inside *erin*

THE AIF COMMUNITY NEWSLETTER

Contents

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.

A Letter From the Editor

Purple Dragon

Welcome to the January Edition of Inside Erin, the very first issue in a bright new year. First of all, I have to apologize for the lateness of this edition. My laptop, which has served me faithfully for the last, oh, 40 or 50 years finally exploded, literally. Well, ok, not literally, but there was some smoke involved (also a fifth of tequila and three Mexican hookers, but that's another story). I had the whole thing just about done, and was only waiting for one last article when it happened. Unfortunately, I have been unable to recover my data at this point, so I had to start over from scratch. It was a long road doing the whole thing twice, but as you see, I was able to pull it together.

This month we have our this month articles, and I'm happy to report that both Nandi and Tera sent us a new article reporting on the happenings at TFGamesSite, and Hypnopics Collective. After a bit of a break, A. Ninny has another installment of The Aphrodite Chronicles. There are a couple of reviews, a few musings by yours truly, and of course, the final installment of Programming Erin.

All in all, not a bad little issue, so sit back, relax, and enjoy the show. I'll see you next month.

This Month In AIF

by BBBen

A bit more activity this month, which is good to see. Hopefully it's all in anticipation of the threesome comp – remember the deadline is the end of January! If you haven't started a game yet there's still time (though I have noticed a couple of people complaining that they don't think they'll have time to get entries finished – come on, guys, I gave you like six months!) and if you have started a game then it's time to get it finished. I'd really like to see a good crop of entries, because it's not clear at this stage whether anyone's going to get anything done in time! It's a unique, one-off comp, so don't miss your chance!
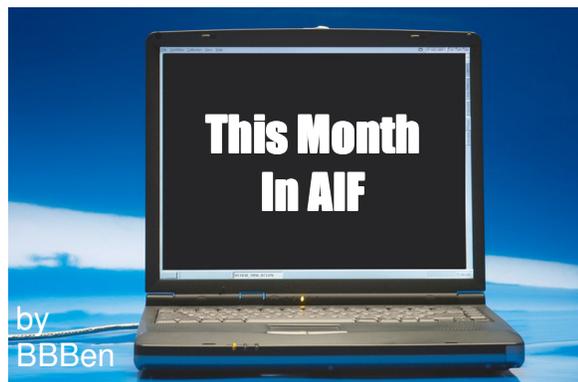
A. Bomire finally released part 2 of his AIF tutorial for TADS 2 this month, which will probably be enormously helpful to anyone wanting to learn to work in that platform, and will compliment well the release of the full Programming Erin series (which is mostly designed to help a new author choose a system to work with).

On the AIF Archive there was some activity about cataloguing those games that have female protagonists. It was a reminder that one day it would be really nice to have some kind of listing whereby players could search for games based on various aspects of the content, and the best suggestion was that such information be put into the ifWiki, which is not a bad idea although the ifWiki is not really focused on AIF.

On AIFGames.com the Live AIF forum continues to see activity, which is a good thing. Also, non-specific plans for some kind of multi-author collaboration game are being tossed about, so we'll have to wait and see if anything comes of that. Suggested ideas have included some kind of Quantum Leap style story so that each author writes in a completely different setting, or some kind of open world in which all the stories are tied together only by the organiser's work. I wish them good luck with this and humbly suggest they not get too ambitious for their own good.

This Month At TF Games Site

By Nandi Bear

As always I had many plans for little pearls of wisdoms to share with you, unfortunately I got sick. Oh, don't worry it's only a cold, but it gave me time to think about the relationship between gamers and designers.

First a confession, I'm a terrible game writer, oh I don't mean the game side (I like to think that in the game leagues I'm kind of semi-pro), I mean in my general style. You see I get bored easy (oh look shiny…), I bounce from game to game, creating half and semi finished games, but hey I'm sure everyone understands, they're happy to wait until I finally finish one of my many mega epics.

But then I'm a player of games, so I wait patently for a game to be finished -- for a couple of days. I just want those games now!! Because obviously the author is working 24/7 on completing the game just for me.

Now I wish I could offer a simple solution to this but I can offer a few helpful suggestions that I'm sure I'll break in the future. The main one for game creators is to never give a date of any release, it locks you into a timeframe (which you'll break) and disappoints when the game isn't out. And pick a couple of projects and stick to them, between little side projects and those pesky contests I stick to two games, jumping between them when I get bored.

And players, please don't bug for release dates, it's a pressure, however friendly it sounds. Compliments and constructive criticism will drive a writer more, like a good director you're never happy with what you've released. These things tend to drive a writer to continue to do good works.

COLLECTIVELY MADE...

This month at the Collective...

By TeraS

Before anything else, I do hope that you all had a wonderful and happy holiday season!

Now onto the Collective happenings for this month…

Firstly and most importantly, Matrix1807 has started development on a new game!

In his words:

The game is called NTJ - Natalie's Transformative Journey. As the wisest of you probably guessed, it's about Natalie (and she TFs). The extremely small game can be downloaded here: http://rapidshare.com/files/176862286/Nat3.rag

Comments and ideas or other things that can help the author of this game, can be posted at the Collective here: http://hypnopics-collective.net/viewtopic.php?f=11&t=14483

Next, DragonTrainer has uploaded a new version of his game called Jigsaw Town II, which can be found here: http://rapidshare.com/files/175937378/Jigsaw_II_-_Trial.rar

Comments and ideas can be posted at the Collective here: http://hypnopics-collective.net/viewtopic.php?f=11&t=13871

In other news, _Z_ posted a link to a site called Dark Translations that creates patch files to translate Japanese games into English. That link is: http://www.darktranslations.com

In the Collective Gallery this month, Ehlanna posted a new version of her game called Debutante here: http://hypnopics-collective.net/cpg132/displayimage.php?pos=-83909

And that concludes the game updates at the Collective for this month…

Please do visit and remember that there is also an active RP section of the Collective and of course the always updating images gallery!

Have a Happy New Year from all the Mods and Admin of the Collective! ◆

This Month in AIF - Continued from page 1

There was an ADRIFT AIF game released this month, and a couple of public beta RAGS games. As usual with the games that are made primarily for the Hypnotics Collective and the TF Games site, I am not 100% sure when the games were originally released or whether they are a final version or even if I should announce them here. In the absence of clearer information I'll announce them as they were announced to the AIF community.

New Games

Career Change by Sally73 for RAGS, released 7th December 2008. You are staying at a friend's place while you reconsider your life, but you may find that more is about to change about you than just your career.

Debutante by Ehlanna for RAGS, released 15th December 2008. After being mugged of the money that you owed a loan shark, you now have to find him a woman, so you set yourself to the goal of scoring his step-daughter for him.

Magic Wishing Fountain by Erus, released 22nd December 2008. You are in a fantasy village with a mysterious stone fountain in the middle, and while generally getting laid about town, you might discover its secret. ◆

TFGames - Continued from page 2

Someone who never seemed to need any help was a game creator named Decker who has since moved on to new pastures. I'm afraid to say I missed his first game, *Summoned*, a clever little game which switches from the sorcerer's point of view to that of his summoned succubus. I didn't miss his second game, *the Jade Ring*. Now I've always thought that Decker was a writer worth pointing out to AIF, and I'm sorry that I didn't. The main reason was that he wrote like an AIF creator, but with TG sensibilities. Let me explain. First, he wrote in ADRIFT, a popular favourite on the boards, second he actually wrote long, thoughtful text, with proper sex scenes, multiple positions, the works. But he also wrote good TF, with multiple bad endings, something you don't really get in vanilla AIF. After three partly finished games, (*Summoned*, *The Jade Ring,* and *Tower of Ariman* [aka *the Tower*]), and a contest entry, (*Hit or Miss*) he moved on to other things. I for one miss his work, if only to find out what goes on in my house of reed (I promise not every entry will be six degrees of Nandi, though he did kindly let me poke around his code).

On the actual forum (after all that's what I get paid for) things have picked up a bit from last month. While the October contest is still ongoing (I can't really comment since I'm in the middle) another contest has been announced for March by Jamehlers and Maelyn. This one is an interesting idea, a large (10+ rooms) game which involves a story with TF, but includes no adult content, something for the PG end of the market so to speak. I think this one looks interesting so I'll keep bugging you until the contest is over.

Also a couple of betas were released one from me (*House of TG*, ADRIFT), one from Sally73 (*Midlife crisis*, RAGS), which you might have already seen, and one from Matrix1807 (*NTJ*, RAGS) which TeraS may bring up in this month at the Collective.

