So You Want to Be a GM

Ariel Segall
Kevin Riggle
Jake Beal
Andrew Menard
Ken Clary

July 14, 2012

The Disclaimers

All Rules are Made to be Broken
(just don't do it unless you mean to)

There Is No One True Way

What We'll be Covering

- So You Want to Write a Game
  - Team selection and dynamics, basic gamewriting logistics, core game ideas
- Plots and Characters
  - Creating them, balancing them, having enough but not too many
- Mechanics
  - Selecting them, balancing them, testing them
- Economies
  - What they are, how to balance them, why not to be scared of them
- Things That Go Wrong and How to Fix Them
- Preparing to Run Your Game
  - Sanity Checking, Apps, Casting, Handout, Runtime, Wrapup

Getting Started

In this section, we'll talk about:

- Creating and managing a GM team
- Zampolits
- Basic gamewriting logistics
- Creating and agreeing upon a game concept

What Do You Need in a Team?

- Shared core vision for the game (we'll get to how to develop that shortly)
- Work well together under stress
- Produce content
- Communicate well

Everything else is optional!

Team Dynamics

- Teams change, and **that's OK**
- GMing is high stress; games can kill friendships. It's better to go your separate ways than try to hang on and have everything explode.
- Good GMs may not all make a good team, or all be good for the same game, and **that's OK**.
- Trust is essential in a GM team.
  - If someone has a task, and other team members insist on micromanaging, the game won't get written.
  - If you don't trust a person with their task, they should be doing something else; or there should be checkpoints and a backup plan.
  - If there's nothing you trust a person with, this might not be the right team for one of you.
  - Equally, you must *be worthy of trust*. Get done what you say you'll do; if you can't, communicate early and often.

The Zampolit

The High Council will require you to have a zampolit in order to be scheduled. You could get a rubber stamp, but you don't want one.

- A good zampolit will read all of the progress on game. He or she will tell the team if you're deluding yourselves about how finished you are.
- A good zampolit will play project manager, and help you set realistic deadlines. . . and hold you to them.
- A good zampolit will be the bad guy if there is a problem GM, so that the rest of the GM team won't be resented.
- A good zampolit will prioritize GM sanity as well as getting the game done. . . unlike many GMs.
- A good zampolit will cancel your game if need be, *and this is a good thing*.

Choosing a Zampolit

Some things you want in your zampolit:

- Someone your team is comfortable accepting criticism from.
- Someone who is willing to criticize your team.
- Someone who is willing to accept *all* of the zamp responsibilities, including reading all of the game-in-progress and cancelling the game if necessary.
- Someone who has read the entirety of Jim's how-to-zamp instructions[1] before agreeing.

[1] http://web.mit.edu/jlwfnord/www/zampolit.pdf

GameTeX and Other Tools (1/2)

So you want to write a game. What will you use to do it?

- GameTeX: the current Guild default, for a reason
  - Pros: Support for all Guild standards, well-documented, modifiable to support game-specific needs, lots of knowledgable people willing to help, all of the document-creation power of LaTeX.
  - Cons: All of the complexity of LaTeX, including the cryptic error messages. Not really database-compatible if you're trying crazy webapp tricks.
  - This is the right default choice; the skills will carry over.
- Gameki, Xavid's bleeding-edge new alpha game-writing tool
  - Pros: Designed for easy player printing and webapp integration, wiki markup language for people who are scared of latex; both web and subversion access; back end in python, not latex.
  - Cons: Doesn't have the full power of latex; minimal support and documentation; **alpha**.
  - You probably don't want this yet. . . but if you're interested in its features, be aware that it's coming, and he's looking for testers.

GameTeX and Other Tools (2/2)

- Your Favorite Word Processor
  - Pros: You probably already know how to use it. Some of them come with, e.g., changelogs that some teams like.
  - Cons: You'll be recreating everything from scratch if you need anything other than basic sheets. Not all are cleanly compatible with version control.
- Wikis
  - Pros: Easy to set up nowadays; easy GM collaboration
  - Cons: Clean printing can be a pain; version control often very dubious

The technology you use doesn't matter, as long as it meets your team's needs.

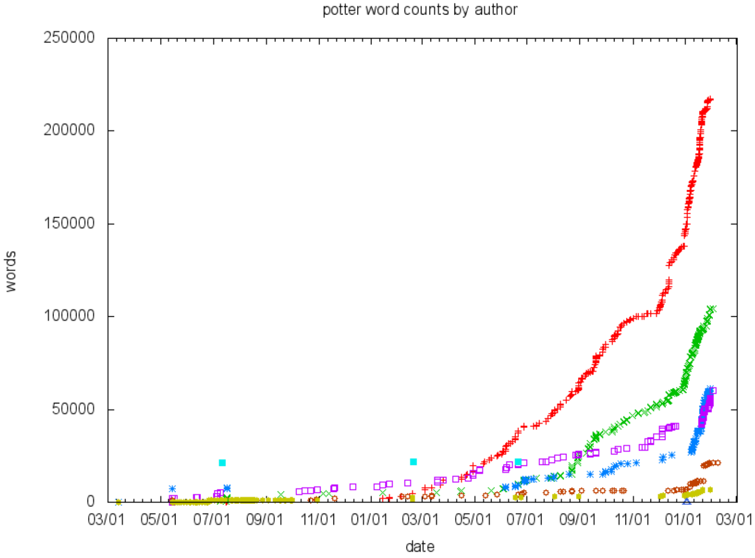
Tools: Version Control

The most essential technical tool for gamewriting is version control.

- Many options; subversion (svn) and git are the most popular now
- Not locked to any given machine
- Multiple people can do overlapping work, with clean merges
- Reduce the chance of accidentally losing work
- Allow restoration and lookup of earlier versions; handy in big projects
- Commit Logs (zephyr, e-mail, web update, whatever) are a really useful tool, especially for your zampolit.

You can look up the details later, but you want one no matter what tech you use to write.
Every GM on the team must be comfortable with the basics of the version control system.

Writing a Game



This is what the word-count checkin graph from HP7 looks like.
Hint: You *don't want* to follow this example.

Getting a Game Written

This is critical. **If you remember nothing else, remember this.**

- Plan your expected writing time. Add slack time. Now double or triple both. No plan survives contact with the enemy.
- Make sure you have sanity check time for every GM to read over every sheet, mechanic, etc. in game.
- Talk to your zampolit about your writing plan. Be sure you all know what you're committing to as individuals; write to that schedule for a few weeks so you know what it feels like. *Then* schedule.
- Commit early, commit often. Don't just wait until a sheet is complete.
- **Do NOT** plan to "get it done in crunch time". You'll have more than enough to do in crunch time. Plan **nothing** but sanity checking during crunch time; then you might stay sane.

Habits of the Healthy GM Team

- Don't surprise your GM team.
- Don't penalize people for saying they can't do something, as long as they're timely. Either way they wouldn't have done it; reward communication, because communication lets you plan.
- Assume good faith. No one wants to let their friends down, but everyone occasionally finds themselves with work than they can handle. Help them triage. This does not mean you should believe them the next time they promise they'll do something for you, however.
- Everyone on the team should know the big picture, even if they don't know the details, and be on the same page.
- Every expert skill should have a backup GM. Every piece of technology should have multiple people capable of using it; preferably everyone.
- Everyone's ego should be invested in turning out the best collective product possible, which will mean putting your own darlings aside to do what's right for the group and the game.

The Core Vision

- Why here?
- Why now?
- Why these people?
- Why only this time window?
- How is this game different from all other games?

Seeding the Game

Scope of your game

- How long? How many players? What game style? (e.g., SIK, high-action, diplomacy, party, spy...)
- Who are your players? (Experienced Guild players or frosh? MIT students, or Intercon attendees?) What's your approximate gamespace? (Anywhere at MIT? One room?)
- Why here? Why now? Why these people?

