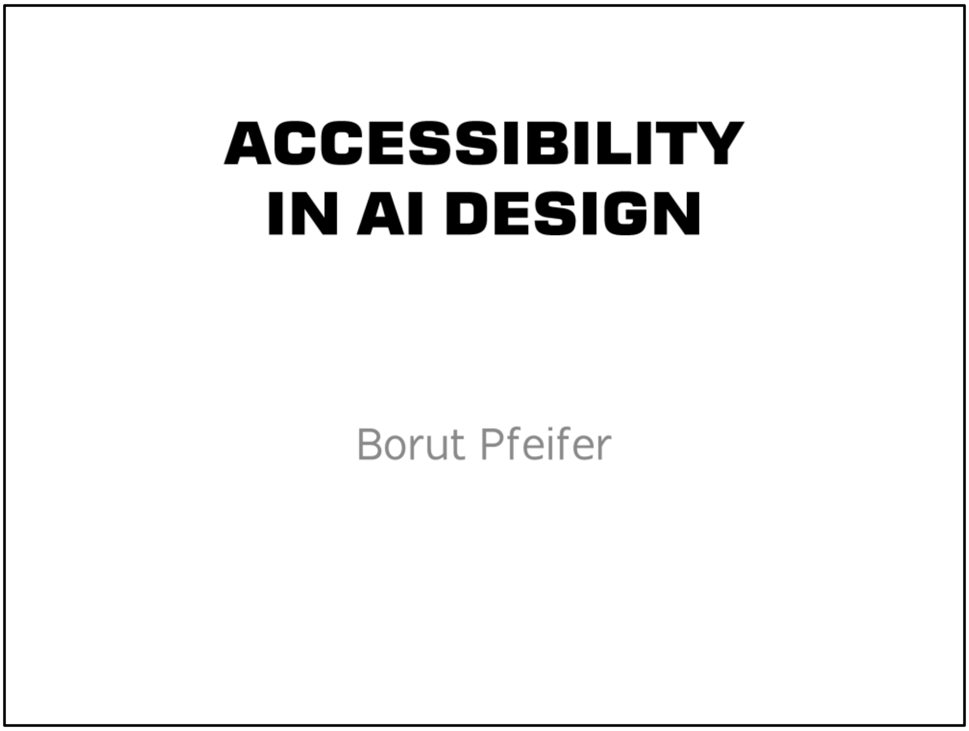
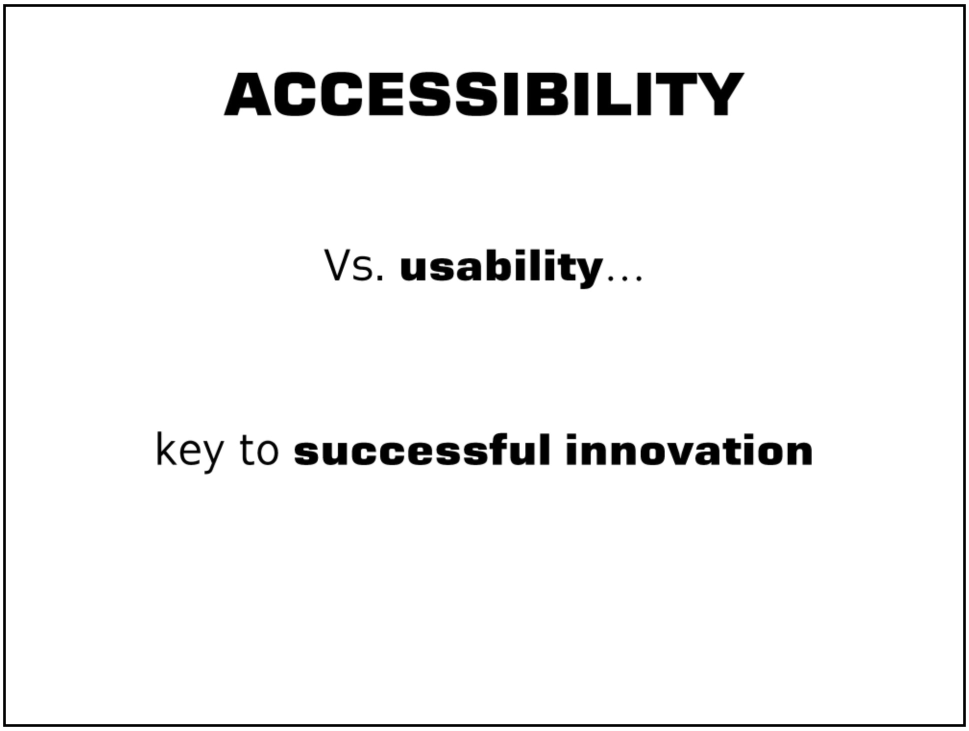


