is a "natural language processor" – or as close to it as I've seen in quite some time. For most people, this would be accomplishment enough – but Graham Nelson (et al) also packed a bunch of new features into this release.

Inform 7 is rules based. What does this mean? Well, a rule is something that defines what happens during the game, usually specific to an action performed by the player (although not always). These rules can be set up in different places on a single action: "Before" the action takes place, "Instead" of performing the action, while you "Carry out" the action, "After" the action takes place, and when you "Report" on what happened during the action. For example:

Before inserting something into the box, try silently opening the box.

Instead of eating the poison apple, say "Are you crazy? That's a poison apple! POISON!"

Carry out attacking the nitroglycerine: say "Boom – you're dead!"; end the game in death.

After taking the diamond for the first time: say "At last – you have the Gribaldi Diamond!"; award 5 points.

Report dropping the crystal vase: say "(taking care not to break it)".

And yes, these are all working examples of Inform 7 source text. (The Inform authors like to refer to it as "source text" instead of "code" to emphasize the use of natural language. I will attempt to do the same, but forgive me if I slip now and again.) Inform authors will recognize "Before" and "After" from Inform 6. However, a new addition is that these rules don't necessarily have to be associated with the item (or items) they reference. Inform 7 allows you to put the rules wherever you think best. (Note: There are a few exceptions.) For example, you can collect all of your text that hands out points into a single section. You can even organize your rules into *rulebooks*.

Inform 7 allows you to define relationships between objects. Some are already defined, such as the physical relationship I alluded to earlier where my wallet is on the dresser. Or, you can define emotional relationships, such as "*Carmen likes the player*". These relationships can be referred to in multiple ways. For example, "*The coin is in the purse*" means the exact same thing as "*The purse contains the coin*". Once defined, you can use these relations in writing your rules and determining the outcome of the action.

Everything in Inform 7 is built from three basic types of object: thing, room or direction. Those of you familiar with Inform 6 will find many of your old favorites still in existence such as “scenery”, “container”, and “supporter” which are all subsets of the three basic objects. This specialization is called a “kind” in Inform 7. These specializations nest, one to an object. For example, “*a bed is a kind of supporter*” and “*a daybed is a kind of bed*”. But you couldn’t, for example, say “*A bed is a kind of supporter and a kind of container*”. This may at first confuse people who are used to using multiple inheritances. Overall, Inform 7 has less than twenty such kinds pre-defined. (In contrast, TADS 3 has over 400!) But, like a box of tinker toys, these bases give you the tools to build the objects you require: “*A chair is a kind of container. A chair is always enterable. A chair is usually scenery.*” You can then reference this in your source text: “*The throne is a chair in the Throne Room.*”

This arrangement provides a lot of flexibility in creating items, and also keeps the Inform 7 source code (oops, I mean “text”) lean and mean (just over 11,000 words). However, there are detractors who balk at re-inventing such basic items as chairs and such over and over. Fortunately, there is the possibility of including “Extensions” just as in Inform 6, which allows other Inform 7 authors to create source text you can use within your own game.

(Note: As of this writing, Shadow Wolf is hard at work on an AIF extension to handle layered clothing, body parts and sexual interaction. Once completed, this will be of tremendous help to AIF authors. See the links at the end of this review.)

Another new feature of Inform 7 is the ability to group things together, such as the rulebook mentioned earlier. Other examples include grouping rooms together into a “region”, or grouping actions together into a “behavior”. Using these groups, you can make blanket rules to affect multiple parts of your game. Look at the following example:

Screwing someone is being sexual. Kissing someone is being sexual. Instead of being sexual in the Public Area, say “It’s too public for that!”

In this example, I’ve grouped some actions into a single behavior (*being sexual*), then defined a prohibition against performing that behavior within a collection of rooms (the *Public Area* region).

