



Contents

A Letter From the Editor	1
This Month in AIF	1
This Month at TF Games	2
This Month at the Collective	2
2009 AIF Mini-comp Results	4
Mini-comp Reviews	5
Coder's Corner	14

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.

I realize that for many of you, July 4th is just another day on the calendar. However, since the editor, several of the staff, and about half the readership of this newsletter are from the US of A, I thought that I would indulge my sense of patriotism a bit.

For many Americans, the fourth of July seems to be just another opportunity to get drunk and make a lot of noise (as if they need an excuse). I'm afraid that many people miss the point of what this day symbolizes. Words like freedom and patriotism are thrown around an awful lot. When they come from the mouth of an elected official (or someone who hopes to be one soon) they are often seen as some kind of act – as calculated, empty words designed to acquire votes or public approval. It is a shame that they are seen that way, and even more of a shame that, all too often, that view is closer to the truth than not.

Freedom is not a gimmick to achieve personal gain, and its purpose is certainly not to ensure a person's right to stay up until three o'clock in the morning shooting off fireworks. I would bet that even many of you who do not live in this country have heard the following words at one time or another.

Continued on page 3



The main event this month was, obviously, the mini-comp. There were six entrants, which is certainly down on previous years but still a good turn-out (especially considering how generally quiet things have been lately). Rip_CPU entered twice, with one of his games being slightly outside the rules (ironically it was called *The Prosperity of Cheaters*), and I'll be interested to see how that is reflected in the voting.



Occasionally there are themes that crop up totally coincidentally in a mini-comp, and if there are any this time then I suppose they would be Superheroes/villains and the breaking of wills through sex, which both appeared in *Obedience*, incidentally. Play the games, you'll see what I mean.

Otherwise the month was generally quiet, with talk still ongoing about SD3, and a little bit of interesting research as to whether there was a correlation between activity on the AIF Archive and the level of turnout for the mini-comp. There wasn't.

New games

Sex is Mental by Rabbi. Released June 10th 2009 for ADRIFT 4.
BSG: Twenty Two by Madquest8. Released June 10th 2009 for ADRIFT 4.
The Mechano-Menace by A. Bomire. Released June 10th 2009 for TADS 2.
Obedience by Rip_CPU. Released June 10th 2009 for Inform 7.
The Prosperity of Cheaters by Rip_CPU. Released June 10th 2009 for Inform 7.
In Darkness by Goblinboy. Released June 10th 2009 for TADS 2. ♦

In at least one way I'm much like the Graffiti artist Banksy. No really. You could walk past both of us in the street and never know or realize who we were. And at heart we're both artists, no not in the pretentious way, just in that we produce something that (hopefully) people enjoy.

But despite the rather strange position of being both know and unknown because of the internet I suffer from the curse of all artist.

I have an Ego!

Anyone who shares anything with the rest of the world, be it virtual or not, must have some confidence that what they produce is good enough for other people to enjoy. On some level they must believe that their stuff is a bit good. This means they at heart want people to say that they like their work. Now there are two ways that this works. The more pessimistic and introverted of us sit back and hope that someone will say they like the work. If they do well great, if not well the work wasn't good enough anyway.

The more extroverted among us however look for column inches, if they don't get any comments they need to know why. They never consider that people are happy, but quiet, with the game. Silence means that they're not good enough and they need to know why.

Traditionally at this point a column would then take the option over which one is the better style. And then try to lecture the other over how to make themselves better. Again, like Banksy, I've never held to following tradition as I can see that both have there own drawbacks. The Introvert has the possibility of being ignored, there games lost among others that are sold better. And the Extrovert can alienate there public by constantly asking, even demanding, more and more attention.

Ironically not ending traditionally I can't end with a nice neat Oprah or Springer sound bite. But being on the introvert / pessimist side it does mean I can babble on in this column month by month with no feedback. For all I know everyone skips this section for the more interesting stuff. With only me and the editor even reading what I've written.

Continued on page 3

Summertime! Yay!

Well... Mostly yay, but game-wise the Collective slowed down last month some...

TinaB posted that she is working on her *House of the Future* game. She has added some sound effects, music and other things, but does not have a release date as yet.

VVrayven, a really wonderful manipper on the Collective posted this on the game she is working on:

"Just a little update: still progressing with the game I'm now simply calling Resist. It's both larger and smaller in scope than I had intended. I'm making a "framework" that I can plug future scripts into - with the idea that a second game would be easier. But this first one is HARD!"

NandiBear and WightWashed posted an update on their game called *The House That Jack Built*. They noted that, about 80% of part 3 been put together, but NadniBear and WW have been comparing notes and making sure everything works the way they want it. No release date as yet posted.

Siriggles posted an update on his game *S.U.B. Mission Spa and Resort* that he is working on it again and is expecting to release a new version of the game sometime in August of 2009.

Benmbdlam posted an update on his game *Rough Landing* that *Rough Landing* as of version 1.3 he considers done and completed. *Rough Landing 2* is going to be a sequel set three years later, leading on from the four potential endings of the first game. Release

Continued on page 3

This Month At

TGames
FSite



COLLECTIVELY MADE...

**This Month
at the
Collective...**

By TeraS



LFTE, continued from page 1

“We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness.”

These words from the Declaration of Independence are perhaps some of the best known words in the English language. They do not talk about the rights of individuals to have riches or power. In fact, they do not talk about the rights of the individual at all. They talk about “all men” having the very basic rights of living, being free, and at least striving for happiness. If something an individual does or says puts those rights in jeopardy for anyone, then that is not freedom, that’s its black sheep cousin. The one that everyone tries to exclude from family reunions, but who nonetheless seems to keep showing up anyway. Abraham Lincoln said, “Those who deny freedom to others deserve it not for themselves.” and truer words have never been spoken.

In this country we enjoy a tremendous amount of personal freedom. Some would no doubt disagree with that, but compared with many countries in the world it certainly seems true to me. Without the freedoms we enjoy there would be no *Inside Erin*, no AIF Mini-comp, no AIF games at all. Many people in this country would call what we do here obscene. Many, many others would not find it so, but would still call it juvenile and pointless. But no matter what they call it, I think that most people (I don’t say all) would defend our right to be here. Maybe that’s what freedom really is. Defending someone’s right to do something even when you don’t personally agree with it.

I know that this letter is way off topic, and I apologize for that – especially to those of you who don’t live in the United States, who have still managed to read through this whole thing. However, don’t say that none of this applies to you just because you live in a different country. I don’t see any mention of the word “American” in the quote above. That “all men” should apply to anyone reading this. We are a multi-national community, but we are all just men and women, and the freedoms I speak of need to apply to everyone, or they don’t really apply to anyone.

Thanks for indulging me. I promise not to do this too often. God bless you all, and God bless America. ♦

TF Games, continued from page 2

Oh well, ever onwards. This month has been rather quite in the TFGamesite forums, as those of us still in school finish all those important exams and essay. So whilst people like Gunny and CryOfPain tease us with games to come we’ve had to entertain ourselves with only a few, excellent as they are, games.

On the RPGmaker’s side, we have the only game I know where the hero is (briefly) a chicken Pojo’s *Quest for a Cure*.

Whilst on the RAGS side we dipped into an experimental idea of Fuearye shared world idea of *TF Wars*. Where twenty people take turns to transform each other on the forums, only to see there results in an ongoing RAGS game.

And finally on the Hypnopic cross over stakes we have another update on Orca’s *Demon Town* (RAGS).

So another month, another column. Please, if you do have any comments, feel free to contact me. ♦

Collective, continued from page 2

Blau is working on a Harem Collection game and is looking for ideas. You can help by visiting this thread: <http://hypnopics-collective.net/viewtopic.php?f=11&t=15636>

Dromdri posted a new game called *For Fear of Little Men*. It is a short game which can be found at this link: <http://hypnopics-collective.net/cpg132/displayimage.php?pos=-94199> And the thread to comment on or help in the game’s development is: <http://hypnopics-collective.net/viewtopic.php?f=11&t=15809>

And finally for this month, Orcha posted a new version of *Demon Town* which you can find here: <http://hypnopics-collective.net/cpg132/displayimage.php?pos=-86846> And bug reports and comments can be placed here: <http://hypnopics-collective.net/viewtopic.php?f=11&t=14673>

And that’s June at the Collective! Have a great summer!

Tera ♦

Here are the results for the 2009 AIF Mini-comp. Congratulations to GoblinBoy for taking the number one position. It was a fairly close race overall, especially between 1st and 2nd place where just a couple of votes made the difference in several of the categories. You will also notice that there was actually a tie for 4th place so if you're one of those people who think that your vote doesn't matter, think again.

I'm sure that you will all join me in thanking all of our participants in this year's comp for some really interesting games. Remember that it is not too late to send some feedback to the authors telling them what you liked (or even what you didn't) about their games. It is really the only payment they receive for their efforts, and in some cases it can mean the difference between them writing another game or not.



