

# Definitive Guide to Writing Guild Games for the Rest of Eternity

Jake Beal, with help from Joe Foley

Version 96

2010-06-22

<http://web.mit.edu/foley/Public/Mech/gamewriting.pdf>

## Changelog

2010/6/22 Fixed typos that Kendra Beckler pointed out.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	A Note On Schizophrenia . . . . .	5
1.2	Do I Really Want To Write a Game? . . . . .	6
1.3	What a Guild Game Isn't . . . . .	6
1.4	Managing Your Time . . . . .	7
<b>2</b>	<b>Forming a GM Team</b>	<b>8</b>
<b>3</b>	<b>Design</b>	<b>10</b>
3.1	Getting Started . . . . .	11
3.2	Philosophies of Writing . . . . .	11
3.3	Other People's Ideas . . . . .	13
3.4	Forming the Universe . . . . .	14
3.5	Game Dynamics . . . . .	15

3.5.1	Spatial Distribution . . . . .	16
3.5.2	Time Distribution . . . . .	17
3.5.3	Combat . . . . .	17
3.5.4	Mortality . . . . .	20
3.5.5	Captivity . . . . .	22
3.5.6	Love-Fest vs. Knife Fight . . . . .	23
3.5.7	Good Guy Mobs . . . . .	23
3.5.8	Suspension of Disbelief . . . . .	24
3.6	Mechanics . . . . .	25
3.6.1	Players vs. Mechanics . . . . .	25
3.6.2	GMs vs. Mechanics . . . . .	26
3.6.3	Other Design Hints . . . . .	27
3.7	Character Design . . . . .	28
3.7.1	How much plot does a character need? . . . . .	29
3.8	Sanity Checks . . . . .	32
<b>4</b>	<b>Writing</b>	<b>32</b>
4.1	Organization . . . . .	32
4.1.1	Time Estimates . . . . .	33
4.1.2	Wrap-Up Document . . . . .	33
4.1.3	Writing Meetings . . . . .	33
4.1.4	Sheet-Reading Parties . . . . .	34
4.2	Technical Issues . . . . .	35
4.2.1	The Template and LaTeX . . . . .	35
4.2.2	Revision Control . . . . .	36
4.2.3	Backups . . . . .	37
4.3	Writing Characters . . . . .	38
4.4	Mechanics . . . . .	39
4.5	Physreps . . . . .	42

---

<b>5</b>	<b>Pregame</b>	<b>43</b>
5.1	Writing A Good App . . . . .	43
5.2	Collecting Applications . . . . .	45
5.2.1	Managing Applications . . . . .	45
5.2.2	Shit, There Aren't Enough Players! . . . . .	45
5.3	Casting . . . . .	46
5.3.1	Dealing With Hard Cases . . . . .	48
5.4	Communicating With the Players . . . . .	49
5.5	Players Communicating With You . . . . .	49
5.6	Reserving Rooms . . . . .	50
5.7	Production . . . . .	50
5.8	Handout . . . . .	52
5.8.1	How Handout Should Be . . . . .	52
5.8.2	Attaining the Ideal . . . . .	53
5.8.3	Dealing with No-Shows . . . . .	55
5.8.4	When Things Go To Hell . . . . .	55
5.9	Shore Leave . . . . .	56
<b>6</b>	<b>Runtime</b>	<b>56</b>
6.1	Setup . . . . .	56
6.2	Running a Short Game . . . . .	57
6.3	Running a Ten Day Game . . . . .	58
6.4	The Role of the GM . . . . .	59
6.5	Managing Ghosts and NPCs . . . . .	60
6.6	Maintaining the World . . . . .	61
6.7	Dealing with Brokenness . . . . .	62
6.8	High Stress Situations . . . . .	62
6.8.1	Some situations you are likely to have to deal with . . . . .	63
6.9	Keeping the Team Together . . . . .	64

<b>7 Postgame</b>	<b>65</b>
7.1 Cleaning Gamespace . . . . .	65
7.2 Wrapup . . . . .	66
7.3 Return to the Office . . . . .	67
7.4 Last Rites . . . . .	67

# 1 Introduction

Due to the encouragement of some people who really should know better, I've decided to write the *Definitive Guide to Writing Guild Games for the Rest of Eternity*. I'm claiming Official Status for the *Guide* until somebody else writes one that people like better, so if you disagree with me, go write your own damn guide! Until then, you'd better do what I say or risk the **Wrath of the Spider-Lords**.<sup>1</sup> Feel free to send me flames at `jakebeal@alum.mit.edu` or post reasonable discussion to `generic@mit.edu`.

Now that that's done with, let's get down to business. Why does this document exist, and why the hell should you care? Well, presumably you're a Guild member who wants to write a game and you want a little advice on how to do it. There's a lot of institutional wisdom out there, and you need a tap into it. More to the point, you want three important things from your game:

1. Your game shouldn't suck.
2. Your game should actually run someday.
3. Writing it shouldn't cause you to flunk.

Notice that I didn't say anything about writing a great game. This document is *not* going to teach you to write a great game any more than a manual of style will teach you to write a great novel. Greatness comes from the cool ideas that you've thought of and that I haven't.

What this document *will* do is try to help keep you from making stupid mistakes, and thereby allow the greatness in your game to shine. There are a lot of easy ways to shoot yourself in the foot, and it only takes a couple shots to turn a good game into a horror story. Personally, I like playing in cool games, and I like them even better when they aren't broken.

That said, let's get down to it. This guide is organized into relatively arbitrary chunks, roughly aligned with the lifecycle of a game. The first couple sections are the inception of a game, when you figure out what you want to do, form your team, and figure out how to attack the problem. After that, I give advice on the main bulk of game writing: structuring your game, designing mechanics that won't break, and avoiding some of the easy pitfalls of game design. Finally, I deal with casting, production, handout, actually running the game, and what to do once it's over.

Enjoy.

## 1.1 A Note On Schizophrenia

There is some schizophrenia in the writing of this document, since it was started as a rant by Jake back in 2000, and then Joe joined and the two decided to finish it in 2005. Anything

---

<sup>1</sup>Don't ask. No, really.

in first person should be taken as Jake's voice, while the rest should be taken as collaborative between Jake and Joe.

## 1.2 Do I Really Want To Write a Game?

Yes, you probably do. You've got a cool idea in your head, and dammit, you should make it happen so the rest of us can share your cool idea!

But you're correct to wonder what sort of task you're taking on. Writing a game is not a small endeavor. In fact, it's larger than you think it is. But you don't have to flunk anything to write it, and you don't have to sacrifice your social life either.

How big is a game really? I figured out how much text I produced for a game, and it comes out to about a novel. Game size ranges approximately from Hardy Boys (SIK game) to War & Peace (particularly convoluted 10-day), and you should expect it to take a similar amount of time to create. It's a lot of work, but it's very doable, and it's a *lot* of fun.

## 1.3 What a Guild Game Isn't

An Assassin's Guild game is not a novel, a movie, or a play. There are many more main characters, so there are generally many more simultaneous plots evolving at once. Moreover, you have little or no control over the plot once the game starts to run.

An Assassin's Guild game is not a tabletop RPG. Usually a lot of important things happen in secret meetings in dark corners away from everybody else. As a result, the GMs often know less about what's actually happening during the game than the players, and most of the game should be self-running without any call for the GM. My ideal for a game is that the GMs say "start"! and then just sit around shooting the breeze in the control room or wander around spectating, since they've become unnecessary.

An Assassin's Guild game is not a board game.<sup>2</sup> Immersion is a key part of the game. Although your players will be using every trick in the rules to win the game, at the same time they should be drawn into your world and feel what their character feels.

An Assassin's Guild game is not Nero, SCA, or a White Wolf LARP. In a guild game, player are usually issued characters rather than making them up, and the game runs once and stops.<sup>3</sup> The Assassin's Guild is also very shy about physical contact, for good historical reasons,<sup>3</sup> though occasionally a wacky-noodle will be used for actually whacking people in a SIK game.

---

<sup>2</sup>Shut up about *Lawn Arrangers*. Just trust that I know whereof I speak.

<sup>3</sup>Shut up about *Partitioning Germany*. Just trust that I know whereof I speak.

## 1.4 Managing Your Time

Something that most people don't realize before writing their first game is just how large an undertaking writing a game is. Not all games take a long time to write, but they obey the 5%/95% rule of building software: the first 95% of the work takes 5% of the time; the last 5% of the work takes 95% of the time. As a result, however long it takes to write a game, there are aspects of it that you will persistently underestimate. I know I still do.<sup>4</sup>

As a result, most GM teams go through “crunch time” in the last few weeks before the game is run, when people are panicking and the list of things still remaining to be done just keeps getting longer and longer. Game escapes from the neat scheduled blocks of time you'd been planning to devote to it and starts eating into your sleep schedule and causing you to skip classes and punt problem sets. By the week before game, you're doing nothing but game and the few things you absolutely can't escape, and even then you're only half doing them because your mind is on game. You become a Guild Zombie. Some people really get off on it, on the whole adrenaline high, racing to meet the all-important deadlines, feeling super-important because there are lots of people depending on you for their entertainment and hoping you'll make it in time.

I hate it. I also fear playing games that required a crunch time. That's because during crunch time, there's no opportunity to sanity check things,<sup>5</sup> and the “less important” parts of the game get triaged or shoddily done.<sup>6</sup> Moreover, in crunch time there's no chance to add the little filigrees and grace notes that turn a good game into a beautiful game.

Your goal is to write a game without a crunch time. Throughout this document, I'll try to give you realistic estimates of how long it takes to do things, so you can factor things into your estimate of how to avoid crunch time. The best heuristic, though, is to give yourself an entire extra month between when you expect to be done and when you ask to be scheduled.

How long does it take to actually write a game? Ten-day games usually take between six months and two years, though *Showdown at the Golden Gate* was a two-month crash project (and showed it) and I believe *Conspiracy* was eight years in the making. One night games usually take somewhere between a month and a year, though we wrote *Berkeley* as a 14-day crash project, from conception to game start, just to prove it could be done.

---

<sup>4</sup>For example, did you realize that it often takes a full hour to enter all the player information after you're done casting? If you've got 60 applicants, 50 of whom are playing, and the applications are in a different order than the entries in the `players.tex` template file, all the little bits add up shockingly.

<sup>5</sup>*Zagreb's* shipping mechanic is a prime example: it was never tested by anybody besides the GM who wrote it until it was inflicted on the players, and was of such hideous complexity that most people simply ignored it while a few spent hours trying to figure out how to make simple shipments.

<sup>6</sup>*Nanopunk: Jersey* had a beautiful tunnel mechanic where a hugely complex maze had descriptions at each node so that people who were supposed to be doing things there could easily follow the descriptions to their goal, while others would simply get lost. Unfortunately, the descriptions got triaged and the people who needed to find things in the tunnels spent literally days swearing at the GMs while stuck in a maze of twisty corridors all alike with a wumpus beating on them. The easiest way to leave the tunnels was to get killed by the wumpus and some players preferred it to navigating the tunnels.

When you're estimating time to complete a game, it's a mistake to think you'll get anything done during vacation or while you're on a trip. Sure, you don't have classes to contend with, and maybe you hate your family and your friends all moved away and so there's nobody you'll be spending time with. But you'll be away from MIT, and the pressure of the environment vanishes. Suddenly, you rediscover the joy of a full night's sleep, and a great nostalgia for Space Invaders and before you know it you're back at MIT and nothing's done.

Having a good estimate of time-to-completion is much less important when it comes to little games, because it usually matters less when they run. For those, I prefer to first get them basically finished, then ask to schedule them — and if you're done before you ask to schedule, you don't need a zampolit either.

## 2 Forming a GM Team

The most important part of your game is not a part of the game at all, but the people you write it and run it with. The most important part of the GM team is the team dynamics. Interpersonal conflicts on a team always spill over to the players, one way or another, whether because the strife keeps you from writing as cool a game as you're capable of or because they're actually getting conflicting rulings from GMs fighting it out via the game. At its worst, team conflicts have caused games to be aborted while actually running.

Pick people you can work with. This doesn't mean you should pick your friends. Your friends are sometimes the worst people to work with because we all pick up blind spots with respect to people we're close to, and you may be tempted to give them too much slack and have unrealistic expectations of their abilities—every parent thinks their baby is a genius.

Be especially careful about working with somebody you're dating. Jeremy Brown's rule of thumb is that you shouldn't be on a GM team with somebody unless you've already been dating for as long as you expect it will take to write the game. GMing is a very intense experience and, much like a long road trip, it is likely to either cement a relationship in place or destroy it completely. Do you really want to gamble?<sup>7</sup>

How many GMs do you need on your team? One nights typically have 1-4 GMs, while ten-day games usually range from 3-6 GMs. A larger team has more heads to brainstorm and debug ideas, more hands to write sheets, and more bodies to deploy and run the game. A smaller team has less opportunities for internal conflict, less chance of GMs being ignorant of parts of the game, and less trouble scheduling meetings.

In the earliest stages of development, you don't have to have your full team together. Once it starts acquiring some weight of notes and feeling like it's really going to happen, fill out

---

<sup>7</sup>I was on a GM team once with a lover whose company I enjoyed immensely. As time went on and our relationship got strained, it started bleeding over into GMing, and the two fed back badly—the emotional shock waves traumatized our newbie GM. My lover eventually quit the team during a huge full-team fight that nearly scuttled the game, and years later she still won't speak to me. It still hurts. Do you want to risk that?



the team as soon as possible. The later a GM is brought on board, the less likely they are to really understand the world and become a full partner on the team.<sup>8</sup> The worst are Runtime GMs: while the extra help is nice, they are often woefully ignorant of the game. Stick to friendly ghosts for your runtime labor—they're less trouble since they can't make insane rulings.

When putting together your team, don't be afraid to ask people who aren't in your circle of friends. If you've admired their work on other games or like the way they talk or play and think they might be an asset, just give them an elevator pitch on your game and ask them straight up if they'd like to be on the team. Quite often, they'll say yes and work out quite well.

Some useful qualities you'd like to have amongst the members of your GM team, no matter the game:

- The **Mechanics Hacker** crafts simple and elegant rule systems, writes them down precisely, debugs them to make them airtight, knows what a player can reasonably do on the fly, and can balance a game worth of statistics, money, and RTIs.
- A **Prose Artist** writes clear, informative, and entertaining characters and bluesheets, and can often churn them out at an incredible rate.
- A **Nitpicker Extraordinaire** ferrets out the dangling threads and hidden conflicts in your design, and is the only person who will notice that the brother's sheet puts his mother's death two years earlier than the sister's sheet.
- The **Bad Idea Generator** is a person who, when asked a question, can brainstorm up twenty mostly insane answers, of which three will become important plots in game.
- A **LaTeX Wizard** is able to make neat new types of cards, pretty computer generated diagrams of things, and make the production process run smoothly.
- The **Voice of Experience** has played or run many games and is a treasure trove of stories about what worked and what failed in games past, not to mention knowing all the weird architecture and hidden places of campus.
- A **Newbie** brings fresh blood to the team, since the Guild always needs more people who write games, and new GMs bring new ideas to gamewriting. Don't squash them.

In addition, you may need a Zampolit, but unless things explode, it makes no difference who the Zampolit is. The ideal Zampolit, in my opinion, is a GM who had to drop off the team previously (e.g. moved away or started failing classes). Another good choice is an experienced player who you know won't want to play your game: after all, the Zampolit is

---

<sup>8</sup>I really wasn't supposed to be able to acquire nuclear weapons in *Macondo*, but the GM had been brought on late and wasn't thoroughly indoctrinated by the other GMs, who eventually decided it could be made to work.

just one more potential player who can't app for your game. Don't pick a Zampolit until you're actually requesting to be scheduled.

Once you've got a team, be sure to set up a regular weekly meeting time. Although people's schedules can make this really hard, any less frequent and you will lose momentum. GMing is a social activity as well as a project. I like meetings that incorporate a meal, especially during the design process when not everybody needs to be working at a computer. Alcohol is a bad idea, although it's traditional for SIK games. Don't worry if you stray and chat about other stuff, especially during the early stages: the team needs to gel and GMs need to develop their working relationships.

As for dealing with GM team problems... I don't have any good advice. I've been on teams that were love fests and teams filled with acrimony, dealt with lame GMs, GMs who couldn't take "NO!" for an answer, GMs undergoing massive self-confidence crises, and even a GM whose work all had to be rewritten because he never did get how Guild games worked. All the problems are hard, the situations all suck, and I'm not much good at social work.

People get extremely invested in games: you spend years investing your heart and soul in this massive project, and something starts going wrong, and suddenly the stakes feel very high. Maybe it is only a game, but it's also a work of art and a work of love.

The best cure I know is prevention. Make sure you spend time having fun together as well as just writing the game together. Investing in a healthy and open relationship between GMs is investing in a safety margin for arguments. When people trust each other emotionally, they can separate an attack on their idea from an attack on them, and it makes it much more possible to find compromises.

Especially take care of the new GMs on the team: it's easy for them to end up feeling disconnected and ignored, washed over by a sea of jargon and back-in-the-day stories that don't include them. It's scary to write your first sheet and wonder if you're doing it wrong. Even though it's hard to let go, you have to give the new GMs real trust and responsibility or they can't learn the craft, and you have to make it safe for them to ask stupid questions or they have to learn everything the hard way.

## 3 Design

The design period of game-writing encapsulates the progress of the game all the way from a single vague sentence to serious sheet-writing. This is the time when the game is most pliable and you can make big changes easily, though it gets harder as your ideas become more developed and begin to interact with one another.