Another way of grouping things together is to create “Scenes”. A close equivalent would be “fuses” in TADS, or “scheduled events” in ADRIFT, and “daemons” in both TADS and Inform 6. However, they are a much more powerful implementation in Inform 7. You can specify when the scene begins, when it ends, and what happens during each phase of the scene (begins, ends, or while it is happening). You can also check to see if a scene is happening, or if it has happened. Even more powerfully, you can specify multiple endings, with different actions taking place in each ending - even starting a new and different scene for each ending! Here’s an example of an opening scene in a game, where the player is supposed to hang about his house waiting for a phone call from his girlfriend. His girlfriend finally calls after 6 turns.

Waiting for a Phone Call is a scene. Waiting for a Phone Call begins when play begins. Waiting for a Phone Call ends when the time since Waiting for a Phone Call began is 6 minutes.

Instead of going through the front door during Waiting for a Phone Call, say “You know as soon as you leave the phone will ring.”

Every turn during Waiting for a Phone Call: say “You sigh. ‘C’mon – ring!’ you say, looking at the phone.”

When Waiting for a Phone Call ends: say “As you putter about the house, the phone finally rings. ‘Finally!’ you mutter to yourself as you pick it up to hear your girlfriend on the other end. The two of you talk briefly, and you are surprised to hear yourself agreeing to meet her at the mall to go shopping! Almost before you know it you are in your car and driving to the mall, where you park and go inside.”; move the player to The Mall.

You’ll note that the time when “Waiting for a Phone Call” ends is specified in “minutes”, not turns. This is because Inform 7 automatically tracks both turns and minutes (and the two are equivalent – 1 minute equals 1 turn). Thus, if you desire to track time within your game, Inform 7 makes this very easy. You can even refer to the time in your source text, such as “*At 4:00 PM.*”, “*18 minutes from now*”, or “*wait until 2:15 AM*” and Inform 7 will know what you mean. By the way, the game begins at 9:00 AM by default, but this is easily set to any time you wish. However, I couldn’t find any way of changing the time rate. It’s 1 turn per minute – no more, no less.

Inform 7 has many “phrases” that it uses to describe or carry out actions that take place within the game. Some of them you’ve seen already, such as “*the time since Waiting for a Phone Call began*” or “*taking the diamond for the first time*”. It even provides the capability for you to create your own phrases and have Inform 7 carry them out. To programmers like me, this is analogous to writing a subroutine:

After giving the boarding pass to the flight attendant, fly to the coast.

To fly to the coast:

say “Giving your pass to the attendant, you board the plane and stow your luggage in the overhead compartment. After finding your seat, you attempt to get comfortable, grimacing as you attempt to ignore the screaming baby three rows back. Surprisingly, you somehow fall asleep and are awakened by the attendant as your flight begins to land. Yawning and stretching, you retrieve your luggage after landing and exit the plane. Looking at the clock in the terminal, you remember to set your watch to the local time.”;

move the player to The Terminal;

change the time of day to 3:00 PM.

As you can see, the creators of Inform 7 have gone to tremendous trouble to create a system of writing IF that is analogous to writing fiction. It isn’t perfect as there is a syntax that you have to follow but that syntax is incredibly flexible. This isn’t really all that different from written language. After all, I could hardly write “stuff found in I box some the” and expect you to know I meant “I found some stuff in the box”.

There are those who complain about the wordiness of any natural language system. There is some truth to that. Saying “*The description of the wallet is ...*” is definitely more wordy than “*ldesc = ...*” Yet I look at a short snippet of text like “*now all of the doors are open*” and think about what that would involve when writing it in TADS. So it all balances.

Some of the syntax makes little or no pretence to be natural language whatsoever, such as defining additional terms for objects. This is done using the “Understand” syntax: *Understand “leather”, “brown” and “beat up” as the wallet.* There are a few other cases (manipulating tables, for example), but other than that it is pretty close to natural language.