Overall Finish Results

1. *In Darkness*
2. *The Mechano-Menace*
3. *Obedience*
4. *BSG: Twenty Two* (Tie for 4th)
4. *Prosperity of Cheaters* (Tie for 4th)
6. *Sex is Mental*

<p>Category: Concept</p> <ol style="list-style-type: none"> 1. <i>In Darkness</i> 2. <i>The Mechano-Menace</i> 3. <i>Prosperity of Cheaters</i> 4. <i>BSG: Twenty Two</i> 5. <i>Obedience</i> 6. <i>Sex is Mental</i> 	<p>Category: Writing</p> <ol style="list-style-type: none"> 1. <i>In Darkness</i> 2. <i>The Mechano-Menace</i> 3. <i>Obedience</i> 4. <i>Prosperity of Cheaters</i> 5. <i>BSG: Twenty Two</i> 6. <i>Sex is Mental</i>
<p>Category: Characters</p> <ol style="list-style-type: none"> 1. <i>In Darkness</i> 2. <i>The Mechano-Menace</i> 3. <i>Obedience</i> 4. <i>BSG: Twenty Two</i> 5. <i>Prosperity of Cheaters</i> 6. <i>Sex is Mental</i> 	<p>Category: Technical</p> <ol style="list-style-type: none"> 1. <i>In Darkness</i> 2. <i>The Mechano-Menace</i> 3. <i>Obedience</i> 4. <i>BSG: Twenty Two</i> 5. <i>Prosperity of Cheaters</i> 6. <i>Sex is Mental</i>
<p>Category: Sex</p> <ol style="list-style-type: none"> 1. <i>The Mechano-Menace</i> 2. <i>In Darkness</i> 3. <i>Obedience</i> 4. <i>BSG: Twenty Two</i> 5. <i>Prosperity of Cheaters</i> 6. <i>Sex is Mental</i> 	<p>Category: Enjoyment</p> <ol style="list-style-type: none"> 1. <i>The Mechano-Menace</i> 2. <i>In Darkness</i> 3. <i>Obedience</i> 4. <i>Prosperity of Cheaters</i> 5. <i>BSG: Twenty Two</i> 6. <i>Sex is Mental</i>

Mini-comp Reviews by A. Ninny

Another year, another mini-comp. I'm happy to see the community still supporting this event. This comp received six entries, which is a good number because otherwise I might not be able to write such detailed and well-considered reviews of all of them. So without further delay, let's get right to these reviews.

BSG: Twenty-two by Madquest8

Overview: This game has a science fiction setting; you play a ruthless space warrior. Humanity is at war with a robotic alien race and your warship has had a bomb concealed on it by a particularly sexy alien lady robot, whom your crew has captured. It's your job to convince the alien to reveal the location of the bomb before it explodes. You have only twenty-two turns.

Review: This is the kind of game that's got such a cheesy story, if it seemed like it was taking itself too seriously it would fall flat. Instead, you get just enough backstory to know your mission and off you go. The whole setup - you're in heightened suspense for 22 turns before the game ends - makes the author's terseness an asset to the feeling of the game. If you waste a lot of time looking for backstory, you'll die. If you spend a lot of time looking at the scenery, you'll die. So best to get straight to the point. It's also a game that you're going to have to play repeatedly before you solve, and so were I the author I'd want to make sure to offer enough variety in actions that player won't get bored. I thought the author got part of the way there on that point. There are a fairly wide variety of options, but not different descriptions for similar actions. I found the sex writing to be pretty much comic book goofy, but that didn't bother me too much given the hokiness of the story. I appreciated that the characters stay consistent to their roles and don't become generic, not that that's too difficult considering how cartoony they are. I didn't really buy the author's choice as to what the solution would be. It felt like he was trying to use something that players wouldn't hit upon in their normal AIF command repertoire, but even though I figured it out relatively quickly, it left me scratching my head as to just why that worked.

Concept: B-. I guess I'm a sucker for goofy Sci-fi stories. This qualifies.

Writing: B. As soon as I saw that this was a space opera, my expectations dropped, but the writing was actually pretty good.

Characters: B-. Both of the characters are cartoonish and flat, but still fun.

Hotness: C. While the sex scene is in character with the rest of the story, it isn't one that did much for me.

Technical: C. There are some bugs, but overall the game works pretty well.

Enjoyment: C+. I think it could have been better if the solution to the puzzle made more sense.

In Darkness by GoblinBoy

Overview: It's hard to describe this game without giving too much away. You start the game with no knowledge about the PC whatsoever and slowly, over several small episodes, the game reveals itself and develops into a horror/mystery story.

Review: GoblinBoy, as usual, has given us a fully realized story, fabulously detailed sex, and realized deliciously complete technical achievement. The story is really frightening, and the PC's confusion and fear are palpable from the first moment of the game. Structurally, the game is set up as a series of flashbacks. You get transported to various points in the PC's past (though not his memory - he has none) in order to learn something about who he is and how he got where he is now. I found it a very immersive way to tell the story. I also felt like the game treated me fairly. I never felt like GoblinBoy was cynically saying 'I know something you don't!' I thought the PC was truly just as in the dark as I was. About halfway through I guessed the answer, and felt like the PC probably did as well - at that point it was more like being drawn through the story by frightening forces unseen. There are two separate sex scenes in the game, both ably written, though not without GoblinBoy's perennial mention of de-virginification that peppers all his games, and, not without the phrase 'anal deflowering' which I can always do without. In his favor, he managed to write a detailed and highly explicit sex scene that really kept the game's confused and frightening tone consistently running through it, and that extra layer of emotion made it that much more hot.

Concept: A. A deeply frightening mystery story, wonderfully executed.

Writing: A. Excellent storytelling.

Characters: B+. The PC is a total enigma, but the NPC's step up and fill in nicely.

Hotness: A-. Great characterization in the sex scenes, and I didn't have my normal feeling with GB games that I'm watching the



kind of porn where the cameras are too close and the lights are too bright.

Technical: A. I found no bugs, and the game was technically ambitious.

Enjoyment: A. A game with rich depth and great suspense, and, almost unnecessarily, pictures.

Obedience by Rip_CPU

Overview: One of two superhero-themed games in this mini-comp; this time you play the supervillain. You've captured a superheroine and have her in your lair. What fun do you think you could have trying to break her?

Review: Like in *In Darkness*, *Obedience* begins with you having no knowledge of your true identity. Unlike *In Darkness*, though, the PC knows exactly who *he* is. So there's no reason - other than to toss in a useless puzzle - for me not to be told right off the bat who I am, who I've captured, and what I'm about to do. Once I sort that I'm a supervillain, there's nothing in my way of getting it on with the superheroine I've captured. Not that she's willing, of course. She gets into it (don't they always?), but essentially this is a rape fantasy.

I had some thoughts on improving this game, the most obvious would be to have another superhero or superheroine that I had to defeat in order to get at my quarry. Another would be to somehow corrupt the superheroine to come to my side and conquer the world (and, as a bonus, initiate action in the sex scene).

Concept: C-. This game is simplistic to the point of being meaningless.

Writing: B-. I didn't have as big a problem with the quality of the writing as the story itself.

Characters: C-. The characters were really generic. Next time, create a bigger difference between the superheroine and the villain.

Hotness: C-. The non-consensual aspect and the lack of emotional quality mostly did this in for me.

Technical: C. Some opportunities to improve the tech in this game were missed. For instance, give the player the opportunity to manipulate the apparatus restraining the girl.

Enjoyment: C. This game simply lacks the impact to make it very enjoyable.

The Mechano-Menace! by A. Bomire

Overview: The other superhero-themed game. In this one you play Aegis, a superhero from ___. Along with the superheroine Marvela, you are hot on the tail of MechanoMan, an evil genius who commands an army of robotic hell hares to try to vanquish you.

Review: This is actually a pretty cool adventure game condensed into mini-comp size, with a hot sex scene to boot. It has a combat system built in - you have two attributes that you track as you battle the evil MechanoMan in his secret hideaway lair: health and shield. If your health falls to 0, you lose the game. Using shields too much drains you so your moves are less effective. This all seems to work quite well, and didn't confuse or hold me up. As to the characters, I'm not familiar with the comic books that they are drawn from, so I can't say how well they conform to their sources, but Bomire has graciously provided background on them within a 'help' menu, and the characters don't behave contrary to these descriptions. From the description, it appears that Marvela doesn't actually have any superhuman powers; all her abilities are gained from training. Since she doesn't have powers, when it comes to the sex scene, an opportunity for superhuman sex is mostly lost, and in the end, ignored. Aegis does have super strength, and Bomire does mention that he uses it to lift her, but I didn't feel like she had much of a reaction. Perhaps she should have emphasized how much she appreciates being fucked by super-strong dudes and that could have been a focus for making the sex scene more in character with the rest of the game.