Well I've babbled on enough, happy New Year and may there be many more AIF, TG and MC games in the oncoming New Year. To 2009! ◆

Chapter 6

*D*ear Mortal Men and Women,

I apologize for the delay between my last letter and this one. In troubled economic times, even the Goddess of Love needs to earn money in the old-fashioned way. And no, *not* old fashioned in the "oldest profession" sense. I just have a regular job.

If you recall my last missive, I briefly mentioned ifeelmyself.com while describing my encounter with Chica, a young woman who has since been featured several times on the site. It seems important for me to praise them once again, so this will be my rant this month, and I will keep it brief. I Feel Myself presents female sexuality in the most honest, beautiful, direct way I have yet seen on the Internet. The content is very simple: videos of real women masturbating and having orgasms. Usually they're on their own, but occasionally they have more than one woman making love to-

The
Aphrodite
Chronicles   by A. Ninny
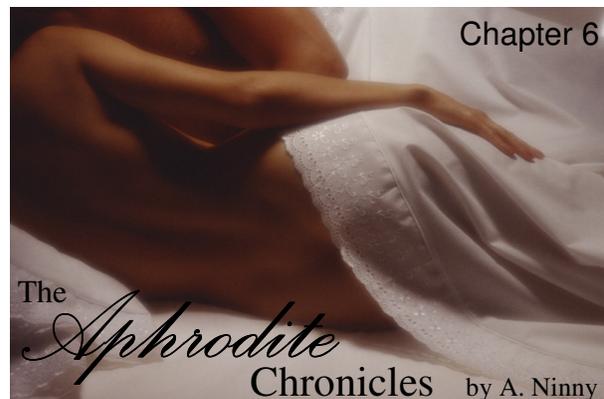
gether. What I like so much about IFM is that where most porn divides women, separating their genitals from the rest of their bodies, IFM takes a holistic approach, presenting women as complete, and their sexuality is displayed as a whole-body experience. The production is run by women, and it shows: the lighting is soft and complimentary, the audio is given equal consideration to the visuals, and the web site itself is simple and no-nonsense.

Now, on to the story I wish to begin to relate today. I mentioned earlier that I have been working a 'day job', and that has led me to commute to work by rail. Obviously, for a Love Goddess, the most interesting thing about commuter rail is that you see the same people every day, some of them twice, and even if you don't become acquainted, you become familiar with them and their habits. Over the course of the past few months, I became aware of two particular people, a man and a woman, that I decided could benefit from an Aphrodite intervention.

The woman is an ice queen, a familiar type for me – many of the warrior goddesses I know suppress their emotional selves to focus on their skills, and become very unapproachable as a result. This woman is really astonishingly attractive. She has a long, slender body that she dresses beautifully, but her clothes are almost always overly modest. She wears turtlenecks and blouses buttoned to her throat, long skirts or trousers. She shows the minimum of skin. She has striking features: clear fair skin, high cheekbones, a long, straight nose and expressive lips that occasionally unfurl into the brightest, warmest smile that shatters her ice queen look (she keeps the smile under wraps most of the time). Her hair is very long, deep red, dead-straight, with harsh straight-across-her-forehead bangs, and always is either in a pony tail or flowing lightly over her chest.

The man is less descript in appearance – the most important thing we need to know about him is that he is obviously infatuated with the ice queen. Each day I watch him board the train and when he enters the car he looks around to see where she is sitting. He always takes a seat where he can see her. Early on I observed he deliberately sat next to her, perhaps hoping (in vain) for an opportunity to break the ice, but later he became discouraged and instead sat where he could watch her without possibility of interaction.

Intervening in an infatuation is delicate business. It was very likely that the ice queen knew of it; many times she didn't have an alternate distracting activity while commuting and it would be hard for her not to notice a man attempting – clumsily – to be discreet (peeking up from his book or laptop) while watching her. I decided the intervention should begin with her since I felt that she would have to thaw in order to make it possible for him to approach her.

I certainly didn't have any problem meeting the ice queen. I simply sat next to her and began talking, behaving as if I'd known her all my life. My goddess charms work on mortal women, ice queens or not (mortal ice queens are merely putting up a front – they're not able to suppress their feelings like goddess ice queens can).

She had a thrillingly beautiful voice – rich and contralto with an eastern European accent, Polish, perhaps. Her name was Sonia, and she smiled easily as we made getting-acquainted small talk about our work, about the train, and about our homes and families. Her hands were animated when she spoke, her long fingers moving expressively. I kept one eye on the man during my conversation with Sonia; he was watching us, no longer able to be discreet while straining to overhear our conversation.

Finally, I got to what I knew was the point. "You seem to have an admirer," I said, flicking my eyes in his direction.
She made a point of deliberately not looking over and agreed. "I know. It's kind of cute, in a pathetic kind of way, I guess. I used to wish he'd introduce himself, but now sometimes I think about finding a different place to sit. I worry he's a stalker."
"Did you ever give him any kind of indication that you wanted him to talk to you?" I asked, ignoring her stalker comment.
"No, I guess not."
"From what I saw, you put on an off-putting air. That could explain his inability to talk to you."
"I didn't do it on purpose," she said thoughtfully.
"Let's get his side of the story," I suggested, and before she could object or even register what I was suggesting, I was beckoning him to come over.

This story gets very interesting, but the rest of it will have to wait for upcoming months, when I will relate it all in detail.

Until then, I wish you all wonderful love.

Aphrodite

E very so often a discussion seems to crop up about what people like in their AIF games. Such a discussion has been going on again recently in a couple of threads over at AIFGames.com. A lot of the discussion seems to be talking about how realistic the women/scenes/situations/whatever should be. Should they be completely realistic, or complete fantasy? Now, that is a question that can really never be answered in any ultimate sense. If you asked two different people, you would likely get two different answers. I also don't think that it will come as a shock to anyone that most people's answers seem to lie somewhere in the middle. For instance, here are a couple of opinions from the boards.

How Real Is Too Real?

By Purple Dragon

Andrei said:
*"My preference is for **magical realism**. The essence is that everything is realistic, *except for one thing.* For example, it could be a realistic world, except that all the girls are easy. Or, it could be a realistic world, except that you have a scary resemblance to a famous gladiator, and nobody can tell you apart. The "realistic world except" bit sums it up... gives you a hook to buy into, and a world that feels real."*

Interesting idea except, "all the girls are easy"??? Isn't that all AIF. Well, here's hoping anyway.

Dudemandude said:
*"Personally I like what I call **exaggerated realism**. I don't really care if things aren't perfectly realistic, but it should have some basis in reality so that I feel I can relate to it. I like a little fantasy, but when things get too far out there it can sometimes be a turn off as I loose the ability to relate to the characters."*

Both good points, and I find myself somewhat agreeing with both of them. Personally, I tend to lean toward realistic women and scenes in the games I play and write, but I have a couple of big qualifications there.

Realism is fine, but I think that too much realism can become tedious. This is one of the reasons that I have never really cared all that much for arousal systems. They claim to promote realism, but unless the author is very careful, they can actually end up having just the opposite effect, repeating the same block of text over and over again until one or both participants reach orgasm, at which point you start all over again with a different action. That is an EXTREME oversimplification, and I should mention that I have greatly enjoyed many games that make use of this system, so don't anyone get their panties in a bunch.

So what about other aspects of the games. Is it realistic for a guy to be able to come a dozen times a night, or for a girl to jump into bed with you at the barest of interactions? No, or at least such has never been my experience in life. But does anyone really want to play a game that forces you to chat up and win over a girl for even a couple of hours (which would still be rather unusual in a real life situation) before getting to the good stuff? I doubt it; I know I wouldn't. So when I say that I like realistic women and situations in my games, I am fully cognizant of my own hypocrisy.

What about types of subject matter. Is it realistic for you to start out raping a girl only to have her be totally into it by the end? I shouldn't even have to answer that one, but here we are talking about personal preferences in games. I really don't get into extreme violence in my games. I learned that when I wrote my second game, and included scene that started out as bondage (which I sometimes do enjoy) but ended up as a torture scene. I don't like it, I won't write another scene like that in a game, and I tend to shy away from games that have that kind of thing in them. But does all that mean that there is no room for such games in the community? Of course not. It's just not my cup of tea, so all I tend to see is how unrealistic the situation is.

Now let me give you an example of something that I do enjoy. I like anal sex in the games that I play, and I have put at least some of it in most of the games I've written. Does everyone like anal sex? No. Do most women like anal sex in real life? I'm afraid not (Damn porn! Lied to us again). However, it's a kink I enjoy, so I happily add it into my games, and look the other way as to realism when playing others.