How was Graham Nelson (et al) able to do this so quickly? (If 3 years can be called “quick”.) Well, mainly because compiling Inform 7 doesn’t actually compile Inform 7. The new interface converts the source text into Inform 6 code, and compiles that using the proven Inform 6 compiler. This means that compiled games can be played on existing Inform interpreters and you can include Inform 6 code within your game if there is a feature that hasn’t been ported over to Inform 7.

While we’re talking about the interface, I should point out that the new Inform 7 language is only half the story. If you are an IF author, you will be very pleased with the new interface. The interface is split down the center with two facing panels, like a book, each adjustable in size to take up as much of the screen as you find comfortable. Any part of the interface (I will go over those parts next) can be accessed in either panel. So, if you prefer to view the documentation on the right side and the source text on the left, or vice-versa, you can do so with ease.

One thing to note is that there are two versions of the interface: Windows and Mac OS X. (Don’t worry Linux/Unix users. Someone is hard at work perfecting the interface for you. Right now, the only version has a Linux IDE using the Windows compiler, requiring WINE.) The two versions are maintained by two different people, and look slightly different. More disconcerting is that features available in the Mac version (the older of the two) sometimes work differently or are even unavailable in the Windows version. Getting these two synchronized is a must in my book.

The documentation is included within the interface, completely searchable and including examples. There are even icons to click on to copy the example text directly into your source text. One feature is a “Recipe Book” that organizes the examples for you. So, if you are working with people, look under PEOPLE for samples dealing with “Yourself”, “Conversation”, “Traveling” and “Scripted Characters” – to name a few. The documentation is quite well written, but it didn’t feel “complete” to me. Some things seemed undocumented, while others didn’t work quite as the documentation suggested. However, keep in mind that I was using the public beta; i.e., nothing was “complete” yet. Independent of the Inform 7 authors, fans have already created a quick reference and index to the documentation. Another requested feature is the ability to search the numerous examples. Hopefully, these can all be included in the final edition.

As you play the game, the interface keeps track of every command you enter in what it calls the “Skein”. If while testing you discover something you wish to change, you can immediately fix it, then press the “Replay” button to compile and run to the point at which you left off. Or, you can select any “knot” (command) in the Skein and simply “run to this point”. The Skein can be trimmed or removed entirely at any point as well. As a final bonus, when you are finished you can use the Skein to automatically generate a walkthrough for your game.

As well as keeping track of your commands, the interface also keeps track of the transcript of the game. (Note: This feature is currently only available for the Mac version. The Windows version has not yet been completed. So, I am describing something from the documentation; I haven’t tried it myself.) When you finish a portion of the game, you can “bless” that portion of the transcript, and if the output ever differs from that it will highlight it and show you the difference. So even if you aren’t observant enough to notice that the word “noise” has been changed to “nose”, the interface will detect it and let you know. Combining this with the Skein should be a powerful testing feature; as I said, I was unable to test the transcript feature.

Not enough features? Okay – there are built-in debugging and tracing features. These are integral to the interface and accessible every time you play, but automatically removed from the released game. You can even include text within your game to test any object automatically. Anyone who has written a game with special code to be used only in testing will enjoy another feature of the Inform 7 language. You can include a notation that a section of code is “not for release”, and it will be ignored when you compile the game for release.

Every time your game is successfully compiled, a clickable index to your game is created, organizing your game components in just about every way imaginable: by rules, by objects (“kinds”), by scenes, etc. This is especially handy once your game starts to get long, as you cannot separate it into different files. This is something I’ve always done to better organize my coding and keep the current code-set manageable. Inform 7 source text is all one big long stream of text. Yikes! As well as the above methods of indexing, you can create your own “Table of Contents” by placing headers in your text to organize it by Volume, Book, Part, Chapter, and Section. It will show up this way in the Index as well, and this helps. One feature requested is the ability to “roll up” the text, and this would help even more. Something I found annoying is that the Index isn’t created until you successfully compile your game – even when re-opening an existing game. So if you intend opening up your huge game and jumping right to the section you last worked on – forget it until you either compile once to create the Index or use the search feature to find what you are looking for.