Scenarios That Don't Make Good Games

- A few spotlight characters and a large supporting cast.
  - NPCs or "Horde PCs" are ok, but people want to know going in that they're playing one.
  - This is one reason that having only a few people in the high-epic plot can go wrong!
- A scripted story that can't be affected.
- Scenarios with minimal conflict.

All rules are made to be broken, but good games with these are built completely around turning these scenarios into real games.

Themes and Motifs

- How is this game different from others in its genre? What's its hook?
- What core ideas should the game be centered around? This is what your players should see to make them feel like they're part of the theme: "not enough to go around" and "sacrifice" for Promises, "wizards" and "school" for HP7. Every character should touch at least one of your themes, and have a relationship to it (not necessarily be a part of, but have an opinion or a shaping experience or a goal.)
- . . . which leads to: What do you expect your players to be doing during game? Keep this in touch with your themes. Star wars → dramatic fight scenes. Dresden → grand magic.

Managing Player Expectations

- **Branding:** Players expect genre conventions to hold unless you explicitly declare that they won't.
  - Don't assume your players will figure out that your Death Eaters aren't evil or your Jedi come in three flavors of morally neutral; they won't.
  - If you stick too closely to genre conventions, your game may feel predictable and generic.
- **Feel:** tragedy vs. comedy; heroic vs. mundane; fantastic vs. realistic.
  - Jokes and puns in a tragic game breaks feel; so does a magical messiah in a spy vs. spy game. Breaking feel breaks people out of character.

The Core Vision, Again

- Why here? Why now? Why these people? Why only this time window?
- How is this game different from all other games?

How do you know if you have these? The blurb test.

The Blurb Test: An Example

The blurb should tell you:

- Who are your characters?
- What is your setting?
- What are your characters doing?

And most importantly: it should get everyone on the team excited and pointed in the same direction.

End of Getting Started

Questions? Comments? Discussion?

Plots and Characters

In this section, we'll be talking about:

- Creating plots
- Creating characters
- Plot balance

Plots

Do you have to do plots first? No, but it's usually easier to generate characters from plots and scenario than plots from characters.

Definition of a plot: thing which people have as a goal which requires action to accomplish.

- If you don't have to spend time on it, it's not a plot
- If you don't know about it, it's not a plot.
- Making a choice is not a plot; and if you have to choose between two courses of action, only one is a plot.
- If you can't make meaningful progress towards the goal, it's not a plot.

Things That Are Not Plots

- Keep Your Secret (Note: not a plot *and* not fun!)
- Drug Addiction
- Black Market Contact
- In Love
- Act Gay/Masculine/Chaotic/Religious. . .

Fixing Things That Are Not Plots: Exercise

- Keep Your Secret
  - You're being blackmailed; get the blackmail, fix the original problem, or neutralize the blackmailer.
  - Race to acquire evidence.
  - Safely pass on the vulnerable baby so you can Reveal Yourself and go back to war. . .
- Drug Addiction
- Black Market Contact
- In Love
- Act Gay/Masculine/Chaotic/Religious. . .

Who Determines Success?

Player-versus-Player (PvP) or Player-versus-Environment (PvE)

- PvP: PCs' actions alone determine who wins (possibly with mechanical assistance)
- PvE: GMs (usually via a mechanic) determine who wins

PCs vs PCs

Usually you want most or all of your plots to be PvP.
PvP advantages:

- Reasonably self-balancing (same resources, same # pcs, approximately even)
- Less for GMs to do
- Great for high-conflict, high-adrenalin, high drama
- PCs will surprise you, and that's great.

PvP disads:

- GGM problem (breaks balance, risks isolating "bad" PCs)
- Solve problems by killing (can be advantage in some games)
- PCs will surprise you, and if you don't plan for it, that can break game.

PCs vs. the World

PvE advantages:

- Can set up plots intended for the GGM (or all of game) to tackle; this can make great unifying game scenarios.
- Good for setting up friendly, low-conflict, high-heroism plots
- Allows for fine control over pacing and story for a given plot. (e.g. research notebooks)

PvE disadvantages:

- Very difficult to balance
- Very work-intensive on GMs
- Can result in a low-conflict (and thus fairly boring) game
- Can result in resentment of GMs if PC expectations are violated in a negative way

Success Criteria

- Defining success is not required for a plot, but it helps.
  - Particularly important when multiple people involved, so they agree whether they've succeeded or not
- "What are we trying to accomplish?" should always be clear
- Multiple tiers of success are often a good idea
  - In PvP plots, can maximize conflict while still making people feel they've accomplished something
- Player-defined success: Fun but not predictable
  - Not generally a good idea if there is opposition or dependencies
  - Great for lightweight schtick (e.g. love plots, morale plots, etc.)
  - Risky for difficult plots; easily turns into frustration.
- **Impossible plots are often not plots!**
  - If success is not possible in game, the activity usually isn't meaningful.
  - Try providing partial success criteria, so there are still goals to be met.

Rewarding Success

Congratulations! You've won your plot! Now what?

- Success feels more meaningful if it results in something concrete.
- **For optimal player happiness, make rewards/effects proportional to effort.**
  - Too much effect for too little effort, and victory seems cheap.
  - Too little reward for too much effort: "Why did I bother?"
- After a world-shaking ritual, *everyone* should see the results.
- If you have a power-gaining plot, you should feel more powerful.
- In long games, in particular, it is often beneficial to give *in-game* rewards.
  - If I won my plot and I can now do something cool because of it, my win feels bigger, *even if it's pure schtick*.
  - Showing off my win to other people means I get in-character victory moments.
  - Note: unique, story-generating rewards can be just as awesome as large mechanical rewards.

Sanity Checking Plots (1/2)

- What does a character do to achieve this plot's goals?
  - If the answer is "Wait until X, then react", you probably have a problem.
  - If the answer is "I don't know, something creative", you have a problem.
  - If the answer is something out of proportion to the plot, you have a problem.
- What mechanics does this plot need?
  - If the answer is "lots", you probably have a problem.
  - If the answer is "this lovely unique thing nothing else uses", you want to reconsider either this plot, or reusing the mechanic.

Sanity Checking Plots (2/2)

How much time needs to be invested in this plot? All plots?

Special guest speaker!

Characters

Before you begin creating characters, think about your core concepts:

- Who are these people?
- Why are they here?
- How do they relate to each other?
- What will they spend their time doing?

Designing Characters

- A character in its simplest form: a bullet list of plots/goals, with a[n optional] personality.
- Working from a bullet list makes balancing characters easier.
  - Quick summaries are easier to review
  - Large imbalances leap out at you the way they don't in text
  - Difficult to fall into the failure mode of "awesome character, nothing to do"
- Also handy: collecting bullet lists in one file makes for easy-to-skim game summaries

A coherent personality and backstory is non-essential. . . but is much of the difference between a good and a great game.

The Plot Matrix

Core idea: characters on one axis, plots on the other.

Name	**House**	**Low**	**Medium**	**High**	**Personal**
Neville	Gry		Defense	DA	Lead Gry
Seamus	Gry	Gob, Duel	Hacker	Con	Band
Parvati	Gry		TFC, ElfW	Norska	
Anthony	Rav	Pref, LSq	Techno, Hacker	NWO	Book Stasis
Wayne	Huf		ElfW,CoM	NWO	
Zivia	Sly		Hacker, Lib	Con	

Plot Balance Using the Plot Matrix

1. Do you have different plot types in your game? e.g., “high-epic”, “low-epic”, “personal”
   - If so, consider making a columns for each type. Not all columns need to be filled in for each character.
   - If there are high-epic plots, *strongly* consider giving one to everyone.
2. How many plots do you want for each character?
   - Two major and a minor for a one-night is a good rule of thumb
   - Larger games scale up significantly, but sometimes in plot size rather than number.
3. Remember: Plots and groups are not quite the same!