Concept: A-. A fun little comic book adventure that comes to life nicely.

Writing: A-. A. Bomire has a great ability to use his writing to create lively settings and characters, and this game is no different.

Characters: B. The characters are written with a definite slant toward humor, and it works well during the battle scene, but less so during the sex.

Hotness: B: The sex writing is immersive and detailed, but I felt it didn't take advantage of the characters well enough.

Technical: A-. This game was well put together, not as ambitious as his WWE game, but it certainly works cleanly and the battle scene is intuitively simple.

Enjoyment: B+. Definitely a fun game, but I wished the sex scene had been a little less ordinary.

***Sex is Mental* by Rabbi**

Overview: You play a patient in a mental institution. You're hot for a nurse and are about to get a new crazy roommate. You need to somehow use the roommate's craziness to get in the nurse's good graces.

Review: I had to suspend my disbelief in a big way just to get into the concept of this game. Usually I don't have a problem with that – I'm good at immersing myself in even a simple game's world, but in *Sex is Mental* I was asked over and over to believe more and more unlikely and outlandish concepts as the game progressed. I'm a patient in a mental hospital and I have the hots for a nurse. I can easily buy that. I get a new roommate who I manage to learn goes bonkers when he hears the word 'taxi'. Well, ok... keep going... Of course, I say 'taxi' and end up in the infirmary with said nurse who is suddenly eager to put out. Riiight, I think you lost me along the way there, sonny. Sorry. There are just too many improbable things going on. Then some weird thing happens at the end that is so poorly explained that I can't figure it out. At least I made it to the end. The whole game is just tossed together from a shake of this and a dab of that, and the characters don't offer much more conviction that they know why they're there, either. Add to these problems the fact that the ADRIFT coding is full of holes and that the spelling and grammar are pretty miserable, and you have a rather unsatisfactory experience.

Concept: D. If this game had limited the oddball ideas to just one, it could have been successful. It just asked me to accept too many.

Writing: C. I appreciated that there were a few humorous moments in the game that helped salvage the writing, but the concept dragged it down.

Characters: D. None of them came to life or seemed realized.

Sex: D. Unfortunately, the sex is very clipped. Precious little redemption here.

Technical: D. The game is buggy. There are a few guess-the-verb issues and lots of grammatical glitches.

Enjoyment: D. Not very enjoyable.

***Prosperity of Cheaters* by Rip_CPU.**

Overview: You've got the hots for a girl, but she's got a boyfriend. You have a girlfriend, so if you want to score you're going to have to cheat and convince her to cheat as well. Now's your chance: it's the wildest party of the year.

Review: I'm going to review this game as it is, ignoring for the moment that it violated the mini-comp rules by using three non-player characters, instead of the maximum two. I penalized it for this when I voted; no need to penalize it twice.

This is a nice little game, not too ambitious, but it works well in an old-fashioned mini-AIF kind of way: a simple setting, a straightforward goal, an endlessly repeating script to give you plenty of time to figure out the solution, and a puzzle that isn't too taxing, but one that has six different outcomes. If this weren't a mini-comp game, instead of two sex endings and four non-sex endings, you could have different pairings and even orgies. Maybe Rip_CPU will go back and add these later – it would make for a fun little T&AIF. The game doesn't concern itself too much with characters or background. The PC expresses his feelings in a basic way about the boyfriend of the target girl (he's a jerk, of course), his own girlfriend (gotta get her out of the way somehow), and the desired girl (hottie), but that's all the depth we get on them. Still, even though we don't get much characterization, it works with the casual sex college kid attitude that permeates the game. The sex writing is decent; there are two different sex scenes, and they're quite different from one another. I think it's worth your while to find both.

Concept: B. It's a simple idea, executed reasonably well.

Writing: B. Concise, simple to follow, writing that doesn't challenge, but still provides enough detail to carry the story. The writing could have been improved if the party atmosphere was noisier – the game has scripts constantly running, why not add some party activity in the background?

Characters: C+. They're certainly not profound, but there's enough here to work with.

Hotness: B. Two different sex scenes, nicely detailed, with several different responses for each action.

Technical: C. I found quite a few bugs, commands that gave no responses, objects that didn't disambiguate properly, and overall I think the implementation could have been more robust.

Enjoyment: B-. Again, I found this to be a simple concept built into a fun little game.

Mini-comp Reviews by ExLibris

Games are listed in the order that I played them, and some of the comments reflect that.

BSG: Twenty Two

Through no fault of its own, *BSG* starts with a few strikes against it simply because of my own my own subjective prejudices. Firstly, as a rule I don't like regular fan-fiction very much, but after playing this game I realised that it's preferable to fan-fiction with the names slightly changed.

Secondly, I'm not too keen on games with time limits, especially when the time is advanced by nearly any action, even something as innocuous as examining an object. Generally that means the annoyance of having to undo a lot, but in this case the annoyance was increased by the fact that undo was disabled. I can sympathise with the desire to create tension, but when it comes to not being able to undo after accidentally wasting a turn through a typo, it's a bit much.

The time limit is an indirect cause of my third quibble, the huge infodump at the beginning. A lot of the stuff in the infodump could have quite happily been placed in a readme file (*BSG* was also the only game without a readme file, which I'm afraid is another strike against it). As far as the story related information goes, I think there are better ways to provide it. *Obedience* was a good example of this, as it drip-fed the required information, allowing the player to slowly submerge into the character, instead of just dumping everything on the player in one load. The load in this case felt particularly heavy as well, not so much nudging the player in the right direction as leading them by the hand.

My prejudices against the game were increased by the fact that Tricia's dialogue was in a different colour to the rest of the text. Normally I'd think that was a good idea but, like probably 50% of the population, I had white set as my background colour. Consequently, making the dialogue yellow was not an inspired choice unless the aim was eyestrain. Changing the colour was easy enough, but it was just one more little annoyance.

Finally, run on sentences are bad, m'kay. Splicing them with commas or chaining them together with ellipses doesn't improve readability either.

The reason I mention all of this, is that I really had to force myself to play this game through to completion. And I'm glad I did as there is a surprising amount of content concealed under that rather unappealing exterior. Tricia is certainly one of the better implemented NPCs in the competition, with a wide variety of responses to both conversation and especially actions. Also, while the execution didn't appeal to me in a lot of ways, I felt that the basic concept was a good one.

Overall, I'd classify this game as something of an ugly duckling. It doesn't quite grow up into a beautiful swan when you get into it, but there are certainly some features of interest.

Prosperity of Cheaters

It says something for the prevailing themes of this minicomp that despite casting the player in the role of someone who happily drugs people and cheats on his girlfriend, *Cheaters* was actually one of the 'lighter' games.

I think it's pretty safe to say that of the two games the author contributed to the minicomp, this is the one that was finished second. Compared to *Obedience*, it feels unfinished and lacks the same degree of polish. Objects mentioned in room descriptions aren't always implemented (a pet peeve of mine) and the sex scene (or at least the one I've found so far) feels fairly perfunctory. There were also a few bugs, mainly actions that could be repeated when they shouldn't (e.g. talking to Beth or using the phone).

Cheaters does breach the rules as far as characters go, and I've reflected that in where I've ranked it. But I think I would have ranked it quite low in any case, as characterisation is one of the areas where the game is lacking. There's no ask/tell system, which wouldn't be so bad except that talking to characters rarely produces anything meaningful either. The only real source of characterisation is a few snippets of conversation that you overhear while standing about in the main room. Of course sometimes a little is enough, and hearing their respective opinions on Twilight was certainly enough to make me want to pick Beth over Kirsten.

A consequence of the limited characterisation is that there is no real relationship built up between the PC and Bethany. The PC

is told that she's hot, but that's about it. If it wasn't for the Twilight thing she and Kirsten would be virtually indistinguishable. Beth really needed more spotlight time than she received to sell her to the player as a desirable lust object. As it was, when the sex scene made its appearance, Beth moved from being 'girl you've just met' to 'girl you're having sex with' very abruptly and unconvincingly. From memory the conversation ran something like this: PC: "Oh dear, my girlfriend and your boyfriend are having sex." Beth: "Yes, that is a shame. Perhaps we should have sex also?" PC: "Okay". I'm exaggerating a bit, but that's the overall impression I have of it.

As mentioned above, the sex scene is comparatively brief and one of the weaker entries in the competition. The lack of characterisation led to a sex scene that had no real power attached to it. If it had been written differently it could have been carried by simple physical lust, but that didn't happen either and the scene felt like something of a dud.

Overall, I'd classify this game as a might have been. Having multiple endings is always a plus, and I feel that the basic concept was a good one, though not necessarily for a minicomp game. It also needed more time and polish to properly realise that concept.