If your compile fails in some manner, you are presented with an error message describing why it failed, complete with a clickable link back to the source text in error. No more scrolling through the text for “line 5247”! Not all of the error messages are as helpful as I’d like. One example was an if-condition that complained about having incorrect syntax. It turned out that the incorrect syntax was that I misspelled the object to which I was referring – the actual syntax was correct. Things like this are irksome, but I have yet to encounter a programming language whose error reports were 100% clear and accurate. Just like all of the rest, you eventually get a feel for what is causing the problems.

To complete the total creation experience, there is even an entire section of the manual and interface devoted to publishing your game. You can use it to create the released version, a walkthrough, even a web page that includes download links which you can add to your website (assuming you have one, of course). You can even define cover art and other files to release along with your game (“feelies”).

One last thing to remember is that at the time of this writing, Inform 7 is still in the beta stage. A new version seems to come out about once a week with corrections and additions. You would be advised to keep up-to-date at rec.arts.int-fiction, if for no other reason than to keep up on bug reports, new extensions, and new versions. An added bonus is the number of helpful examples being posted there almost daily.

In conclusion, here is my take on Inform 7:

The “natural language” isn’t as natural as they hype it to be, but it is a hell of a lot more natural than anything I’ve used in the past. Although it is very easy to sit down and bang out rooms, items, and descriptions, creating some of the more complex aspects of the game requires some thought and planning – just as in any other authoring system. And documentation that is more complete will certainly help. A lot of the learning process is through looking at examples, many of

which are filled with standard source text not fully explained in the documentation. Finally, the two versions of the interface need to be synchronized.

Will I be switching from faithful TADS to Inform 7? Probably not. I find myself more comfortable using traditional programming languages such as TADS. They seem to be more precise than the intentionally loose system of Inform 7. And I fully admit and am comfortable with the fact that this may be a baseless prejudice on my part. TADS also allows more access to the “nuts and bolts” of the system. Inform 7 does as well, but you have to use Inform 6 code to do it. However, that doesn’t mean that I can’t see the power and potential of Inform 7 for others.

Overall, I think it is an excellent alternative to those authors who are intimidated by the somewhat cryptic nature of TADS and Inform 6, but are frustrated by some of the missing capabilities of point-and-click systems like ADRIFT. As in all things, you will be the best judge. I suggest you at least try it out because it is quite frankly like nothing I’ve ever seen before.

For more information, look to these links:

Inform 7 home page:

<http://www.inform-fiction.org/>

Stephen Granade’s review at The Brass Lantern:

<http://brasslantern.org/writers/howto/i7intro.html>

The SPAG interview of Graham Nelson and Emily Short:

<http://www.sparkynet.com/spag/backissues/SPAG44>

The Yahoo discussion group for Shadow Wolf’s AIF extension:

<http://groups.yahoo.com/group/inform7aif>

Seven Seas of Thēah: Episode 7 by Christopher Cole

OPTIONS: At the end of this story each month, you will be given a number of options. Choose the option that you like and vote in the poll at the Yahoo AIF Archive. The option that gets the most votes will determine how the story continues in next month’s newsletter.

NOTE: You can read background information and other tidbits about this story here: <http://ccole.aftermath.cx/theah.htm>.

Magnus leaned back against one of the crates, licking the fruit juices from his fingers. He looked at Violetta and smiled. Her face was grim, perhaps still thinking about the near miss with the mage.

“Don’t worry,” he said. “We’re safe now. They won’t realize we’ve stowed away on the ship. They’re probably scouring the forest right now.”

“I’ll feel better when we’re fifty leagues out,” she replied, wincing as she ran a hand down to her left calf.

“Are you alright?” Magnus asked, reaching over and taking her leg onto his lap.