Plot Matrix Plus Plot Hours

Combining the plot matrix with your earlier plot hours calculation makes it easy to spot underplotted or vastly overplotted characters and groups.

Name	**Hrs.**	**House**	**Low**	**Medium**	**High**	**Personal**
Neville	24	Gry		Defense	DA	Lead Gry
Seamus	25	Gry	Gob, Duel	Hacker	Con	Band
Parvati	34	Gry		TFC, ElfW	Norska	
Anthony	26	Rav	Pref, LSq	Techno, Hacker	NWO	Book Stasis
Wayne	38	Huf		ElfW,CoM	NWO	
Zivia	21	Sly		Hacker, Lib	Con	

Fixing Plot Balance Problems: Underplotting

- **One of the most common failure modes of games.**
- **Underplotted characters lead to bored players.**
- To fix:
- Ask “What else might this character/group be up to?”
- Take their plots and add complexity
- Shorten game or combine characters, if it’s widespread!

Fixing Plot Balance Problems: Overplotting

- Rare!
- Overplotted characters have too much to do in too little time. They will drop plots, which can to failed GM expectations, and/or to their groups being screwed.
- **Overplotted players stop having fun when they feel forced to drop plots they'd like to do, feel obliged to stay in game when they don't want to be, or feel that their character is just going to lose because they can't finish.**
- To fix:
- Drop plots for them (making the reduction predictable)
- Give the characters tiered priorities, so the players feel it's OK to drop bottom-tier plots
- Simplify plot requirements to reduce time spent
- Ensure that groups do not require all individuals' participation to succeed

End of Characters/Plots

Questions? Comments? Discussion?

Mechanics

In this section, we'll discuss:

- How to decide what mechanics you need
- How to select appropriate mechanics
- Mechanics design (in brief)
- Player interactions with mechanics
- Balancing mechanics
- Mechanics testing

“So what’s your $FOO mechanic?”

There is no such thing as a ”required” rule or mechanic.

- The standard rules are a convenience, not a set of guidelines.
- Your game should start with an empty rules set and add what you need, not start with the standard rules and replace or add.
- Your game does *not* need rules for everything; if whether characters have sex is irrelevant except to characters, just leave it up to the players to decide if it happens; don’t add a seduction mechanic.

What Mechanics are For

- Mechanics serve one of the following purposes:
  - to allow players to do things that can't be done safely, legally, cheaply, ethically, publicly.... in real life
  - to give players something to roleplay over or with (e.g., money to bribe someone with, objects to acquire, deaths to cause or mourn...)
  - to resolve "what happens" when a player wants to do something, or when multiple players want something that conflicts.
- Ideally, mechanics are there to allow (and encourage) in-genre behavior.
- Two major categories of mechanics: PvE (do I succed when interacting with the world?) and PvP (which of us wins?)

Do we need a mechanic for...? (PvP)

How do I expect my players to interact with each other?

- Fighting each other? You need a combat mechanic.
- Killing each other? You need a killing mechanic–which may or may not be the same thing.
- Being diplomatic with each other? This might involve simple roleplaying, charm abilities, or might involve any of a monetary economy, a voting mechanic, a public opinion or political power mechanic....
- Espionage? Flirtation? Seduction? Mind control?

Ideally, you want to only provide mechanics for things which are:

- common in-genre behaviors
- critical to completing plots
- otherwise essential for game.

Mechanic or Roleplaying? (1/2)

Should it be roleplayed or mechaniced? You usually want a mechanic for:

- Things that are not *universally* ethical and appropriate for one player to do to another. For example: combat, seduction, flirtation.
- Things which are impossible.
- Things which would freak out a clueless watcher if they saw it happen out-of-context. (Combat is most common, but consider Nazi uniforms as a very different example.)

Mechanics or Roleplaying? (2/2)

- Things where one player may wish to do something to a non-cooperative other player. If there are negative mechanical consequences, there's a good chance you want a mechanic.
  - Persuading someone to cooperate with something with a negative mechanical consequence is different!
  - "Helpless or willing" requirements allow either mechanical or roleplaying solutions. This is often a good thing.
- Things where there is a mechanical effect, positive or negative, which players may wish to game.
  - Any long-term state change should have either a set of mechanical requirements to cause it, or mechanical limits on how often it can occur.
  - Example: Pantheon's "Act according to your deity's nature to get power" mechanic. Players could decide what qualified, but could only get benefits up to 3 times per hour.
  - **One of the most powerful tools in the GM arsenal is providing a mechanical benefit for desired behavior.**

PvE Mechanics

This covers everything which doesn't exclusively involve other players. For example:

- A race for office where the winner is determined in part by popular opinion of NPCs, which can be manipulated via bribes or the media
- A shadowrun, where PCs wish to bypass traps
- A combat module where PCs are fighting NPCs to achieve a plot goal
- A research notebook
- Shops which will sell you items
- A sign on the wall that tells you what you see.

Do I need a mechanic for...? (PvE)

When do I need a PvE mechanic?

- When I want there to be elements of the world my players can interact with which are more than what the players themselves see (items that aren't foam or plastic; locations not on the MIT campus....)
- When I want the world to feel larger than the limited set of PCs (an isolated spaceship game needs very few PvE elements; a game that takes place in Washington, DC in the middle of a presidential election, or at Hogwarts, probably needs a lot.)
- If it's entirely up to the GMs to determine whether a plot succeeds or fails.

PvE mechanics are harder to balance than PvP ones; the opposition is, effectively, the GMs. Before deciding to add a PvE mechanic, it's a good idea to ask yourself what role it serves in your game, whether you actually need it, and whether there's a way to make it actually PvP. (see next section for examples.)

Blurring the PvP vs PvE Line

Some things blur the line.

- Wargames usually involve opposing PCs interacting on a "game board" outside of game space, using external armies as an intermediary.
- Popularity mechanics often involve both PC and NPC votes.
- Competing corporations involve lots of PC cat-herding and negotiation, but with major mechanical or economic support indicating how the larger corporation is doing.
- Shops with limited stock have elements of both: what's available is affected by what other players buy, and some players will stock up to block the opposition.

Mechanics where players interact with each other via the environment are actually very common–but it's nevertheless a handy abstraction to keep in mind.

So I need a $FOO mechanic... (1)

Now you know you want a mechanic, what should it be? When to stick to the standards:

- It's really fundamental and you don't need to change it (e.g., How to Play the Game, sheet colors, item cards)
- You need something (e.g., death/killing mechanic) but the details don't matter.
- You want something like the standard; don't change the details unless you have very good reason, because people go on autopilot. If you do, highlight them; if you have a non-standard killing blow incant or non-standard count, people *will* screw it up.

So I need a $FOO mechanic... (2)

Rules of thumb, to be violated whenever appropriate:

- Design your mechanic to fit your game, not your game to fit your mechanic. (First decide what you need, *then* design a mechanic to get it.)
- A good mechanic involves no GM interaction.
- A good mechanic is self-documenting. (e.g., poker chips instead of stat changes.)
- A good mechanic is lightweight.
  - Sometimes mechanics need to be complicated, but they should be *no more* complicated than necessary.
  - Areas to optimize: State players need to track. Items players need to juggle. Numbers players need to do math on. Number of people who need to be informed of results. Places a player needs to go.
  - Players should have to remember as little as possible in combat time, and should never need to handle math with two-digit numbers under stress.

Selecting a Mechanic: An Example (1/2)

Let's say we have a game that wants people killing each other for increased excitement and paranoia. We need a combat mechanic. First, let's figure out what we want out of it:

- Fast and stealthy, for back-alley kills?
- Slow and dramatic for public duels?
- Mostly one-on-one, or allowing large group fights?
- Player skill, or character skills instead?
- Predictable outcomes, or semi-random?