Some people advocate a very short design process, focussing on sheet-writing as the main area of effort. While one should never underestimate the time required by sheet-writing, I hold that you should expect to invest about half your time in the design phase before a single sheet is ever written. If you start writing sheets prematurely, you'll suffer the heartbreak of

having to discard your loving works of prose.

During design, you'll produce prodigious quantities of notes that will later form the skeleton of every sheet you write. Right now, however, you want to just record everything in whatever structure is most convenient for organizing it, without thought as to what information is public and private, or all the polish and niceties that attend proper sheets.

The other thing that characterizes the design period of gamewriting is that most meetings will have the whole team working on the same problem as a team. Later, during the writing process, meetings will usually have each person working on their own independent task, and having everybody in the same room is only necessary for moral support and so people can ask each other questions.

### 3.1 Getting Started

You'll want to schedule regular weekly meetings with your team.<sup>9</sup> Meeting less often than once a week is bad, because you will lose momentum. Choose a place that you won't have to worry about potential players overhearing you or about getting kicked out by NPs.<sup>10</sup> Dinner meetings are nice, as long as people can avoid being too distracted, and eating in is usually more productive than eating out.

Get a mailing list for your GM team, so that nobody accidentally gets omitted from any discussion. If you've got the technology, then set up archiving for mail to the list. While you're at it, set up a place online where people can share files—if you're handy with version control, now's a good time to set up a project archive (See Section 4.2.2).

Always establish an agenda before you begin a meeting. Even a rough agenda is better than nothing—you just want some idea of what the topic is so you can come back to it when you stray. If you set the agenda in advance, people can think about the problems beforehand and come in with some ideas ready to start.

At every meeting, somebody should be in charge of taking notes. I recommend taking notes electronically, because far too often hand-written notes are never transcribed and get lost. Somebody on your GM team probably has a laptop with wireless net, so you should be able to work just about anywhere on campus.

### 3.2 Philosophies of Writing

There are many ways to understand how your game is divided up and how the pieces will fit together. Different ones work better for different people, so I'm going to enumerate some of the main ways people understand guild games.

---

<sup>9</sup>You can have them more often if you're gung-ho or on a tight schedule, but beware of burn-out.

<sup>10</sup>Once you're far enough along, you can ask the Grandmaster to reserve a room for you if you want—we did this with *Caer Phaedria*.

Orson Scott Card, in his book *How to Write Science Fiction and Fantasy*, identifies four elements out of which all stories are composed.

- The **Milieu** is the world in which the story takes place, encompassing culture, physics, environment, and other large, lasting elements.
- The **Idea** is the information which is revealed throughout the course of the story.
- **Character** refers to the individuals a story may focus on, and the transformation of their self and their role during the course of the story.
- The **Event** is the disruption in the world that defines the bounds of the story, creating a state of flux that allows the story to occur.

Most written stories focus on one of these elements, drawing coherence from that centering. For example, taking three stories by Niven, *Lucifer's Hammer* is an event story about Earth being struck by an asteroid, while *The Integral Trees* is a milieu story about life in the accretion disk of a neutron star and *The Legacy of Heorot* is an idea story about the peculiar ecology of Grendels.

While individual characters often fit this categorization quite well, guild games as a whole are almost always event stories because of the plurality of perspective between players. With many players, it's a bad idea to have one be the focal character, it's difficult to sustain interest in an idea story for any but the shortest of games, and the milieu works only at a meta-level.

Within the Guild, players are often categorized into two types: play to roleplay and play to win, dividing players by whether their rewards for playing the game come at the player level or the character level.

- **Play to Win** players don't need much character, just a laundry list of plots and a splash to paint to explain why this collection is their list. They generally prefer simple plots (kill this person, find this item, do this mechanic, etc) and have a strong desire to win win win—"The ends justify the means." They often perform triage on their plots, punting plots that don't matter or are impossible to complete in the span of game.
- **Play to Roleplay** players care more about their character than the goals you give them. This does *not* mean that you don't need to give them plot. What this does mean is that you will need to give them well-linked and clearly motivated plots. They need to have more in-depth character sheets because they want to know the why of their character. These players are likely to punt plot if they feel their character wouldn't pursue it, but are often happy to pursue impossible goals if the pursuit is dramatic.

I prefer to parse the way that players pursue game rewards more finely and less perjoratively. Most players have a preferred type of game reward, and it's important that your app make it clear what types you're providing, and give players a way to tell you which they want.

- **Conflict** is beating another player in some aspect of the game.
- **Drama** is being the center of attention and entertaining other players.
- **Persona** is trying out what it would feel like to be your character.
- **Exploration** is figuring out the workings of the universe the GMs have created.

If you put a persona player into a part where they have to spend all their time fighting against conflict players, they're likely to have a terrible time. Similarly, the conflict players may be disappointed that the persona player didn't put up a better fight. Think about what sorts of play style your game is going to focus on, and how (or whether!) you are going to support players who prefer different styles.<sup>11</sup>

When game design begins, it usually gels around one of three aspects. Concentrating on one aspect is often helpful, because everything else can flow naturally from that piece when you understand it.

- **World Building:** What is the universe like? How does physics work here?<sup>12</sup> What sort of history informs the game? What are the big mechanics?
- **Group Structure:** What organizations are carrying out what plots in opposition to one another, and how are they balanced?<sup>13</sup>
- **Characters:** What personalities are you bringing together in game, and what sorts of histories, needs, and desires do they have?<sup>14</sup>

### 3.3 Other People's Ideas

Imitation is not only the sincerest form of flattery, it's also a great way to get a leg up on design. Never be shy about stealing other people's ideas when you're writing fiction—it's a long and honorable tradition—but think extremely carefully about whether you want the reference to be visible.

In our culture, a secret agent will always echo slightly of James Bond and Ian Fleming's work has lots of great inspirations in it. But if you say "Bond" then immediately everybody assumes gadgets, philandery, and shaken not stirred. This is very powerful, and can be used

<sup>11</sup>SIK games are proudly infamous for discarding persona entirely.

<sup>12</sup>This is often most important in games that involve magic: *Caer Phaedria* spent a long, long time developing this.

<sup>13</sup>*L5* is an extreme example of this: we brainstormed a set of groups, assigned them in a balanced way to a set of characters, then spackled together characters to try to justify the groups they were in. This led to some extremely strange characters.

<sup>14</sup>*Reality Check IV* was written around the collection of 50s archetypes we wanted in game—the mad scientist, the Hardy Boys, the gentle atomic monster, the visitors from SPACE, etc.

to simplify game-writing: tell somebody they're playing James Bond or Princess Leia and you barely need to write a character sheet, just tell them what their plots are.

Once you do this, however, you're stuck with precisely that character. If Princess Leia is secretly an agent of the Empire, it's jarring and people feel cheated. Worse, because this happens so often in Guild games, players often try to guess how the GMs have perverted each of the archetypes, and you're now talking to the players on a meta-level. In a parody game, where people know you're going to play with the archetypes, it works, but I strongly recommend against it for a serious game.

It's much less dangerous to blatantly steal world and groups than characters. The Star Wars universe is large, and MI6 has lots of agents nothing like Bond. Again, though, you're locked into your players understanding of the reference.<sup>15</sup>

Most of the time, you can steal something and conceal it fairly easily just by changing names and a few semi-relevant details. Rather than stealing Indiana Jones directly, I could make him Russian, call him Ivan of Astrakhan, and have him be an academic forced into the espionage service by the Politburo. He is being blackmailed because he knows too much about religion, yet that very knowledge is what they need him for, since they suspect that the Nazis are hunting for a novel war machine hidden in da Vinci's lost notebooks. Ivan, strong and fit from his childhood on the steppes, heads for Italy with his trusty pistol and a beautiful but likely treacherous KGB agent by his side. Although he still looks a lot like Jones to the writer, it's much less obvious to the players.

You can also steal a lot of useful ideas from other Guild games. An invaluable resource for design is the `a-archives` locker, available on Athena. Archived versions of most of the games ever run by the Guild are in this locker. I recommend at least browsing through a few to see how other people approach game-writing, and the wide range of products they've produced.

### 3.4 Forming the Universe

As you start building your game, there are some high-level questions about the universe that need answering. You can change these answers later if you need to, but it's good to have a working answer at all times.

- Is the game a recognizable genre? A genre lets people know what to expect and gives you archetypes to work with, but constrains your options.
- How long is the game? Typical lengths are one-night (3-4 hours), one-day (8-10 hours), three-day, and ten-day. The shorter the game, the less time your players have to be creative or to get bored.

---

<sup>15</sup>In *Nexus*, there were Jedi. Not the Star Wars type, because there was a whole other mythos and explanation, and grey Jedi who balanced the light and dark sides, but they were still called Jedi and the title really didn't fit what they actually were.

- Is the game expected to be high-death or low-death? Anything over about 1/3 of characters dying is a high-death game. Killing is exciting, and worrying about getting killed adds enjoyable adrenaline to the playing experience, but the more people that die, the more plots that die with them.
- Are there good guys and bad guys, or complicated morality? Villains are easy to write and generate lots of plot for games, but their plots are more predictable and less satisfying than a beautifully complicated knot where everybody's doing the best they can. It's also very difficult for bad guys to survive in game, because the good guys will mob them when it becomes obvious who they are.<sup>16</sup>
- Is the game silly? If not, puns and blatant rip-offs are completely banned. One Dr. Goodthighs and the whole affair reeks of silliness.
- Why is your game occurring here and now? Why does game start? Why will it end? Why can't characters solve their plots post-game?<sup>17</sup>
- Is game-space open or closed? In a closed game, there is some sort of barrier that prevents players and materials from moving in and out of game easily. This lets you make things like rope and hair brushes valuable game items, but requires a good explanation. An open game is often more realistic, but means players may try to import missile-launchers or fly off to Prague to check out a lead.<sup>18</sup>

If you're not using a cookie-cutter universe, spend some time working out the physics and, more importantly, deciding what parts you aren't going to work out. Although it's preferable not to kludge things, be sure to clearly identify your kludges where you do use them: there will probably actually be some physicists playing your game, they'll figure out the holes, try to exploit them, and be rightfully pissed if you only tell them it's a kludge after they've spent an hour working out the math.<sup>19</sup>

### 3.5 Game Dynamics

There are big game dynamics that you need to be aware of when designing your game, because the decisions you make will affect what type of game you're running. If you mis-

<sup>16</sup>In *Baldur's Gate*, the main bad-guy was toned down from the super-powers he had in the video game and placed in a paladin-rich environment. Unsurprisingly, he died quickly and pointlessly.

<sup>17</sup>Overwhelming force gets used a lot: at the end of the game the marines arrive, or the alien fleet blows up the planet, or the gates of hell open all the way. It's trite at this point.

<sup>18</sup>One-nights are almost always closed and generally need little explanation: either time is so short—who's going to leave the peace conference to pick up a hair-brush at the drug store?—or the scenario is inherently limiting—"You're trapped on an out-of-control space-ship", or pretty much anything containing the word "zombie".

<sup>19</sup>I'm not kidding. It can go the other way too: one of the *Caer Phaedria* GMs was a physicist, and with her fact-checking, we were confident enough that we could give out "figure out how the universe works" as an almost mechanic-free plot: a couple of players had an absolute blast working things out from first principles and coming up with experiments that they brought to the GMs to get results from.

estimate them, then you won't be able to balance the game properly, and some players will end up with parts that suck because they're boring or unintentionally doomed.

### 3.5.1 Spatial Distribution

Unless motivated to disperse, players will tend to cluster, for several reasons. They cluster because they're looking for the characters they need to talk to, because Guild games are a social activity and that's where the other players are, and because dramatic scenes they can watch are more likely where there's an audience.

Most long games embrace this and provide an explicitly labelled common room, which acts as a rendezvous point for players and a place to socialize when one doesn't have anything more pressing to do. I have never seen a 10-day game that could sustain other than one common room. If there are more, the players abandon all but one after a few days and go to the others only when they have to for a mechanic. If there is no common room, the players create one.

You also need to give your players reasons that they want to go away from the common room. Off alone in little groups is where the most interesting bits of a game happen: secret meetings, betrayals, killings, chance encounters. A player's adrenaline picks up and they start wondering if they trust the people they're with, or if their enemy noticed them leaving and is lying in wait around a corner somewhere. But players are pragmatic and won't leave safety without a good reason.<sup>20</sup> Some good motivators:

- Secret groups that don't want their existence or membership known
- Exploration to see what the GMs put in game
- Harvesting resources and looting treasures
- Stashing items to keep them safe
- Rituals that can only be done in a specific place
- Public events planned for a specific place
- Riddle trails, shadowruns, and tunnel systems

If you succeed too well in getting your players out of the common room, however, it'll become a wastland occupied only by stragglers trying to rendezvous and run off to the wilderness again.<sup>21</sup>

Public events are one way to balance players dispersing and coming together, plus provide some entertainment in the bargain. Good ones become rendezvous points that people build

---

<sup>20</sup>*Maelstrom* disallowed combat except in the basements, so the players simply never went to the basements.

<sup>21</sup>This happened to *Nanopunk: Tokyo*, where most of the game action was off in the shadowruns.



their evenings around, scheduling their secret meetings before and after.<sup>22</sup> Bad ones are too frequent or un compelling or too long and cast a pall over your whole game: nothing's as depressing as being one of the few people to attend a public event that everybody else has abandoned.

Go out and actually walk around the areas where you're thinking of running your game to refresh your memory for scale, atmosphere, and odd features of the buildings. MIT campus is large and varied and some parts have gotten very weird during the continual cycle of renovations. Think about your alternatives as you explore the space, if you can't get the rooms you want.

### 3.5.2 Time Distribution

Another way that smart players can wreck your game is by being too strategic about time. Given something risky to do, a lot of players will rightly decide to either get it over with immediately or put it off until the end so nobody can screw them before end of game.<sup>23</sup> How do you encourage people to act before the end of the game? This is especially a problem in 10-day games, where the weekdays tend to be very flabby, but can be bad in one-nights too.

Correcting this is much harder and more delicate than spatial distribution, because you don't want to make the players feel railroaded. Early deadlines<sup>24</sup> move resolution up, but don't change the dynamic. A better approach is a carrot for possession<sup>25</sup> or a stick like increasing likelihood of betrayal with time. Another way is to have a resolution which extends over a long period of time, as often happens in brainwashing plots, so that the players start early so they can build towards the finish, although this often makes the finish anticlimactic.

### 3.5.3 Combat

Most games have two types of combat: martial combat and ranged combat. If guns are effective, martial combat generally goes completely unused. This is really hard to balance against, because almost all martial combat systems require significant time to resolve, causing an attacker to lose the advantage of surprise and end up surrounded by allies while he's still trying to dispatch the first victim.<sup>26</sup>

---

<sup>22</sup>In *Macondo*, everybody was Catholic and went to Mass every night. It was brief (usually about 30 minutes all told), the two PC priests put in real work on the sermons and switched off who was leading.

<sup>23</sup>In *Caer Phaedria*, we invested a lot of effort building a revolution system, but forgot to give an incentive for being in charge except at the very end. As a result, all the revolutionary groups held off until the end, when deaths had shattered many of them, so there was only a single abortive attempt and the status quo won.

<sup>24</sup>Like *Macondo's* Friday SIK game

<sup>25</sup>*Reign of Terror* gave significant rewards for taking over the government, even if you were tossed out a couple days later

<sup>26</sup>The *Darkwater* combat variants are the conspicuous exception, and work by essentially turning martial combat into ranged combat. This sometimes defeats the purpose.

To deal with this, ask why you've got the two types of combat. You don't have to have both ranged and martial combat, of course. Ranged combat is easy: if nobody's got guns, then there's no ranged combat, and there's lots of reasonable excuses to take everybody's guns away. It just rarely happens because players really love shooting each other, and GMs like to make the players happy—plus guns are simple and *scary* after you've played a couple games.

I think some games have a martial combat system just because they can't think of a good excuse to disarm people of their fists. If there's no point in it, though, why use it? Just add to the rules, "There is no martial combat in this game. None of you is particularly good at hitting other people, so if you get into a fistfight you'll just end up with bruises and skinned knuckles." Then if there's a couple people who actually are competent, you can give them special "I beat you up" ability cards.

**Martial Combat** If possible, use a martial combat system directly out of the Standard Rules. Players are at least vaguely familiar with them, and they can usually be fudged to get the effect you want with relatively little effort. Even if you do this, though, there's several to choose from, and you need to figure out what you want out of martial combat.

Often, the real reason you want to have martial combat is something other than killing people, and killing people is a side benefit it can be used for on occasion. Often, there is some system of duels for honor, or a boxing tournament, or some other thing where the combat system is being used to show off in front of a crowd. If this is the case, you can make it as showy and complicated as you want, because there's not really any time-pressure on the combatants. If they actually want to kill somebody, they'll get a gun.

Think about how your combat system will wear over the course of your game, particularly for ten-days. If the duellists have to fight 50 times over ten days, the shiny-new-toy aspect of the mechanic will be entirely gone.<sup>27</sup> If combat is a boring game with lovely paint, they'll be bored. If the underlying game has some interest, however, it's likely to be complex enough that they'll have a hard time playing the first couple combats. Balancing these two aspects is tricky.