When it all comes down to it, I don't think it's really about realistic or unrealistic at all, but rather about what types of unrealism people enjoy. Something that might be stupid and unrealistic to one person, could very well be one of the hottest thing ever written to another. So in the end, you should probably just write what you enjoy, and rest assured that someone out there will enjoy it right along with you. ◆

Debtutante
Review by A. Bomire

Game info: Debtutante, v1080.01
Author: Ehlanna
Platform: RAGS
Size: 2.5MB
Type: PF
Content: mf, Transgender/Transformation
Length: medium
Reviewed: January 2009
Extras: Pictures

Game Reviews

Basic Story

You play a male character who has fallen on hard times and had to take a loan from a loan shark to get a new car. Unfortunately, as you are going to pay back the loan your money is stolen from you and you make a deal with the loan shark to provide him with a slut to make up for your missing funds.

Overall Thoughts

This game was originally intended as an entry into a competition for the TFGamesSite.com website. This website (and community) is devoted to games dealing with transgender/transformation - men into women or vice-versa. According to the author, this game just wasn't finished on time for the competition, and so the game was expanded and released in this larger form.

Like all games which come from the TFGames website, this game has an element of transgender/transformation to it, whereby the player can change himself into a girl during the game (actually, 2 girls). But, he can also change himself back again if he wants, so it isn't a permanent problem. Nor is it forced upon the player either; you make your own decision regarding this option. You can also transform other girls within the game, although they change into different girls and not into men. Most of the game revolves around wandering about town, doing research and obtaining materials in order to accomplish your transformations, and to find/create the slut for the loan shark.

What I liked most about this game wasn't the play or the transformations, it was the humor spread throughout the game. The author does a good job of working in some clever replies to actions by the player, and to provide a variety of options for the player to try. An example is attempting to get away from the thugs who mug you. Some of the options include are quite humorous, such as just flailing wildly at them.

Puzzles/Game Play

The puzzles in this game are fairly straightforward. The author provides plenty of clues as to what you are supposed to do to get started on your quest, and as you proceed enough hints are dropped that you never quite feel like you're in a position where you don't know what to do next - although actually doing it may take some time. I like the idea that the game takes place over several days, reflecting real life situations in doing research at the local library and then studying your research notes. I found it a bit tedious waiting around for the library to open, but you are given an option to skip directly to the time that the library opens so it really isn't that bad. There is money in the game, but if you lose it you are still able to obtain the materials you need by stealing. There is an element of randomness as to whether or not you accomplish your theft, so if you are going to steal then you definitely want to save the game first.

Sex

The sex, that I found, was pretty tame. As a RAGS game, there are lots of pictures of naked women, some of them performing sex acts, but hardly any of them are interacting with you (the player). If you decide to opt to turn yourself into a woman, then some more options become available, but it is still move visually oriented than textually well described.

Technical

There are the normal amounts of typos and misspellings scattered throughout the text of the game, although one or two scenes seemed to be rife with them. I never got stuck (except when expected to, such as working on a puzzle), and the game didn't

lock up on me. As is typical with RAGS games, you may need to download the current version of RAGS to play the game correctly. At the time of this writing, it is version 1.0.8.0.

Final Thoughts

As transgender/transformation games go, this one is pretty tame compared to some of the others from the TFGames group. As such, the average AIF player will probably enjoy it more than some of the other games from that site. The game is not bad as a puzzle game, but if you are looking for hot sex, other than some pictures, then you might be disappointed.

Rating: C+

The Magic Wishing Fountain

Game Info: The Magic Wishing Fountain
Author: Erus
Platform: ADRIFT 4.0
Size: 48KB
Type: T&AIF
Content: mf, ff, mff, mmf, BDSM
Length: Short
Reviewed: January 2009
Extras: None
Average Rating: C

Basic Plot

You are a villager in a village with a magic wishing fountain in the center. The fountain is rumored to grant a wish to someone who can give the fountain "what it wants". Your task is to discover just what this is, and to find out what others want as well and satisfy them (as well as yourself).

A Review by A. Bomire

Overall Thoughts

This is the first game by this author, and as such it plays like many first games. It has a very straightforward plot which is the standard treasure hunt: find this and give it to that person, get a reward of this other thing which then goes to that person, etc.

Puzzles/Game Play

As I mentioned, the game play is very linear and straightforward. You find a series of objects, each of which is given to you as a reward for performing some action. You get the objects one-at-a-time and pass them on to the right person to get another object. This cycle continues until you reach the end of the game. Each person wants one and only one thing, so when you find something you just go from person to person and hand it to them. The right person will accept it and offer you a reward. The downside to this is that sometimes it isn't obvious just which objects go with each person, and you can sometimes find yourself wandering from room to room trying to figure out where to get the next object.

Erus, the author, has helpfully placed the important objects in bold print, and you should be sure to ask each person about the important objects in their location as this provides hints about what that person is looking for.

Sex

I was surprised at the sex scenes in this game. Surprised because they are actually pretty well written, which is not something I usually expect from a first time author. Each scene is fairly long, with plenty of details and actions. And there is enough variety to satisfy just about every kink - male/female, female/female, threesomes (both two girls and two guys), and even some spanking/BDSM. In most cases, once you perform a sex act, you cannot repeat it and get a message indicating that it's time to move on to something else. But there are times when you can repeat your actions, although they typically just repeat the same description as well.

Technical

This game is fairly technically sound, although there were a couple of minor bugs. The author has used a technique that I often find in ADRIFT whereby a girls body parts are associated with a specific room, and you can get some weird responses in trying to interact with those body parts even when the girl isn't there. Also, there were a couple of places where to advance the game along you needed to perform an action that took a little trying to figure out. They aren't necessarily "guess the verb" situations, because they needed common verbs to complete. It was more that the action you needed to take didn't seem obvious.

Final Thoughts

As I said, this game is a first for the author, and it has much in common with other first-time authors. However, it showed a lot of promise (especially with the sex scenes). I think I would have enjoyed the game a little more if the puzzles hadn't been so straightforward and linear, with the typical treasure hunt theme.

Rating: C-

A Review by Purple Dragon

Overall Thoughts

It's not easy to take a series of completely unrealistic situations and make them seem at least plausible, but somehow this game gave me the tools to do just that. While there was certainly room for improvement, there were enough good things here to let me really enjoy myself.

Puzzles/Game Play

The puzzles are of the standard, get item…give item…get sex type that you tend to see so often. While there is nothing necessarily wrong with that, it would have been nice to have a bit more variety. Still, even though the type of puzzle is pretty straightforward, I did appreciate the fact that in a couple of cases, the item in question had at least some place in the overall story, rather than just being, "Here's a glass of water." "Thanks, lets have sex."

Sex

The sex scenes are definitely where this games shines, and since it's probably safe to say that that is a major reason why we play these games, the shine is nice to see. The scenes are well written, detailed, and hot. There is also a lot of variety between them and you never get the feeling that you're just going over the same scene with a different girl. In fact, the sex is so varied, that a couple of times the game thrust me into situations that I normally wouldn't have expected to find particularly arousing. It's credit to the author that he held my attention even here.

If I had to say something bad about the sex scenes, it would be that they really needed better responses when the player tries a command a second time. Sometimes it just repeats the same text, which is actually okay in my opinion. What kind of bugs me is being told something like, "No, try something else." Then again, it might just be me, and at any rate, it's not a huge deal.

Technical

This game is a bit hard to pin down from a technical standpoint. There are really no major problems with the game. No game killing bugs, no major spelling or grammar issues, no glaring errors. Where the game would loose points in my book is not for the negatives it has, but for the positives it doesn't. Let me give a couple of examples of what I mean by that.

At times the game feels very undirected. This is especially apparent at the beginning where the player is just dropped into the village square with no direction at all. Once you get things going, it's not a very complicated game, so this isn't as big an issue as it could have been. But still, some kind of introduction to get the game moving would have been very nice to see.

I also noticed that you can't examine body parts except during sex scenes when the girl is naked. The game doesn't really have a clothing system, and clothes are just done away with at the beginning of each scene. I personally believe that this a perfectly reasonable way to handle it, especially for an author's first game. However, even if every girl is just going to be either completely clothed, or buck naked, I really like to see that second description of her and her bits and pieces. On the

subject of body parts, there are also a couple of times where you get some really odd responses dealing with them when the girl isn't in the room. This has to do with the way they were programmed, and it's not too hard to overlook the discrepancies.

Like I said, most of the problem areas I found were minor. There is nothing that will stop you from playing to the end, and enjoying yourself along the way, just a lot of little things that really could have kicked things up a notch.