The nice thing about combat is we have lots of existing choices. For example:

Player-skill emphasis	Patrol; Question Game
Fast, stealthy, predictable, minimal	Darkwater
Fast, predictable, allows styles and escape	MA/MD/MΔ
Fast, partially random	RPS
Showy, slow, optional player skill	Cinematic.

Selecting a Mechanic: An Example (2/2)

Combat Goals and Examples

- HP7: Want wand combat (ranged); but no need for melee; Patrol variant
- Right of Kings: Distinguish between shankings in an alley (Darkwater) and noble duels (dart guns).
- Hive: Melee combat should be big, important, and have an audience; Cinematic. Ranged combat is inaccurate and simple; disc guns.
- Synch: Modern gun-focused combat (Patrol-based); details varied between fragile mortals (early game) and demigods (late game). Death very separate from combat, and somewhat counteracted this!

Combat, or Non-Combat?

Many direct PvP mechanics can be thought of as combat mechanics!

- Seduction attacks? If one person can lose, that's combat.
- Battle of Wits? Losing face rather than blood, but the core is similar.
- Drinking contest? Chess? Court cases?

They may be colored differently, but you can reuse well-tested mechanics if you want similar properties.

- Note: this works in the other direction too!

Selecting Non-Combat Mechanics

If interactions are indirect, consider desired flavor.

- Lots of interaction with other PCs? Consider PC-tied resources (voting, charges, skills, memories...)
- Race dynamic? You want mechanics with limited resources that are claimed on access.
- Environmental interaction? Look into dot hunts, picture trails, shadowruns, reading signs. . .
- PCs acting as individuals groups? Some mechanics limit number of people involved; others have minimum thresholds
- Competetive or cooperative?
- Many approaches to success, or only one?
- Optimizable in advance, or real-time reaction?
- How much time do you want players to spend on this? (Both in-game, and pre-game making sense of it.)

Player Behavior: Optimization

If you give players a mechanic that they can optimize, they will.

Corollaries:

- If the optimal behavior is unfun, players will optimize it and then complain that they're not having fun; or they'll drop the mechanic.
- Every mechanic should be designed so that optimal behavior is a) what you want to see and b) fun.

Figure out what behaviour you want to encourage, and base your mechanics around that.

Player Behavior: Resisting Failure Modes

- Can a player high on adrenalin get an advantage by being physically unsafe or making others physically unsafe?
- Can a player find corner cases in the rules that are within the letter but against the spirit and screw over other players?
- Does an optimal solution for the mechanic exist?
- Can a clueless player screw up the person initiating the mechanic by having to look up how to respond?

If any of these is true, you usually want to reconsider.

So Now What?

You should now have a list of mechanics you want, and which properties you want those mechanics to have. Now what?

1. Do any existing mechanics meet your needs? Places to look:
   - The Standard Rules, both current version included with GameTeX and Jamie Morris' old version
   - /mit/assassin/Archive
   - xavid.scripts.mit.edu/mechanics
   - Ask around!
2. Can you adapt anything familiar to get the effect you want?
3. Come up with something, anything, that you think might work. Talk to your co-GMs about it; compare with your goals. Test it. Revise it. Iterate as needed.

Adapting a Mechanic: BoH Example

Battle of Hogwarts wanted a shadowrun mechanic.
Goals:

- No spell hands like HP7; that's too much state to track.
- Player skill involvement; it's fun.
- Show off character skills and spells
- More sense of space than packets on the wall
- No less than five minutes the first time; no more than 30.
- Doable in groups, but not mobbable.
- Can be set up in corridors if we can't get rooms for three days.

Designing a Mechanic: Buffy the Vampire Slayer Plot Points

- GMs wanted game to feel like an episode of the show:
  - People run around doing their schtick for most of the episode
  - Early confrontations between named chars are *never* lethal and involve talking and maybe inconclusive fights
  - Near the end, people who've been getting screen time get to do their big ritual or kill/permanently injure opposition.
- Characters had schtick they could do to get a Plot Point, with limits on frequency
  - Vampires snacked on MIT students; Buffy and her team staked nameless vampires; various people cleaned up corpses.
- Rituals required Plot Points to do; characters required Plot Points to kill proportional to their importance
- Going after plot points caused game to look like the show background; confrontations caused plot-point-generating epic duels or snarky conversations through the bulk of game; and at the climax, Stuff Happened and people got permanently hurt.

Balancing Mechanics

- There is no such thing as a perfect way to balance a mechanic.
- You are always balancing for **particular effects**.
- Know the effects you're going for, and verify that when applied to your game and your characters (and your players) it meets them.

Balancing PvP Mechanics

PvP:

- Figure out who you expect to oppose each other.
- Figure out what results you want when direct opponents clash.
- Figure out what results you want when people team up.
- Assign values accordingly. (Often, this will be “same numbers for each opposing plot.”)
- Test variants to make sure results are within expectation.

Balacing PvE Mechanics

PvE:

- Figure out how hard you want success to be. What does minimal effort look like? Moderate effort? Total dedication to the plot? Which should succeed?
  - How much time should people spend?
  - What resources do they need to invest?
  - How many other people must help them?
  - What don't they have at game start? (Items? Locations? Knowledge?)
- What happens if more people than expected get involved?
  - Can this mechanic be Good Guy Mobbed? Is that OK?
  - If not, what prevents that? Options include participation ceilings, diminishing returns, Sketch Factor...
- What warning signs do the GMs have if the mechanic turns out to be unbalanced?

Example: Balancing Darkwater

Let's look at two different games that are both using Darkwater: a superhero game, and a gritty noir game with warring gangs.

- Should weak people (assisting) be able to take down a strong person? If so, how many?
- Should assisting be generic or not? Should assists give combat stat or generic $+1$?

Superhero game: Let's say three mooks need to team up to take down Batman, two Batmans to take Superman. Can two mooks plus Batman take Superman, or not?

- Having decided three mooks can take Batman, what about two mooks with knives? One mook with a chainsaw?
- Can anyone wound, or does it take an edged weapon? How common are they?
- Are knockouts possible? How about with edged weapons?

Sanity Checking Your Mechanic

Some good rules of thumb to keep in mind:

- If it has a mechanical effect, it should have a cost.
  - This could be in time, resources (including limited uses), and/or social capital.
  - Otherwise, the mechanic can be "spammed" and break.
- The cost should generally be proportional to the effect.
  - Relative cost is more critical than absolute cost; a good ability that costs less than a weak ability makes the weak ability useless.
  - Cost should be balanced based on the desired frequency of use.
  - Corollary: high-cost mechanics will only be used if their effects are considered to be worth it.
- If there is opposition, does any side have a noticable advantage?
  - If so, what counterbalances it?

Running a Mechanics Test (1/2)

The basic idea:

- Hand someone the rules and any materials (cards, notebooks, etc) that are required.
- Watch while they attempt to implement it.
- Take copious notes on where they have trouble, or where their interpretation isn't yours.
- Never correct them unless you are *also* making a correction in your notes; the goal is to run the mechanic without the GM in the end.

Running a Mechanics Test (2/2)

1. Start with your GM team. If members of your team can't understand the mechanic well enough to run through it, your players won't either.
2. Get a set of volunteers who are not on the team. Have them play through it a few times.
   - Use people who are as similar as possible to your casting target. If this mechanic will be used by all of game, get both inexperienced people and expert rules-lawyers.
   - If the *mechanic* must be secret, aim for people who won't be playing; otherwise, you can almost always anonymize the secret portions. (e.g., take "Jedi" off of the name of the combat card.)
3. Revise the rules or balance based on feedback; repeat as necessary.

Note: most mechanics tests *don't* need to be formal, scheduled events!

End of Mechanics

Questions? Comments? Discussion?