If you want martial combat to have a real application for characters to kill one another, however, you need to keep it as fast and simple as possible. Rock-Paper-Scissors is good because everybody remembers how to play it and it doesn't require pulling cards out of your pocket; Darkwater is also good and fast. You may also want to dumb down guns, making them limited shot, slow reloading, etc—no martial combat system can survive the application of 20-shot magazines of discs.<sup>28</sup>

Another balance issue is the relative ability of characters. At one extreme, everybody is essentially equal in martial combat and who wins a fight depends on what weapons they've brought to it. At the other extreme, every character has a rank and can automatically beat

---

<sup>27</sup>Some people fought more often than this in *Caer Phaedria*, trying to improve their skills.

<sup>28</sup>In both *Reign of Terror* and *Caer Phaedria* guns were bulky (thus nonconcealable) and represented by dart guns that were loaded by passing them around your waist several times. This meant that if your first shot didn't hit, you probably wouldn't get another off, one way or another.

everybody with a lower rank and is automatically beaten by everybody with a higher rank. You need to decide how you are going to balance between these two extremes: the more that combat is ranked, the less likely anybody but the top few people is to attack another person without a good estimate of that person's skill; the more that combat is equal, the less reasonable it feels to have a karate master lose to a feeble old scientist.

Unfortunately, every martial combat system has to deal with what happens if more than two people are involved in a fight. This is often where the rules become most insane and tangled, and it almost never happens. Keep your rules for this as simple as possible, even at the expense of realism and rationality, or you're just going to end up with a bunch of confused players. The Standard Rules combat systems have some reasonable ways to deal.

**Ranged Combat** Although ranged combat is considerably simpler than martial combat, you've still got some decisions to make when figuring it out.

- Will there be bullet-proof vests (or equivalent abilities) that allow a character to ignore the first few shots? Spy games often feature three-shot vests that are useless after they've been hit three times, or a "super-spy dodge" ability that lets a person ignore one shot per day.
- Does it take one hit or several to drop a person? While multi-shot gunfights are cool, they require math. It's also notoriously difficult to count how many discs you've been hit by, since they're both hard to see and have relatively light impact.<sup>29</sup>
- Are people knocked to the ground by the first shot that hits (regardless of vest) or can they run away? Knockdown is often added to give attackers a chance against a person wearing a vest.
- Does it matter what sort of gun you are shot with? It's relatively easy to tell darts from discs from ping-pong balls, but much harder to tell which of several guns the dart that hit you was fired from.<sup>30</sup>
- Does it matter where the shot hits?<sup>31</sup>

**Interactions** Besides figuring out how they work on their own, you need to think about how the your combat mechanics will interact with each other and with the rest of game. Here are some points you will want to consider:

---

<sup>29</sup>This is why some SIK games have switched to using disc guns as flamethrowers, where all that matters is that you've been hit by at least one disc.

<sup>30</sup>*Macondo* differentiated guns by the type of PVC pipe attached to them: pistols were disc guns, .44 magnums had a 12-inch pipe, uzis had an 18-inch pipe, and assault rifles had a 2.5-foot pipe of a thicker gauge. Each type did different damage, so combat was occasionally confusing, but usually not and it looked really cool and big guns were scary and easy to see.

<sup>31</sup>This used to be common, leading to lovely stories like the Politburo accidentally executing James Bond with a shot to the back of the head in *Midnight in Moscow*

- What happens when you shoot a person in martial combat? If you shoot the person you shoot, then a team of two can execute an enemy by having one martial attack and the other shoot at point blank range while the victim is “stuck”. One solution to this is to have the shot either affect everybody in the combat or a random person in the combat.
- Can you shoot a person who just called martial attack against you? Does your gun already have to be out?<sup>32</sup>
- Can you draw a weapon when martial attacked? Usually the answer is no, to favor the prepared and make a drawn weapon feel more threatening.
- Do martial and ranged combat draw from the same resources, like hit points? Does the bullet-proof vest make it harder for me to punch you? Does shooting you make it easier for me to knock you out, or hitting you make it easier to gun you down?
- How hard is it to recover from being hurt in combat, and are there long-term effects from combat? Are they different for martial combat and ranged combat? At one end of the scale, the hit points you lose in martial combat return the instant the combat is over. On the other end of the scale, you get shot once and you’re stuck with a limp or other reduced abilities for the rest of game.
- In most games, losing a combat of either type ends with you unconscious and helpless before the victors for some amount of time. What happens if an unconscious person is shot, or shot again? My preferred way to handle this is to have unconscious from shooting and unconscious from beating be different and have their clocks run at the same time on the victim. It’s a little odd, but means you don’t have to have a rule for how they interact.
- Are there other combat-like mechanics that you need to worry about interacting with the combat system? Common examples are waylay, pick-pocket, and mobs.

#### 3.5.4 Mortality

Death is a difficult issue in Guild games. On the one hand, killing is an adrenaline rush, the possibility of being killed can make just walking down the corridor exciting, and if you can’t kill your enemies there’s often no good way to stop them from succeeding. On the other hand, killing shreds groups and plots because of the way that Guild games tend to be interwoven: if a character is involved in four plots and one gets him killed, the other three plots are now missing a person too.

Over a certain critical threshold of deaths (somewhere between 1/4 and 1/2, in my experience) the game will start grinding to a halt because almost every plot has lost key people and

---

<sup>32</sup>In *L5*, you could shoot if you did it within 2 seconds. The grossest combat monster in game was gunned down hard by a quickdraw.

the overall resources available to game have been reduced. Deaths also reduce the number of players hanging around game, which reduces how much fun it is to be hanging out and socializing: not only are there less people, but the people who are left become skittish about being easy to find.

I usually aim for a kill rate somewhere between 5% and 25% in the games I write, although some have aimed as high as 50%. It can be hard to predict how many deaths will happen, because so much depends on the individual players and the random factors in how the game dynamic develops.<sup>33</sup> While you can't control how many people die, you can at least control the factors which tend to promote or suppress killing.

Death in guild games depends on four factors—predictability, likelihood of being caught, likelihood of an accidental kill, and explicit goals—plus a healthy dose of player arbitrariness. Some players are just more likely to solve their problems by gunning people down, and if you know who they are, you should make sure they get appropriate parts.

When a player can predict the outcome of a combat, they are more likely to try to start one. This is one of the reasons that martial combat doesn't get used much in game: you can usually tell whether a player has a gun, and can always tell when they've got it out; you can't tell whether the little old guy you're going to roll is Mr. Miyagi. Players planning to take down a target will usually assume the target has whatever defenses are available, as well, so the stronger the bulletproof vests available, the harder it is for people to kill somebody. This applies whether or not the victim has a vest, because the only way to find out is to start shooting. Any system that gives an advantage to surprise will also be more deadly, since players can usually contrive to attain surprise (and that's a big part of the fun of stalking a victim).

Another big unpredictability inhibitor is the presence of wierdshit in game:<sup>34</sup> if there's wierdshit, players will be nervous that the person they gun down will unexpectedly turn out to be immortal<sup>35</sup> or able to blast them with lightning or something of the sort. Even if the players can figure out all the defenses of their target, they may have a hard time killing them just because they are wierdshit.<sup>36</sup>

Likelihood of being caught depends on how long the combat lasts,<sup>37</sup> and whether a victim is likely to be able to ID you as the killer. Darkwater martial combat tends to be more deadly than other forms largely because you can “point and shoot” with it just like a gun, and don't end up spending a long time standing there with your victim. Likewise, if you have to put a dozen shots into a victim, you are more likely to get caught than with a one-shot-one-kill

---

<sup>33</sup>One run of *Krazny Oktyabr* had an 85% kill rate. Another had zero deaths and only two or three discs fired.

<sup>34</sup>Wierdshit isn't the same as magic or superscience. Wierdshit means you don't know what you're dealing with until it bites you.

<sup>35</sup>This happened to several people in *Macondo*.

<sup>36</sup>In *Nexus*, the Dread Pirate Warlock accumulated enough wierdshit toys that he was only able to be taken down by a group of 10 after more than an hour of contingency planning for how to run the combat. The next day his allies used other wierdshit to bring him back from the dead.

<sup>37</sup>*Nanopunk Jersey* had a 30-second killing blow, which also made it significantly harder to kill people.

gun. Forensics and wierdshit also up the chance of being caught, either by letting the dead identify the killer, forcing the killer to spend extra time disposing of the body properly, or having the dead turn out to not be dead after all.

Details of your combat mechanic can increase the deadliness of your game by making it easier to for defeat to turn into death, intentionally or otherwise. Overkill rules are one way this is often done: more bullets, or bullets that hit the wrong place can kill off a character who was only supposed to be captured. Similarly, if wounded characters bleed out without appropriate medical attention, combats are much more likely to become deadly, doubly so if they don't regain consciousness without appropriate medical attention.

Except for PC arbitrariness, killing is actually not the preferred way players try to solve most plots, since the person you think is your opposition might turn out not to be, and even turn out to have vital resources for one of your other plots. If you don't put explicit goals like "stop by any means necessary" or "kill these people", your non-bloodthirsty players will often not resort to killing, which may or may not be what you want. Beware also that you don't accidentally set up a character for death unless you mean to—this particularly applies to characters introduced partway through game, who may be taken for NPC kill-fodder.<sup>38</sup>

One last consideration is whether you want an "aftergame" for dead characters. Death is less of an issue if they can continue to play as a ghost, a zombie, or an NPC. Be warned, however, that an aftergame doesn't make up for losing your primary character. Players will also sometimes be upset at being killed, and it can be dangerous to send them back out again to play a different part around their killer.

### 3.5.5 Captivity

Prisoners are a problem with no good solution in Guild games. Characters rightly want to lock up other characters who they're suspicious of and have probable cause to try to keep away from the nukes, and it's hard to give a reason why there's no way to keep somebody prisoner indefinitely. A player whose character is being held captive rightly gets extremely bored and stops having fun very quickly.

The most common solution is Schroedinger's rope. As long as there are other players nearby who are hanging around doing things and effectively keeping the prisoner company, they can't escape. If the captors go away for five minutes, the ropes loosen and they escape. This can be applied with slightly different paint to almost every situation, but means players are more likely to kill people they are suspicious of, since there's no safe way of holding them captive. You can also have an out-of-game jail so the player is sent out of game when their character is processed in and allowed to return when the character is released. Either way, the other PCs are usually deciding to end the game for that player.

---

<sup>38</sup>This fate befell the Spaceballs in *Airplane 2*. Dark Helmet's goal was to invade the ship, scare people so they'd leave him alone, and watch the in-flight movie. Instead, the crew, who'd been battling vespids, killed him 30 seconds after he appeared and tried to be menacing.

Unconsciousness is like being tied up, only worse, because you can't even bitch and moan at your captors.<sup>39</sup> You can't really do anything to keep players from holding another unconscious if there's any way they can make it start. Some games give an unconscious person the option of falling into a coma and dying if they are kept unconscious continuously for more than 30 minutes, as a way of enforcing politeness on the part of their captors.

### 3.5.6 Love-Fest vs. Knife Fight

Love-fests are when the players work out a win-win scenario (or at least one where nobody explicitly loses) and adopt it by consensus. It makes the players happy to have avoided losing, but it's not very exciting and dramatic. In a knife fight, on the other hand, it's not over until at least one faction loses badly. Love-fests are also closely related to 20th Century Morality, the thing which causes the players in a fantasy game to concede to the demands of the peasants, and the players in a sci-fi game to replace the Dread Emperor with a representative democracy rather than a Space Tyrant. Both are extremely difficult to prevent.

Some ways to promote a knife-fight include keeping information secret, so people don't know enough to work out compromises they can trust, writing in explicitly unresolvable animosity between parties, having hard limits for losing and not enough resources to fulfill anybody's unless somebody loses badly, and using fear as a motivator rather than greed. There is no cure for 20th Century Morality besides suspension of disbelief.

### 3.5.7 Good Guy Mobs

Good Guy Mobs are the most obnoxious manifestation of group-think in a Guild game. They tend to form around a charismatic leader or established power, who decides that something is Bad and people should team up against the bad thing. Peer pressure and McCarthyism bring people in: if you're not joining the fight against Bad and you don't have a sterling excuse, you must be allied with Bad! Pretty soon, the GGM has formed and tends to move around in a paranoid clump, suspecting any member who leaves it.

People start sharing their suspicions about what other Bad things might be going on in game, and the GGM starts having mission creep and becoming an all-purpose plot suppression device. They're not even particularly effective, because any smart villain will join the GGM and try to persuade it to kill the opposition and help with their evil plans.<sup>40</sup>

GGMs are closely tied with epic-level problems, because higher epic level always trumps

---

<sup>39</sup>The night the gods died in *Showdown at the Golden Gate*, one of the two ordinary humans in game was accidentally swept up in the initial raid. As the two groups fought back and forth and the area changed hands, whichever side was in possession kept shooting him with tranquilizer every 5 minutes because he wasn't one of their allies. The player sat there unable to talk or participate in any way for most of the night.

<sup>40</sup>In *Showdown at the Golden Gate*, the aliens joined the GGM and got it to help them build their device for blowing up the sun. They then set off a solar flare, effectively destroying the world.

lower epic level. It is an established guild maxim that CIA and KGB will team up to fight SWORD, and SWORD and SHIELD will team up to fight Chthulu. For this reason, epic plots are unbalancing and tend to unite a team of unlikely allies against the threat, forming the kernel of a GGM.

Epic plots also tend to make a small group of characters into the main characters of the game, which sucks for all of the other players: if your game runs the gamut from janitors to gods, the player cast as the janitor will probably feel insignificant and wonder why you made the part a PC at all. It can be done, but it's a very difficult balancing act.

Epic plots tend to be a symptom of lazy writing: it's always easy to come up with something bigger and badder than ever before, as any fan of Golden Age scifi or anime surely knows. Moreover, the characters involved in epic plots end up with shallow motivations: it's easy to understand why somebody wants the world to not blow up, and they're just not going to worry about whether they keep their job after it's resolved.

### 3.5.8 Suspension of Disbelief

Players live in the meta-game, while characters exist in in-game reality. It's hard for players to suspend disbelief all the time, especially in a 10-day, and it's especially hard for a player to separate in-game knowledge from meta-game knowledge. So while James Bond doesn't know who bonked him over the back of the head, Jake knows damned well that it was Joe, who's playing Jaws. As much as possible, keep in-game and meta-game information parallel, so that you never have to tell a player, "Your character doesn't know that."

You can help suspension of disbelief by adding lots of good paint. Paint is the in-game explanations that go along with meta-game rules and information, and when it fits together well it's easier for players to know how to relate to their experiences in character. Painting things well is a difficult and subtle art.

Another good tool are the rules of three and five. The rule of three is that people like triplets. Triples are memorable, satisfying, and feel both complicated and graspable. For things that should be ungraspably correct, use the rule of five instead: systems with five components are hard to grasp and figure out unless they can be reduced to smaller components (beyond five is usually diminishing returns).

Many GM teams also like to lay meta-tricks and meta-traps for players, so that somebody who makes assumptions based on the fact that they're playing a Guild game will get screwed over or waste time running down a blind alley.<sup>41</sup>

---

<sup>41</sup> *Caer Phaedria* contained an easter-egg that was also a meta-trap: if you investigated the basement of the Citadel you could walk down the stairs in the Green Building five times to get to a mysterious silvery pool. The pool was tied up in the weird physics of the universe, but if you touched it you died instantly. Nobody had any in-game reason to investigate it, and nobody ever did.



## 3.6 Mechanics

The most important consideration when designing your rules and mechanics is load. How much GM time will the mechanics consume? How many things will the players have to remember at once? How thick will your rules be, and how hard to sort out the parts that matter from the parts they can look up when they need them?

The ideal mechanic requires no GM intervention and is completely intuitive for the players. Simulating ranged combat with disc guns is an excellent example: one player shoots at the other, and when you feel a disc bounce off you, you know your character has been hit.<sup>42</sup>

### 3.6.1 Players vs. Mechanics

On the player side, the key limiting factors are how many numbers a player has to remember, how many responses they need primed, and whether they have to do math under stress. Categorize your mechanics into always active (e.g. combat) and swappable (e.g. economy or riddle trails): the always active ones count as an immediate burden on your players.

You can safely assume that recognizing when you've been shot, responding to "Martial Attack" with "Martial Defense" (or generally "X Attack" with "X Defense") and dropping when waylaid are automatic and without burden for anybody who's played a couple of games.

Most things that make it into the Standard Rules are this way, although you'll find lots of little holes in player's knowledge—for example, few people really remember the Phi martial combat system or whether dead bodies are 2 or 3 dots bulky.

Anything else counts as a burden on the players, and the lower the burden the less likely your carefully designed mechanic will be screwed up. This is a good place to apply the rule of three: no more than three numbers and three responses that need to be remembered with less than 30 seconds notice.<sup>43</sup>

Another good trick is non-interacting mechanics. It's OK if a player is unconscious two different ways at once if that means they don't have to remember a special rule for what happens when shot after being bonked on the head.

Math is completely forbidden for time-sensitive mechanics. You will be surprised at how little math people can do under stress. Almost everybody can count to three, many can count to 10, but only a few hardy souls can subtract 4 from 13 during a gunfight.

Off-line mechanics are less of a problem, but you still don't want to load more than a couple

---

<sup>42</sup>Not to say you can't make it difficult. *Showdown at the Golden Gate* had about a dozen different types of projectile damage which had different effects on different characters, so you needed to shoot gods with tranq, vampires with crossbow bolts, werewolves with silver, etc. I like it when one disc means you fall down and start bleeding—it's so easy to know what to do.