In Darkness

Wow, what a downer. Although looking at Goblinboy's other games, delivering a happy ending doesn't seem to be very high on his list of priorities. Actually, calling *In Darkness* a game is a bit of a stretch. There aren't any real choices to make or puzzles to solve. The player merely has to work out what action will trigger the next 'page' of the story, making it interactive fiction in the most literal sense. It's somewhat subjective, but I think that the storyline would have worked better as prose rather than a 'game'. IF has a much closer identification of reader with protagonist than traditional fiction does. Consequently as a player I enjoy games where I feel in charge of the PC's destiny, rather than merely trudging towards a depressing and predestined fate without any hope of reprieve. While playing *In Darkness* you don't even get the chance to struggle against the inevitable. The decisions the player makes don't have any effect on the outcome, things just happen the way they happened. It didn't help that it was fairly obvious how the game was going to end from quite early on, which cast a shadow over what would normally be the fun parts of AIF. I'm not completely opposed to telling a story like this in IF, but because of the way it was executed I didn't find it particularly enjoyable. On the other hand you do get to kill Mike... er I mean Jeff, so it's not all bad.

In technical terms, it's what we've come to expect from a Goblinboy game. That would normally be high praise, but after his nth well executed game it doesn't get him as much credit as it might. Actually, something that probably decreased my enjoyment is how similar *In Darkness* is to previous Goblinboy games, primarily thematically but also in some specific ways. As I've already implied, Jeff is a similar character to Mike from the *School Dreams* series. Additionally, the basic premise is quite similar to *Ghost* and the downer ending is reminiscent of *The Casabian Virus*. The difference here is that as the protagonist I felt restricted, even more so than in *The Casabian Virus* or *The Key to Eternity* where the ending is predestined and the opposite of what the PC wants. When playing *In Darkness* I wasn't really conscious of being able to push against the flow of the story at all. I felt more like a spectator than a participant.

One final nitpick. Although I'm no expert in American law, it doesn't seem that likely to me that the PC would get the chair for what are two textbook cases of voluntary manslaughter. More dramatic, yes. More accurate, no. Oh, and the electric chair isn't actually used very much any more either. Suicide would have been a believable alternative ending, which probably sums up how depressing I found this game.

Obedience

First things first. The comics geek in me wants to point out that Ben Urich was the author of 'Legacy of Evil', not Ben Ulrich. Sorry about that.

Obedience was one of two superhero themed games in this year's minicomp, though it takes the opposite tack to *The Mechano-Menace!*. Coincidentally it also has a very similar basic premise to *BSG 22*, but I think it managed to execute it a lot better. In particular, I found the way in which information about the PC and his aims is slowly drip-fed to the player to be very effective, and a prime example of the virtues of showing rather than telling. The subject matter isn't something that would immediately appeal to me, but this introduction allowed me to slowly submerge myself in the character. This was something that didn't happen with *BSG 22*'s abrupt infodump, and consequently I was more resistant to that game.

Having so many items hidden under other items did get a bit repetitive, but there were plenty of warnings about that so I'm not going to complain too much. Objects mentioned in the room descriptions were also consistently implemented in the game, which

is always nice to see. *Obedience* also delivered one of the more satisfying puzzles of the competition, in the shape of finding your way into the PC's secret lair.

There wasn't a good deal of obvious characterisation, but what there was went a long way. Due to the way in which the PC was characterised I was never in doubt about his motivations or desires. The character of Polaris was a little flat, but given her role in the game that was perfectly sufficient. The premise of the game also gives the player concrete reasons for wanting her (reasons other than just the fact that she's hot, I mean), rather than just letting the relationship between the principals fend for itself, as seemed to happen in *Prosperity of Cheaters* or *The Mechano-Menace!*

The sex scene was among the better written, and was possibly the best implemented as well. The addition of the PC's 'tools' gave it that x-factor required to distinguish it from the rest of the pack. The many different possible variations and outcomes added extra interest as well, and kept the scene from becoming mechanical or predictable.

Overall, I think *Obedience* is a good game rather than an outstanding one, but it's a game that is consistently good in every area. The only thing I can really say against it is that it had a few too many typos.

Sex Is Mental

Somewhat disappointingly, this was the only game by a new author in this year's minicomp. Rather more disappointingly, it also played very much like a first game as well, with a number of rookie mistakes on display.

As an example, on my first playthrough I got as far as the ward before giving up. The problem was that the description of the room failed to mention any of the objects within it. I did correctly guess that there was a bed, which was described when examined (unlike either of the beds in the first room). The object that is needed to progress from this point is briefly mentioned in a cut scene, but unless you pick up on that clue and investigate you'll be a bit stuck. Like I was.

The second barrier is that even if you know you're supposed to be looking for a glass of water, you won't find it unless you 'x glass of water'. Neither 'glass' or 'water' are apparently listed as synonyms for the item. The third barrier is that old classic, 'guess the verb'. You then have to do something with the glass of water, and it involves a specific verb and none other. Reasonable synonyms such as 'push' don't work. To be fair, these are the kind of mistakes that a first-time author is likely to make and they're nothing that can't be remedied with experience.

More subjectively, there were some formatting choices that made the game less readable for me. In particular there was an overabundance of blank lines and white space. Another thing that I found interfered with readability was prefixing each line of dialogue with the name of the person speaking it. That just looks weird to me, and made the dialogue seem even more stilted and artificial than it was. A number of the games in the minicomp suffered from insufficient proof-reading, but this was the worst offender. I noticed many instances of missing punctuation, random capitalisation, typos and so on. Annoying, but reasonably straight-forward to fix.

On the plus side, I thought the concept was an original one and quite well supported by some little touches in the writing. And all credit to the author for having the perseverance to finish writing his game. That's possibly the most important talent that an author needs, everything after that is just polish and can be learned. I hope the author is strong enough to take any criticism he receives and use it constructively in the next game he produces.

The Mechano-Menace!

The only game in this year's minicomp to cast the PC in the role of a 'good guy'! Given that I see the entertainment value of AIF primarily in terms of escapism and wish fulfilment, and my wishes don't run to finding my best friend having sex with my wife (to pick just one example), this was a welcome relief.

The greatest strength of *The Mechano-Menace!* was the depth and inventiveness of its technical implementation. The most obvious example is the PC's shield power, and the shield and health stats that are tracked in the statusbar at the top of the screen. However, there were a lot of other little touches that made the game a pleasure to play, from the extensive help system to alternate means of solving problems to the implementation of amusing responses for various actions a superhero might try. Even better, the game ran perfectly smoothly with no bugs or problems observable. I didn't even notice any spelling mistakes.

The two-part combat with the robots was also very well implemented. I was a little disappointed that the solution to the robot problem could be as simple as just hitting things, although that is certainly in keeping with the superhero genre. The villain is suitably mad, and when you add in robot rabbit ninjas and Herr Hare, this scene was a lot of fun.

Unfortunately in comparison to Mechano, the character of Marvela seems a little... bland. Marvela doesn't get a great deal of dialogue and what she does get is largely reactive, especially in the first half of the game so, she never builds up much of a 'presence'. The relationship between her and the PC isn't given any real development either, so when the story shifts from beating up the bad guy to Marvela's bedroom it feels quite abrupt. This something that I also noticed in one of A.Bomire's games from last year's minicomp (*Working Late*). What was really needed was another scene of some kind to develop the relationship between Aegis and Marvela, or at least something to suggest they were attracted to each other.

The sex scene itself was very well written, probably the best of the minicomp. However, as mentioned above, the lack of any real build-up robbed it of some of the punch it could have had. Something else I would have liked to have seen was something a little unique to make the scene more memorable, especially given the fact that both of the participants were superheroes. I'm sure there's a few interesting things Aegis could do with his shield power...

Overall though, *The Mechano-Menace!* was one of my favourite games in this year's minicomp. A.Bomire is one of the best when it comes to both writing and technical polish, and it's a pity that it's been so long since he's released a full length game (hint hint)

Mini-comp Reviews by Gary Plume

BSG: Twenty Two

This game was annoying. Violent domination sex is not my cup of tea, and the threat of being bombed into oblivion put another damper on my enjoyment. The NPC laughing at the PC was actually a powerful technique, but it unfortunately increased my revulsion of the game. The main puzzle seemed to be a conversational maze that you were supposed to follow, but the constant restarts to try to find a brute force way of exploring all dead ends annoyed me. Eventually, I gave up because the path that seemed to be correct didn't seem to win the game and then I had to postulate following combinations of pathways which wasn't fun. What does this game want from me? The titillation over frustration quotient was too low to make me come back for more after each bomb blast. I gave up trying to win. I don't remember where I read it recently, but someone recommended NOT trying to quote the onomatopoeia of an orgasm. This game shows me exactly what a terrible technique it is.