Technology and Your Game

In this section, we'll discuss:

- Technologies your players may wish to use
- Technologies you may wish to use in your game

Technology and Your Game: Player Technology

- This is the 21st century. Your players have a lot of cool toys.
- There's a section of the standard rules covering what kinds of technology are appropriate in-game.
  - Medieval fantasy game? Probably no cell phones.
  - 23rd-century spy game? Cell phones are probably fine.
  - OTOH, sometimes cell phones will mechanic telepathy or sending in the medieval fantasy game.
  - Or it's a post-apocalyptic 23rd-century spy game, and technology has regressed to Victorian levels.
- Sometimes you need to add constraints, eg.
  - "cc: all in-game mail to the GMs; no phones or texting," eg. if someone may be spying
  - or, "phones can be used to coordinate in-person meetings but not for in-game conversations"
- Your players will want to know. You should have an answer.

Things to consider:

- Not everyone has an iPad.
- Not everyone has a smartphone.
- Not everyone's phone has a camera.
- Not everyone has a cell phone.
- Not everyone has an Athena account.
- Not everyone has a laptop, or wants to bring it to game.
- Cell reception on campus is erratic.
- Wifi reception on campus sucks in certain places.
- Devices run out of power, break or get left home.
- Do you really want your players spending all their time in-game in an Athena cluster or staring at their laptop?
- No one can shoot you through a phone or computer: for most players, this is optimal; for most games, this is probably bad.

Technology and Your Game: Game Technology

Your ten-day does not need a webapp.

Your three-day does not need a webapp.

Your one-night definitely does not need a webapp, unless you've designed game around it specifically. (See Derelict.)

Your players don't care about your webapp

- Unless your game is designed to showcase it, your webapp will not make game.
- Your webapp may **break** game.
- A webapp, no matter how brilliant, can't make a bad game into a good game.
- **Your goal is for your players to have an enjoyable time.**

The Biggest Danger of Webapps

- LARPs are built on personal interactions between players.
- Webapps generally discourage personal interactions.
- If your webapp doesn't promote desired personal interactions between players (including, depending on the game, getting shanked), your webapp will make your game worse, not better.
- If I wanted to spend all my time playing with a webapp, I'd go play Echo Bazaar, not a LARP.

How do your technology choices serve your goals?

Your goals are to (ordered from hardest to easiest to work around):

- Run this game (to promote more interesting competition between players).
- Tell this story (to promote better stories between players).
- Build this world (to promote more immersive interactions between your players).

Your goal should not be to learn web technologies; do that on your own time.

- If the tech doesn't enhance the game out of proportion to the time spent on it, don't do it.
- If the time spent on the webapp would be better spent writing and editing sheets, skip the tech.

If the tech doesn't make the game better, don't do it.

Sufficiently Advanced

- Web technology is a tool.
- Web technology is not magic.
- A common failure mode for a mechanic which is too complicated and not well-motivated: “Eh, we’ll put it in the webapp!”
- If you can’t write a greensheet describing it, you can’t write a webapp to implement it.
- **Web technology takes time.**

What is your complexity budget?

A small webapp is a project on the same level of complexity as a one-night Guild game.

Timing

- Your webapp will take about as much time and manpower as a one-night. Plan appropriately.
- What else is your webapp team doing? (Sheets? Day job? Keeping the team from killing each other?) Take that into account.
- Your webapp should be feature-complete, if not fully tested, on the same timescale as your greensheets.
- Ideally, nothing but testing and bugfixes will happen in your san-check period.
- Your deadlines and resources (ie. number of GMs) are largely fixed, so when you fall behind, you need to cut features.
- **NO NEW FEATURES ONCE GAME STARTS!**

Non-webapp technologies

- When I say “technology,” you think webapp.
- Consider a typical Guild game:
  - Signs: technology
  - Packets: technology
  - Research notebook: technology
  - Mem packet: technology
  - GameTeX: technology
  - Blackboard and chalk: technology
  - Paper and pencil: technology
- Ask yourself: “Is there something simpler which could accomplish the same goals?”
- If the answer is “yes,” you don’t need a webapp.
- Try the simplest thing that could possibly work, and get more complex from there, rather than starting with an all-singing all-dancing webapp and simplifying.

Pre-existing technologies

- Raw HTML files hosted on Athena (bonus points for version control)
- E-mail, mailing lists, zephyr, AIM
- Blog engines (Wordpress, Movable Type, Blogger, Tumblr)
- Wikis (MediaWiki, Google Sites, GitHub Pages)
- Social networks (Twitter, LiveJournal)
- Video sharing sites (YouTube, Justin.tv)
- Document sharing sites (Google Docs, Google Spreadsheets)
- Whitelabel projects

So what are webapps good for?

Webapps are good for:

- Automating costly, repetitive, or complicated computation
- Hiding the details from your players
- Sharing widely-distributed state and allowing remote access (though here, consider connection issues) [2]

[2] … though consider Jake Beal's talks on resolving distributed consensus problems in Guild games for some simpler approaches:
http://web.mit.edu/jakebeal/www/Talks/assassin-slides.pdf
http://web.mit.edu/jakebeal/www/Publications/MIT-CSAIL-TR-2006-038.pdf
http://web.mit.edu/jakebeal/www/Talks/CSW-2006.pdf
http://web.mit.edu/jakebeal/www/Publications/CSW-2006.pdf

Questions to ask:

- Does the mechanic really need to be that complicated? (Too complicated for players is too complicated for GMs.)
- Do the details really need to be hidden from your players? (They'll figure it out anyway.)
- Do players really need to interact with the mechanic remotely? (Easier to get shot in the common room.)
- Do players really need to be touching the mechanic themselves, rather than through the GMs/Box/NPCs? (Trade-off: GM time during writing versus GM time during runtime.)

So you've decided you want a webapp...

- Who uses it, and what do they want to do?
- Sometimes it's useful to think in terms of "stories" (cf. agile development), eg.:
  - "I am a **researcher**, and I want to **see how many Research Points I have today**."
  - "I am a **researcher**, and I want to **allocate my daily Research Points to projects**."
  - "I am a **Captain of Industry**, and I want to **see how my researchers are spending their Research Points**."
- Your deadlines are largely fixed, as are your resources, so...
- You can't add anything to your initial plan without cutting something else.
- When you've fallen behind, you need to be prepared to cut nonessential things.

Descoping without pain

- If you can do 10% of the work and get 90% of the benefit, do that.
- Plan the order in which you will cut or simplify things if you're running out of time.
- It's okay if this changes, but people need to be clear when it does and why.
- This can't happen too often, or nothing will ever get done to completion.
- You need better backup plans for the things that are more likely to get cut, or the willingness to drop them from game entirely.
- If you may drop them, you can't be relying on them in essential mechanics, for example, without a good plan for what you'll put in their place.

Building your webapp

- Ideally, everyone does everything.
- In practice, you **must** have multiple GMs who are capable of using the webapp, debugging the webapp, and fixing the webapp.
- This means you need multiple people on any ACLs, including the hosting and database ACLs.
- Know your limits: be extremely careful modifying or deleting data.
- Security is not a primary concern, although it is a concern to be aware of, and your concern is more automated than sophisticated attacks, so take precautions accordingly.
- 90% of security between players is accomplished by a “don’t break the webapp” rule.
- You **must** have a backup plan. Don’t use it unless you have to, but be ready for when you do.
- Good software engineering practice: version control, backups, crash logging, audit logging, testing!

Five-minute software engineering practice:

- Version control: absolutely non-negotiable. The source of truth.
- Never deploy code not committed to version control.
- Backups: in particular, database backups.
- Crash logging: how do you know if something is breaking or a user is having problems?
- Audit logging: how do you roll back if a change is made in error, or track who made a change when?
- **Testing!** (next slide. . .)

Software testing: Automated

By definition, your first big test is your production deployment. You don't get a beta period. Anything you can do ahead of time to minimize surprises in production is a win.