“Just a pinch,” she said. “I twisted it slightly jumping from the window.”

Magnus slowly untied the lower part of her leather leggings, rolling it up to expose her left calf. He began to lightly massage her muscle, slightly amazed at her skin’s smoothness. There were rumours that Violetta had faerie blood in her and sometimes, staring at her beauty, he believed it.

She stopped wincing as his rough hands worked a delicate magic.

“That feels better,” she said.

“Probably just a slight strain,” Magnus replied. “Just need to work it out a little more.”

His hands began to work their way up and under her leggings slightly, sliding over the bottom of her thigh. Violetta just watched him with a slight, crooked smirk on her face.

“We haven’t had a close call like that in a while,” Magnus went on talking as he rubbed the Raven’s leg. “Really gets the heart racing and the blood pumping. It’s easy to get lazy and not be prepared for those instances. Best to work out all the kinks.”

He had her left legging undone all the way to her hip and his hands were working her entire leg, from ankle to the very top.

Violetta raised her right leg and placed her foot on Magnus’ chest. She slowly pushed him back so his back pressed against a barrel. She kept her booted foot there, just under his chin. His hands still held onto the calf of her left leg, which was completely exposed. She stared at him, that smirk still on her face.

WHAT SHOULD MAGNUS DO?

- 1) Reach up and start massaging her right leg?**
- 2) Try to continue with her left leg?**
- 3) Move her leg and kiss her?**
- 4) Stop altogether and wait for her to make a move?**
- 5) Stop and apologize?**

AIF 101: Brainstorming by Johnny Freebase

Editor’s note:

We are re-publishing Johnny Freebase’s excellent AIF 101 tutorials that were originally posted on alt.games.xtrek. It’s our feeling that there are enough new community members who may not be aware of the existence of this guide. The tutorial was originally posted as a series of articles and we will maintain that construct here. This is the first installment.

* * *

Often, new authors are given the advice that the first thing they should do is pick an IF authoring language. I disagree. In my opinion, the language used is nothing more than a tool, the media through which you get your game across. Different languages are suited for different tasks, so how can you choose which one to use if you don’t know your needs yet?

The very first step is to develop an idea. You need to figure out what your game is going to be about. If you’re lucky, you might have some kind of idea in mind. Many authors can just pull inspiration out of their daily experiences, dreams, and fantasies.

If nothing comes to mind, pick something at random. Choose one of Polti’s basic 36 plots. Draw some Tarot cards. Grab a random CD, put it on shuffle, and use whatever song comes up. Whatever you have to do to get at least the germ of an idea.

Here are some examples. Feel free to use any of these in a game.

- Polsti’s 36 plots: DELIVERANCE: Appearance of a Rescuer to the Condemned.
- Tarot: The Moon - A yearning for fulfillment. Unforeseen perils; threatening situations, psychic forces.
- Song: Jimi Hendrix - Manic Depression.

Some people like to make fan games. Those can be fun... but all they give you is setting and characters. You still need a plot. Sometimes you can weave plot around an interesting character idea as well.

So, you have a very basic idea. Next, you need to brainstorm that into an actual plot... the basis of what happens in the story. You need to figure out who does what, and to whom.

For the above examples,

- Polsti’s: Deliverance. Well, perhaps the PC is rescuing someone, say a generic princess.
- Tarot: The Moon. Psychic powers. Hmm... perhaps a mind-control game wherein the player and his loved ones are threatened!
- Song: Manic Depression. The song is about a guy wishing he could make love to music.

Perhaps this could be someone who meets and falls in love with his muse. Or some other Greek mythological figure... perhaps the PC (Hercules?) has to seduce his way through the Goddesses!

And Xena!

See that last one? Don't be too attached to your original idea... they tend to mutate. Just get an idea and flow with it. You don't need to figure out how the sex fits in just yet... it should arise naturally from the plot, but you can come up with some ideas. So, I'll take and flesh out the second idea, the Tarot Moon card.