ACCESSIBILITY IN AI DESIGN

Borut Pfeifer



ACCESSIBILITY

Vs. **usability**…

key to **successful innovation**

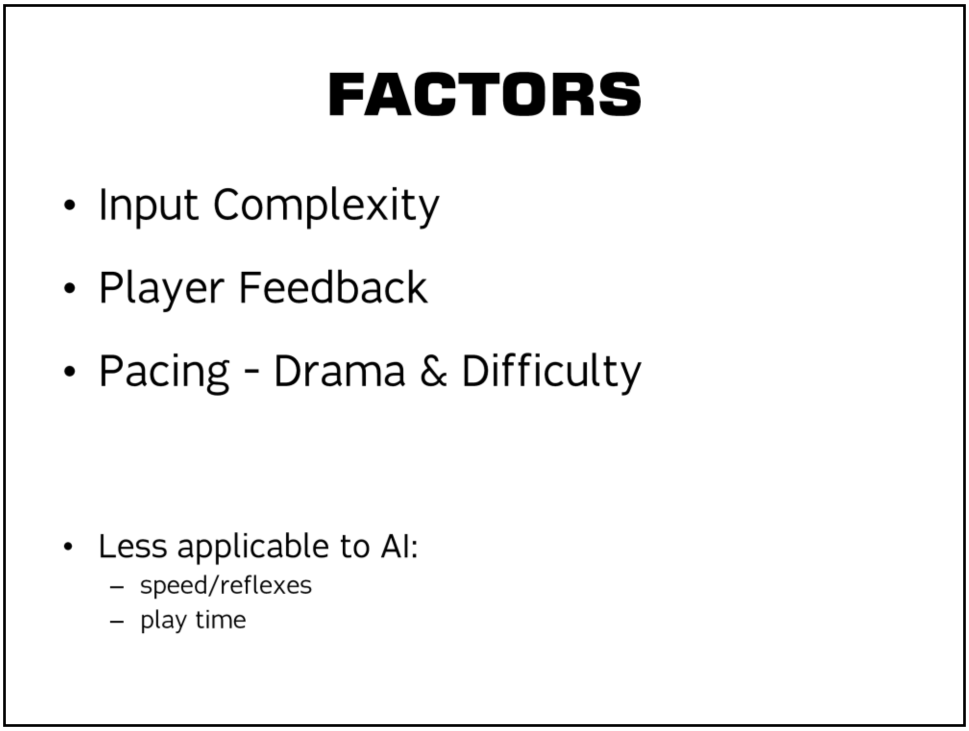
First, about accessibility. It’s separate from usability – that’s making an interface/product friendly to one group of people/audience. Accessibility is focused on allowing more diverse audiences be able to appreciate something. So there’s certainly overlap in the sense stuff you do to make the game easier to understand by one audience may make it easier to approach by another audience, but that’s not always true when looking at usability.

As we try to take new-fangled approaches to AI in games, we have to consider how to make these approaches communicate to players in a way that can reach as many players as possible. Even if it’s not a primary goal to reach more players, communicating straightforwardly is important to making these sorts of innovations actually work.