- Test validity of your models; data, computation, interactions
- Testing is repetitive
- Computers are repetitive
- Write the test, watch it fail, then write the feature
- When you find a critical bug, write a failing test first, fix the bug, make sure the test now passes
- Feels time-consuming, but is often a win

Software Testing: Manual

- Test your user interface
- Automated ways to do this, but they're not a win at this scale
- File off the secret bits; get some friends to run some normal use cases, a couple to try and break it
- Load testing: if the site gets slow with 5 friends, it'll die under 50
- (this does not mean you need more computrons, necessarily)
- Pay particular attention to places where and how information is being exposed; "can't unsee that"
- Unavoidably time-consuming; treat it like your san-check period.

Expect failure

- Build your tech to fail **visibly**.
- Ideally, maintain somewhere a second **human-readable** copy of all the data.
  - Note: this includes runtime changes; logging is wise.
- If your players believe that information was revealed to them through a webapp bug, they won't act on it.
- This means that the errors you need to worry most about are the ones which don't look like errors.
- Platonically, all problems should be solvable in real time without coding.
- Do as little as possible, so if it breaks, as little as possible breaks.
- Four small webapps may be better than one big webapp.
- Only the players who need it should have access to it.
  - It's perfectly reasonable to have a computer-based wargame that only 10 out of 50 players interact with (though it may not be worth six months of a GM's time)
- It doesn't need to be pretty, it needs to **work**.

“Mistakes” users make (partial list):

You will make mistakes. Your players will make mistakes. In order to keep these from cascading, you need to handle things like the following:

- The webapp hangs, and they resubmit the form, thereby causing whatever action they were taking to be done twice.
- They forget if they updated the form, so they do it again.
- Two members of the same group each think they're the only one updating values.
- They screw up a RL mechanic, and update the webapp before it's discovered

You need to be able to see all the changes and arbitrarily roll some back.

End of Technology

Questions? Comments? Discussion?

Economies

In this section, we'll cover:

- What an economy is
- What you may want out of your economy
- Identifying parts of your game that interact with one
- How to balance one

How to Recognize an Economy

- It is some resource which is transferrable between people
- It has multiple sources and multiple sinks
  - “Hunt for Excalibur” is not an economy. There is only one of it; think of it as a McGuffin, not an economy.
- Does *not* need to be painted as money

Economy or Not?

Definitely Economy:

- Cash, Items, Charges (Synch), Focus Points (Harry Potter), Computrons (Arcadia),

Unusual Economy:

- Op-Eds (Synch) : not transferrable directly, but provided transferrable benefit; limited lifespan

Fuzzy Edge of Economy:

- Votes: If they're blind, probably not; if they're visible or transferrable, probably
- Prestige (Synch): Multiple sources and sinks; not transferrable, but somewhat visible

Not an Economy:

- Rich guy with millions of dollars; various recipients who each want it all. (This is a money-colored macguffin.)

Things You Need to Consider About Your Economy

- What are the sources of resources in your economy?
- What are the sinks in your economy?
- What is the friction in your economy?

Sources

- Starting resources
- Steady flow of income of resources
- Resources retreivable through plot or mechanic
  - Shadowruns with stuff at the end
  - Blackmail NPCs to get money
  - Win plot to get rewards

Sources, Cont.

Each of these give you a different flavor.

- Starting resources: People may spend quickly and drop out of the economy; they're done.
- Steady Flow Income: Lots of small deals over time, because I keep having more to trade; but it's predictable, so I can make deals early for future resources.
- Plot or Mechanic Income: Deals happen over time, but no forward-looking deals, because no one knows what they'll have later.

Sinks

- Known, Fixed Sinks
  - The Infrastructure Committee has a greensheet saying exactly what we need over time; it's predictable and finite over game.
  - The GMs know and can balance how much the sink will take.
- Open-Ended Sink
  - Examples: A Bidding War with no ceiling or floor on bids. Resources that can be invested in combat.
  - This is powerful: it can break you (maybe they spend everything on this plot) or it can fix you (absorbing excesses).
- Unpredictable but Finite
  - It will take some number of charges to get through this shadowrun; you don't know ahead of time. GMs can plan, but players can't...

Friction

Some things in your game will be sinks even if they aren't really. For example:

- Expiration
  - If a resource has an expiration date, some number of them will expire ever day. For example: daily charges.
- Hoarding
  - Some people will hold onto resources for fear of future; because they don't know what it does; because they forgot about it; because they stashed it and then died... Or because ending game with resources is often a character win!
  - Especially a problem if many sinks are sketchy!
- Transaction Costs
  - Negotiating resource transfers cost time; some (like writing op-eds) cost a lot. The more painful it is, the more likely that people who doesn't care will let their resources go to waste.
- Carrying Costs
  - If you can be mugged and lose your cash, you have an incentive to stash it (adding transaction time and loss rate).

Balance

- Figure out what all of your sources are
  - What are the maximal number of resources entering the economy?
  - What is the range of possibility for mechanics that may bring resources in?
- Figure out what your sinks are
  - What do you expect people to spend?
  - What is the upper bound on what people are capable of spending?
- Figure out how tight you want your economy to be.
- Estimate the magnitude of friction.
- Write down this equation written down in GM notes, so you can see how small change (adding resources, say) need to be balanced by adding sinks or friction
- You're going to also need this at run time, because the economy may break – if you know who should source X, who should sink X, you can figure out if the problem is a broken source, a broken sink, or unexpected friction.

Time is an Economy

- Each player has some number of hours they are putting into your game. [source]
- Each plot takes a certain number of hours. [sink]
- One way to look at research notebooks: Each notebook of sudoku is x hours per night. [sink/friction]
- One way to look at location and item placement: The time required to walk across campus is using a player's time investment. [friction]

If you want time to *not* be an economy (such that hardcore people investing hours win), reduce the ability for time invested to be transferrable; e.g., make research notebooks require a particular player's investment, make dots only visible to one person...

- This doesn't necessarily make for a more fun game!

A Worked Example: HP7 Focus Economy

Each character had a certain amount of focus per day.

We had large plot-rituals and small plot rituals; we estimated that we wanted it possible to do most of them before resources got tight, but the big ones would only likely happen near the end of game because it took time to build up focus (and do prereq rituals).

We had combat and shadowrun spells that required focus; this gave a tradeoff that if you wanted to be a badass, you might need to have other people do rituals for you, or might face a choice of "do I kill my enemies or do my plot?". This was openended, but we knew roughly how many NPCs the players would have to fight.

We assumed a fairly low rate of friction; we expected that at the end of game, anyone with leftover focus would likely be willing to put it into rituals for friends/housemates/whatever, and that there would be relatively few deaths or punts taking focus income out.

End of Economies

Questions? Comments? Discussion?

Miscellaneous Topics

This is the section for things that didn't fit anywhere else.

- Choosing game space
- Twenty-First Century College Students
- The power of the kludge

The Importance of the Right Game Space

Your choice of game space has many impacts on your game.

- A game space that is large reduces player interaction.
  - Players are less likely to run into each other… and less likely to interrupt sketchy activity.
- A game space that has broken lines of sight reduces player interaction.
  - Rooms, doors, and corners reduces players' ability to notice and react to interesting activity.
- A game space that is small increases the likelihood of mob activity.
  - Everyone can see or hear what's going on, so everyone gets involved.
- A game space that is highly visible reduces the ability to do secret activity.
  - It doesn't *eliminate* sketchy activity… but it dramatically increases the risk, so decreases the likelihood.

If you have unusual gamespace constraints, reserve your rooms early! (Best to do this anyway.)

The Twenty-First Century College Student Problem

- Your players are, by and large, liberal college students.
    - Racism and other discriminatory opinions are generally seen as bad. Unless explicitly told to, no one will play them up.
    - Bullying, gratuitous violence, and sexual harassment (let alone rape) are seen as Bad Guy activity; Bad Guys face ostracism.
    - If diplomacy can solve a problem, that's almost always superior to fighting.
- If you want your setting to involve radically different morality (especially the above) you need to take this into account.
- Setting and scenario statments are *not* sufficient to overcome deep-seated instincts. You *must* address violations of standard morality in individual sheets as well if you want them to be played.
- **Anything that seriously violates "good behavior" either in play or in backstory should have an app question.**

More generally: If there is something that everyone in the world has an opinion on, don't put it in the scenario. Give everyone an individual opinion; then they'll roleplay it.