TAROT: Moon

The player somehow begins to develop telepathic powers. This brings him to the attention of other psychics, who threaten his family and loved ones. One focus of the game will be the development of the player's psychic abilities.

POSSIBLE SEX SCENES: Various sorts of mind control, voyeurism, possession, etc.

Game Reviews

Paradise Hotel, a review by A. Ninny

Game info:	<i>Paradise Hotel</i> released March 27, 2006
Author:	Blue Meanie
Platform:	ADRIFT 4.0
Size:	105 kb
Content:	m/f, voyeurism, m/m (optional, mild)
Game Type:	T&AIF
Length:	Medium
Reviewed:	May 2006
Extras:	None

Basic Plot:

As the assistant manager of a posh resort on an island in the Pacific, you're in charge of guest satisfaction. You must provide satisfaction to all the guests in order to get a shot to have sex with your beautiful boss, Marsha.

Overall Thoughts:

This game is the quintessential sex romp. It's the game someone is always talking about making: behind every door a girl and with every girl at least one sex scene. In games like these, the story is always a bit thin and the motivation for each character to have sex is certainly a stretch, but that's the nature of the genre, and for better or worse this game is no different. Still, it's fun. The sex is quite well written and each NPC is sufficiently different so that the sex doesn't get boring.

Puzzles/Gameplay:

The puzzles in this game are refreshingly easy and well-implemented. Generally, all the puzzles involve getting people alone by hooking up their companions with other NPC's. You'd think it might be difficult to figure out who to hook up with whom, but it isn't. There is one character whose job it is to provide information about each character's basic needs, and Blue Meanie's narration frequently acts to provide useful hints, like "You should talk to so and so before you tell her about that." As a result, I never found myself trying to guess what a character needed to go to the next stage in our relationship.

Sex:

The sex in the story is quite enjoyable and well-written, and considering the immense number of partners and locations, reasonably varied. It helps that the characters are quite varied from one another in age, marital status, nationality, body type and sexual interests. He exploits these differences to try to make the experience with the NPC at least somewhat different in each scene. Of course, being true to a proper romp, the sex is dripping with cliché: The PC's penis is immense, a teenage girl goes for anal sex, etc, etc, but that's ok in this game.

There is one homosexual interaction in the game. Blue Meanie gives the player a warning and the option to skip it.

Technical:

Recalling the diabolical guess-the-verb sequence in Blue Meanie's previous game *The Dinner Party*, in which many a player tore out his hair trying to pour and serve the wine, I feared for my sanity when I opened this game and began playing it. My fear was unfounded. This game plays quite cleanly, with no serious verb-guessing and precious few other bugs. It does have a few very picky little bugs (for instance, if I try to 'lick pussy' with Ann in the Jacuzzi, the game responds "Tina's not here"), inconsistencies (example: how would I know that Diane's a natural blonde if she's shaved between her legs?), I frequently get told off for swearing (especially galling if I type 'NPC fuck me' during a sex scene) and it could use more sexual synonyms (for example, I cannot 'suck tits' or 'kiss tits').

Intangibles:

In several instances, it is necessary to examine a NPC's body part in order to get past a certain point in the game. This occurs fairly late (at least it did in my play-throughs - the order other players choose to work through the different women may vary), when many players may have tired of examining every pussy, tit and ass. Players should keep this in mind.

Final Thoughts:

This is a fun, lightweight romp, from an author who's making a name for himself through the creation of fun romps. It a bit hackneyed at times, and maybe a bit too simple, but overall a good effort.