<sup>43</sup>Peter Litwack and Ken Clary promulgate the "fall down and start bleeding" solution. Whenever confused, a player should fall down unconscious and start bleeding. Often, that's what they should be doing anyway, and while unconscious they can read the rules to figure out what actually just happened.

on any given player. Remember: time playing with a mechanic is often time not spent in character interacting with other characters.

### 3.6.2 GMs vs. Mechanics

There are two types of mechanic burden on the GMs: writing burden and run-time burden. You need to be worried about both.

When designing a mechanic, think about how many players will be using it. The amount of GM effort needed to write and run the mechanic should generally be proportional, since time you spend working on a mechanic is time you aren't spending working on the parts of the game that other players get to play.

Writing clear rules is, of course, vital. You should draft them more than once and expect writing a set of rules for any but the most trivial mechanic to take at least four hours. Filling in the numbers and writing the flavor text for a mechanic must not be discounted as easy: unconstrained, this can easily expand to fill up several full days of GM time for a single moderately complicated mechanic. Programming is even worse: you should expect a system that involves writing code to take an additional order of magnitude in time. When possible, write mechanics that can be produced using the existing formats in the Template (or minor variants thereof)—playing with formatting counts as coding.

Anything that requires a GM to interact with a mechanic while the game is running should be treated with great suspicion. You want extremely low GM time consumption from mechanics because you want to save time to deal with the emergencies that come up during game. Every point that requires GM intervention is a point where your game can get stuck waiting for a GM to be available.

Sometimes you can't avoid it, but you'd better at least be cognizant of what you're freighting yourselves down with. A few of the common things that eat GM time:

- “Ask a GM what happens.” This little tagline adds a sense of importance and ominousness to anything it's attached to. Besides being useful for the stuff that really *needs* a GM, it's great for laughs and meta-traps. It also leads to players saying, “We're going to go open the portal soon, can we have a GM go to building 66 with us for the next five hours while we squabble about whether we're actually doing it?”
- A mechanic where you respond to queries.<sup>44</sup> Even if you're doing written queries rather than verbal, so you can process them during slack times, and if you're responding using precalculated information so you don't have to be creative, it's still going to take about five minutes to process a query when you count in figuring out the player's handwriting, finding the right answer, and writing the response. A couple queries per day is fine,

---

<sup>44</sup>*Maelstrom* had a background check mechanic, but we got so behind on responding that it became nearly worthless.

but if you've got 50 players who can each submit one query per day, that's more than four hours of dedicated GM work to keep up.

- Mechanics with hidden information. Economies and wargames are the classic examples here: you don't want the players min-maxing or knowing who controls the Barbizon Armada, so you have a GM do the processing instead. Even when well designed, one of these mechanics generally takes at least an hour to process, and often much more.
- Maintaining state in game space. Anything you've put out somewhere in game space can vanish on you. Dots lose adhesion and fall off the walls, grad students down packets out of spite, and physplant grunges randomly. Players know this too, and whenever they can't find something, they'll assume it's been grunged and come running to you to check. In a widely dispersed game, it can take half an hour to check a dot, and even worse if you have to trek from main campus over to E51. Big packets are usually easier, since they're more obvious and less likely to produce false alarms, but they usually need more work to reconstruct when they're grunged. Moreover, you can accidentally duplicate items if you recreate something that somebody stole before the packet disappeared!

The worst part is, if you screw up and can't keep up with the players' needs, they'll adapt and play the game anyway—but part of your game will be getting written off as useless, and the people who've been counting on those plots will lose out and it'll be *all your fault* and they'll resent you for screwing up writing the game so that you unfairly doomed them to lose. And they'll be right.

### 3.6.3 Other Design Hints

Here are four important heuristics to test your mechanics against—the answers to all four questions should be no.<sup>45</sup>

- Can an adrenalized player gain advantage by hurdling a table and shoving other players aside?
- Can a delusional player re-interpret the mechanic to screw over other players?
- Does an optimal solution for the mechanic exist?
- Can a clueless player screw over the person initiating the mechanic by having to look up how to respond?

A good sanity check for mechanic complexity is to have a non-mechanic-writing GM read the rules without help from the mechanic-happy GMs. If they don't get it immediately, it's too complicated.

---

<sup>45</sup>These are properly named for four infamous players, but I'm feeling polite right now.

Another consideration in mechanic design is player skill versus character skill. WYSIWYG mechanics using player skill are often simpler and more intuitive. Mechanics which factor in more character skill, on the other hand, give players more chances to be somebody they aren't normally. Although it's drifting away from this, some Guild games have even had rules keeping stash sites low so short players can find them and prohibiting running so slow players can play fast characters.

Think about whether your mechanics will depend on getting particular rooms, or access to MIT-specific resources. Alumni and non-MIT players usually don't have access to the internet on campus, let alone Athena. If you need a particular room, reserve early and have a backup plan. If you must have these resources, you can try to cast around the limitation or make provisions to give access to non-students.

Will the mechanic make use of physreps? Phyreps are wonderful for increasing immersion and adding WYSIWYG to a mechanic, but can be in short supply or a pain to lug around.

Finally, remember that at MIT you're likely to run into wildly ranging skill levels, including some people with frightening abilities. There are players who can do things like read substitution ciphers without translating them or factor large numbers in their heads.

### 3.7 Character Design

Every character has a personality, a history, relationships, resources and plots.<sup>46</sup> Some characters start as a skeleton of plots and groups (relationships) and grow a personality and history to bind them together. Others start with a history or a personality and evolve the rest from that base. No character should ever start as resources and grow the rest: it's demeaning to be a living, breathing mechanic.

It's a good idea to develop all of your characters the same way. Plot-based characters have lots of things to do, but can end up with incoherent characters, while personality or history-based characters are believable people but can be completely without things to do.<sup>47</sup> Developing all your characters in the same direction makes it less likely you'll overlook potential deficits or make a worst-of-both-worlds character: a strong personality and a set of weak plots that have nothing to do with it.

The most important thing in designing characters is that each character should have the potential to be the star of the game. If they can't be the star, they aren't a player character, and maybe shouldn't even be represented in your game. Most games have hordes of invisible bit-part NPCs around, and so if you make the janitor a character, you should be ready for everybody to assume that the janitor is not what he appears and to try to figure out what he's really up to—after all, if he's just a janitor, nobody would want to play the part.

Every player should get to feel important. Every character should be a favorite of the GM

---

<sup>46</sup>Except in SIK games, where they generally get a plot and maybe a personality if they're lucky.

<sup>47</sup>In *Sandman*, Sexton was an interesting and believable character with little personal plot and an apathetic personality. The player was bored stiff for 10 days.

team, and one that at least one of the GMs would be willing to play. That doesn't mean they have to be homogenous, though: the president usually is more important than the scruffy arms dealer, though it might turn out that the arms dealer brings down the government.

The style of characters can range wildly, to accommodate all sorts of players. In addition to the player types discussed in Section 3.2, for ten-day games it is important to be aware that some players are hard-core, and will stay up around the clock running riddle trails for the first 72 hours of game, while others are primarily social and will come in for a few hours in the evening to roleplay with others and advance their plots. Your game should be able to accommodate both types, without making the social players automatically lose to the hard-core players. In my experience, you can expect about 10% scarily hard-core players, 10% quasi-punting social players, and the rest somewhere in the middle.

Two common problems to be wary of: whipping boys and super characters. These are the result of group-think that happens to a GM team over time. A whipping boy is a character that is the butt of half the jokes in game, the character that the GMs laugh at or want to make despicable, and so every time an additional bit of crap becomes available they stick it to the whipping boy. The super character is the result of the same process, but positive, accumulating nifty abilities and toys. A whipping boy isn't fun to play, and a super character isn't fun for anybody else to play with.

### 3.7.1 How much plot does a character need?

The longer the game, the more plot each character needs. In a two-hour game, even a character with no plot at all can be fun to play. In a ten-day game, a character without half a dozen things to do can end up bored stiff.

How do you measure plot? Let's start with some negatives. Shtick is not plot. Personality is not plot. Love plots are not plot. Backstory is not plot. Avoiding somebody or something is not plot. Roleplaying is not plot. These are all great to give to characters, but they're the icing, not the substance. Plot is the work a player can do to accomplish her character's goals.

Jeremy Brown identifies six types of plots by the target of the plot:

- RTI hunt
- Unique item hunt
- Information hunt
- Person hunt
- Riddle trail
- Diplomacy/committee (convince masses)

Unfortunately, this doesn't actually say much about what a player is doing with the plot, only the criteria for a win. I prefer to analyze plot in terms of how a player goes about accomplishing the goals:

- Organizing people: also known as herding cats, the difficulty of organizing a group of people depends on how many other things the other players are trying to do, how specific the membership of the group needs to be, and how many people are in the group. Organizing brings attention to a player and puts them in control of a situation, which is a reward in itself for some players.
- Collecting resources: the scavenger hunt aspect of a game is fun when there is some reason to hope it can be accomplished, and is a mechanism for generating conflict when there aren't enough resources to be used.<sup>48</sup>
- Going places: not generally fun in and of itself, it is nevertheless a key component and time sink in most plots. The excitement in going places is when a player gets to go somewhere that's not normally accessible and might be the first to loot it or see secrets of the game.
- Solving puzzles: hard under time pressure, relatively straightforward in the common room or sitting at a computer. Puzzle solving is an intellectual challenge most players enjoy in moderate doses.
- Warfare: popular as the big adrenaline source in games. Warfare is anything that has the players doing tactical thinking, including non-lethal things like trying to capture or subdue a character, and worrying about being ambushed by others.
- Gathering information: this can be from other players, which most players are always trying to do anyway, or from locations and other sources in game. Figuring out secrets is a big rush for most players, but many people find it hard to effectively pump information from other players.
- Negotiation: wheeling, dealing, and persuading other players who aren't necessarily on your side. Negotiation can be very hard to do unless the other side has a good reason to come to the table.<sup>49</sup>
- Performance: whether public events or rituals, performance is about acting goofy in front of other people and getting attention for it.

Plots can be analyzed in terms of what time investments are required to complete them. For example, a typical riddle trail requires 1-2 hours of mixed puzzle solving and going places

---

<sup>48</sup>In *Peace in Our Time*, most of game was massacred in the common room by characters searching for Grey RTIs.

<sup>49</sup>For example, "blood drive" plots often fail simply because there's a significant chunk of Guild members who won't give out blood/whatever samples because the plot's usually evil.

per dot, while a typical shadowrun requires about 2 hours of work, split between organizing people, collecting resources, going places, and solving puzzles.

Plots come in two types: open ended and laundry list. A laundry list gives a set of things to accomplish which will fulfill the plot;<sup>50</sup> an open ended plot has the goal, but leaves it up to the player to decide how to achieve it.<sup>51</sup> The best open ended plots have goals that you can exceed as well as accomplish, so that a player never ends up done, with nothing to bother sticking around for. Give players a mix of open ended and laundry list plots: the laundry list plots are nice because they don't end up stumped, and the open ended plots are nice because they let players be creative.

A good rule of thumb is to give each player at least two plots that they are in charge of—a personal plot, leadership of a group, or leadership of a particular plot. Then you should add more plot in which they are a participant, but don't have to be the leader. Knowing the rules, you can bend them of course: some characters can get by with a single simple open-ended plot with the right player and a good push, especially in short games,<sup>52</sup> others can be put in charge of half the world and enjoy the madness.

Too much plot can be as much of a problem as too little,<sup>53</sup> but is usually less of a worry, since players will triage things when necessary. Also, many players enjoy feeling caught up in a whirlwind of too many things going on at once. In general, don't worry about giving people too much to do.

Be sure that nobody's game depends on one other character. Besides dying, the other character might be busy with other things or even just be played by a lame fool, and now you've ruined somebody else's game in the bargain. Generalizing this slightly, you should check the structure of your plots to be sure every player has at least some doable plots after 10% of game is killed, and that no portions of game are completely detached from the rest of game.<sup>54</sup> Remember that most plots will fail, for one reason or another, and that people need to be entertained by the few which manage to succeed, either directly because they participate, or indirectly because they are involved in the public fallout of the plot.

---

<sup>50</sup>e.g. You want the amulet of Yendor. To get it, you must locate the Mines of Moria, penetrate to the 12th level, and defeat the Balrog.

<sup>51</sup>e.g. You want to legally own the amulet of Yendor, which is currently owned by the Duchess. Figure out some way to get her to give it to you or sell it to you.

<sup>52</sup>In *Nanopunk: Tranquility Base*, Billy was a stowaway kid who wanted to get involved in something exciting and had no resources except a gun. The player, known for his ability to make shit up, kicked ass, took names, and made out like a bandit.

<sup>53</sup>In *Caer Phaedria*, characters had an average of eight bluesheets, and many plots collapsed because people were too busy. Since players prioritized differently, this meant that some players had all their plots collapse.

<sup>54</sup>In *Macondo*, the immortals plots had nothing to do with anybody else's plots, and so they didn't get to really interact with most of game.

## 3.8 Sanity Checks

Sanity check things with the other GMs and any outside people you can. Group-think is easy, and seeing the holes in your brilliant idea is hard.

One excellent heuristic, which I believe comes from Peter Litwack, is that any sentence beginning with “wouldn’t it be funny if we made the players...” is always wrong. Another is that you can’t stop players from doing something by making it annoying: if they want the reward, they’ll do it anyway and curse you the whole time.

I tend to believe that really incredibly bad ideas have to be expressed somewhere—it’s like lancing a boil. Stick a `plots.rejected` file somewhere in your notes and put the really terrible ideas here.<sup>55</sup> Get ’em down where you can laugh at them and move on, otherwise they’re going to keep bugging you and trying to wiggle out.

## 4 Writing

Once the design has shaped up and you know mostly what your game is going to look like, it’s time to start writing. Don’t worry about your design being perfect before you start writing—actually realizing the design will always involve lots of tweaking, filling in details, and often one or two serious changes. Don’t start too early, though—nothing’s more discouraging than having to throw out or completely rewrite half a dozen sheets.

### 4.1 Organization

Writing a game is a monumental task, so you need to break it down into bite-sized chunks. Make a to-do list of every sheet in game that needs to be written, broken down into sections, with a place to write the owner and status of the sheet. My sheets generally go from blank to `claimed` to `draft` to `checked` to `final`.

In addition to all the sheets, you’ll want to put down other tasks like assigning stats, writing abilities, making riddle trails, and gathering props. Some of these will have their own sub-lists as well, like a prop checklist.

This may seem really anal, but you need it for all but the smallest games. For one thing, every time you upgrade the status of a sheet, it’s a feeling of accomplishment, and it gives you a concrete sense of progress—as well as a realistic assessment of what’s left to go. Also, a Guild game is too large to fit into your head, and all these lists will be a reference for you and your fellow GMs to cross-check each other and make sure nothing gets forgotten.

As you write, make sure that everything with secret information has an updated GM version

---

<sup>55</sup>*Nanopunk: Jersey’s* includes “bessie the cybercow,” “wumpuses are actually pokemon,” and “would you like to trade males?”



somewhere in the notes, typing it all together. This will be important later when you check whether things are coherent or get confused about what things which player should know about a complicated plot.

Above all else when writing, remember that **if it isn't in the repository, it doesn't exist**. Write your sheets whenever and however you like, but check them in where they're safe and everybody else can see them.

#### 4.1.1 Time Estimates

Assuming you've got design pretty much under control, it will take two to six GM-hours to write a typical sheet. Extraordinary sheets like the rules or a character with a convoluted history can take much longer—the rules for a ten-day can easily consume several GM-days of effort.

In estimating total time to write, remember that writing is a marathon, not a sprint, and that you'll perform best when you aren't under stress. Teams I've been on typically set a goal of two sheets per GM per week and estimate time to completion as one sheet per GM per week, plus a month to deal with the final stages.

#### 4.1.2 Wrap-Up Document

A tool that often is forgotten but can be extremely helpful in the initial writing of a sheet is the wrap-up document. In the wrap-up document, you detail all the interesting plots, items, and back-story for the groups and characters. This forces you to catalog the entire game (which often shows serious shortcomings and underplotted characters) and organize it into a format that is easy to read. Often the characters that look sad on the wrap-up document really are sad.

This document is a nice way to check that you have not missed anything important<sup>56</sup> on a sheet while you're writing it.

#### 4.1.3 Writing Meetings

While writing, meetings take on a different character. Generally, each person is working on their own individual project, rather than there being group discussions. It's still good to have meetings, however, rather than just send everybody off to write in their own little cubbyholes. MIT people are usually hosed, and meetings prop open a piece of time dedicated to working on the game, as well as providing good moral support when tackling icky problems in the sheets.

When you're all in the same room, it's also easy to discuss the problems that you run across

---

<sup>56</sup>Since the important stuff is what you put on the wrap-up document.

while writing, so everybody remains informed as the game continues to change. It is nice to meet somewhere that has a whiteboard or chalkboard to write on so that important notes have someplace to go and be seen by everybody.

If you want to do a big push, hold a sheet-writing party. Basically, you pick a large chunk of time such as a holiday weekend, and work on the writing sheets for the majority of the time. The nice thing about this is that the team now has a good chance to really learn about the game and to get a lot done. Also, if a particular GM has exclusive information that has been lost or that they've taken custody of, they're around for others to ask about it. Much of the best bonding around the game happens here. In the past when personal computers were less common, GMs would setup in a cluster to do this.<sup>57</sup> In the modern day where laptops are everywhere, setting up in someone's room is not a bad idea as there is easier access to food and other useful supplies.<sup>58</sup>

These parties often happen around crunch-time, which is unfortunate as having them happen earlier gives you more time to get other things done like pass your classes. These parties help ensure that your game doesn't suck by getting work done earlier. During the parties is also when the worst problems in your game are likely to get noticed and fixed.