The Power of the Kludge

Kludge, in this context: A thing that is true because of GM fiat.

- Kludges have a mixed reputation, but are powerful tools.
- A kludge makes it clear to players not to bother trying to find a way around a constraint.
- A kludge makes it easy for GMs to not worry about rules lawyers.
- In other words, *kludges regularize expectations*.
- Kludges should be consistent; like any other ruling, if you tell one person it's a kludge, you should tell anyone else involved as well.

Things that Go Wrong and How to Fix Them

In this section, we'll cover:

- Common failure modes of games and GM teams
- Advice for handling them
- When and why to consider punting

The Game Won't Gel; Or, This Isn't the Game You Thought It Was (1/2)

You had a brilliant idea, and you were all psyched! But somehow, when you tried to pin it down, it just didn't quite work. . .

- This is when you step back and revisit your core vision.
- Has the GM team developed disagreements? Bring them out in the open and try to solve them.
  - Nebulous disagreements can be highlighted by proposing blurb variants that emphasize what different people want from the game; that will give you a point to start talking from.
  - If you can't solve them, consider whether you should continue as one GM team, shed members, split into multiple teams, or punt. *None of these is something to be embarassed about.*

The Game Won't Gel; Or, This Isn't the Game You Thought It Was (2/2)

- Was there an area of it that people just didn't know what to do with, or weren't excited about?
- Do you need to consider a shorter game, if you're way low on plot? A longer one, if you're feeling cramped by the constraints?
- Can you propose variants that avoid the problem? Go back to the blurb test.
- Consider having each team member sleep on it, and come in with **their** best suggestions for what would make the game work for them. This will at least jump-start discussion.

You're Behind Schedule (1/2)

…and by more than the flex time you already built in.

- Are you on the schedule? If so, you either need to reschedule, or revisit your deadlines.
  - Talk to your zamp. If you revisit deadlines, you and they need to agree about when game is called.
  - **When setting ambitious deadlines, remember: GM sanity and sleep matter.**
- Is your schedule realistic?
  - Revisit your assumptions. If you've been working productively, you may just have a classic underestimation problem. Scale the rest of your schedule based on current progress.

You're Behind Schedule (2/2)

- What's actually being written?
  - Is everyone ignoring one kind of sheet? If so, figure out whether you need another GM, or how to get people comfortable with writing them.
  - Is writing not happening? Consider having writing meetings, so you can encourage each other.
  - Are some areas being perfected while others are neglected? Meet to reestablish priorities.
  - Are some GMs not writing? Figure out what can be done to support them; e.g., provide skeleton rituals, parameters for notebook contents, switch them to a different section of game, etc.

Common Problem GMs (1/2)

You'll probably encounter one of these. Try not to be one.

- The Ghost GM: somehow, they're just never there.
  - First, find out whether there's a root problem, or if they're just a flake.
  - If they don't shape up, either treat them as non-existent or ask them to leave the team.
- The "No" GM: They demoralize the team by blocking progress or ideas
  - Do they have a different idea of what's going on than the rest of you?
  - Work on establishing trust in the rest of the team, or better communication
- The Mismatched GM: Silly on a serious team, mechanics-heavy on a party game...
  - Revisit the core game vision, this time addressing the points of disagreement.
  - If you can't find a compromise or understanding, this is usually time to split.

Common Problem GMs (2/2)

- The Empty Promise GM: They promise a lot, and never deliver
  - Remind them of the importance of communication, and that the team would rather know than not.
  - If you're patient, teach time management.
  - Restrict them to only parts of the game that are optional, or ask them to leave the team.
- The Distractible GM: They found a shiny, and they're doing that first regardless.
  - Talk to them about priorities. Show them *why* their cool thing has to wait, or propose tradeoffs.
  - If that doesn't work, you must assume they will only do their shiny. Use it, or ignore them.

Your Zampolit should be the bad guy if it comes to a hard talk.

When to Punt (1/2)

Sometimes, punting your game is the right idea. There are two categories of punt: a schedule punt, and a total punt.

- Schedule Punt: You have a run date, but you're going to cancel.
  - If the game really isn't finished and the GMs will lose sanity, grades or jobs if they try to finish? **Punt**
  - If you realize that the game is severely broken and can't be fixed in time? Consider punting and revisiting later. Earlier is better!
  - If you realize that the game you have isn't the game you wanted to run and isn't up to your standards? Consider punting and revisiting later.

When to Punt (2/2)

- Total Punt: This game will never happen. Period.
  - Sometimes, you need to cut your losses. Your time matters; write another game.
  - If the GM team cannot work together? Punt, and talk about whether the remnants can be reused.
  - If the game has a fundamental flaw, and you can't find a way to fix it? Punt.
  - If you've fallen into a perma-funk where you talk a lot and nothing happens? Either jump-start it with deadlines, or punt.

End of Troubleshooting

Questions? Comments? Discussion?

Running Your Game

In this section, we'll cover:

- Sanity checking
- Writing an App
- Casting
- Production
- Rules Seminars/Handout
- Runtime
- Wrapup

The Importance of Sanity Checking

- Sanity checking takes time, but is one of the differences between good and great games.
  - Inconsistencies break immersion and balance.
  - GM oversights lead to player frustration.
- For best results, sanity check in stages during writing.
  - Particularly for long games.
  - The earlier you find problems, the easier it is to fix them.
  - At a minimum, note things to check during the writing stage.
- Sanity check *everything*; not just sheets.
- Sanity checking needs to be done by multiple people.
  - Writers tend to see what they intended, not what's actually there.
  - But if you don't know what should be there, you won't see what's missing.

Sanity Checking Your Game

Cross checking is the most critical sanity checking step.

- When reading a sheet (or notebook, or mempacket, or. . .):
  - Whenever another component of the game is mentioned, directly or indirectly, pull it up and **look at them together**. Even if you've done it in reverse.
  - Unless *explicitly* irrelevant or secret, a character should know everything other people know about them.
  - When cross-checking items or places, don't just check text. Check that they're actually placed in game, and how they can be accessed. Can everyone that should have access get to it? Are there enough? Too many?

Writing An App (1/2)

- First, for each character: What questions would we need to ask to determine whether someone would be a good fit?
- **Be aware of triggers**. Topics like rape, abuse, bullying, abortion, mental illness, racial or sexual discrimination, even religion, are things players can be sensitive to.
  - *Always* ask if someone's willing to play a character with a trigger. If you're not sure if it's a trigger, play it safe and ask.
  - You often want to distinguish between victim and perpetrator; many people are OK with one but not the other.

Writing An App (2/2)

- Once you have a list, consolidate and simplify. (e.g., burglar and conman might become "criminal")
- Add generically important questions, such as preferred character gender, romance plot gender, etc.
  - Remember to ask aobut both *player* and *character* gender for romance plots; some people care.
  - This probably means you need to ask about player gender; we recommend allowing freeform responses.
- Always ask for contact info, including e-mail (default) and phone (for when you're tracking down someone late to game)
- A freeform "Is there anything else you want to tell us?" is always a good idea.

There are many schools of thought about freeform vs. numeric vs. multiple choice responses, and no data about what works best. Use what you're most comfortable with.

Common Casting Strategies

- Best Fit Per Role
- First Come First Served
- Younger/Newer Players First
- Students Before Cruft
- Best Overall Game Balance (GM defined)

Try to avoid “My friends first”; it leaves new players out in the cold.