Prosperity of Cheaters

This had nice gameplay if a bit simplistic in manipulating humans and timing their routines. I didn't mind the rule violation because I'm not a stickler by nature. Sex with the object of your affections seemed anticlimactic. It was fun trying the game over to reach all six endings: a nice device to keep me coming back. There is a technical problem with talking to Beth in the bathroom in scenario 6 - her discovery of the boyfriend's infidelity repeats. In post-mini-comp modifications, wouldn't mind playing for a menage-a-troi ending or even a lesbo voyeur ending. I was annoyed that the punch couldn't be used to spike drinks. Perhaps a solution could be a warning that "The purple color of the punch is going to be a dead giveaway if you spike drinks with it." "Ask ____ about ____" provoked puzzling nonresponses.

In Darkness

Interesting start concept from initial scene and buildup of background to romantic love that goes beyond "You want to bone this textual chick because she's hot." Flashback device with objects were linear and not too adventuresome but an understandable and effectively jarring device in the minicomp limitations. I appreciated the effort to make the flashbacks different, especially the braid, with some variation in how the charms disintegrated afterward. The first sex scene felt very hot to me mostly because of the tentativeness and restraint of the PC and the corresponding flirtatiousness of the NPC. A delicate tug-of-war. The game itself functioned like one of its talismans and transported me back to a time when I was a horny teenage virgin with nothing but fantasies. The part where the PC creams in his jeans and the NPC giggles was the most memorable scene for me out of all the entries. Contributing to the heat was the challenge of doing/saying the right thing without getting too guess-what-author-wants, although I almost gave up getting my pants off. The sad and mostly predictable ending was a boner-killer, but evoked in this

reader the intended feelings of jealousy, betrayal, rage. I guess the author should be applauded for creating pathos, but I would've just liked a happy ending. In fact, I was so inspired, that I am now porting the first sex scene into Inform, partly to see how difficult it would be to render the NPC. Technical: got stuck after touching braid the second time since the replay didn't succeed in "ask jemma about friend" - kinda like getting stuck in the "friends zone". I somehow successfully did "fuck pussy" before final ending which made it confusing as the plot seemed to depend on this being refused.

Mechano-Menace

This was a funny and well coded and well written game. I liked the novelty of the spicy latina superhero and her Spanish exclamations. The enemies were quite funny. The Health/Shield meter was an interesting technical achievement, but not really a management challenge during the fight. It would have been fun to work in the scene where the supervillain talks too much and gives away the lynchpin of his plan. I have no idea where those damn last 10 points are. It's probably because I only used brute force on Herr Hare without anything more clever. The sex was well written, but picturing two musclebound goody two-shoes together created a comic distance that shielded me from some of the hotness.

Obedience

This was a fun segue from mundane to extraordinary. Again, domination is not my cup of tea, but this was certainly better than BSG. Being the supervillain created some fun, but the toys seemed to become useless too quickly. Aren't there more ways to use my supervillain powers? Sex was hot but again not really consensual. The scale of obedience that the XYZZY-summoned author seemed to allude to was too subtle for me to grasp; perhaps a status bar meter would help me understand what was helping or hurting my cause. The "look under ____" puzzles seemed awfully simplistic for game play. I'd have preferred a secret narcissistic closet/shrine for the uniform.

Sex Is Mental

This was a weird twist on an old fantasy. What red-blooded male hasn't wanted a nurse fantasy? Even though we had very little interaction with the first NPC, I got some nice insight into the dog-eat-dog institutional life and the craziness of the PC. I found this the weakest game in terms of technical implementation. What about groin as a synonym for balls? Annoying grammatical errors and inexplicably capitalized words (Jacket, Guards, Corner, Dirty, Knock) bugged me. I got stuck in the ward and almost gave up trying to find a legitimate excuse for Mary to stay - I'm going to blame it on guess-the-verb and a nearly invisible cup of water. Sex was okay, if somewhat perfunctory. Technical: Where's my bandage? In an empty room, I feel more acutely aware of the scarcity of resources. The least I should be able to do is use the broom to sweep the floor, drink the water, lie on the bed, and masturbate. Some amusing bugs show up if you try doing some things with the NPC while she's out of the room.

A Case for In Darkness by Softiron

I'm not near as grumpy as I was after playing last year's mini-comp, and the reason is GoblinBoy. New and aspiring authors, please take note: you don't need to make an epic to create a game that will be praised and remembered.

How does *In Darkness* prove this? Let me count the ways:

1. There is no pre-game info dump. It is very tempting as an author to provide exposition to bring the player up to speed with the characters and the plot. In most cases, this is entirely unnecessary and defeats the purpose of the medium. This is "interactive" fiction. GoblinBoy allows the player to gradually learn about the main character by exploring his surroundings and talking to other characters.
2. The few puzzles that exist fit into the story. They are clued, but not so much that they cease to be puzzles. Take, for example, a minor paragraph early in the game:

You look round, taking it all in. It appears to be late spring. There are flower beds with multi-coloured blossoms. Lots and lots of grass, of course, a deep, rich green. There are a couple of trees nearby, and it seems there must have been some strong winds recently, as a fairly large branch has fallen to the ground at the base of a nearby tree. Various people move around in the distance, doing all the things that people do in a park.

It's a fairly innocuous passage, there to provide setting. Initially, one would not think a clue to a puzzle would be here, and that's the way it should be. Imagine if it instead read like this:

There are a couple of trees nearby, and it seems there must have been some strong winds recently, as a fairly large BRANCH has fallen to the ground at the base of a nearby tree.

Not only is the first version not insultingly easy, it forces the player to share the motivations of the PC. It's now MY idea to pick up the branch, and not the author's. This helps build identification which is key in bringing the player into the story.

3. The sex serves the plot. There is a reason I think *Basic Instinct* is a better movie than *Pirates!* Now, the latter is a fine movie, but all the costumes and set decorations and cinematography is all there to serve the sex. There's nothing wrong with this; I like my straight porn just as much as the next guy. But there's just something special about a great, realistic story that is enhanced with adult situations. The sex in *In Darkness* is never gratuitous, but that doesn't mean it is any less sexy. In fact, I would contend the opposite is true. My only criticism of the entire game is that the primary sex scene uses the chick.t sex daemon, which disrupts mimesis.

4. Nothing feels excessive. The best songs always seem too short. The best movies are over too quickly. And some of the best fiction is over before you can blink. There is nary a wasted word or unnecessary subplot to be found. And despite only being allowed three rooms and three characters, GoblinBoy made his world seem much larger. This did not feel like a competition game, yet it is the type of game that is attainable for any new author.

Of course, just because a game is solidly made does not mean it will be to everybody's taste. If you prefer AIF that acts more as a role-playing game (e.g. *School Dreams 3*), then *In Darkness* will certainly not be to your taste. GoblinBoy has written a dark piece of adult fiction and programmed it into TADS. Like reading an actual piece of fiction, you have no real control over how the story unfolds or how it ends. But the experience is more than simply turning pages. By having the reader manually explore the environment and interact with it, the story can be internalized, thereby providing more emotional impact. For those who played Adam Cadre's *Photopia*, it is quite apparent that story would have failed as static fiction. It is much the same here, and I applaud GoblinBoy for not treating his work of fiction like a toy box, allowing the player to mess with it at will.

This may be a mean thing to say in light of how much work was put into SD3, but due to the reasons listed above, I strongly believe this is the best game GoblinBoy has made, and it has supplanted *Blowjob Drifter* and *Memories Are Made Of These* as my favorite AIF game of all-time.

And now, for a couple of my own awards.

Hottest Individual Sex Scene: This would go to me nailing an "influenced" Bethany in front of several partygoers. Suggestion to Rip-CPU: more games like this one, tightly-packed with multiple endings. If not for all the misspellings, grammar issues, and bugs, this would have finished in 2nd place on my ballot.

Paul O'Brian Award: To Mechano-Menace, for rivaling the fun factor of *Another Earth*, *Another Sky*, only adding some gratuitous sex at the end. Can't complain about that! Or the upcoming sequel, I hope. ♦

There have been more than a few examples of windows in AIF games, but more often than not they seem to be just static objects with a single description. They do not have to be that way. It is possible to create a window which functions like they real thing. Something that can be used to look into the other room and show the true state of that room as well as the people or objects it contains. That was the task that I set for our authors this month.

Create a window between two locations that the player can look through to see the other location. Both locations should be actual "rooms" that the player can get to. In addition, the description of looking through the window should reflect changes in the other location. For instance, if the player takes something from one room to the other and drops it, he should be able to look back into that room later and see the new object. I'll leave the particulars in your imaginative hands.



Coder's Corner

TADS 2 Segment by A. Bomire

In this month's "Coder's Corner", we will be exploring seeing objects that are not in the same location as the player; i.e, in another room. This can be handy for spying on sunbathing women through a window, watching someone through a security camera, or just in general looking at something that is not in the same location. There are a couple of ways of handling this: 1) The easy way, or 2) The harder but much more effective way. Guess which one we are going to do? Sorry, that was a trick question. I'm actually going to describe both methods!