Final Thoughts

All in all, a solid little game by most standards, but adding in that this is the first game by this author raises the whole in my estimation. A bit more attention to detail could certainly have made it better, but very enjoyable nonetheless, and I look forward to his next one.

Rating: C+

W elcome to the seventh and final installment of Programming Erin. This month we're going to talk a bit about clothing. Clothing systems in AIF can be one of the more complicated things to program, and we will not even be attempting to cover all the options. Remember that our purpose here is not really to write a tutorial, but to compare how to do some basic things in all the different systems.

Speaking of purposes, you may recall that I have mentioned a couple of times that we were planning to release the source code for the games when we were all done. By the time you're reading this, that should already have happened. I have posted a single zip file called "Programming_Erin" in the author resources section of both the AIF Archive, and AIFGames.com. Included in the file is the full text of this series of articles, the source code, and a compiled game for each of the platforms. Check out the "read_this_first" file in the package for more information. We hope you've enjoyed the feature, and whether you have or not, we would love to hear what you thought about it. Thanks for sticking with us to the end, and now, on with the show.
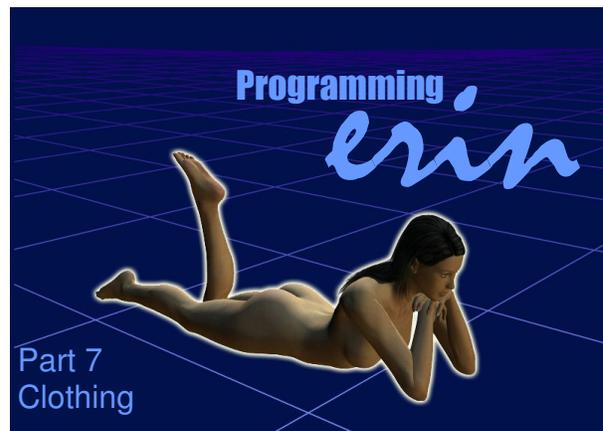
TADS 3 Segment by Knight Errant

By now our little demonstration game is finally nearing completion. We have the basic plotline laid out, and the player is able to go through and have sex with Erin. Now, we simply need to add the ending and some "finishing touches". First is clothes. Many games only include clothes on women, some include clothes for men as well. In this case, we'll just demonstrate clothes for Erin and if you would like to add in clothes for the player as well, go ahead.

Clothing can either be very simple or very complicated, depending on how much of a simulation you would like it to be. Layered clothing can get quite complicated, and as such it's a little beyond the scope of what we want to do here. We'll simply give Erin a suit, as one object.

```
+ suit: Wearable 'sexy suit' 'suit'
    "<<isWorn ? 'Erin is wearing a nice suit.  In particular, her tight skirt
shows off her tight ass and slender legs.' : 'Erin\'s suit is in a crumpled
heap.'>>"
    wornBy = erin
    owner = erin
    dobjFor(Wear)
    {
        check()
        {
            if(gActor!=owner)
                failCheck('That wouldn\'t look good on {you/him}.');
        }
    }
;
```

The fact that clothing has several possible states means it has fairly complex handling. First we have to introduce a new syntax. Normally, double-quoted strings simply print as they read, but sometimes we want it to change based on the current game state. We can add programming statements into double-quotes strings between << and >> brackets. What we have in the description is a special syntax for an if statement. The part before the ? is the variable to be checked. If the variable equates to True, then the first string is used. If the variable equates to Nil, then only the second string is used. In this example, if suit.isWorn then it displays one description. If not, it displays the second.

The next line, wornBy, means that at the beginning of the game Erin is wearing her suit. If we didn't define this line, she'd simply be portrayed as carrying it. After that, we set Erin as the owner of the suit. This is a property which can be quite useful, among other things it attaches "Erin's" to the vocabulary. This way, the player can refer to it as "Erin's suit". We can also use that for disambiguation, but since it's the only suit in the game we don't need to. Finally, we add a bit of code to prevent anyone but the owner from wearing it. Notice that we have {you/him} at the end instead of just "you". This is called a message parameter substitution, it uses predefined rules to determine what to insert into that part of the message. In this case, it chooses based on the actor: if the actor is the PC, then it prints "That wouldn't look good on you." If the actor is not the PC, then it used "him" or "her" as appropriate. It would also use the appropriate pronoun if we decided to switch the game to first or third person.

Now that we have her clothing, we need to have alternate descriptions of Erin's body parts when the suit is worn. That can be done with a suit.isWorn check, much like we did for the suit's description. We can also use a suit.isWorn check to prevent certain sex actions when her suit is worn.

```
+ pussy: Component 'pussy/snatch/vagina' 'pussy'
    "<<suit.isWorn ? 'You can\'t see her pussy through her skirt.' : 'Her
pussy is shaved bare.'>>"
    dobjFor(Fuck)
    {
        action()
        {
        "<q>If you want those pictures back,</q> you whisper, <q>you'll have
to let me fuck you.</q>  You turn Erin around and force her over her desk.
She yelps in surprise and moans when you reach between her legs, discovering
that she's not wearing panties.  You unbuckle your pants and slowly slide your
cock into her dripping pussy.  <q>You're such a naughty girl, fucking your co-
worker in your office.</q> Erin just moans in response, gripping the desk and
grinding her ass back at you.  <.p>
        <q>Oh yes, I'm your dirty girl!  Oh god, fuck me!</q>  Her words fade
into moans, her body writhing against you as you pound your cock deep into
her, gripping the swell of her hips for leverage.  She arches her back,
screaming as she climaxes.  Once she calms down from her orgasm, Erin turns
around and expertly sucks your cock until your climax.";
        }
    }

    dobjFor(Lick)
    {
        check()
        {
            if(suit.isWorn)
            {
                failCheck('You\'ll have to take her suit off first.');
            }
        }
        action()
        {
            "You lick her pussy.  ";
        }
    }
;
```

Finally, we need to allow the player to remove her suit. By default, TADS 3 prevents the player from taking anything from NPCs, so we need to override Erin's checkTakeFromInventory method, like so:

```
    erin: Person 'lovely young coworker erin' 'Erin' @ erinChair
        "Erin is your lovely coworker.  Her slender body and long legs have been
    the object of many of your fantasies.  "
        isHer =  true
        posture = sitting
        isProperName = true
        checkTakeFromInventory(actor, obj)
        {
            if(actor==gPlayerChar && obj==suit)
                {"You slowly strip off her suit."; suit.makeWornBy(nil); suit.
    moveInto(erin.location);}
            else inherited;
        }
    ;
```

Now the player can >take suit, but TADS generally doesn't allow "remove" to work for clothes you're not currently wearing.
I've cut out the parts that we've already coded.

```
    + suit: Wearable 'sexy suit' 'suit'
        dobjFor(Remove)
        {
            verify(){}
            check()
            {
                if(!isWorn)
                {failCheck('No one\'s wearing that.'); }
            }
            action()
            {
                if(isWorn)
                { makeWornBy(nil); "You slowly strip off her suit."; moveInto
    (erin.location);  }
            }
        }
    ;
```

Finally, we can code the end of the game. For simplicity's sake, we'll have the end triggered after the fuck command. The
ending cutscene gives us an excellent opportunity to tie up the plot, such as it is.

```
        dobjFor(Fuck)
        {
            action()
            {
            "<q>If you want those pictures back,</q> you whisper, <q>you'll have
    to let me fuck you.</q>  You turn Erin around and force her over her desk.
    She yelps in surprise and moans when you reach between her legs, discovering
    that she's not wearing panties.  You unbuckle your pants and slowly slide your
    cock into her dripping pussy.  <q>You're such a naughty girl, fucking your co-
    worker in your office.</q> Erin just moans in response, gripping the desk and
    grinding her ass back at you.  <.p>
            <q>Oh yes, I'm your dirty girl!  Oh god, fuck me!</q>  Her words fade
    into moans, her body writhing against you as you pound your cock deep into
    her, gripping the swell of her hips for leverage.  She arches her back,
    screaming as she climaxes.  Once she calms down from her orgasm, Erin turns
    around and expertly sucks your cock until your climax.";

            "<q>Thanks for doing this for me,</q> Erin says, as both of you
    straighten up.  <q>I've always fantasized about being taken advantage of at
    work, and my boyfriend really wanted to see me fucked by another man.</q>/p
            <q>Wait, what?</q> you ask, surprised.\p
            <q>Yeah, you played your part perfectly.  Thanks!  See you Monday!</
```

```
q>  With that, she leaves.\p
        <<finshGame(ftVictory, [finishOptionUndo,finishOptionFullScore])>>";
        }
    }
```

And that's the game!