ME

There's certainly a lot of study and metrics around usability, and accessibility in as much as where they overlap. But trying to reach broader audiences is still much more art than science. You can ultimately never be sure of anything being absolutely accessible vs. inaccessible, because this is all about playing between the tensions of simplification in communication (via the game), and saying something complex enough that it is interesting to lots of people.

So this talk is a result of all my learning on this topic over the course of my career – nothing I say here should be taken as a rule in stone, these are just guidelines and rules of thumb. I encourage to think about these problems more yourself and develop your own rules of thumb through practice.



FACTORS

- Input Complexity
- Player Feedback
- Pacing - Drama & Difficulty

- Less applicable to AI:
  - speed/reflexes
  - play time

There many factors to consider in accessibility, but three have the biggest impact on AI.

The complexity of player input is at the lowest level, the number of buttons or controls the player has. When dealing with AI, it's a question of how many tools does the player have to affect the AI for their benefit?

Player feedback – this requires instantaneous, continuous feedback as to how the player's actions are progressing. The difficulty here is the more actions, goals, and subgoals (game directed or player chosen) a player can have, the more feedback you have to give. From there it turns into a big visual and AI design challenge as to how much feedback you give when. How we model our underlying simulations has a big impact on the type and quality of feedback we can give.

Pacing is critical to any accessibility (whether your making a mass market movie or what).

PLAYER INPUT

By player input to AI, I'm referring to the actions the player can take that affect AI meaningfully. They cause the AI to change the state of the world in ways that allow the player to reach their goals (whether those goals are given by the game or player-selected).

The Sims series does a good job, generally, of managing the range of complex ways the player can interact with the AI.

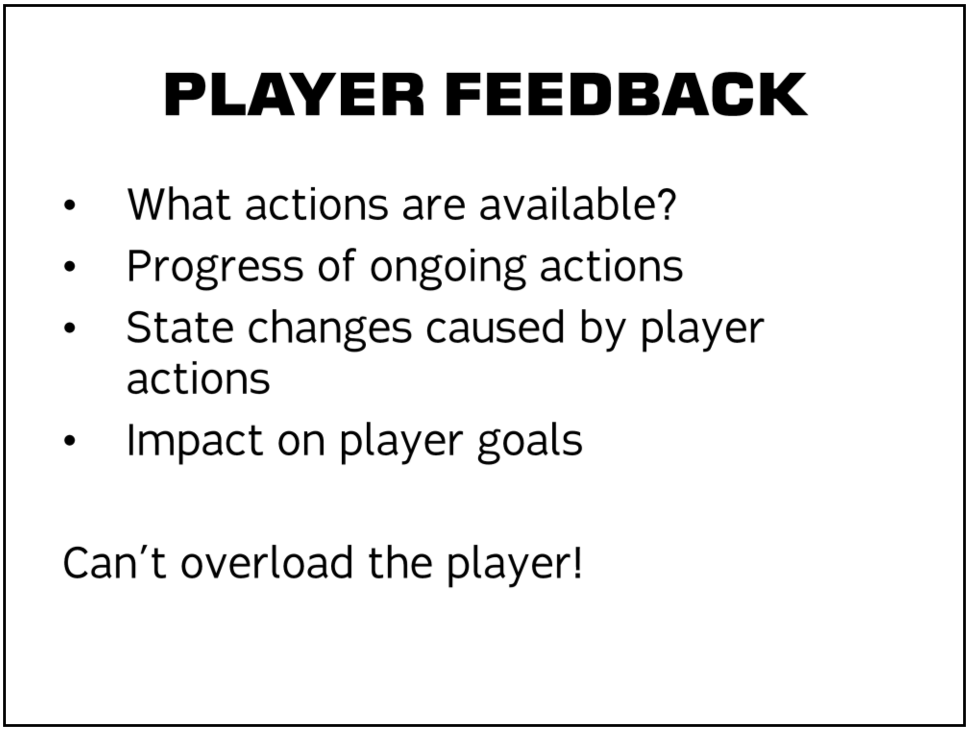
First order actions the player can perform are easily discoverable – menus about objects & characters that display simple verbs of what can be performed.