#### 4.1.4 Sheet-Reading Parties

After all the writing is done, you get to have another type of party which is often even more fun: reading the sheets. You should aim to have a sheet-reading party one to two months before game. While it's not technically necessary, every game I've had one of these for has come out significantly more coherent, less buggy, and better understood by each individual GM.

Everyone should read all of the sheets in your game, twice if the game is small. This will take about a day.<sup>59</sup> Everyone gets a different colored pen, and looks for problems: grammar and spelling, problems in writing style, coherence between sheets, insufficient or incoherent plot, and general sanity-checking. After a GM finishes with a seat, they initial at the top and stick it on the next GM's pile. When everybody's initials are on all the sheets, you're done.

Once this is done, the sheets are redistributed, and the edits go in. Unfortunately sometimes this is when another problem appears, when two GMs disagree on the interpretation of a character. Resolving this depends upon how your group manages conflict in general. Possibilities include giving the sheet to a third person to interpret or debating it as a group.

You should aim to go through this cycle twice: once when all the bluesheets, greensheets,

---

<sup>57</sup>After you've been consecutively logged on for 48 hours or so, the Athena Cops will often check on you and sometimes log an abuse report. We've had to explain to befuddled techies that we really have been sitting there typing for the last two days straight.

<sup>58</sup>If you're really lucky, friends, significant others, or potential players will contribute to your effort in the form of food and drink. Be judicious in accepting bribes.

<sup>59</sup>Although *Caer Phaedria's* took two days and even then not everybody had read everything.

and character sheets have a first draft and again when all the sheets are in their final draft, with all the details filled in as well. The second cycle goes much faster, but is critical to consistency checking.

## 4.2 Technical Issues

Above all else, make sure everything is Athena compatible. You're running game on campus, and if your laptop catches fire you don't want to be out of luck. Besides this, there are a few technical steps you can take as insurance against unnecessary pain and suffering.

If not all of your GMs are students, you can still usually use Athena, either by setting up your repository so the non-Athena person can still access it, or by asking the High Council to get them a temporary Athena account while they work on the game. This latter option should be used rarely, and only by people who have a track record of work on games already—it's great for the crusty alum serving as your Voice of Experience.

### 4.2.1 The Template and LaTeX

When writing your sheets, you have a couple of options of how to do it so you can actually print them when it comes time.

- Write all of them in Word/PageMaker/Illustrator or some other GUI tool you know.
- Write all of the sheets in text files
- Jamie Morris's Template system
- Ken Clary's GameTex system

You might think that writing them in GUI-based tools is the best way, because you don't have to type any weird characters, and everything looks like how you typed it in. You're wrong. Most word processors do not give you any way to keep the various forms of information consistent. LaTeX based systems do, so you can do things like enter your casting decisions in one compact file and have player names spread properly through the entire game. If you have a game where this does not matter, such as a very simple less than 10 character game, then maybe it is ok. Even then, one of the other options might be better.

Text files have the same problem as the GUI tools in that it's hard to keep the various pieces of information up to date. Also, they don't look as good. If you're going for very fast, it might work.

Jamie Morris's Template system<sup>60</sup> is a combination of perl scripts and LaTeX magic that allows you to produce nice looking sheets and make sure that character to player mappings

---

<sup>60</sup>Actually a collaboration of authors, [template-gms@mit.edu](mailto:template-gms@mit.edu)

work out right. You can also do some interesting modifications to the system to produce things like custom cards and packets. Another neat feature is the ability to have it generate encodings into item numbers for “magic” items or other attributes you’d like to hide. A downside to the system is that it is somewhat complex to use, but it does have decent documentation.

Ken Clary’s GameTeX system tries a different approach. The entire system is written in TeX. Which means you don’t have to run any weird scripts, you just latex the file, and out comes sheets, packets, and other good things. It’s pretty amazing and very simple to use. Unfortunately, when it breaks, unless you are an expert in TeX, you might have difficulty tracking down the bug.

Both of the commonly used systems break the users of the systems into two groups: the Production Czar and everyone else. The Czar has to know enough LaTeX to be able to set up the production system you choose. The Czar should also create some basic templates and config files so the rest of the team can just get to writing sheets. Mind you, this is not an excuse for the rest of the team to be totally LaTeX illiterate—they don’t need to know the ugly inner workings, just how to write sheets and how to run the scripts to actually produce them. LaTeX is about as hard as HTML, and there are good quick-start guides like the OLH guide *LaTeX on Athena*, SIPB’s *Inessential Guide to LaTeX*, and even the Template example files.

Once you have things setup, you should latex your sheets as you work on them to make sure there aren’t bugs. A great way to lose some sanity is waiting until the last minute to produce sheets only discover that there is some subtle bug preventing anything from being printed.

One other nice feature of LaTeX: you can have information in the document which doesn’t actually show up in the printed version. Comments (preceded by %) are an excellent place to leave GM notes, cross-references, and reminders about secret information. Some teams I have been on also designate a comment-marker (e.g. %!) for flagging items that need to be dealt with. Then simply searching for the flag turns up all the known bugs that still need solving.

### 4.2.2 Revision Control

At some point, you might have a GM go insane or just be dumb. They might decide that a character might be funny to just be blank or to delete the entirety of game. If you don’t have revision control, and the person deletes the sheet you spent 10 hours writing, you’re likely to go insane too and the killing will begin. We don’t want that to happen.<sup>61</sup>

One of the tools that can help with this problem is revision control. Revision control was once only thought of as a method for keeping track of changes in code being revised by

---

<sup>61</sup>Regarding the scenario, we’re only partially kidding. A number of games have had major problems when a GM deleted the sheets by accident.

programmers. These systems were designed so that a team of programmers could modify code without worrying about losing information. The Revision Control system keeps track of many different versions of the same file so that a developer can see how changes progress, and by whom. You can even undo changes or develop a file in parallel with other versions. The cooler systems even allowed multiple programmers to work on the same file simultaneously.

LaTeX files are close enough to code that you can reap the benefits of the Revision Control systems. Currently, the three most popular systems are RCS, CVS, and SVN. RCS (Revision Control System) is the simplest. It has a number of small programs that take care of the various tasks such as checking in and checking out files. When a file is checked in, the changes are appended to a separate file usually kept in a separate RCS directory. Then when files are checked out, the changes are applied to produce the latest version of the file. There's also the ability to lock a file so that only the current user can edit a file to keep people from modifying the same file at the same time. The major detraction to this system is that all modification is done in the same directory. This means if someone runs "rm -rf" in the gamewriting directory, you lose everything<sup>62</sup>.

To understand CVS (Concurrent Version System) or SVN (SubVersioN), we'll have to slightly change our concepts. Under these systems, there is a master database (or filetree) that contains all information about all files and their various versions. This is called the Repository. When a person wants to work on a project, they "checkout" a local copy of the Repository, often called a Sandbox. The person does all of the changes they want to make to the Sandbox, then "checks in" the changes. To make sure that the Sandbox stays up to date, you run the "update" command to resync it. You may discover "conflicts" with someone else who has edited a file at the same time as you, then you need to manually go through the file and pick the changes that are most relevant, delimited with a bunch of ">>>>", "====" and "<<<<".

The difference between the two systems is that CVS is built on top of RCS and has some of its limitations. In particular, this means that CVS deals very poorly with directories and moving files around. SVN was built to deal with these limitations and is more full-featured. Unfortunately, until recently it was annoyingly painful to setup and use.

### 4.2.3 Backups

It sounds obvious, but make backups of the important information of your game often. If your files are stored in AFS on Athena, then a daily backup happens anyway. The previous day's files are stored in a separate read-only directory called `OldFiles`.

If you're storing the files on a personal machine or on `assassin.mit.edu`, then it's important that you occasionally (or automatically) make a backup copy on another machine or AFS.

---

<sup>62</sup>Unless you made a backup, which we'll talk about later.

### 4.3 Writing Characters

A good character sheet has three Cs: consistency, completeness, and coherence. Consistency means that any significant piece of information in one character sheet should show up the same in the sheet of every other character who would know. If there's a reason they'd understand it differently, you need to make it explicit in the sheet that the others may not agree—otherwise, the players will assume it's a GM goof and write it off. Leave cross-reference notes in the sheets so that you can make sure that they are consistent, and maintain a central timeline of when pre-game events happened.

Completeness means that you didn't forget anything. Sheet-reading parties and good notes help. It also helps to include a bullet-list of goals and key information at the end of the text part of the sheet, before the lists of bluesheet, greensheets, stats, etc.

Coherence means that every aspect of the character fits together into a believable human being, or else that it is clearly explained to the player that there are parts which deliberately appear to not fit—again, if you don't make it explicit, the player will assume it's a badly written character rather than a cool mystery for them to pursue in game.

A character sheet needs to convey five things:

- Personality, including roleplaying hints and psychlims
- Significant history
- Significant relationships with other characters (cross-referenced)
- Goals and motivations for each goal
- Available resources

As long as the sheet conveys these, you can put it together however you want. There are as many ways of writing good character sheets as there are GMs. Develop your own style—first person, second person, third person, whatever works to get the information across. Some character sheets are rich stories, others are bullet-lists of information. They don't have to be consistent between characters either—whatever works is the rule.<sup>63</sup>

Be clear with the player about whether to expect their character to be deluded and/or insane. Most players don't mind playing an insane nutcase as long as they know it, but it burns to think you're playing well and get told that actually, it was all in your character's head.

Write the text of the sheets first, leaving the extractables (abilities, stats, items, etc.) for later—feel free, though, to stick in special things if you think of them while writing the text. You need to do this because the resources allocated to characters need to be balanced across game, and that's best done as a group. After everything else is done, make a pass for each class of extractable separately, distributing resources and balancing stats. It takes about one GM-day to deal with each collection of extractables in the character sheets.

---

<sup>63</sup>In *Reality Check IV*, several of the characters were written as a chapter or two from a young adult novel.

## 4.4 Mechanics

The previous discussion of mechanics has focussed on high-level design properties and how they affect your game. This section looks at some of the issues in implementing your mechanics so that they operate the way you've designed them to.

**Innovation** Most of your mechanics should be tried and true mechanics that you don't change at all. People are familiar with them, there's technical support for building them, and you've probably experienced them as a player and know something about what's hard, what's easy, what's fun and what sucks. Plus, if your new mechanics crash and burn, your old ones will still work and game won't be sunk.

Don't be afraid to vary some of your mechanics or try a few totally new ones, though, especially if you've got an idea about how to solve a problem with pre-existing mechanics.<sup>64</sup> Specialty mechanics for a game are also a great source of fun and stories for players, particularly if a lot of people get to play with them.<sup>65</sup> Your new mechanics may also end up getting widely adopted—S-packets, Darkwater combat, shadowruns, and waylay are a few of the recent additions that have become widely used.

When you do introduce a new mechanic, make sure you playtest it ahead of time, within the GM team or with potential players if the mechanic won't give things away. Some people have even gone so far as to run an additional short game for the specific purpose of playtesting a mechanic before the game it is intended for.

Make use of your MIT education. If you know something technical very well, why not use it? If you're a cryptographer, you can apply it to hide things and have verifiably correct mechanics. If you're a mechanical engineer or an artist, build cool props or physical puzzles.<sup>66</sup>

**Hiding Information** The easiest way to keep information secret is to leave it with the GMs, but that puts extra burden on them. Ways to selectively reveal information without the GMs includes packets, encryption, stats with an interpretation key, memory packets, and physical hiding.

Simplest of all, of course, is the hierarchy of overriding rules, so only people with access to the more specific rules know how they work or the information they reveal. Ordinary rules are superceded by greensheets and signs that instruct the player otherwise. These, in turn, are generally superceded by abilities, items, and memory packets, which may supercede one another. You don't need to say, "unless you know otherwise" on the lower levels unless you want make people concerned about those who know otherwise, but the upper ones should

---

<sup>64</sup>The Darkwater combat system is an excellent example of this, and was a major driving force behind the *Pirates of Dark Water* game

<sup>65</sup>*Calypso's* sailing mechanic and *Krazny Oktyabr's* driving-the-boat scripts are specialty mechanics that players enjoyed greatly and that show up in some of the lasting stories from the game.

<sup>66</sup>*Amnesia* was written as an excuse to use Jeff Foley's bomb toy; the nuke mechanic in *L5* was because we found a concrete cylinder in the hallway labelled "property of the Dukakis campaign"

be specific about the fact that they override. Give people with higher levels the lower level information as well—the doctor should get the first aid ability too, because she’d know how it works. Be sure there is a GM version that includes all the information, from bottom to top, so you don’t get confused about what’s possible.

**Distributed Execution** Having things that run themselves without GM interference is nice, but potentially brings in all the horrible complexity of distributed computing theory. The problem is basically one of transactions: let’s say physplant grunges the vault and a player comes along to try to steal the diamond and finds the packet missing. How do the GMs know what to do? If somebody else already stole the diamond and they recreate it, then there are two diamonds floating around; if it wasn’t already stolen and they don’t recreate it, then the diamond has vanished from game.

There are only a few ways to deal with this, none of them particularly good.

- Require a GM to be there for every transaction.
- Make the players get all important items from the GMs.
- Make the players log everything important in the mechanics room.
- Put everything on hold until you find out whether somebody already stole the diamond.
- Guess which is right, and retro everything later if you’re wrong.

With most mechanics, this isn’t a problem very often, because the GMs will often know in advance when somebody’s planning to do something important, and will almost always be informed shortly afterward by happy, bragging players.<sup>67</sup> Most items won’t matter if they’ve been duplicated or lost—sure, it messes with your balance a little bit, but if it matters deeply whether there are 25 widgets or 28 widgets then the mechanic is too fragile anyway. If a mechanic has players manipulating state often or if time-delays are significant, however, you need to adapt one of the solutions above.

Obviously, it’s more of a problem in ten-days than one-nights, because game tends to be more spread out in space, as well as time, and the GMs are less omnipresent. In a one-night game, it’s not uncommon to have a GM present at *every* important event in game.

One nights often run twice in quick succession, however, which creates a related duplication problem. Some items from the first run will inevitably still be floating around game space during the second run, where they can be noticed and picked up by players. Template and GameTeX both have support for changing the date on all the items, so that players can distinguish Friday items from Saturday items, but the text is small and will often go unnoticed. Again, the best solution is usually to just shrug and allow the extra items to

---

<sup>67</sup>Allowing people to keep you informed without having to risk bumping into their enemies is one of the reasons that you want to keep the GM room moderately distant from the common room.



enter game, with the exception of the few most important. The important ones are good candidates for physrepping, because physrepped items will be collected at the end of game much more reliably, and it's easier for you to verify that they've been cleared out properly.

A related problem is non-local effects from mechanics. If somebody presses the “blow up world” button, ending the game immediately, how long will it take for people spread out all over campus to find out that they should stop playing? Again, this can often simply be ignored as a problem: if it takes five minutes to send people running through game space spreading the word, the time-lag just probably isn't significant. If players might not hear for hours, it's a problem. When simultaneity is important, add a visible count-down to the mechanic<sup>68</sup> so that everybody can respond appropriately; this also has the advantage of letting players respond to the big event coming up.

**Difficulty** Make mechanics easier than your first inclination. By at least a factor of two, and usually a full order of magnitude. Unless you have long experience designing puzzles, you will tend to make them much harder than you intended.

Also, never count on puzzle difficulty as a timing mechanism: some people will see through hard things instantly, others will get hung up for days on simple problems. If you want to make something impossible, constrain the resources it needs; if you want something to happen after specific time, use triggering resources that aren't available until then.

**Balance** In general, gaussian and thermal distributions are your friends when allocating a power scale. The magic granularities are, surprise surprise, three and five! Divide into three to organize power/resource levels—low, medium, high—then add intermediate values to make five. A gaussian distribution wants medium-low and medium-high, a thermal distribution wants medium-high and super-high.

For example, when assigning martial combat ability in a 45 person game, I would use a thermal distribution with average being the mode. The rule of three means people are basically divided into weak, average, and tough fighting ability, with extra values super-tough and sorta-tough. Then I'd assign people to end up with 5 weak, 20 average, 10 medium tough, 7 tough, and 3 supertough. The thermal distribution means that nearly half of the game gets to feel special, and the few weaklings can get other sorts of toys to keep them happy.

Don't add super-characters or trump-cards to game unless you're giving the character psychlims that prevent them from stomping all over game just because they can.

**Kludgite** Use kludgite materials when you need them to prevent players from breaking something, but be sure to label it as such. Kludgite is particularly useful for removing interaction between mechanics, a notorious source of unexpected brokenness in game. Don't

---

<sup>68</sup>e.g. the sun is pulsing and will go nova at 9:53pm.

overuse it, because kludgite breaks suspension of disbelief. Kludgite should never be used to patch weaknesses in plots, only weaknesses in mechanics caused by your inability to perfectly simulate the game world.

**Human Interface** Plan to rewrite your rules and greensheets at least once. Clear, concise explanations are more important here than anywhere else in game, particularly for the rules that people need to remember all the time. It is entirely reasonable for the rules to consume a GM-week of time before they are done. Sprinkle examples throughout the rules.