TADS 2 Segment by A. Bomire

This is the final chapter in our tutorial on writing a game using the TADS 2 authoring language. There is obviously a lot left out of the demonstration game that we've written, and I don't plan on covering all of it here. But some of the things left to cover will be found within this final section. The first is clothing. TADS has a built-in *clothingItem*-class of objects specifically designed for characters to wear and remove. It is a fairly simple object, and as such cannot handle more complex tasks such as layered clothing (i.e., the bra is under the blouse, and thus not visible when the blouse is worn, and you cannot wear the bra over the blouse). Creating layered clothing is a little beyond the scope of this tutorial.

Our female character, Erin, can be wearing all kinds of clothing, but to make things simple we'll have her wear a single article that we'll design and describe as a full set of clothes. This is one easy way of handling it, and I often use it for the player in my own games. For an office environment, we'll have Erin wear a business suit:

```
ErinsSuit: clothingItem
   location = Erin
   sdesc = "business suit"
   ldesc = "<<self.isworn ? "Erin is wearing a nice suit.  In particular, her
tight skirt shows off her tight ass and slender legs." : "Erin's suit lies in a
crumpled heap.">> "
   noun = 'suit' 'skirt'
   adjective = 'business' 'Erin\'s' 'nice' 'tight'
   isworn = true
   verDoWear(actor) =
   {
     if (self.isworn)
       "It's already being worn! ";
     else if (actor = Me)
       "That wouldn't look good on you! ";
   }
;
```

Since Erin's suit can be in several states, we need to worry about changing the description based upon those states. For our demonstration game, we will only worry about whether the clothing is worn or not (and ignore the case of it being worn but carried by Erin, or not worn but carefully folded, or other such things). This limits our description to two values - worn or not worn – which makes it an excellent candidate for some special TADS processing called embedded expressions.

An embedded expression can be a fairly complex piece of code, but we're using it in its simplest, and most common, form. Think of an embedded expression as a short-cut to writing an if-condition, similar to this:

```
if (condition)
 result if true
else
 result if false
```

The embedded expression takes all of that and turns it into this: <<condition ? result if true : result if false>>. The best part is that you can embed this within output text so that you don't have to "break out" of your output text in order to evaluate a condition and display some additional text.

You'll note that I refer to a property of Erin's suit called *isworn*. This is a property of all *clothingItem*-class objects that does just what it says: keeps track of whether or not the clothing is worn. When you say "wear suit", it changes this property to true, and when you say "remove suit" it changes it to false. This is handled automatically, and you needn't worry about it. It is very convenient, however, as you can see that I also use it to perform some verification. We want Erin, and only Erin, to be able to wear the suit. So, when the player issues a command to wear it, we check to see if it is already worn, and if it isn't we

check to see if it is it the player who is trying to wear it. If so, we give a nice error saying that the suit won't fit the player. If you had additional characters within your game, you would probably want to extend this check to cover them as well – unless the clothing was such that it fit other characters as well.

Now that we have clothing for Erin, we need to modify some other descriptions. For example, when she is wearing it, Erin will look different then when she is not. And so will her associated body parts. For our girl Erin, we previously created her breasts as a body part. Just for difference, let's create a different body part, her pussy, and alter the description based upon whether or not Erin is dressed. We'll also redirect any attempts by the player to fuck her pussy to the already defined ability to fuck Erin (as the two actions are synonymous):

```
ErinsPussy: fixeditem
      location = Erin
      sdesc = "Erin's pussy"
      adesc = self.sdesc
      thedesc = self.sdesc
      ldesc = "<<ErinsSuit.isworn ? "You can't see her pussy through her
      skirt. " : "Her pussy is shaved bare. ">> "
       noun = 'pussy' 'snatch' 'vagina'
      adjective = 'Erin\'s'
      doFuck -> Erin
  ;
```

Most of this should be stuff you've seen before, but there is something new here as well. Notice that the last line of her pussy definition is "doFuck -> Erin". These two characters, "->", tell TADS that when it is processing the "doFuck" directive (technically, the direct object action for the *fuckVerb* we previously defined), to redirect it to another object – in this case, Erin. It is functionally equivalent to these two lines:

```
      verDoFuck(actor) = {  Erin.verDoFuck(actor); }
      doFuck(actor) = { Erin.doFuck(actor); }
```

This is very convenient for things which are part of other things ("components" is the term used in the game authoring language), or for other cases when you want some action to redirect to another object. To demonstrate checking Erin's dressed condition before allowing sex, let's add in an additional sexual activity – giving Erin oral pleasure. For this, we'll include the syntax for normal interaction: 'lick'. And, to make things simple, we'll simply add this syntax to an existing TADS verb: *eatVerb*. We do this by using TADS modify on the existing verb:

```
   modify eatVerb
     verb = 'lick'
   ;
```

We've seen modify already when adding completely new code for existing objects, but here we're making a slightly different use of it. We are simply adding a new word to the verb list for an existing object, in this case *eatVerb*. TADS will simply add 'lick' to the existing list of words so that now we can use 'lick pussy' and TADS will interpret that to mean invoke the doEat directive on the pussy object. So, let's make a modification to the pussy object we created earlier to disallow this action unless Erin is both "in the mood" (much as we did for having sex with her) and that she is nude. If both are true, we'll write a description of giving Erin oral pleasure:

```
ErinsPussy: fixeditem
      location = Erin
      sdesc = "Erin's pussy"
      adesc = self.sdesc
      thedesc = self.sdesc
      ldesc = "<<ErinsSuit.isworn ? "You can't see her pussy through her
      skirt. " : "Her pussy is shaved bare. ">> "
      noun = 'pussy' 'snatch' 'vagina'
      adjective = 'Erin\'s'
      doFuck -> Erin
      verDoEat(actor) =
      {
            if (not Erin.isSexy)
                 "Not now! ";
```

```
                  else if (ErinsSuit.isworn)
                    "You'll have to remove her skirt first. ";
        }
        doEat(actor) =
        {
        "You lick her pussy. ";
        }
        ;
```

Obviously, in your own game the description of licking Erin's pussy would be much more titilating, but you get the idea.

The last thing we need to do is to take care of getting the suit off of Erin. By default, as Erin is wearing the suit the player will be unable to remove it. TADS assumes that any command to remove the suit (i.e., "remove suit") will be an attempt by the player to remove the suit from himself. We'll need to alter some of the verification routines used to prevent this in order to allow it. Just as putting on the suit is handled by "Wear", taking it off is handled by "Unwear" (which is a little odd grammatically, but that is how Mike Roberts [the author of TADS] coded it):

```
ErinsSuit: clothingItem
   location = Erin
   sdesc = "business suit"
   ldesc = "<<self.isworn ? "Erin is wearing a nice suit.  In particular, her
tight skirt shows off her tight ass and slender legs." : "Erin's suit lies in
a crumpled heap.">> "
   noun = 'suit' 'skirt'
   adjective = 'business' 'Erin\'s' 'nice' 'tight'
   isworn = true
   verDoWear(actor) =
   {
     if (self.isworn)
       "It's already being worn! ";
     else if (actor = Me)
       "That wouldn't look good on you! ";
   }
   verDoUnwear(actor) =
   {
     if (not self.isworn)
       "It's not being worn! ";
     else if (not Erin.isSexy)
       "Not now! ";
   }
   doUnwear(actor) =
   {
    if (actor = Me)
     "You help Erin strip off the business suit. ";
    else
     "Erin strips off the business suit. ";
    self.isworn := nil;
    self.moveInto(ErinsOffice);
   }
 ;
```

You'll notice that I also provided our own description of removing Erin's suit, as once again TADS assumes that any clothing item being removed is currently worn by the actor removing it. So, by default you get a description reading "Okay, you're no longer wearing the business suit." Writing our own description will also require us to alter the suit's isworn property as well as moving it into the office instead of the player's inventory.

Last but not least, let's tie up the whole game with an end scene. This is another choice of the author. I've done some games where the sex scenes continue on and on until the player finally get's tired of it and quits on his own. Others I've written where a specific sex scene causes the game to end. There are advantages to either method. For our demonstration game, let's cause the sex scene with Erin to finish the game, wrapping things up in a nice little bow. As with licking her, we'll also prevent the player from initiating sex until Erin is naked. This will involve altering the doFuck directive for the Erin object.