To begin with, let's set the stage. In our sample game, the player character is involved in an investigation, and the game takes place in your typical interrogation room. There is an observation room with a piece of one-way glass looking into the interrogation room. From the interrogation room, the glass appears to be a mirror but from the observation room the player can see everything that is happening in the interrogation room. Let's create our two rooms:

```

startroom: room
    sdesc = "Observation room"
    ldesc = "This is the observation room, where you can observe
        what is going on in the interrogation room to the south.
        There is a large window made of one-way glass on the
        south wall, allowing you to see into the other room. "
    south = Interrogation
;

window: fixeditem, seethruItem
    location = startroom
    sdesc = "window"
    ldesc = "You can look through this window to see what is going
        on in the interrogation room, but from there the window appears
        to be a mirror. "
    noun = 'window'
    thrudesc = Interrogation.lookAround(true)
;

Interrogation: room
    sdesc = "Interrogation"
    ldesc = "This is the room that is used to interrogate suspects.
        There is a large mirror on the north wall next to the door
        leading out. The room has a simple table and chair, used
        in performing interrogations. "
    north = startroom
;

```

```

mirror: fixeditem, seethruItem
  location = Interrogation
  sdesc = "mirror"
  ldesc = "This big mirror is actually a window through which
    observers in the outer room can look to see what is going
    on in this room. "
  noun = 'mirror'
  thrudesc = "You just see yourself. "
;

table: beditem
  location = Interrogation
  sdesc = "table"
  ldesc = "This is just a simple metal table - nothing fancy that
    a potential suspect can use as a weapon. "
  noun = 'table'
  adjective = 'simple' 'metal'
;

chair: chairitem
  location = Interrogation
  sdesc = "chair"
  ldesc = "This metal chair goes with the metal table, and is also
    of heavy construction to keep a suspect from breaking it. "
  noun = 'chair'
  adjective = 'metal' 'heavy'
;

```

Pretty simple, right? You'll notice that I've placed a window in one room and a mirror in the other, both of which are objects of TADS class *seethruItem*. A *seethruItem* is a special object class for which TADS has predefined the ability to "look through". When the player types "look through window", TADS will find the window object and execute/display whatever is defined for that object's *thrudesc* property. For the mirror, I defined this to simply tell the player that he can see himself. For the window, I have TADS display what the player would see if he were in the interrogation room and executed the "look" command. This is the easy method of performing this task. However, it is of very limited effectiveness. To begin with, looking through the window displays something like this:

Interrogation

This is the room that is used to interrogate suspects. There is a large mirror on the north wall next to the door leading out. The room has a simple table and chair, used in performing interrogations.

While this accurately displays the room, and any visible objects in the room, it really isn't a good description. After all, it describes the mirror - something which the player cannot see from where he is (on the other side of the mirror). Furthermore, any objects in the interrogation room aren't really "visible" in the sense that the player can examine them individually. For example, from the outer room the player cannot "x table" or "x chair". If he does, he will get a message indicating "I don't see that here."

A much more effective method of doing this would be to make all of the objects within the Interrogation room visible from beyond that location. To do this, we would need to play around with a property of TADS objects called *isVisible*. (For more information about this, please see the TADS manual, Chapter 4 - Parsing Fundamentals. Also, see Chapter 6 - The Parsing Sequence, and Chapter 7 - Parser Summary and Quick Reference.) The *isVisible* property is accessed like this:

```
object.isVisible(vantage)
```

Where *object* is the object which we are checking to see is visible, and *vantage* is the vantage-point or viewpoint from which we are checking. Usually, *vantage* is a character (and almost always the player), but you could use it to see if something is visible to a completely different object. For example, in our scenario above, the chair is visible to the table and vice-versa, but not to the player.

This property returns a boolean (true/false [nil]) response. How does TADS determine this? Well, it goes through a series of tests to see if an object is in the same room as the vantage (let's just say the player to make this simpler), is that object within another object and if so is that object open (like an open box containing things), and other tests like this. It can get pretty complicated. For us, we are going to make it simple. If the player is in the observation room, and the object is in the Interrogation room, then that object is visible to the player. If the object isn't directly in the Interrogation room, then normal *isVisible* processing will be done. The good thing about this is that if an object is in/on another object (say, on the table), then the normal processing for TADS will be to check to see if the table, for example, is visible, and if it is then the object resting on the table will be as well. So, we don't have to worry about drilling down into object contents - it will automatically be handled. We could modify every object in the game, but instead we are going to modify the big cahoona of the TADS object hierarchy, the *thing* object:

```

modify thing
    isVisible(vantage) =
    {
        if (vantage = Me and Me.location = startroom and self.location =
Interrogation)
            return true;
        else pass isVisible;
    }
;

```

Now, if we make that change, the table and chair should be visible from the observation/startroom. However, testing reveals that we are still receiving the message "I don't see that here.". Why is this? Because TADS is attempting to work faster. Whenever the player types a command that acts upon other objects, TADS goes through all of the objects that match the *noun* phrase used, and whittles them down to just objects that are in the same room as the player. From those, it then whittles away the ones that are out of reach - for example inside a closed, transparent container. This is something that we talked about in this newsletter's earlier series "Programming Erin", when we attempted to place a phone call to Erin while she was out of the room. But, it doesn't hurt to go over this again. The list of objects that are in the same room as the player is the *validDoList* (for Valid Direct Object List). Now here's the somewhat tricky part - if you set the value of this list to nil, then you would assume that this means there are NO objects that are valid. This isn't true. Instead, TADS then assumes that ALL objects are valid, and begins checking every object in the game to see if it is within the player's reach. Odd, right? Yes, but it is done this way for the exact reason that we are using it - to allow us to bypass the visibility of an object and allow the player to act upon it if we want, without having to rewrite a bunch of code. (For more information on the use of *validDoList* and its companion *validIoList*, see the in-code documentation in ADV.T and the TADS manual, Chapter 4 - Parsing Fundamentals and Chapter 6 - The Parsing Sequence.)

As we did earlier, we can modify the *validDoList* for all verbs within the game by modifying the root *deepverb*. However, in this case this may not be what we want. After all, Mike Roberts designed the *validDoList* and *validIoList* functions to act as they do for a reason. We will not be allowing the player to act upon any object in the Interrogation room other than to view it, so the only verb we really need to alter is the one that is used when the player examines something (ex. "x table"). This is the *inspectVerb*:

```

modify inspectVerb
    validDoList(actor, prep, iobj) = nil
;

```

Lastly, we need to alter what is displayed when the player looks through the window in the observation room. Instead of just parroting what the player would see if he were actually within that room, we will customize it. We'll need to paraphrase the room's description, and then list any objects of interest in the room. This can be somewhat involved. After all, we have the room itself, then any objects which may be on the table, and any objects which may be in the chair. If we had more furniture in the room, we would need to worry about that as well. And we also may need to worry about things inside of other things, such as a picture inside of a box that is sitting on the table. These can all add complications to your viewing. Or not, depending upon how detailed you wish to be. For us, we are going to worry only about superficial items; e.g., in the above example we would say that a box is on the table, but not the contents of the box.

TADS has built-in routines to list the contents of objects, but they would not necessarily work for us. For one thing, the routines do not list actors/characters, so any characters we have in our room wouldn't be listed. So, we are instead going to write our own routine, which I am going to call *listObjects*. This will involve writing a function that goes through a list and outputs the individual entries, with a comma after every entry except the last one. But, we don't want to list any of the Interrogation room's normal contents, which are the mirror, the table and the chair. So, we will ignore them. First, let's write a routine to print out a list of items:


```
listObjects: function (objList)
{
  local i, tot;

  tot := length(objList);
  if (tot = 0) return;

  for (i:=1; i<= tot; i++)
  {
    "<<objList[i].adesc>>";
    if (i = tot - 1) ", and ";
    else if (i < tot) ", ";
  }
}
```

Now, we can use this function to list the contents of the Interrogation room. We will first look in the room itself, then what is on the table, and finally what is in the chair. The objects within all of these will be listed in the *contents* property of each object. All of this will be done when the player looks through the window:

```
window: fixeditem, seethruItem
  location = startroom
  sdesc = "window"
  ldesc = "You can look through this window to see what is going
    on in the interrogation room, but from there the window appears
    to be a mirror. "
  noun = 'window' 'mirror'
  thrudesc =
  {
    local i, objects := [ ], x;

    "Through the window you can see the interior of the interrogation room,
    including the table and chair. ";

    //List the contents of the room, if there are any
    for (i:=1; i<= length(Interrogation.contents); i++)
    {
      x := Interrogation.contents[i];
      if (not (x = mirror or x = table or x = chair) )
        objects := objects + x;
    }

    if (length(objects) > 0)
      "Inside the room you can see <<listObjects(objects)>>. ";

    //List the contents of the table, if there are any
    objects := [ ];
    for (i:=1; i<= length(table.contents); i++)
    {
      x := table.contents[i];
      objects := objects + x;
    }

    if (length(objects) > 0)
      "Sitting on the table is <<listObjects(objects)>>. ";

    //List the contents of the chair, if there are any
```

```

objects := [ ];
for (i:=1; i<= length(chair.contents); i++)
{
    x := chair.contents[i];
    objects := objects + x;
}

if (length(objects) > 0)
    "Sitting in the chair is <<listObjects(objects)>>. ";
}
;

```

Now that we've written a routine to see into the other room, we need some objects to view. An appropriate addition would be someone who needs to be interrogated, like Erin. And of course, the ubiquitous glass of water:

```

Erin: Actor
    location = chair
    sdesc = "Erin"
    adesc = self.sdesc
    thedesc = self.sdesc
    ldesc = "Erin is a lovely young woman. "
    noun = 'erin'
;

glass: item
    location = table
    sdesc = "glass of water"
    ldesc = "This is an ordinary glass of water. "
    noun = 'water'
    adjective = 'glass' 'glass of'
;

```

If everything has been done right, the player should be able to see both of these objects through the window, as well as examining them individually.