If a mechanic seems complex, you've done it wrong—mechanics should always seem simple to the user, or else half of them will end up breaking the rules accidentally. That doesn't necessarily mean they're easy—a puzzle is a simple mechanic that's hard to solve. It also doesn't mean the dynamics are simple, only that the part that the person interacts with is. If you can't figure out how to explain it simply, throw it out and rebuild from scratch.

One way to reduce the complexity of a mechanic is to break its stats into different types of information. Colors, numbers, and letters are a good example of orthogonal information that is easier for people to deal with in parallel—for most people, it's easier to remember A-9-Red than 1-9-2.

**Computer-Driven Mechanics** Stay away from the complex computerized stuff: anything that can be done without programming is better because it's easier for you to notice the bugs and it's unlikely to expand into a general excursion into the mysteries of Perl. If you must use a computer-dependent mechanics, it's important to make sure that any GM can operate and print the mechanic without help. Make sure that all files are in the repository, usable on Athena, and have READMEs giving step-by-step instructions for operation and printing.

**Riddle-Trails and Other Spatial Beasts** You should allocate a whole day, well before the game starts, for deploying your trails and other spatially distributed mechanics. I recommend walking around campus as a team of two, one writing riddles and taking pictures, the other placing dots and recording locations. When you're done, you can sit down at a computer, enter everything into the repository and decide which dots go with which trails.<sup>69</sup>

## 4.5 *Physreps*

Props are your best aid in suspension of disbelief. Just holding something physical in your hands goes a long way to bridging the gap between make-believe and real. They can be an

---

<sup>69</sup>Don't be as evil as I've been in hiding dots, or as evil as Joe in writing riddles. Joe wrote a riddle, "30 seconds. Bang!" that referred to a specific place in the tunnels where a pipe banged once every thirty seconds. I hid a dot underneath the lip of the bottom step in 6-120 lobby so you could only see it if you lay prone at the right spot and squinted. We've since learned these are no fun for the players.

incredible pain for the GMs because they require lots of time in preparation and setup, but simple physreps are usually worth it.<sup>70</sup>

Choose props that have the properties you want to physrep. A few examples:

- *L5* physrepped nuclear warheads with three-foot long concrete cylinders: annoying, bulky and difficult to hide.<sup>71</sup>
- *Krazny Oktyabr* used a big pink foam sheet as lead plates. Although not so heavy, it was awkward and nearly impossible to conceal.
- *Macondo* showed the size and deadliness of a gun by the length of the PVC pipe attached to it. Easy to see at a distance, harder to conceal and handle a large weapon.
- *Nexus: Operation Toronnen Horse* indicated ship size by hat size. Carriers were baseball hats, and the huge space station was a massive ten-gallon hat—easily visible at a long distance.
- *Krazny Oktyabr* did bomb disarmament with a circuit board. When you flipped the switch, if the green lights came on you could move the item, and if a red light came on it blew up. Other players watched nervously from a distance as the people interacting with the mechanic shivered with anxiety and demanded total concentration.

## 5 Pregame

Pregame begins when you send out the application and players start getting involved in your game. This is the first critical test of all the work you did designing and writing the game, and the point where people start to form an impression of your game. With a good pregame, your players will trust you more, be more willing to suspend their disbelief, and actively help you make your game work. With a bad pregame, you lose credibility and your players will come in preparing to triage plots that the GMs have broken.

If you've planned well, you should be done writing at this point, and able to spend some time checking details and adding grace notes. Otherwise, you've just entered crunch time and you're going to have to finish writing and run pregame simultaneously.

### 5.1 Writing A Good App

Don't fret too much about the blurb and the application. If you've got the time, you can make it a masterwork, but if you're rushed, don't worry about it too much: people are going

---

<sup>70</sup>*Krazny Oktyabr* required 26 mid-sized cardboard boxes in addition to two shopping carts filled with props and several more boxes besides. It hurt, but the players loved it.

<sup>71</sup>Some people would claim we used them because we're sadistic bastards who enjoyed abusing the players. We remain tactfully silent on the subject.

to play your game anyway. Those opening sequences are cool, but they're not what people are playing for. In general, if it takes you more than an hour to write your app, you're probably being too clever.

This is the info that your app needs to give players:

- When the game will run, and for how long
- A vague idea of the game's genre/wierdshit-level/size

And this is the information that you need to collect:

- Player vital stats: name, email, etc.
- Hotbutton questions: Nazis, pregnancy, sexuality, religion, stalking and other touchy issues
- Player-player conflicts: things that cause screaming fits or lawsuits<sup>72</sup>
- An open ended "tell me what I need to know" question

That's it. Everything else is gravy. Half your players will send useless apps anyway, in one of three types:

- **I'm a new player and know nothing.** Some new players know a lot about the Guild and LARPing. Many don't. The ones that don't sometimes conscientiously fill out their apps with essentially meaningless answers because they just don't have the information you want.
- **You know me, just give me a part I'd like.** FEAR FEAR FEAR! Usually this causes me to give somebody exactly the part they'll hate most because I'm wrong about their tastes. This is not the same as, "you know me, give me anything." Those apps are pure solid gold for casting flexibility.
- **Look at me! I'm cool and give wacky-wacky answers to your appidty-dap!** These people are being assholes. Deal with them however you please.

Don't make your application too long. I once sent out a 300-line application, and people threatened to not play my game for the sole reason that the app was too long.

---

<sup>72</sup>I'm serious here. There's required cautionary reading when you become Grandmaster.

## 5.2 Collecting Applications

Back in the day, maybe you could send out an app two days before handout and get 150 players. But that's not the case today. Your app should go out 3 to 5 weeks in advance of the game, depending on how many players you need. IAP 10-days may wish to send their app out even earlier, before classes end, because finals and holidays are great distractors.

You'll generally get a flood for the first few days and then things will trail off to a trickle. Send the app out again occasionally until you've got enough players and a healthy waitlist. Encourage people to apply for the waitlist even after the game is full: some people will punt, and you'll need a good waitlist. You want a waitlist about 15% of the size of your game: a 20 person game casting two runs wants a waitlist of at least 6 people.

Set the deadline for the app to the day before you want to cast. That way there's a little slop time for the last-minute folks to show up before you cast.

If you can, post rules and public background sheets for the game online so people can get a sense of what they're applying to play. It'll help whet people's appetites and create a buzz around your game that will drag their friends in as well.

### 5.2.1 Managing Applications

You don't want to get your apps screwed up in your mailbox. Add an Apps directory inside your game directory and store them there so everybody has access and the same view of the collection. I usually name the apps `appnumber.username`, so the if the 5th application comes from Joe Foley, it'll be `05.foley`, which makes it easy to see the order they came in.

### 5.2.2 Shit, There Aren't Enough Players!

It happens. Sometimes your app deadline passes and you just don't have enough apps, or you've already cast but then too many people punt. There's only a few good ways to deal with this:

**Frantic Plea For Apps** If you're only a couple parts short, you can send out another app, frantically pleading for just a few more people to apply. Say it loud and clear: "We need a couple more people, and we need them NOW!" This won't get you more than a couple more apps.

**Personal Arm Twisting** If you know somebody who you think should be playing, but hasn't applied, track them down and ask them! Be warned that this isn't a good tactic to push too hard on: it's easy to piss people off or frighten people away by being too obnoxious, and that's not good for your friendships or for the Guild.

**Cut Feeble Parts** Maybe there's one or two parts that are kinda lame and that nobody is expected to depend on. Kill them.

*WARNING: This is a very dangerous thing to do, and should only be used as a final last resort. You balanced the game for all the parts, and cutting some is likely to break your balance in subtle ways.*

**Reschedule** You've probably noticed that these tactics generally won't get you more than a few people - that seems low, but it's what you should expect.<sup>73</sup> If you're short more than about half a dozen people, *don't run your game!* You didn't design it to run with that few people, and it'll probably suffer a lot for the lack.

If you're coming up on handout and find yourself down that much, you aren't going to be able to make it up, and you aren't going to be able to adequately patch your game in time. Postpone the game, get a better slot or shrink the game, but don't flush months of effort down the drain so that you can have a bad experience watching your game stagger around like a three-legged dog.<sup>74</sup>

## 5.3 Casting

Casting should be about a week in advance of your game. I recommend casting on Friday the weekend before your game, so that you can use the rest of the weekend for production.

Make sure you have your whole GM team there for casting. Plan the time way, way in advance and reserve at least six hours. *SIX HOURS!?! you say* - it can't possibly take that long to cast, can it?

Yes it can. And it probably will, if you aren't running a SIK game. If you're anything like the GM teams I've been on, you'll get about 80% of the parts cast quickly, and then you'll be down to the really hard decisions. After a while, you'll start reconsidering parts that you were sure of at first, and maybe they'll change or maybe they won't. Eventually everything settles down and you can take your final hour to compose the casting email, the waitlist email, and the NPC/plant email and type in every bloody email address.

Remember, it's vitally important to have all of your GMs approving of the casting, not just argued into submission. You want a good consensus here coming into the final stretch before runtime.

Here are the heuristics which I use for casting a game, listed in order of descending importance. I'm sure this'll attract flames, but I think it's a good policy nonetheless.

---

<sup>73</sup>A conspicuous exception to this rule was the 4th run of *Krazny Oktyabr*, where Joe added a 5th run by somehow producing 25 additional applicants over the course of three days.

<sup>74</sup>*Nexus* aborted when they couldn't get enough applications and proceeded to squish their game to eventually remove about 35(!) parts. It ran the following year and went pretty well.

**1. Preapps and GM-Preapps** First thing, figure out which parts are already taken. Preapps will fill some parts because, after all, those are the people who the parts are written for. Theoretically, that's all there is for precast parts, but GM teams are human. You've probably been saying things like "Wouldn't Bob be cool as the Shaolin Death-Monk?" and "Suzy is just totally a Dread Lord of Flan..." for a while. **Don't do this.** If you discover that you have, don't fight it. Just check to see that their app is compatible and give it to them, you jerk.

**2. Give Cool Stuff to Newbies** This is the most important heuristic in this section. Even if you hate everything else I've said and are planning to napalm me back to the Stone Age, **CAST THE NEWBIES IN COOL PARTS!** The Guild lives and dies on whether new players come back for more games.

Give *really* cool parts to new players. Remember, you want their first couple games to be really cool so they'll get hooked and keep on coming back. Good newbie parts generally have fun abilities, team-mates, not too many traitors on their teams, and an experienced player or two in their vicinity who they can trust to help show them the ropes. One combination that often works well is a newbie team leader with an experienced loyal lieutenant, although in this case the lieutenant needs to be somebody who isn't going to entirely usurp leadership.

Good parts for newbies are generally heavy in plot, character, and skills and light in painful mechanics. Having lots of plot, character and/or skills means they're never going to be left sitting around the common room wondering why nothing cool is happening. Either they'll be doing the cool things themselves, or they'll have other people asking them to help do the cool things. Lightening the mechanics means they get to focus on interacting with people, rather than getting trounced by experienced gamers in an economics plot or a wargame. I personally like to save a couple of my very favorite characters in the game for the newbies.

*Remember - you don't have any clue how incredibly cool a new player might turn out to be. Give them a part that lets them shine.*

**3. Cast Sob Stories** If Bob Smith hasn't gotten cast in the last two games he applied for, give him a break and probably even a good part. If he likes it, he's more likely to keep playing guild games. If you skip over him too, he's going to assume you're a bunch of petty, vain assholes—and he'll be right.

**4. Cast the Difficult Parts** A few parts may need delicate play or require skills that few players can supply. Pick among the few who fit them now, so you don't find yourself stuck later.

**5. Cast the Difficult People** Some people are just difficult to cast. Maybe they don't like certain people, or a lot of people don't like them. Maybe they only like to play religious traitor diplomats. Maybe they won't play a Nazi and your game is set in Berlin in 1941.

Whatever it is, some people are harder to cast than others. If you know who the hard ones are, figure out where you're going to put them first. It's a lot easier than casting the easy people first and then realizing that you have to recast everybody because you can't fit your hard person in anywhere.

**6. Give Power to the Powerful** Pick out the people on your list who are powerful personalities, the sort who warp the game around them. You know that no matter where you cast them, they're going to assume leadership positions, make things happen and get involved with plots and power plays. Your game almost certainly has characters who do this too. Distribute your powerful, game-warping players relatively evenly among the powerful game-warping characters. If you don't, they're just going to take over anyway. Unless you *want* an unbalanced game, though, don't put too many where they can easily ally.

**7. Minimize Crosscasting** Most people don't like being crosscast, plus it will tend to confuse the hell out of your players when male characters are played by females and vice versa.<sup>75</sup> Try to minimize crosscasting. If you can, have a couple characters whose gender just doesn't matter that much. One thing I have noticed in the games I've run is that there seem to be less men willing to be crosscast than women, so it's better to err on the side of too many male parts. Another good rule of thumb is that there seem to be about twice as many men as women in the Guild, so if you make 30-40% of your parts female you're likely to be able to avoid crosscasting.<sup>76</sup> Don't be too shy about casting somebody who applied later in favor of a slightly earlier applicant if it allows you to avoid crosscasting.

**8. Cast Early Applicants** People who respond promptly to your app should be rewarded.

### 5.3.1 Dealing With Hard Cases

Both MIT and the Assassin's Guild attract people who just don't know how to interact with other people, whether it's due to a bizarre upbringing, autism, or whatever else. These people are nightmares to cast because they tend to piss off other players and also the GMs.

It's really tempting to just blackball such people. Don't. Not unless they have a well established history of violating basic rules of game-play.<sup>77</sup> Frankly, my first response is to hope they applied late, so that I can put them on the waitlist and avoid the issue. That's usually not a viable solution, though.

---

<sup>75</sup>In one run of Krazny Oktyabr, every single attempt at seduction failed because the attackers realized too late that they were trying to seduce cross-cast characters

<sup>76</sup>This is now outdated, the gender ratios are more balanced and sometimes applicants are mostly women (2010-06-22).

<sup>77</sup>I have blackballed precisely one person: after more than a year of continuous bad behavior including trying to get other players to conspire with him to cheat and applying using a false name, our GM team decided that if he applied, we wouldn't cast him. Fortunately, he didn't apply.



Instead, look for a part where they will do minimal damage if they start acting up; if necessary, you may even create a part for the specific purpose of casting them somewhere that they won't screw up your game.<sup>78</sup>

## 5.4 Communicating With the Players

Have empty *game-players* and *game-waitlist* mailing lists prepared in advance of casting. As soon as you're done casting, fill them in and let people know whether they've been cast, along with when and where handout will be. This is also the right time to mail out costuming hints.

Once you've let people know that they're cast, you can settle back and wait for them to punt. People are going to punt on you - there's nothing you can possibly do to prevent it. All you can do is pray that they'll remember to let you know.

When somebody punts, send a nice little note thanking them for letting you know and saying you're sorry they can't play. Then pick the person off the waitlist who fits the part best and let them know they've been cast as fast as you can.

This all sounds like common sense, and it is, but it's really hard to remember when you're exhausted and all you want is to die a peaceful death of sleep.

Send out extra reminders about handout and game start. Hit the players on the head with the place and time for both in every email you send. That way less of them will screw up and give you a heart attack by not showing up.

When people tell you that they won't make it to handout, force them to send the note via email and keep it all in the same place. You don't want to accidentally give away their packet because *you* were an asshole.<sup>79</sup>

## 5.5 Players Communicating With You

From the moment you send out casting notices, you'll start getting emails back from the players. They need prompt, coherent responses, and you'll quickly determine that some players are much needier than others. Usually, the best way to deal with this is to assign the task of talking with the players to one GM, and lighten up all of their other responsibilities.

If you have the capability, set up archiving for all messages sent to the GM mailing list. Anyone writing to a player should copy the message to the GM mailing list and put the

---

<sup>78</sup>I have done this precisely once. I often, however, make sure there's a few parts which form a relatively self-contained unit so that hard cases and personality conflicts can be dealt with.

<sup>79</sup>We actually did this to a *Krazny Oktyabr* player. Unfortunately, he'd already seen the part via our electronic distribution, so we couldn't recast him to another part. There was nothing left to do but tell him to come to both runs and hope that the person with his part punted. We felt terrible.

mailing list in the reply-to field of the email. Resourceful players will still screw it up, but you do your best.

The Spokes-GM's job is to defend the rest of the GM team from the players. They should answer simple questions immediately, and ask for patience while they consult on hard ones. Hard ones then go on a list, waiting to be run by the other GMs in a batch once every couple days. Any question asked twice should be added to a web-accessible FAQ if its answer won't blow any secrets.

As the players ask questions, they will reveal lots of little flaws you never noticed before. While this may cause you to correct ability cards and bluesheets and the like, *DO NOT FIX THE RULES DOCUMENT*. The last thing you want is conflicting versions of the rules floating around. Rather, there should be a web-accessible errata list detailing all the fixes.

## 5.6 Reserving Rooms

The Grand Master is responsible for reserving the rooms you want, but can't do a damned thing until you ask. Send a list of what you need—either specific rooms or relative arrangements.<sup>80</sup> The earlier you request, the more likely you are to get good rooms: I recommend asking a month in advance, or longer if you're running a 10-day or during IAP. Be prepared to be screwed.

Some areas of campus are easier to get rooms in than others. Buildings 12, 26, and 36 are usually pretty good. Sometimes you can get all of 24, but it's sometimes just completely gone. As of this writing, the Guild has a special deal worked out with Course 6 that allows us to reserve rooms in 34 that are not normally reservable, so those are almost always available. Patrol has 36-1 reserved every Saturday evening. You can sometimes take Patrol's rooms, but shouldn't if you can possibly help it.