Common Casting Problems (1/2)

- Player A and Player B don't want to work with each other
  - **Keep this sort of request confidential.**
  - If this is just "don't work with", cast them in characters that don't have any reason to interact.
  - If this is "don't put me in the same game as":
    - First: Try to put them in separate runs.
    - If that isn't possible: Find out whether this can be downgraded to "don't work with". ("Would you rather not play the game, or play in the same run as X?")
    - If that isn't possible: You will have to pick one or the other to cast (or cast neither) *even if your game is short on players*. Use your best judgment; these situations suck.
  - You may need to clarify with the players; some people say the first and mean the second.

Common Casting Problems (2/2)

- A doesn't fit any character in game!
  - If you can fill without them, do so.
  - If you need or want them to play anyway, get back in touch and ask them questions.
- A is a good fit for the role by app, but GMs are dubious.
  - Ask yourselves: Why are you dubious?
    - Sometimes, this is a player trying to break typecast. This should be supported!
    - Sometimes, this is a new or weak player in a difficult role. Could you provide support in other roles?
    - Very rarely, they've apped for a role that's a bad fit; put them elsewhere. But in general, player preferences should trump.
  - Beware the "My friends/the cruft would do this best!" trap.

Production: Overview

Production comes in three stages:

1. Printing and Stuffing for Handout
2. Setting Up Game
3. Runtime Production

Planning and technology can make all three easier.

Production: Printing and Stuffing

Why do Handouts start late? Why are pieces missing?

- Can be as short as 1 man-hour.
- Of course, can be much, much longer than that: 5 people doing prod for days straight, plus some "helpful ghosts," is not unheard of.
- Schedule one full day for prod, at least one day before Handout, preferably the weekend before Handout.
- Relying on "helpful ghosts" to save man-hours is ill-advised (and sometimes presumptuous). Can get you out of a hole.

Production: Printing and Stuffing

Planning ahead is your friend.

- LaTeX / test all of your documents beforehand, early and often.
- Don't slack, even if you're writing at the last minute.
- Partial dry runs are good proofreading opportunities (*not* on the day of).
- Some early design decision can have a huge impact on prod: think ahead!
- Too many memory packets, notebooks, or other funky constructs can ruin your day.

Production: Setup

It is common (but not necessary) to have a "second prod" for game area: signs plus items and other things that don't start in a character packet.

- Even moderate amounts of packets-on-the-wall, extra notebooks, and signs can ruin your day, if not your week. (Looking ahead, things get grunged.)
- So planning ahead is a even bigger deal here. Give yourself limits ("no more than 50 things to tape up, ever." "Little to no state in signs and their envelopes.")
- If game area is serious, plan a full day of prod for it as well.

Production: Setup

Walking through gamespace and putting things up takes time.

- Given a well-organized set of signs, an optimal route, and 1-2 signs per building-floor, it takes about two hours to cover all of main campus.
- Complex setup, backtracking, and dis-organization can a make this much longer.
- Working in pairs helps (one person to tear/stick the tape).
- *Don't use duct tape.* It can peel the paint.

Production: Runtime

Some runtime production is planned.

- You may have late documents to distribute.
- There may be item stocks to replenish.
- You might have modules to set up.
- You'll be supplying physreps.

Production: Runtime

Some runtime production is a surprise (“known unknowns”).

- Signs will get grunged. This is somewhat random, but bigger and more complicated things are more likely to fall off the wall and get swept or get noticed by people who tear them down.
- People will lose their documents and name badges.
- You may need to print new versions of sheets or abilities.

Production: Lessons

- Don't panic.
- Triage may be necessary. Be prepared for it, but don't jump the gun.
- No solution can make production effortless. Sometimes a pen and a stack of index cards really is useful.
- Everything is better if at least someone is well-rested, well-fed, and has done their homework.
- The moment Handout ends, you have to deal with the players. This will eat far more time than you think, and slow down everything else you're trying to do.

End of Production

Questions? Comments? Discussion?

Running a Rules Seminar

- **Remember, your players are literate.** Just summarize and highlight.
- Anything that differs from the norm should be called out.
- Unless you're running a Rush game, concentrate on your game-specific rules.
- Show, don't tell: Spend the time you've freed up to actually demonstrate your mechanics.
- **Take notes whenever someone asks a question**, and e-mail clarifications to the whole game
- Be honest about what, if anything, didn't get prodded.
- If you have a complex mechanic, consider giving players a chance to test it out with example characters.

Handling Player Questions and Problems (1/3)

Your players want to enjoy your game. You want your players to enjoy your game.

This means you're on the same side!

- **It is always acceptable to say "Hold on a moment, I need to talk to my fellow GMs."**
- **Unless you have no other choice, never try changing a mechanic on the fly, especially not solo.** Predictable but broken is almost always better than something which is unpredictable; players can plan around it.

Handling Player Questions and Problems (2/3)

- Try to avoid just saying no to players. Unless they're a rules-lawyer, chances are, they've come to you because they have a problem.
  - First question to ask yourself: "Why is the player asking this?"
  - If it seems to be a "stupid question", you probably didn't write the rules/mechanic/sheet as clearly as you think. Be patient, and clarify to other players if it seems necessary.
  - If the player seems frustrated, see if you can find out why. They may only have come to you with one problem, but a frustrated player probably has hit several; learning what they are may be able to prevent future problems.
  - "Yes, but...", "No, but...", and "No, because..." are better than a flat "no" when handing frustrated players.
    - These mean the GMs are at least listening.
    - Be careful not to reward the squeaky wheels over the players who try not to bother the GMs; but a squeaky wheel often means there are others with the same problem.

Handling Player Questions and Problems (3/3)

- “Can I do X?” is a dangerous question.
  - Do you know what X implies in your game? If you’re at all unsure, ask the player: “What effect do you expect from X?”
  - Can X screw over any other characters or plots? If so, and X is not already in the mechanics, the answer is almost certainly no.
  - If X is unclear in the mechanics, make a ruling, and then **immediately notify all other parties with the mechanic**.
  - “Yes, but this has no mechanical effect” is a tremendously useful line to remember.

Running Wrapup

- Keep it short! Let the best stories get told at the post-game dinner mob.
- Tell people how mechanics resolved – Who did win the house cup, anyway?
- Tell people the things they didn't already know. Reveal the secrets. Don't give me a five minute talk about each character (in a 60-character game, that's a five-hour block of characters, and you haven't even hit groups yet!)
- Keep it moving; don't let a player get wrapped up telling a story that takes more than a couple minutes.
- A wrapup doc is optional, and based on your GM cheatsheet; everything except outcomes, awards, and quotes can be created pregame.
  - You won't have time to update and print in a short game; if you want a wrapup doc, just do it during prod.

GM Zen

- “A game is never finished; it merely runs, then dies” – kenclary
- There is no game so good that *everybody* is completely happy.
  - The longer the game, the more true this is.
  - Corollary: There are always punts. Don’t take them personally.
- There is no game[3] so bad that *nobody* has fun.

[3] If you’ve heard of Spin Cycle, that wasn’t a game. That was performance art.

If You Take Nothing Else Away...

The most critical GMing skill is asking questions.

- Even if you don't always get the right answers, you'll at least be aiming in the right direction.
- What do you want to accomplish?
- What are your players doing in game?
- How do your mechanics, rules, and space support your players and your game feel?
- What do you *really* need to make the game work, and what is extraneous?

Further Resources

- These slides: http://web.mit.edu/ariels/Public/sywtbagm-slides.pdf
- http://web.mit.edu/jakebeal/Public/gamewriting.pdf
- The Old Standard Rules: http://web.mit.edu/jemorris/Public/Rules/standard-rules.pdf
- Jim's must-read Zampolit talk: http://web.mit.edu/jlwfnord/www/zampolit.pdf
- Old games: /mit/assassin/Archive
- Mechanics discussions: xavid.scripts.mit.edu/mechanics
- assassin-writers mailing list