TADS 3 Segment by Knight Errant

Welcome once again to Coder's Corner. This week's programming issue is windows. Unlike doors, TADS 3 doesn't have a built-in class for windows. Instead, we'll have to make do with a combination of a SenseConnector (to connect the two locations) and Fixture (because we don't want to allow the player to remove the window). First, create two rooms that you'll want to connect ... for example, an indoor and an outdoor room. Indoor rooms come complete with walls, so we'll just place the inside half of the window on one of the walls. We'll need a Fixture in the outside room to represent the building, that we can put the outside half of the window in. It can work like this.

```

Window : SenseConnector, Fixture 'window' 'window'
    "It's an example of a window."
    connectorMaterial = glass
    locationList = [outsideRoom, insideRoom]
;

```

For any sense connector, the connectorMaterial determines what senses can travel across the connector. "Glass" allows sight but nothing else. "Adventium" blocks all senses. "CoarseMesh" allows all senses including touch, but does not allow movement of objects across. "FineMesh" is transparent to all senses save touch. "Paper" blocks sight and touch, but allows sound and smell. Here, we want glass.

With the sense connector in place, we can now see from room to room, and objects in each room will be visible to each other. However, any object viewed through the window will use its `remoteDesc` or `remoteSpecialDesc`. That means you'll have to customize these properties for any objects that may be viewed from room to room. The easiest way to deal with this is to modify `Thing` to have an appropriate `remoteDesc`, like this.

```
modify Thing
    remoteDesc = "<<theName>> is in <<location>>"
```

Of course, even if you modify `Thing` to give a decent generic description, you can always also give an individual item a more specific `remoteDesc` of its own.

That's all it takes for a window you can look through. If you want to make it something you can open, add an `Openable` class. If you want to be able to climb through it, make it a door as well.

Until next month!

Inform 6 Segment by 'trix

Hello, marionettes.

This week, a window into another room. Perhaps a metaphor for ... Anyway, this is one of the things that TADS 3 is really good at. But we can make do in I6.

Essentially it's a scoping problem, so here is my IF Beginner's Introduction to Scope. Scope is one of the basic things that IF authors need to get to know. When the player has typed a command that seems to contain references to objects, the parser goes through all the objects in scope to find the best match. Things out of scope are not considered. Mostly, scope just means "stuff the pc can see", which more or less means "stuff in the room"; but certain contexts require variations. For instance, if you're issuing orders to an NPC in another room, telling him to pick up an ostrich, then scope needs to consider what the NPC can see, not what the pc can see. And if the pc is in a dark room, but it has been established that there is a light switch that the pc can find and reach, that needs to be in scope. Or, if you want to write a debugging verb that will move any named object in the game into the pc's inventory, scope needs to include every object in the game. (That debugging action does exist, and is called 'purloin'.)

Going through scope is part of parsing, which is the process of working out what the player's command means. Before parsing is complete, you can't check the values of variables like `action` and `noun`, because they haven't been decided yet. So any alterations to scope need to be dealt with before any action processing, and need to use other methods to work out what's going on. Fortunately I6 has enough scope entry points to make that fairly straightforward.

Any object that is in scope by the normal scope rules can provide the property `add_to_scope`, to add a bunch of other objects into scope. `add_to_scope` can either be an array of objects, or it can be a routine, which should add objects to scope with the command `AddToScope(obj)`.

If you put a standalone routine called `InScope` in your code, then it can override the normal scoping rules according to your own whims.

If you want a special scope for a particular action, you can use `scope=Routine` in the verb grammar, as I think I did for one of these articles a couple of months ago. Who can keep track.

OK. Here's the rooms with a window. I'll make it work later.

```
Constant STORY "A Window Into Something";
Constant HEADLINE "^Perhaps a metaphor for ...^";

Include "Parser";
Include "VerbLib";

! Yes, it's good old TwoDoor again
Class TwoDoor
```

```

with directions,
  door_dir
  [;
    if (parent(self)==self.&found_in-->0)
      return self.&directions-->0;
    return self.&directions-->1;
  ],
door_to
  [;
    if (parent(self)==self.&found_in-->0)
      return self.&found_in-->1;
    return self.&found_in-->0;
  ],
has static door;

Object viewroom "Viewing Room"
  with description "This is the room wherein you can view
    things in the interview room. There's a big
    window on the north wall, and a black door to
    the west.",
  w_to viewdoor,
  out_to viewdoor,
  has light;

Object -> window "window"
  with name 'big' 'glass' 'window',
  has scenery;

TwoDoor -> viewdoor "black door"
  with name 'black' 'door',
  found_in viewroom corridor,
  directions w_to e_to,
  when_open "The black door is open.",
  when_closed "The black door is closed.",
  has openable;

Object corridor "Corridor"
  with description "A short, curiously sparse corridor.
    There is a black door to the east, and a white
    door to the north.",
  e_to viewdoor,
  n_to interviewdoor,
  has light;

TwoDoor -> interviewdoor "white door"
  with name 'white' 'door',
  found_in corridor interviewroom,
  directions n_to s_to,
  when_open "The white door is open.",
  when_closed "The white door is closed.",
  has openable;

Object interviewroom "Interview Room"
  with description "A big white room with a wooden table,
    and a white door leading south. On the south wall
    is a big mirror.",

```



```

        s_to interviewdoor,
        out_to interviewdoor,
    has light;

Object -> table "table"
    with name 'wood' 'wooden' 'table',
    has scenery supporter;

Object -> -> notepad "notepad"
    with name 'notepad' 'pad'
;

Object -> -> pencil "pencil"
    with name 'pencil'
;

Object -> mirror "mirror"
    with name 'big' 'mirror',
        description "Damn you look good.",
    has scenery;

[ Initialise;
    location = viewroom;
];

Include "Grammar";

```

Now, when you look in the window, you should get a description of the interview room and its contents. There's two ways to approach this: we can print the text we want, and customise it ourselves according to the objects in the room; or we can cheat and get the standard looking rules to do it for us. Unremarkably, I'm opting for the latter option. Let's try this:

```

Object -> window "window"
    with name 'big' 'glass' 'window',
    before
    [ loc;
        Examine, Search:
            loc = parent(player);
            PlayerTo(InterviewRoom, 1);
            <Look>;
            PlayerTo(loc, 1);
            rtrue;
    ],
    has scenery;

```

That's moving the player (silently) to the other room, looking, and then moving her back. That works, after a fashion, but you'll get the room announcement at the top, which isn't really appropriate, and the room description isn't quite what we want. What we want is to skip the announcement and the normal description, print a customised description, and then do the standard ##Look listing of the room's contents. Fortunately, listing a room's contents is in a routine called `Locale`, which we can call ourselves.

```

Object -> window "window"
    with name 'big' 'glass' 'window',
    before
    [ loc;
        Examine, Search:
            print "Through the window, you can see the interview room.
                It is a big white room with a wooden table in the centre.^";
            loc = parent(player);

```

```

        PlayerTo(InterviewRoom, 1);
        Locale(InterviewRoom);
        PlayerTo(loc, 1);
        rtrue;
    ],
    has scenery;

```

We intercept ##Search as well as ##Examine because the command “Look in mirror” will generate a Search action.