I'm not going to repeat the entire code for Erin, just the part we are changing:

```
        Erin: Actor
          code cut for clarity
         verDoFuck(actor) =
          {
             if (not self.isSexy)
                "Not now! ";
             else if (ErinsSuit.isworn)
                "You'll have to remove her skirt first. ";
          }
          doFuck(actor) =
          {
             "\"If you want those pictures back,\" you whisper, \"you'll have
             to let me fuck you.\"  You turn Erin around and force her over her
             desk.  She yelps in surprise and moans when you reach between her
             legs, discovering that she's not wearing panties.  You unbuckle your
             pants and slowly slide your cock into her dripping pussy.  \"You're
             such a naughty girl, fucking your coworker in your office.\" Erin
             just moans in response, gripping the desk and grinding her ass back
             at you.
             \b \"Oh yes, I'm your dirty girl!  Oh god, fuck me!\"  Her words fade into
        moans, her body writhing against you as you pound your cock deep into her,
        gripping the swell of her hips for leverage.  She arches her back, screaming
        as she climaxes.  Once she calms down from her orgasm, Erin turns around and
        expertly sucks your cock until your climax.";
             "\"Thanks for doing this for me,\" Erin says, as both of you
             straighten up.  \"I've always fantasized about being taken advantage
             of at work, and my boyfriend really wanted to see me fucked by
             another man.\"
             \b\"Wait, what?\" you ask, surprised.
             \b\"Yeah, you played your part perfectly.  Thanks!  See you
             Monday!\"  With that, she leaves.
                 \b* * * You have won! * * * \b";
                 scoreRank();
                 quit();
          }
        ;
```

Here we've let the player know that he has won, and showed him his current score using TADS built-in *scoreRank* function. Finally, we end the game using the built-in *quit* function. Of course, this ends the game rather abruptly, without giving the player the opportunity to UNDO or RESTART. Adding in these possibilities is something that can be done, but it involves some additional programming and testing of the player's input which is a little beyond the scope of this demonstration.

And that's the end of our little game. There are plenty of holes in it, and I encourage you to attempt to fill them in yourself using the techniques discussed within the "Programming Erin" feature. If you run into trouble, please remember that I, as well as the rest of the authors taking part in this feature, are always available to answer any questions.

ADRIFT Segment by BBBen

I'm going to diverge a little from what KE is talking about in his article this month, because stripping is a more complex issue for ADRIFT than many of our previous subjects. Generally speaking I'm actually going to recommend that you do not use the default clothing system for ADRIFT, as it's not really designed with AIF in mind. I don't have time for a full treatment of all the issues involved, so I'll tell you a basic way to deal with it (as I did in my early games) and I'll tell you where to start with a more advanced stripping system that will be the best way to go when you're making more sophisticated games.

First up you'll have to be aware that examining individual body parts (like "x tits") is not really supported in ADRIFT, so you'll have to do something about that. You could try to create objects to represent them, but a better way is to create a number of tasks called "x tits", "x ass", etc. For the simple system, however, we won't worry about this. Again I point out that we will NOT be using clothes as real objects for this stripping system. We'll just be describing them.

Now, the basic system isn't too hard; just create a task called something like "strip" or "remove erin's suit" or whatever. Next, in the character description box for Erin, write the description for her when she's clothed. There's another box below, so use the drop-down menu to select "But if task <remove erin's suit> is complete then show" and in the box below that, write the description for her when she's naked.

That's the simplest way to make a stripping system in ADRIFT, but it's a bit restricting and inadequate for advanced sex scenes, and if you're willing to devote a bit more time and effort to it you can do much better. I'll give you some idea where to start with the following advanced instructions.

Basically the advanced stripping system is going to be all about variables and the ALR file. I'm not going to explain how all that works right now; there are other articles out there on how to use the ALR file, so I'll just assume you have an idea how to use it. So, start out by setting up a variable called something like "clothing". Next you've got to decide how layered the clothing will be; I tend to go for one layer (i.e., either dressed or naked), but in theory you can use this system to be as complex as TADS stripping system (it's just a lot of work for a questionable amount of reward). Let's say a relatively simple three state, two-layer system – Erin can be wearing a suit, just wearing underwear, or naked. With the "clothing" variable, number 0 will represent being clothed, 1 will represent underwear and 2 will represent buck nakedness.

You'll need to have two different tasks, "remove suit" and "erin remove suit", and aside from giving different descriptions each of these will both do the same thing – change the "clothing" variable from 0 to 1. If you really wanted to, you could also have the task move an object to represent the suit onto the floor of the room, so that the player could see the clothes being removed and you would give the illusion of a more TADS-like stripping system, but I don't see the need. Then you do the same thing for another pair of tasks, "remove underwear" and "Erin remove underwear", and these tasks will move the "clothing" variable from 1 to 2. Remember to put in restrictions so that "remove suit" and "erin remove suit" will only work when "clothing" is at 0 (and do the same for "remove underwear" and "erin remove underwear", so that they will only work when "clothing" is at 1).

In the ALR file, you'll need lines that look something like this:

```
erindescription0|Erin is dressed in her suit.
erindescription1|Erin is wearing underwear.
erindescription2|Erin is naked.
erintits0|Erin's tits look big in her suit.
erintits1|Erin is wearing a bra.
erintits2|Erin's naked tits are great.
erinass0|Erin's suit makes her ass look good.
erinass1|Erin's panties make her ass look good.
erinass2|Erin's naked ass looks great.
erinpussy0|You can't see it.
erinpussy1|You can see the shape through her panties.
erinpussy2|Erin's naked pussy is wet.
```

…though obviously you'll want to go into more detail than that. Then in Erin's character description put the entry "erindescription%clothing%", and for the "x tits", "x ass" and "x pussy" tasks, put in the descriptions "erintits%clothing%", "erinass%clothing%" and "erinpussy%clothing%" respectively. We won't be using the "But if task <remove suit> is complete then show:" method at all. The description will only be changed in the ALR file.

The result will be that if the player types "remove suit" then examines Erin they will see the description of Erin in her underwear, and "x tits" will show the description of Erin in her bra. Since the same variable is used for all the descriptions, they will all change at once.

As you can probably see with this system it would be possible to have separate descriptions for removing a top and bottom and leaving the other parts on, but it would require you to use several variables instead of just one "clothing" variable, and it would make the overall description harder to do, as you would need several different ALR commands within the description of Erin alone, rather than just the one "erindescription%clothing%" entry.

Anyway, that is, as I said, just the beginning of an advanced system. You can get much more complicated, but in ADRIFT it will be a bit of work, and I just don't think it's worth it to have to go through all that every time. I would, however, recommend that you use some form of the 'advanced' system if you understand how it works, as it will allow for better interactivity with individual body parts and will also allow you to set the *player's* clothing states as well (which is a subject that I haven't addressed here, but it would use a similar system to what I just described).

Inform 6 Segment by 'trix

First on the agenda this months is clothing. Erin's clothing is going to comprise a suit, which makes things nice and easy. In a big game, if you have any inclination towards simulationism, you'd probably want to find or write a layered clothing extension, but in this case things are pretty straightforward.

In Erin's inventory, define an object to represent her clothing.

```
    Object erinSuit "suit" erin
      with owner erin,
           name 'suit' 'skirt',
           adj 'sexy' 'nice' 'tight',
           articles "Erin's" "Erin's" "a",
           description
           [;
              if (self in erin)
                 "Erin is wearing a nice suit. In particular,
                  her tight skirt shows off her tight ass
                  and slender legs.";
              "Erin's suit is in a crumpled heap.";
           ],
           before
           [;
             Wear: "That wouldn't look good on you.";
           ],
      has  clothing;
```

Note here that I'm making a simplifying assumption: Erin is wearing the suit if it is in her inventory, and no one is wearing it otherwise. In a complete clothing lib you would need to write some code to handle NPCs wearing items, since the default I6 behaviour only really deals with things being worn by the PC.