Rating: B-

***Gifts of Phallius 1: The Baron's Plot*, a review by Grimm Sharlak**

Game info:	<i>Gifts of Phallius 1: The Baron's Plot</i> released May 9, 2006
Author:	GoblinBoy
Platform:	TADS 2
Size:	229 kb
Content:	m/f, f/f, voyeurism, bestiality (mild)
Game Type:	T&AIF
Length:	Medium
Reviewed:	May 2006
Extras:	None

Basic Plot:

Set in a medieval/fantasy style setting, you play the role of a guard for Baron Jalafar, as part of a spying mission for King Goldik. The oft-suspicious Baron has been acting even shiftier lately, and the King has assigned you to keep an eye on him. The action's been mostly in watching over doorways so far, but in recent weeks it's become clear that the Baron has been planning something, and it's beginning to be put into action...

Overall Thoughts:

For a first game, *Plot* is a great effort. It has some interesting sexual scenes, decent puzzles and genuine humour. Progress can be difficult at times due to the linearity of the game, which can lead to frustration. The sex scenes, however, are quite good and aside from the last, are finite due to plot considerations, which is a nice touch.

Puzzles/Gameplay:

To be honest, the biggest problem with *Plot* for me was "why can't I do this?" As it turns out, the question should instead have been "why can't I do this *yet*?" The puzzles, while straightforward, are essentially linear. The most baffling for me was trying to hook up a guard so I could get him out of my way, but I couldn't do anything about it until I had completed some unrelated tasks. While it doesn't make the game unplayable, it can be quite frustrating to be faced with a puzzle you can't complete (even though the game hints that you can) because you'll upset the plot of the game.

Other than that gripe, however, the puzzles are mostly straightforward and won't tax players too much. Getting all the points, however, is more of a challenge and it's good to see some inventive sex commands being included.

Sex:

Speaking of which, the sex in *Plot* is of a high quality. This is a sex-charged game, and despite the political intrigue you're embroiled in, the main character is still a horny bastard. Each scene is quite detailed, and of a decent length, especially the final scene, which is a good reward for completing the game. The scene with the Baroness may cause some folk distress, however, as it involves transforming your man parts into... non-man parts. However, it's not a big part of the scene and just another "spell" in the game.

Technical:

Aside from the linearity issue, I encountered no real bugs in this game. A very good level from a first game.

Intangibles:

This game has a little something for everyone, a bit of humour, a bit of political intrigue, some interesting transformations and even a couple of areas that are pitch black, in good old IF tradition. No grues, though.

There are three separate endings, but you select them at the end, rather than your actions affecting the outcome.

Final Thoughts:

This is a great first effort from GoblinBoy, and I'm definitely looking forward to *Gifts of Phallius 2*.

Rating: B



This is Erin! comic strip #18. All previous strips can be viewed on the AIF Newsletter web site and on -3-'s new site: <http://comics.aifcommunity.org>. Don't miss them!

AIF Wants You

If you can write game reviews, articles, opinion pieces, humorous essays, or endless blather, we want you. Contact the Editor for suggested content or just write what you want and send it to us.

Staff

Editor A Ninny is an AIF player, author of two AIF games and frequent beta-tester. His *Parlour* received an Erin for Best "One Night Stand" game in 2004.

Web Master Darc Nite is a newcomer to the AIF scene. He is an avid gamer who heard the call for help with the AIF Newsletter.

Staff Writers A Bomire is the author of several TADS AIF games, including *Dexter Dixon: In Search of the Prussian Pussy*, *Tomorrow Never Comes* and *The Backlot*. His games have won numerous awards and Erin nominations.

BBBen is an AIF author. His games have received two Erin awards, numerous nominations and first place in A. Bomire's 2004 mini-comp.

Grimm Sharlak is the author of two AIF games: *Breakout* and *Of Masters and Mistresses: Abduction*.

Christopher Cole has written many popular ADRIFT AIF games, including *Camp Windy Lake*, *Gamma Gals*, and *Mount Voluptuous*. He is the 2005 winner of AIF's Badman Memorial Lifetime Achievement Award.

Submitting your work to "Inside Erin"

Please direct all comments, articles, reviews, discussion and art to the Editor, A. Ninny, at aifsubmissions@gmail.com.