There’s one subtle issue here to mention: the white door (in the interview room) is on the south wall, so it shouldn’t be mentioned (“The white door is closed.”) in the locale text. So here’s how we stop it: set a flag when we’re looking through the window, and have the white door not announce its presence when the flag is on.

```

Object -> window "window"
  with name 'big' 'glass' 'window',
  before
  [ loc;
    Examine, Search:
      give self general;
      print "Through the window, you can see the interview room.
        It is a big white room with a wooden table in the centre.^";
      loc = parent(player);
      PlayerTo(InterviewRoom, 1);
      Locale(InterviewRoom);
      PlayerTo(loc, 1);
      give self ~general;
      rtrue;
    ],
  has scenery;

! ....

TwoDoor -> interviewdoor "white door"
  with name 'white' 'door',
  found_in corridor interviewroom,
  directions n_to s_to,
  describe
  [;
    if (window has general) rtrue;
    if (self has open) "^The white door is open.";
    "^The white door is closed.";
  ],
  has openable;

```

describe is always the property to use if you want the option of hiding an object entirely, because if it returns true, the room contents list prints nothing related to the object, not even a new line.

Next, the scope stuff. The quick way to add the contents of the interview room into scope is using ScopeWithin. Put this anywhere before Include “Grammar”:

```

[ InScope person;
  if (person in viewroom)
  {
    ScopeWithin(interviewroom);
  }
  rfalse;
];

```

ScopeWithin put the contents of the given object in scope, recursing down by containment.

Returning false at the end of the routine means that the parser can carry on and put everything in scope that would normally be in scope.

Now that's the quick way to put everything in scope. But not everything should be in scope: the white door and the mirror aren't visible through the window. So instead we'll have to go through the contents of the room and add the stuff we want to be added.

```
[ InScope person x;
  if (person in viewroom)
  {
    objectloop (x in interviewroom)
    {
      if (x ~= mirror or interviewdoor)
      {
        PlaceInScope(x);
        ScopeWithin(x);
      }
    }
  }
  rfalse;
];
```

Now anything (other than things specifically excluded) in the interview room will be in scope from the viewing room. That makes them available for examining (which is what we want), and conceivably for other actions, like opening and closing. How do you prevent that?

One way is to write a before property on the viewing room that intercepts any unsuitable action on an object in the other room. That's not as much trouble as it sounds. But there is a possibly easier way: every action involving physical contact will call the routine ObjectIsUntouchable. This checks if the player is separated from a given object, for instance by being in a sealed glass case. If we replace this routine, we can add the extra check we need. To do this, we can copy the routine from out of VerLibM, paste it into our code, and put some extra stuff in it to check what we need to check.

```
Include "Parser";
Replace ObjectIsUntouchable;
Include "VerLibM";

[ ObjectIsUntouchable item flag1 flag2 ancestor i;
  ! NEW STUFF BEGINS HERE
  if (IndirectlyContains(viewroom, player)
      && IndirectlyContains(interviewroom, item))
  {
    if (flag1) rtrue;
    print_ret (CTheyreOrThats) item, " in another room.";
  }
  ! NEW STUFF ENDS HERE. Everything below this is just
  ! copied and pasted from VerLibM.h.

  ! Determine if there's any barrier preventing the player from moving
  ! things to "item". Return false if no barrier; otherwise print a
  ! suitable message and return true.
  ! If flag1 is set, do not print any message.
  ! If flag2 is set, also apply Take/Remove restrictions.

  ! If the item has been added to scope by something, it's first necessary
  ! for that something to be touchable.

  ancestor = CommonAncestor(player, item);
  if (ancestor == 0) {
```

```

    ancestor = item;
    while (ancestor && (i = ObjectScopedBySomething(ancestor)) == 0)
        ancestor = parent(ancestor);
    if (i ~= 0) {
        if (ObjectIsUntouchable(i, flag1, flag2)) return;
        ! An item immediately added to scope
    }
}
else

```

! First, a barrier between the player and the ancestor. The player
! can only be in a sequence of enterable objects, and only closed
! containers form a barrier.

```

if (player ~= ancestor) {
    i = parent(player);
    while (i ~= ancestor) {
        if (i has container && i hasnt open) {
            if (flag1) rtrue;
            return L__M(##Take, 9, i);
        }
        i = parent(i);
    }
}

```

! Second, a barrier between the item and the ancestor. The item can
! be carried by someone, part of a piece of machinery, in or on top
! of something and so on.

```

if (item ~= ancestor) {
    i = parent(item);
    while (i ~= ancestor) {
        if (flag2 && i hasnt container && i hasnt supporter) {
            if (i has animate) {
                if (flag1) rtrue;
                return L__M(##Take, 6, i);
            }
            if (i has transparent) {
                if (flag1) rtrue;
                return L__M(##Take, 7, i);
            }
            if (flag1) rtrue;
            return L__M(##Take, 8, item);
        }
        if (i has container && i hasnt open) {
            if (flag1) rtrue;
            return L__M(##Take, 9, i);
        }
        i = parent(i);
    }
}
rfalse;
];

```

You should maintain the practice of checking `ObjectIsUntouchable` in any new actions that you write so that this sort of thing will work consistently.

That's about it from me this month. Happy programming, hypothetical readers.

Inform 7 Segment by Dudeman

Another month, another coding problem for us to tackle here at coder's corner. This month we will be dealing with windows. Though this can be a literal glass window, from a coding stand point a window can be anything that allows the player to see into a room without actually being in that room. However, to keep things simple we will just be dealing with a basic everyday window, like one that might be in your kitchen that looks out into your backyard.

```
the Kitchen is a room. "A standard kitchen with all the standard kitchen
fixtures. To the north is the backyard with a large window looking out into it."
```

```
the Backyard is north of the kitchen. "The backyard is fairly large with a small
garden running along the side. To the south is the entrance to the kitchen with
a window along the wall looking into it."
```

```
Jill is in the backyard. The description of Jill is "Jill is your wife of only
a few months and the two of you are still very much still in the honeymoon
phase. Her hot, tight young body still makes you want to jump her every time
you see her."
```

```
a window is a backdrop. It is in the kitchen and the backyard.
The window can be openable. The window can be open. The window is openable and
closed.
```

Now that we have our basic set up, we now want to make it so that the player can look into the backyard from the kitchen and vice-versa as would be expected by the presence of the window. Just like last month, this feature of games is so common that it is actually addressed by the Recipe Book that comes with the Inform 7 software. Specifically, chapter 3 part 5 of the recipe book titled "Windows" covers various situations revolving around windows in a game. In this case, the best example give would be "Dinner is Served" which gives you a basic idea of how to tackle this in our little game. Using this example, we can fit it into our game with only minor tweaks and taking out the parts we don't need like being about to reach through the window since we don't need that for our example.

```
After deciding the scope of the player:
```

```
    if the player is in the kitchen, place the backyard in scope;
    if the player is in the backyard, place the kitchen in scope.
```

```
Instead of examining the backyard while in the kitchen:
```

```
say "Looking through the window into the backyard you can see [a list of visible
things in the backyard]."
```

```
Instead of examining the kitchen while in the backyard:
```

```
say "Looking through the window into the kitchen you can see [a list of visible
things in the kitchen]."
```

```
Instead of examining the window:
```

```
if in the kitchen, try examining the backyard;
if in the backyard, try examining the kitchen.
```

And that gives the player the ability to see into the kitchen from the backyard and vice-versa through the window, but will not be able to otherwise interact with things in the other room, behavior that is expected of a window. This is because the "scope of the player" as used in the first rule is the term Inform 7 uses to describe what the player can interact with, but there are other rules in place that prevent physical interactions from happening between rooms that would be illogical. You can learn more about scope in Chapter 17 part 27 of the Inform 7 documentation.

Now to test this out we can add a few random objects to the rooms such as;

```
an apple is a thing in the kitchen. The description of the apple is "A healthy,
nutritious red apple. Not much more to say about it."
```

```
A watering can is in the backyard. The description of the watering can is "A
large watering can Jill uses to water her garden every day."
```

Now you can try playing around with these objects, moving them from room to room and you can see that any changes will be reflected when you look through the window. Now this is just very basic behavior and can always be expanded upon, but for now I feel this should give you an idea of how Inform 7 handles things like windows.

ADRIFT Segment by BBBen

This month's challenge is, I'm pleased to say, not too challenging in ADRIFT. In fact, there's a nice, easy way to take care of it. Let's say we have two rooms, a living room and a garden, and a window that looks out from the living room onto the garden.

1. Create the rooms, and an object for the window.
2. Put that object in the living room. In the window object give a description along the lines of "This window looks out onto the garden. You can **look through** it."
3. Next, create a task called "look through window". Make it repeatable, and make it completable only in the living room.
4. Now, instead of putting any description in the task, just below the "message upon completion" box is a drop-down menu labelled "Then show description for room:". All you have to do is select the room "garden".

And that's the extent of it! Now when the player types "look through window" in the living room they will get a response that is just like a "look" command in the "garden" room. ◆

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.



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.

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*.

Dudeman has released one game and is working on a second. He has also released an impressive Inform 7 sex extension to help make it easier for others to write games of their own.

Knight Errant is an AIF player who has released two games 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.