Now we have clothes, we can address the issue of whether Erin's anatomy is concealed or displayed.

```
    Pussy erinPussy erin
     with article "Erin's" "Erin's" "Erin's",
          adj 'shaved' 'shaven' 'bare',
          description
          [;
            if (erinSuit in erin)
                "You can't see her pussy through her skirt.";
            "Her pussy is shaved bare.";
          ],
          before
          [;
            Fuck:
              "~If you want those pictures back,~ you whisper,
               ~you'll have to let me fuck you.~^You turn Erin around
               and force her over the desk. She yelps in surprise,
               and moans when you reach between her legs, discovering
               that she's not wearing any panties. You unbuckle
               your pants and slowly slide your cock into her
               pussy.^~You're such a naughty girl,~ you tell her,
               ~fucking your coworker in your office.~
               ^In response, Erin moans and grips the desk,
               grinding her ass back at you.
               ^~Oh yes, I'm your dirty girl! Oh God, fuck me!~
               ^Her words fade back into moans, her body writhing
               against you as you pound your cock deep into her,
               gripping the swell of her hips for leverage. She
               arches her back, screaming as she climaxes.
               ^Once she calms down from her orgasm, Erin turns
```

```
                around and expertly sucks your cock until you
                climax.";
              Taste:
                if (erinSuit in erin)
                   "You'll have to take her suit off first.";
                "You lick her pussy.";
            ];
```

Since clothing in the standard I6 lib can only be worn by the PC, the standard "removing clothes action" (which is called Disrobe) can only be applied to things in the PC's inventory. To fix this we need to amend the verb grammar.

```
    Extend 'remove' replace
        * noun                                  -> Disrobe
        * multi                                 -> Take
        * multiinside 'from' noun               -> Remove;
    Extend 'shed' replace
        * noun                                  -> Disrobe;
    Extend 'take' replace
        * multi                                 -> Take
        * 'off' noun                            -> Disrobe
        * multiinside 'from' noun               -> Remove
        * multiinside 'off' noun                -> Remove
        * 'inventory'                           -> Inv;
```

All I've done above is copy from Grammar.h the grammar for each verb that refers to the `Disrobe` action, and change the noun tokens to plain noun.

Now we can intercept the `Disrobe` action on Erin's suit and write some appropriate behaviour.

```
    ! In the definition for Erin's suit
          before
          [;
            Wear: "That wouldn't look good on you.";
            Take, Remove: if (self in erin) <<Disrobe self>>;
                "You shouldn't walk off with Erin's suit.";
            Disrobe:
               if (self notin erin)
                  "No one is wearing that.";
               if (erin.check_sex()) rtrue;
               move self to real_location;
               "You slowly strip off her suit.";
          ],
```

Now, what about if the player wants Erin to remove her suit herself? Since this is the last month, and it's worth an article just by itself, I'm not going to go into the whole mechanism of programming NPC actions and allowing the player to give orders. But in summary: you give the NPC a property called `orders` and write the code in there.
Here is Erin's `orders` property.

```
      orders
      [;
        Disrobe, Drop:
           if (noun == erinSuit && erinSuit in self)
           {
              if (erin.check_sex()) rtrue;
              move erinSuit to real_location;
              "Erin slowly removes her suit.";
           }
      ],
```

A small wrinkle on the subject of clothes: the `GInv` lib which I wrote for this tutorial assumes the game is using clothing consistently, and remarks "You are naked" when the player takes inventory. Since we are not bothering to implement clothing for

the PC, this isn't suitable. So chuck out `GInv` for this project, and let's custom-write `InvSub` to do what we need for this game.

```
[ InvSub nh x;
   if (inventory_style == 0) return InvTallSub();
   inventory_style = inventory_style | CONCEAL_BIT;
   objectloop (x in player) if (x hasnt scenery)
   {
     give x ~concealed;
     ++nh;
#ifndef MANUAL_PRONOUNS;
      PronounNotice(x);
#endif;
    }
   if (nh==0)
   {
      L__M(##Inv, 1);
   }
   else
   {
      L__M(##Inv, 2);
      if (inventory_style & NEWLINE_BIT) L__M(##Inv, 3);
      else print (char)' ';
      WriteListFrom(child(player), inventory_style, 1);
      if (inventory_style & ENGLISH_BIT) L__M(##Inv, 4);
   }
   AfterRoutines();
];
```

This is pretty much the version in `GInv`, but with all the stuff about clothing removed, since it's not necessary in this game.

Also, since I'm tidying up things from previous months: my `TwoDoor` class my show some aberrant behaviour. The global variable `real_location` is supposed to be the current room, but it seems that this isn't updated soon enough to be used in `TwoDoor`'s `found_in` property. So if you're using that class, change it to check location instead of `real_location`.

ALSO, anyone who has found their object containment tree behaving in weird ways, it seems that the -> notation doesn't have quite the versatility I thought. Anyone who has found Erin's inventory wandering around the map will need to change their code so that the location is listed explicitly in the Object declaration (as in the declarations in this article).

The last fairly (but not completely) essential feature is an ending to the game. This is going to be triggered by fucking Erin, so we need to add some stuff into the Fuck interception for Erin's pussy. The way to end the game is to set the global variable `deadflag`. Setting it to 1 kills the PC and ends the game with the message "You have died". Setting it to 2 ends with the message "You have won".

```
Pussy erinPussy erin
 with article "Erin's" "Erin's" "Erin's",
     adj 'shaved' 'shaven' 'bare',
     description
     [;
       if (erinSuit in erin)
          "You can't see her pussy through her skirt.";
       "Her pussy is shaved bare.";
     ],
     before
     [;
       Fuck:
         print "~If you want those pictures back,~ you whisper,
         ~you'll have to let me fuck you.~^You turn Erin around
         and force her over the desk. She yelps in surprise,
         and moans when you reach between her legs, discovering
         that she's not wearing any panties. You unbuckle
```

```
                  your pants and slowly slide your cock into her
                  pussy.^~You're such a naughty girl,~ you tell her,
                  ~fucking your coworker in your office.~
                  ^In response, Erin moans and grips the desk,
                  grinding her ass back at you.
                  ^~Oh yes, I'm your dirty girl! Oh God, fuck me!~
                  ^Her words fade back into moans, her body writhing
                  against you as you pound your cock deep into her,
                  gripping the swell of her hips for leverage. She
                  arches her back, screaming as she climaxes.
                  ^Once she calms down from her orgasm, Erin turns
                  around and expertly sucks your cock until you
                  climax.";
                  deadflag = 2;
                  "^~Thanks for doing this for me,~ Erin says, as you
                  both straighten up. ~I've always fantasised about
                  being taken advantage of at work, and my boyfriend
                  really wanted to see me fucked by another man.~
                  ^~Wait, what?~ you ask, surprised.
                  ^~Yeah, you played your part perfectly. Thanks!
                  See you Monday!~
                  ^With that, she leaves.";

            Taste:
              if (erinSuit in erin)
                "You'll have to take her suit off first.";
              "You lick her pussy.";
          ];
```

Barring unforeseen problems, that's it. There are a few other additions in the full version of the game that I haven't described in the articles, mostly to make the game consistent with the other versions, but they should follow naturally from the rest of the code.

Happy new year!

Inform 7 Segment by Purple Dragon

I personally believe that clothing on women is a waste of time (did I say that out loud?) but since you often find women dressed in games as well as real life, here is how you might handle it. In reality, it would almost certainly be more complicated than this, but here we will give Erin only one piece of clothing, just as an example to test the basic concept.

```
    Erin's suit is worn by Erin.  The description is "[if Erin's suit is worn by
    erin]Erin is wearing a nice suit.  In particular, her tight skirt shows off
    her tight ass and slender legs[otherwise]Erin's suit is in a crumpled heap[end
    if]."
    Understand "skirt/blouse/shirt" as the suit.
```

Notice that we don't implicitly say that "Erin's suit" is clothing. Inform knows that only clothing items can be worn by people, so creating the suit this way kills two birds with one stone. Since you might expect the description of the suit to change depending on whether or not Erin is wearing it at the moment, we use a text substitution in the description to handle that. Finally, we give the suit a couple of synonyms. The description mentions a skirt, so it is quite possible that the player will try to refer to it that way.

```
    Erin's pussy is part of Erin.  The description is "[If Erin is wearing Erin's
    suit]You can't see her pussy through her skirt[otherwise]Her pussy is shaved
    bare[end if]."
```

As well as the description of the suit changing when worn or not, the description of Erin's body parts would as well. In this simple example, another basic text substitution will handle things fine. If you got into things like layered clothing and multiple outfits, you would probably need to come up with a better way.

You may have noticed that the condition here and the one in the suit description seem to be different. In the suit description we test for, "if Erin's suit is worn by Erin" and here, "If Erin is wearing Erin's suit." These two conditions are actually testing the exact same thing, and the only reason I made them different is to show you two different ways to word the same condition. Inform will often let you do things like this, choosing the wording for the rule that sounds better to you. Of course, there are also times when the wording needs to be annoyingly precise, so care should be shown.

The next thing is to allow the player to remove Erin's suit, but there are a couple of big problems here. The first is that the action, which here is called "Taking off" applies to items of clothing worn by the player. The player is obviously not wearing Erin's suit, so Inform will very helpfully try to get the players hands on it to see if it can salvage the action. This runs us into the second problem, which is that the player cannot normally take something that is carried (or worn) by another character.