You can sometimes get away with using rooms that you haven't reserved, but should never count on it, particularly for important rooms in your game.

Don't forget to reserve a room for handout. GM teams forget this a lot.

## 5.7 Production

You want to print immediately after you cast. Ideally, you will cast on Friday night, and do your production on Saturday, spilling over into Sunday if necessary, for a Wednesday handout. Printing and stuffing both take a surprisingly long amount of time, and you want to be at ease while you're doing it, or you risk stuffing a SWORD Bluesheet into GAUNTLET's packet.

---

<sup>80</sup>For example, *Panoramic Steam Intercontinental* requested "about a dozen rooms in a linear area like a train"

Get a nice, big area you can take over and set up camp. At the moment, the Guild office is an ideal location. Other good locations include the w20 and building 37 clusters.

You will need:

- High speed printing capability (one kickass printer or several normal printers)
- Manilla envelopes and dime packets
- Colored paper (generally green, blue, yellow and red)
- Papercutters, scissors, masking tape, scotch tape, staplers, lots of staples, a sharpie
- Props, props, props

If you've got the time and inclination, you can print item cards and ability cards on cardstock. Most printers these days are incapable of duplexing card stock, however, so don't plan on it unless you've specifically tested everything in advance.

You'll want three production areas: one for packets, one for props, and one for cutting, taping, and stapling. Put the player and character names on the packet so that when the flap is folded the player name shows but the character name doesn't. Spread the character packets across the floor in the same order as the character files in the directory: this is the order things will come off the printer, and it'll make distributing them much faster.

The Template will let you print sheets by type or a whole character at a time. Print things by type: all the character sheets, then all the bluesheets, etc. This means you don't have to worry about juggling a bunch of paper colors at once. Although Template (and I believe GameTeX as well) support telling printers which tray to pull paper out of, so you could load different colors into different trays, my experience has been that the printers often ignore it and pull whatever paper they want.

As things come off the printer, make piles on top of each character packet. You may want to print memory packets and ingestibles first, so that you can get a start on fabricating them—in many games, these are the biggest bottleneck. If you've got enough little things to cut, fold, staple and tape, set up an assembly line.

Once all the character packets are finished and stuffed, print everything else in game, gather up and fabricate props, and load everything into carts and boxes ready to transport to game. Be profligate in your expenditure of paper here: it's better to have everything all organized up and ready to go, even if some things don't get used, than to make a player wait while you try to print the death-ray they just built.

In addition to game-specific props and all the stuff you've just fabricated, be sure to pack:

- Several full rolls of masking tape
- Scissors, scotch tape, pens, stapler, and other sundry office supplies

- Badge holders
- Loaner and sale guns and ammo of the appropriate types
- First aid kit
- Printouts of the playerlist and your room reservations
- Extra dots, envelopes, and whatever else you need to repair parts of game space that get grunged.
- Radios (optional)

## 5.8 Handout

Handout is the first real glimpse any of your players get of your game, and first impressions count. If handout goes well and everybody gets their stuff without much of a hassle, things are more likely to get off to a good, enthusiastic beginning when game actually starts. If everything's late and screwed up, it's not a complete disaster: people won't hold it against you. It will, however, mean that they aren't feeling as confident about the quality of your game, and that means you have less leeway to screw up once things actually start. And believe me, you *will* need every bit of slack you can get.

Yeah, yeah, I know there's a tradition of GM teams being late and sending in only a single sacrificial victim and having to stall for three hours while the other GMs are frantically stuffing packets.

It sucks.

I can say, from experience, that it is *very* satisfying to be lounging around in the handout room, waiting for the players to show up, a big crate of packets by your side. This is the ideal *and it can happen!* I know it's possible to have everything done on time because I've done it many times. You can do it too.

So here's what should happen, how to make it happen, and how to deal if it can't.

### 5.8.1 How Handout Should Be

You walk into the handout room about fifteen minutes before the players are supposed to show up. You hang out with your fellow GMs, chatting and going over last minute details.

With you in the room are:

- **All the Character Packets**
- **Rules, Scenario, and other handouts (a copy for every player)**

- **Cool Props**
- **A list of people who can't make it to handout**

When a critical mass of players has arrived, about five minutes after the scheduled time, you say a few words welcoming people to handout and then start the rules session.

The rules session takes about forty-five minutes, as you read off all of the nifty special rules for your game. Somewhere in the middle, you get to show off your nifty props and shoot each other a few times.

Now you hand out the packets in some sort of random order, setting aside the packets for people who aren't there. People start digging in.<sup>81</sup>

You sort the true no-shows from the people who told you they weren't going to make it, and immediately hand them out to the waitlist people in the audience on a first come-first served basis.

After you've distributed all the packets, you let people come and ask questions about their packets and handout slowly breaks up.

### 5.8.2 Attaining the Ideal

Assuming your game is finished, it's not too hard to have an ideal handout. All you have to do is start early.

**Getting There On Time** Plan to show up thirty minutes early, and plan to gather your props and move junk thirty minutes before that. That might be enough extra time to allow for all of the last-minute hassles that you're going to have to deal with. (This is considerably reduced if you've done production right, and just have a big box labelled "HANDOUT" with everything you need in it.)

**Killing Trees** Don't be afraid of killing trees: print one copy of every handout document for every player in your game, even if they aren't going to be at handout. They're just going to need it later, and you've got time to make it now. Sure, some people will print stuff out, especially if you've put a copy of the rules online, but only a couple of them will remember to bring a copy to handout. If you're feeling guilty about the dead trees, recycle the extras.

Trust me, the pain is less this way.

**Plan the Rules-Talk** You've almost certainly got more rules than you have time to talk about. You're going to need to figure out how to skim over the ones that aren't tricky or important.

---

<sup>81</sup>Never hand out packets before the end: people will read their packet instead of listening to you.

Plan your rules-talk ahead of time, and strictly limit the time it takes. You are not the President, and nobody's going to listen to you talk for three hours. Your target is a 30 minute rules talk, but you can extend it up to an hour if you have to. Remember, people will be asking questions too, which will take up some of your time.

**Newbie Rules Session** If there's more than a few new players in the game, hold a newbie rules session before the main session, in which you explain the Standard Rules. Allot 30 minutes for this session, more if you need to explain a combat mechanic.

**Post Rules Ahead of Time** You should try to post the rules online around the time that you send out casting email. That way players can read them ahead of time. This simplifies your life at handout two ways.

First, people will have a chance to figure out the rules somewhat. This will cut down on the number of dumb questions you have to deal with at handout.

Second, people will be able to point out some of the Big Horrible Holes in your rules so you can patch them before handout starts. There's nothing like getting caught with your pants around your ankles when somebody points out that your combat system allows Bill Gates to kill the Terminator with a single blow. Much better to deal with the really bad stuff beforehand. It'll also help you plan where to spend the most time when you give your rules talk.

**Sleep** Get a good night's sleep before handout. You'll be able to think on your feet and you're much less likely to do something stupid. I mean it. Eight hours. Say it with me. Eight hours of sleep before handout.

**Managing the Audience** They'll want to participate and probably get rowdy. Give them enough latitude to joke, but not enough to sidetrack you. Similarly, if somebody is being an asshole and asking pointless questions, feel free to tell them to shut up. Call on people so it's not just the same couple loudmouths talking, and don't let the experienced players intimidate you.<sup>82</sup>

**When to Shut Up** Don't talk forever. If you have to, cut yourself short, hand out the packets, and fill in the details for people who want to stick around. Don't torture the players by talking for two hours before letting them have their packets and get on with their lives.

Also, don't allow question dialogs to get out of hand. It's a good thing when somebody points out a bug in the rules, but don't try to patch it on the spot. Nothing's going to be done except waste time if you let forty people debate how your Bazooka mechanic should work. Just say, "We'll fix that. It'll be in the errata," and go on.

---

<sup>82</sup>Some seem to take pride in breaking GMs.

### 5.8.3 Dealing with No-Shows

If somebody doesn't tell you they can't make handout, it's your prerogative to recast them on the spot if they don't show. Take advantage of it.

If they didn't show up and didn't tell you, they're probably a lot less enthusiastic about the part than the waitlist person who's been waiting in the audience for the mere chance of snagging your deadbeat's packet.

Knowing who to recast where is a harder problem. The person you're giving the part to was on the waitlist in the first place because you thought somebody else fit this particular part better. If you don't have precisely one waitlist person and one deadbeat, you'd better have some sort of plan for how you're going to distribute the parts, because the last thing you want to do is have waitlist people sitting there while you dither about who to give what parts to.

I don't have a really good answer to this one, but I recommend a "first-come, first-served" approach. Hold up the packet, yell "Deadbeat! Who wants a packet?" and give it to the first waitlisted person who calls out. It's a better method than anything you'll come up with on short notice.

### 5.8.4 When Things Go To Hell

If you've followed my advice above, you shouldn't need this section.<sup>83</sup> However, we're all human, and you might have screwed up and not have everything all nice and ready to go on time. Here's some hints to help you avoid sucking any more than you absolutely have to:

- **Don't Use CopyTech** Contrary to some beliefs, an Athena printer *is* a copier. It's much more efficient to print duplicates than print a single copy and send runners to xerox it.
- **Print Single-Sided** Most printers print two single-sided pages significantly faster than they print a single double-sided page.
- **Character Sheets, Blue Sheets, Green Sheets** These (plus Ability Cards if you can manage it) are the things you *must* have stuffed for handout. If you can get nothing else, make sure these are there and put everything else in the Box for game start.
- **Send the Sacrificial GM** Have somebody start giving the rules session on time and as per schedule. Don't show that you're completely screwed until you have to - maybe you'll make it after all.
- **Admit You Screwed Up** When things really go to hell, send the players home. Don't keep them up until midnight hoping you'll give them a packet.<sup>84</sup>

---

<sup>83</sup>I've always wanted to say that.

<sup>84</sup>There is a special place in hell reserved for *Zagreb*.

## 5.9 Shore Leave

And on the seventh day God ended his work which he had made; and he rested on the seventh day from all his work which he had made.

If you've followed this prescription, you'll notice that Monday, Tuesday, and Thursday don't have any scheduled pregame work. Every GM should have at least one day completely off. You're about to run the game and you need to be in top form.

## 6 Runtime

You've survived all of the preparation, and you're all caught up on sleep. Now the big day is here and you have avoid blowing it as you run your game. You need to complete setup in a timely fashion so you can start your game on time. How you start and run the game will then depend on whether it's a short game or a 10-day. Always, however, you will be faced with the problems of keeping your team together, making coherent rulings that don't break game, and dealing with the strange ways that your players will break the game.

### 6.1 Setup

Setup can take a surprisingly long amount of time, particularly if you've got tunnels and riddle trails, or if your game is highly physrepped and signed. A good rule of thumb is to allow yourself a base of two hours before game (one for finding your pants and moving stuff over from the office, one for last minute panic and dealing with arriving players) plus fifteen person-minutes per simple floor to set up and one person-hour per complex floor to set up.<sup>85</sup> These may seem extremely long, but when you factor in time to walk back and forth, being confused, fixing mistakes, reprinting broken things, running back to the Guild office for forgotten bits and pieces, etc, you'll soon realize just how quickly the time goes by.

Some things like riddle trails can be deployed long before the game starts.<sup>86</sup>

Hold back expensive props like radios and attractive ones like guns for deployment about an hour before game start. These are more likely to walk off than other props, and you want to minimize the time when they are exposed before the game starts running (once the game is running, player presence makes it less likely they'll walk away). For 10-day games, try to avoid deploying any expensive or attractive props openly—there's just too much chance they'll disappear on you.

---

<sup>85</sup>*Krazny Oktyabr* used 24-1, 24-3, and 34-3 and they were all complicated, so that's 3 person-hours of labor, plus 2 base hours. Thus with 3 GMs, we should start setup 3 hours before game. A SIK game putting signs all though 24, 26, 34, 36, and 38 would expect to take 6 person-hours plus 2 hours to set up, so a team of 3 GMs should start setup 4 hours before game.

<sup>86</sup>We ran *L5* six months after we had originally planned, and all but a couple riddle trail dots I had deployed were still up.



Setup is also a good time for negotiating with NPs. Let people know politely about rooms you've got reserved and that there will be an Assassin's Guild game running in the area soon. Sometimes NPs will ask if they can share the area—this happens especially often with ballroom dancers in 34-3. Warn them about yelling and flying projectiles, and offer some alternative areas (the ballroom dancers are often willing to move to 36-8, for example). If they're cool with the game running on around them, though, by all means let them stay and let the players know they don't have to stop for those NPs.

## 6.2 Running a Short Game

For a short game, and especially for a one-night game, time is a critical issue and always in short supply. Getting the game off and running on time can be difficult, especially when players show up moments before start with piles and piles of questions.

I generally insist that players show up at the GM room at least 30 minutes before the scheduled game start, and let them know GMs will be on hand to answer questions starting an hour before game start.

Set up a queue on the board where people can sign up to ask questions; this means you won't get mobbed and lets you distribute question-answering time fairly. There will usually be a couple of needy players who will fill up all available time with questions if you let them—give them the time they need, just use the queue to make sure everybody else gets the help they need. You may need to defer some of the needy people's questions for answering while the game runs.

One GM is probably going to need to be on printer duty, to fix the things that players have discovered are missing from their packets. If you can bring a portable printing setup to the GM room, do so!

Be subtle when giving incriminating props to players. Mostly people just won't notice or care, but you shouldn't get careless.

Have one GM keeping an attendance list so you know who isn't there. Fifteen minutes before game start, you can start panicking and calling the missing players to demand where they are. Usually it's the more experienced players who are late—maybe it's just that they historically expect games to not start on time? An experienced player who is late will probably show at the last possible moment and piss you off; a new player who is late is cause for serious worry. Although it's possible to recast at the last moment, it usually causes a big mess and a deeply confused player and should be avoided if at all possible.

Thirty minutes before game start, a GM should synchronize everybody to the same time. It's often not strictly necessary, but most short games care about timing in one way or another and synchronizing eliminates one more source of confusion.

After synchronizing, go over announcements, rule clarification, and errata. You will likely have a few things added to this by the questions players have asked you earlier.

Once you've got everybody there (or have made arrangements to deal with the missing people), you'll want to place people and get started. Assume it will take 5 minutes plus 2 minutes per building to place a group of PCs: in most short games, 10 minutes is enough time to place everybody.<sup>87</sup> Choose your actual start time based on this and announce it to everybody: once you start placing people, you cannot change when game starts.<sup>88</sup>

The end of your game is likely on a timer as well. Be extremely wary of violating this, even if neat things are happening, because players will have built their strategies around the expected stop time and if you change it without a damned good reason, you're breaking the game contract with them. Unless you've moved game start back a long way, don't change game stop. Just let your game be a little bit shorter—it's much less confusing this way.

### 6.3 Running a Ten Day Game

Most ten-days are much less time-sensitive than short games, and you should expect some people to roll in hours after game start unless there's a specific reason they must be there for the start.

Follow the same procedures as a short game where you need them and abandon them where you don't. For a ten-day, the critical thing is to sustain communication with the players over the long run.

Most ten-day games have two GM-related rooms. One is a GM room that is off-limits to players unless they are invited in, and contains props, log books, and secret materials of all sorts. The other is a box/mechanics room and ideally is adjacent to the GM room. If possible, you want both of these rooms reserved 24/7 for the entire game so you don't have to clean up at the end of each night.

The Box room is your nexus of communication with the players. Errata and public announcements should go on the chalkboard in the Box room (also the common room, if your game has one). The board should also contain a queue to talk with the GMs and information about when which GMs will be around game. You may wish to have a GM designated as "on call" when no GMs are in game, so that players can find you in an emergency.<sup>89</sup>

The Box itself is a banker's box sized object with a folder for every player and one for the GMs. Private communications between players and GMs should go into the Box, as should any small things you need to deliver to the players.<sup>90</sup> Many games also put folders for particular mechanics in the Box, so they can be dealt with separately by the GMs. Check

---

<sup>87</sup>*Reality Check IV* was a conspicuous exception. It took us nearly an hour to place everybody since they were going all over main campus.

<sup>88</sup>Sometimes you can have even better control, like in *Krazny Oktyabr* where everybody read from a script at game start which included a radio message broadcast to all locations that started the game.

<sup>89</sup>During *Zagreb* we waited in a halt for two hours, starting at 4am, as we cudgeled a GM into leaving bed and coming to tell us what we happened when we discovered the UFO and I immediately blasted it with a pile of C4.

<sup>90</sup>e.g. replacement ability cards, blueprints won at the end of a research mechanic, drugs shipments, etc.

the GM folder in the Box when you come into game and every couple hours when you have a lull.

For most games, you actually don't want much email communication with the players. Players can only send and receive email when they aren't in game, and the GM room generally doesn't have email access unless you bring in a laptop. The plus side of email is that if somebody wants to ask a long, confusing and difficult question, email forces them to write it down clearly.

You should still send a copy of important announcements over email, but post them in game first and put "POSTED IN THE BOX ROOM" at the top of the email so players know it's a duplicate of information available at game.

## 6.4 The Role of the GM

The role of the GM in a Guild game is completely counterintuitive for tabletop roleplayers. During game, a GM should almost never have take an active role, and GMs should never play NPCs. You are not a participant in the game, you are the referee.

Wander around game, watching, listening, and being available for questions and stories. Players will often want to tell you something neat they just saw or did—this is one of your rewards for running the game! You should also always carry a notebook so you can record good stories and neat quotes for sharing at wrap-up.

Try to be present for difficult situations, and always be sure there's a GM findable by the players. When a big group of people goes off to do something, send a GM with them. When you know somebody's likely to be tortured, go so you can answer clarifying questions. Always have at least one GM in the GM room,<sup>91</sup> and try to have others hanging around the common room or wandering places where there are lots of players. Those are where you're most likely to be needed, and also where you're likely to see interesting things happen in game.

Practice your poker face and maintain it at all times. Players often deeply believe in theories that are completely ridiculous or make wild guesses that are startlingly accurate. You don't want to spoil their fun or give them an unfair advantage by giving away something by mistake. A poker face isn't necessarily blank, just unreadable—a consistent amusement at the player's folly is just as impenetrable as being completely expressionless. Never confirm anything, even when the player already knows it: they might start wondering about it again later on.

Sometimes a player will want to use you for a bluff. The basic form of a GM-bluff is as follows: a player talks to the GM, then says something outlandish to other players. If the other players are metagaming, then they assume that the outlandish thing is real since the person just talked to the GMs. This is a legitimate tactic, as long as the GM makes it clear

---

<sup>91</sup>Always having a GM in the GM room also means it's safe to leave bags there.

what the *characters* are actually seeing or hearing, so as not to lend unwarranted legitimacy to the bluff.<sup>92</sup>

Playing an NPC is bad for two reasons. Ordinarily, the GM only interacts with players and never with characters; when you play an NPC, a player is also interacting with you as a character. It's extremely hard to keep the two personas completely separate, both for you and the player. The player may get angry at you as the GM because his character is angry at your character, or worry (justifiably!) that the secrets she tells the GM will come back to haunt her through the GM's NPC persona. The other reason a GM should never play an NPC is that they are creating an artificial GM shortage since they aren't as available for questions. If you must for some reason play an NPC, at least never play one that can shoot and kill a PC.

## 6.5 Managing Ghosts and NPCs

When people show up to watch your game, make sure they get a white headband and know about ghosting etiquette. Try to get the white headbands back—the Guild is constantly running out of them because ghosts walk off with them. If there aren't any white headbands, they can fashion a substitute from masking tape.

Most of ghosting etiquette also applies to GMs, so I'll list it here.

- Do not speak unless spoken to.
- Speak quietly when you speak: you are not here.
- Do not pass on in-game information.
- Don't block places people may need to run or shoot through: stay out of doorways, at the side of corridors, and on the side at the back of a mob.
- Run and get a GM when the players need one.
- If the players tell you to go away, then go away.

As GM, you have the right to banish an obnoxious ghost. Don't do this. Usually, you can at least find scut work to keep them out of people's hair.

When things are chill, whisper secrets and funny stories to the ghosts in the corner. Ghosts who hang around for a while can become extremely useful helpers, especially as they get to

---

<sup>92</sup>In one run of *Krazny Oktyabr*, the radio officer ran a bluff pretending to receive launch orders on the radio, which was publicly known to be broken. The GM present informed players, "The lights are off and the screen isn't showing anything, but he's definitely pushing buttons." Nobody thought to ask to hear the message themselves, or wonder how the radio officer was receiving anything from something with "BROKEN" still written on the sign in big letters.

know what's going on. Ghosts should never make rulings, though players can ask them to make calls about things like whether a disc hit somebody.

When somebody dies, let them come back to ghost as quickly as is reasonable. It's often nice to have a time-limit after which bodies are discovered for this reason. In a confined short game, be even more lenient: after somebody's been lying around dead for 10 minutes, it's probably OK to let them out to wander.

Ghosts and NPCs can both help a lot with running a game, doing setup and providing extra color characters for a world. Make sure they are briefed precisely before you send them out, however, and never give them significant power or abilities unless it's been planned with other GMs in advance—they're there to enhance the player's experience, not disrupt the flow of events.

## 6.6 Maintaining the World

Things get grunged and disappear, particularly in a ten-day. In a short game, you don't have to worry about this much, but in a ten-day, much of your time will be spent repairing broken or supposedly broken parts of game space. Have extra copies of everything either on-hand or easy to print, and check that important high-use things<sup>93</sup> are still where they're supposed to be on a daily basis. Players will come and report more things to you constantly, especially if you have a tunnel mechanic, shadowruns, or riddle trails, and sometimes the stuff is missing and sometimes the players just didn't find it.

Expect each mechanic distributed through game-space to require an hour a day of maintenance. You can reduce the number of times players just fail to find something by placing objects in predictable locations.<sup>94</sup> You can reduce the amount of grunging by reducing the amount that things are noticed by NPs: things at eye level are more likely to be grunged, as are things in high-traffic areas or on doors, and small packets are less likely to be grunged than large packets. Also, don't put things in areas where temperature varies often, like next to external doors: adhesive doesn't like temperature stress and your dots and packets will drop like flies.

Maintenance is a bit tricky when you're dealing with hidden dots and other secret information, since you don't want to let the player know whether they've really found the location or not. Tell the player you're going to go verify the dot, then go to the place by a slightly circuitous route, leaving the player behind. Once you've found or repaired the dot, come back and let the player know. Hunting for dots can be discouraging: it's usually a good idea to have something in the game that lets players get the location of a dot they've been stumped on for a day.

---

<sup>93</sup>e.g. a black market where players can get guns and bullet-proof vests

<sup>94</sup>e.g. the shadowrun sign is always 2 feet above the floor

## 6.7 Dealing with Brokenness

Your game is broken. You don't know it yet, but the players will find the broken parts you've overlooked, and you'll have to figure out how to fix them while the game is running. Sometimes the problem is with the rules, and those almost always need to be fixed. Other times the problem is with game balance, and you may need to try to gently ameliorate it or just let it stand.

First of all, don't panic. If you panic and make a hasty ruling, you'll compound the problem, screw things up even more, and confuse the hell out of the players when you have to patch it again. If you have to, just declare something off limits for a little while—your players will understand and appreciate that you're taking the time to fix it correctly.

Ask yourself if this brokenness needs a patch, or just an apology. Once something's running, it's hellishly hard to patch, and you may be better just acknowledging that it's broken and running with it as is. This is especially the case if players have figured out a reasonable way to use your mistakes to their advantage.<sup>95</sup>

When you do patch, get consensus in the GM team and propagate the patch in the fastest and least disruptive way you can. If the patch is a rules change, you may want a lag before the patch kicks in so that different parts of game aren't running on different rules. NPCs are often a good resource for patching, and provide a way to introduce resources that the game is missing.<sup>96</sup>

## 6.8 High Stress Situations

Something's gone wrong, and suddenly people are on edge and you're in the middle of it. You have all the authority in the world, but if you make a wrong move with it, you'll fuck up everybody's experience. How do you handle it?

This is a serious test of your ability as a GM. You have to think on your feet and have enough of your game in your head that you can avoid breaking anything as you deal on the fly. There's no universal answer for this, but I'll give a few pointers from my experience and some example of situations you're likely to encounter.

Principles of crisis management:

- **Don't panic:** Take as long as you need to take to fix things. Holding up game for a

---

<sup>95</sup>In *Nanopunk: Tranquility Base*, the nanoplague was supposed to be halted in the first hour of game, but the good doctor died fifteen minutes in, and every time we tried to patch things by adding another way to cure the plague, the evil doctor who caused the plague would get ahold of the new fix and sit on it. Eventually, we had to give up and accept that the game we were running was a different game than we thought we had written.

<sup>96</sup>In *L5*, all the research plots bogged down because too many players with research skills had died for unrelated reasons. We sent in a few researcher NPCs so players could talk to them, discover they had the skills they needed, and bribe them to help.

little while is better than fucking up game for good.

- **Acknowledge your dilemmas:** Be honest with the players when things are screwed up. Players understand that shit happens, and when you're straight with them, it's easier for them to give you the benefit of the doubt.
- **Accept player help:** Players, especially those with lots of experience, often volunteer ways that feel fair to them for sorting out a problem. Don't be proud; accept their help when it's useful.
- **Appeal to impartial authorities:** Avoid favoritism and the appearance of favoritism by using things like a coin-toss to resolve ambiguous situations.
- **Triage:** Get as much of the game up and running again quickly as you can. If necessary, you can separate out a few people and keep them on ice in the GM room while you figure out how to sort things out.
- **Minimize Halts:** If you need to halt the game, try to get everything sorted at once so you won't have to halt it again. A dramatic situation can often survive one halt, but rarely two.
- **Don't Erase Memory:** It's a rare player that can effectively act as though they haven't learned something. Erasing character memory is a solution of last resort.
- **GMs Don't Disagree:** Sort out all serious GM disagreements behind closed doors so you can present a united front to the players. Never undermine another GM's authority, or make players feel like they're going to get different rulings from different GMs.
- **Attain Resolution:** Sometimes there's nothing that will satisfy all parties. Pick a reasonable resolution and stick with it, even if it's going to annoy one of the parties.<sup>97</sup>

### 6.8.1 Some situations you are likely to have to deal with

**Sorting out mass confusion** Several bombs just went off a few seconds apart in the common room and everybody's shouting at each other trying to figure out who's dead and who's alive. Call a halt if it hasn't been called already, and take charge. Sort out the bombs one by one, asking people how they got where they currently are, accepting corrections from players who saw things better than you. For questionable cases, flip a coin, being a little forgiving, and rule on the spot. Check to make sure people are ready, then call the resume.

---

<sup>97</sup>During *Tenchi Muyo*, an ambiguous situation meant it was unclear whether my character escaped or not. The GM choose to resolve this by making two alternate endings, one in which I did and one in which I didn't. Everybody involved was less satisfied than if the GM had just picked one, whether it favored them or not.

**Mass NP invasion** The whole common room has come to besiege a secret meeting on the first floor, and NPs keep walking through halting things. Call a halt and tell people things are going to be virtualled in a different class-room one floor up. Let the people inside move up first, bringing secret things out under cover if necessary, then move the mob up, get everybody into reasonable positions a little farther back than they were, and call a resume.

**When you find out a player has fucked up** First, assume innocence. Most of the time, a player screwing up the rules just requires a stern talking to and a warning not to do it again. Sometimes, however, you will need to penalize them, because they've been getting too much advantage from their screw-up. Be extremely careful with this.<sup>98</sup>

**Dealing with an angry player** If it's in the middle of game action, call a halt and give them time to cool off quietly. Remind everybody that it's just a game and that people can get worked up sometimes. If it's not in the middle of action (often as a result of getting killed) bring the player into the GM room and let them rant and rave and explain to you why they're so angry. Listen. No, really listen. Often, that's what they need, and your honest sympathy for how they've been done wrong by that asshole who killed them will let them calm down and figure out how to deal. Call for backup if you need it, and especially if you find yourself getting worked up. Never yell at a player.

**When another GM has fucked up** Another GM made a completely crackheaded ruling, and the affected players are ripshit. Ask the players to wait while you sort things out and call a GM conference. When you've figured out what to do, let the offending GM announce that *we* have come to a different decision on the ruling.

**When you're the one who fucked up** First and most important, apologize and take responsibility. You need all the forgiveness you can get, and being straight with players is the best way to get it. Call for support from the other GMs, and have them help you sort things out as best you can.

## 6.9 Keeping the Team Together

Keeping the GM team friendly and synchronized during game is always important, but especially in a ten-day.

For a short one-night, you can usually keep everything in your head for the whole game. For anything longer, keep a log book in the GM room where you write down all rulings and

---

<sup>98</sup>We found out that a player had misinterpreted rules in *L5* and been able to brainwash several characters as a result, and some of the players were pissed when they realized what had happened. After a lot of agonizing, we decided to resolve the situation by having the brainwashing slowly wear off, having first warned the brainwasher so she could have a head-start on the lynch mob.



important events. Give every GM a notebook so they can record things that need to be put into the log book.

Radios and cell-phones can be a great boon to synchronization, but if you rely on them you can get badly screwed. MIT's buildings are fairly impervious to radio signals, and even HAM radio and cell-phones have dead spots in unexpected parts of campus. Family band radios are cheap and easy, but don't have good range and often get interference from other people. HAM radios are excellent, but you have to have a license. Cell phones are good, but you can't get the sort of free-form multi-way chatter that you can with a radio. The other problem, especially with radios, is discretion: if I call over the radio, "Joe, can you go sit in on the Chthulu ritual in 4-270?" and he doesn't have an earpiece, then I've just told all the players nearby.

Make sure you rotate GMs through game space: don't leave one person stuck in the GM room for the whole game. Drop in on the GM room every hour or two, just in case: the person on duty there may have had to run off and resolve a crisis elsewhere, or may just be going stir crazy and need to trade off. Conversely, important rulings or events may have occurred that you won't find out about if you don't check in at the GM room.

When another GM comes in to game, sit down together and brief each other on what's been going on in the other's absence. Tell stories and laugh about the players behind their backs. Hang out in the GM room and talk about things besides game, when you're capable of it. Listen to each other's annoyances, and try to be generous: game is a high-stress environment, and you don't want your teammates to fall apart on you or start to hate you.

Above all, remember it's a game and it will be over far too soon.

## 7 Postgame

Your game is done, some players are feeling good and some are feeling shitty, everybody's exhausted. Time to go home? Not yet: you've got some important responsibilities left. These won't make or break your game—that's already been done—but they are duties to the Guild and to your players which you need to fulfill.

If you're running a short game, you'll want to do cleanup and wrapup immediately after the game ends. For ten-days, the game usually ends at noon on Sunday and wrapup happens an hour or two afterwards: most people aren't actually in game at the end and appreciate the extra time to sleep in.

### 7.1 Cleaning Gamespace

Before you let the players sit down for wrapup, make them clean up the game space. Remember how hard setup was? Cleanup is even worse because people have moved everything around, and you really don't want to have to do it yourself.

Cleanup is also very important for the Guild's reputation: when there are signs all over the place saying they'll be torn down a month ago, it pisses off the NPs and makes them likely to tear down signs while games are running.

Make a plan beforehand as to which players will clean which areas. It often helps to organize things by groups and assign buildings to each group (e.g. CIA gets buildings 4 and 12, KGB gets building 24, FBI gets building 26). If you can, supply each group with a box to fill with junk they collect and a list of what they should expect to be in their area. Make a list of secret locations likely to be missed by players and check those yourself.<sup>99</sup>

Yell at people to clean up their stash sites. A good stash site will never be found except by the person who put it there, or occasionally other players five years later.

Don't worry about dots and other micro-sized cleanup: your main goal is to recover all the props and make sure all the signs come down.

Don't overflow the trashcans. If necessary, send runners out to put trash in other cans. You can also often find extra clean trash bags underneath the current bag in a trash can. Especially make sure that all food is cleaned up from the common room, GM room, and other places where people were lounging. Failure to clean up jeopardizes the Guild's room reservation privileges.<sup>100</sup>

## 7.2 **Wrapup**

No matter how awesome your game was, people don't want to sit there listening to you tell about it forever. You want the part of wrapup where you stand there and talk to last no more than about 30 minutes.

Your wrapup document helps you in many ways. You can stay inside the 30 minute limit by skimming over detail (especially complex world background) that's spelled out in the document. It also helps you remember all the characters and all the plots in game.

Be sure to mention every character and something neat about them during wrapup. Nothing makes a player feel crappy like having the GM either forget their character or say, "Oh, and Carol's character pretty much sucked, but we made her play it anyway."

Tell a few of the neat stories you got to watch unfolding during game, read off all the quotes you recorded, and give out awards if you're planning to (awards usually only happen for ten-days). Let the players tell a few stories too, but don't let it go very long: there's be time for more storytelling afterwards, and not everybody wants to stick around that long.

Finally, declare the game dead or alive. If the game is alive, you expect to run it again within a year's time, and players should not tell stories to people who haven't played yet. Storytelling is an important part of Guild recruiting, so you should declare games dead

---

<sup>99</sup>*Reality Check IV* had pseudo-character sheets which we used to set up each building and which we gave to players so they wouldn't miss secret locations during cleanup.

<sup>100</sup>The Guild currently has special deals that let us make reservations that other groups aren't allowed to.

promptly.

Once you're done, open the floor up for shameless plugs. People will get up and tell about upcoming games, whether scheduled, in progress, or looking for more GMs. In addition, be sure to give shameless plugs for the following things:

- Patrol (“running every Saturday”)
- Upcoming Guild meetings
- Players writing their own games
- The `assassin-writers` and generic mailing lists.

### 7.3 Return to the Office

Now quickly, before the players disperse, get them to help you carry everything back to the Guild office! Once you're there, put everything away right then and there, using all your extra hands. If you don't, you're just going to have to come back later and do it on your own. Of course, you could be an asshole and just leave the office a mess, but then I'd have to mock you and give you a wedgie.

When everything's done, reward yourselves by going out to eat and tell stories. Bask in the glow of a job well done.

### 7.4 Last Rites

There's just a couple last things to do in the week following game. Prepare your game for inclusion in the `a-archives` by deleting all the applications (they contain confidential information like people's hate-lists) and latex turds (`.dvi`, `.aux`, `.log`, etc), then zip it up and send it to the Guild secretary.

Send an email to your players thanking them for playing, reiterating the game's status as alive or dead, and opening it up for discussion on `generic`.

If anybody had a truly stellar performance, consider emailing the High Council and nominating them to become a Master Assassin.

You're done, and I hope this document has been helpful. Now, what will you write next?