Inform has several testing features built in. One of the most helpful is the "Actions" command. Simply type "actions" when testing the game to turn this feature on (note that this will automatically not work when you compile the game for release. It is for testing only). Now whenever a command is entered, Inform will run through which actions are being attempted and their result. So, with the actions feature on, here is what happens when we try to remove the suit.

```
>remove suit
[taking off Erin's suit]
(first taking Erin's suit)
[(1) taking Erin's suit - silently]
That seems to belong to Erin.
[(1) taking Erin's suit - silently - failed the can't take people's possessions rule]
[taking off Erin's suit - failed the carrying requirements rule]
```

What's going on here is pretty much what I just described. [taking off Erin's suit] is the action that is being attempted. Inform notices that the player doesn't have the suit, so it tries to take it. When you see a number like (1) it means that a separate action is being tried in the middle of the original one. That action fails, and it even very helpfully tells you why it failed (because you can't take other people's possessions). And finally, since after all that the player still does not have the suit, the original action fails as well. It might all look kind of complicated, but it is very helpful when something in your game isn't working right and you want to find out exactly where it's failing.

There are several ways to override this and make the command work. If there was going to be a lot of this type of thing in your game, you might want to actually re-write the rules themselves to allow the player to undress other people. However, since we only have a single instance here, the brute force method will work just fine.

```
        Before taking off erin's suit:
              If erin's suit is worn by erin begin;
                    If erin is ready for sex begin;
                          move erin's suit to the location;
                    say "You slowly strip off her suit." instead;
                    otherwise;
                          say "That's a good way to get fired." instead;
                    end if;
              end if.

        Instead of wearing erin's suit:
              say "That wouldn't look good on you."
```

Using a before rule let's us get in the first punch with this action. Before Inform gets around to checking whether or not the player has the suit, it first goes though our before rule, and the way we have written it insures that it will never make it any further. We check to make sure that Erin is ready for sex, and if so, we move the suit to the location (remember this is the location of the player, which just means it puts it into the room, not carried or worn by anyone) and print out what happens.

Note the word instead after the say command. This is very important, since if you didn't put that, the rule would run through to the end and then Inform would continue trying to carry out the action, which isn't what we want. The word instead makes sure that, once a true condition is reached, the action will stop. And finally, we put in a rule to make sure the player can't wear Erin's suit himself.

So that handles the player undressing Erin, but what if we want to let her undress herself? As it stands, if the player tried that, this is what he would get.

>Erin, remove suit
Erin has better things to do.

Which certainly isn't what we want. Actually, even if we didn't want Erin to be able to remove her own suit we would probably still want to rewrite that. In Inform, like in real life, getting someone to do something that you want them to do comes down to persuasion. By default, all characters will refuse to obey any commands given by the player. If we want something different, we have to step in and take action.

```
A persuasion rule for asking erin to try taking off the suit:
        Persuasion succeeds.
```

Here we give one exception to Erin obeying the players commands. If he asks her to take off the suit, she will now gladly oblige. Just by writing that, she will remove her suit just fine, but it doesn't really work how we want it to. First, when she removes the suit she will keep possession of it, which isn't terrible really, but when people strip down for sex, they rarely want their hands full of clothing.

Next, the response that's generated by this would be, "Erin takes off Erin's suit." which is a bit generic. These two are certainly less than ideal, but depending on the situation, you might be able to put up with it. However, the third problem is a big one. As it is now written, Erin will remove the suit at any time during the game. So when she and the player are in her office before he shows her the pictures, and she is yelling at him, demanding to know what the hell he is doing snooping around, the player types, "Erin, remove suit" and she happily strips down. Not what we want, to say the least.

The following is a bit more complicated than most of what we have done so far, and probably not, strictly speaking, all necessary, but it shows how to use and manipulate the "check," "Carry out," and "Report" rules that all actions have.

```
Check erin taking off the suit (this is the erin isn't ready rule):
        If erin is not ready for sex, stop the action.

Unsuccessful attempt by erin taking off the suit:
        if the reason the action failed is the erin isn't ready rule, say
"Suggesting that is likely to get you slapped or fired.";
        otherwise say "Erin can't do that at the moment."
```

This is the most complicated bit. Basically, what we are doing in the check rule is making sure that Erin is ready for sex, and stopping things in their tracks if she isn't. The words in the parentheses are not necessary, but it gives us a way to refer to this later. Normally, when a character is asked to do something that they can't do, you would just get something like "Erin is unable to do that." Not only is that generic, but in this case it's just plain wrong. It's not that she can't do it, but rather that she doesn't want to.

We can rewrite this text, which is why we needed to name that check rule earlier. In reality, we would probably also want to include some of the other reasons this rule could fail as well, but this will do for an example.

```
Carry out erin taking off the suit:
        Move the suit to the location.

Report erin taking off the suit:
        say "Keeping her eyes locked on yours, Erin slowly removes her clothes
and kicks them aside." instead.
```

Once we get past the check rules, the rest should be fairly self-explanatory. The carry out rule moves the suit to the location, and the report rule says what happened. Notice the word instead again. Without that, our text would print and then it would go ahead and print the default, "Erin takes off Erin's suit." which would look a bit funny.

By the way, I just noticed that I have taken to referring to Erin's suit as "the suit." This is fine since there is only one suit in the game, but if the player had a suit as well, you would need to be more explicit when writing these rules.

Now that she is naked and ready for sex, there's only one thing left to do. Oh yeah, you know what I'm talking about.

```
Instead of fucking erin's pussy:
        If Erin is not in the location begin;
                say "Erin isn't here at the moment you know.";
```

```
        otherwise if erin is not ready for sex;
            say "Without warming Erin up to the idea first, that's only likely
    to get you fired.";
        otherwise if erin is wearing the suit;
            say "It would be easier if she didn't have all those clothes on.";
        otherwise;
            say "'If you want those pictures back,' you whisper, 'you'll have
    to let me fuck you.'  You turn Erin around and force her over her desk.  She
    yelps in surprise and moans when you reach between her legs, discovering that
    she's not wearing panties.  You unbuckle your pants and slowly slide your cock
    into her dripping pussy.  'You're such a naughty girl, fucking your coworker
    in your office.' Erin just moans in response, gripping the desk and grinding
    her ass back at you.[paragraph break]'Oh yes, I'm your dirty girl!  Oh god,
    fuck me!'  Her words fade into moans, her body writhing against you as you
    pound your cock deep into her, gripping the swell of her hips for leverage.
    She arches her back, screaming as she climaxes.  Once she calms down from her
    orgasm, Erin turns around and expertly sucks your cock until your climax.
    [paragraph break]'Thanks for doing this for me,' Erin says, as both of you
    straighten up.  'I've always fantasized about being taken advantage of at
    work, and my boyfriend really wanted to see me fucked by another
    man.'[paragraph break]'Wait, what?' you ask, surprised.[paragraph break]'Yeah,
    you played your part perfectly.  Thanks!  See you Monday!'  With that, she
    leaves.";

        End the game in victory;
        End if.
```

This is the same command that we did last month with two exceptions. First, I added in a check to make sure that Erin is naked. And second, we have decided that fucking her is the ultimate goal, and will win the game. It's simple enough to do, just add in "end the game in victory" within the rule. That's all there is to it.

And that, as they say, is that. It's still not much of a game, but hopefully you have learned a thing or two and enjoyed the opportunity to see how different platforms handle the same basic actions. Thanks for reading, and good luck on your upcoming game. ◆

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.

Submitting your work to Inside Erin:

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

AIF
Wants
You

Editor:
Purple Dragon has written six AIF games including *Archie's Birthday - Chapter 1: Reggie's Gift, A Dream Come True*, and *Time in the Dark.* He has received one Erin award and been nominated for several others.

Staff:
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. He was the co-recipient of the Badman Memorial Lifetime Achievement Award in 2006.

Staff

A Ninny is an AIF player, author of four AIF games and frequent beta-tester. His *Parlour* received an Erin for Best "One Night Stand" game in 2004 and his most recent game, *HORSE* walked away with three Erins at the 2007 awards show.

BBBen is an author of a number of Adrift AIF games. His games have received numerous Erin awards and nominations and first place in A. Bomire's 2004 mini-comp. He was also the recipient of the 2007 Badman Memorial Lifetime Achievement Award.

Bitterfrost is a longtime IF/AIF player working on his first (and last) game, *How I Got Syphilix*.

Knight Errant is an AIF player who has released one game and is currently working on a couple of others.

'trix has released one game, *Casting*, which was written in Inform 6, and is sporadically working on another in TADS 3.