Second order effects, the events & behavior that occur because of these actions, are layered over time. The player discovers new actions over time, and can slowly modeling the increasing number of second order effects by observation.

Also, some actions change utility over longer periods of time (sleeping being more useful when a Sim is tired).

Furthermore, hard failure (having a Sim die) is very rare – many actions do not have any failure case, and the rest cause soft failure (minor set back or only something that can lead to hard failure if aggregated).

You can use the AI to guide the player to new input actions (seeing a behavior and figuring out how to replicate or trigger it).



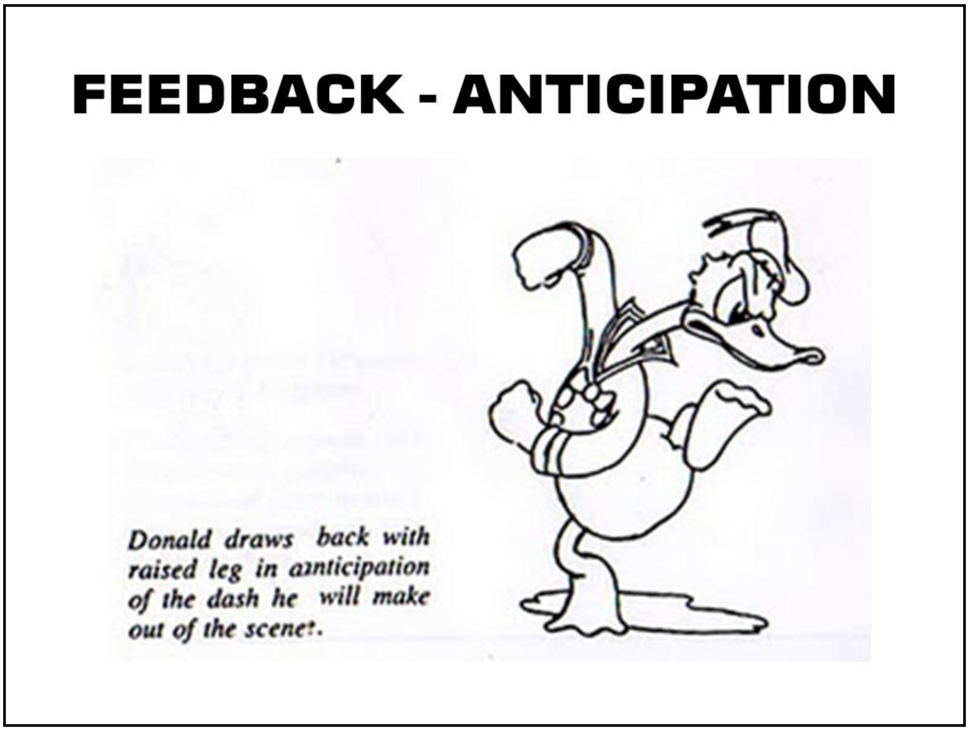
PLAYER FEEDBACK

- What actions are available?
- Progress of ongoing actions
- State changes caused by player actions
- Impact on player goals

Can’t overload the player!

There’s a lot we need to communicate for the player to be able to successfully use and interact with AI – however this is inherently difficult because there’s so much going on, we have to choose only a very small number of things to focus on communicating at any given time. Also, we can find ways to reinforce communication through background/lower priority channels, to reduce the signal load coming at the player and make it easier for them to parse.

So now lets look at some key elements of how that communication must be handled (anticipation), and how our underlying AI design can impact it.



FEEDBACK - ANTICIPATION

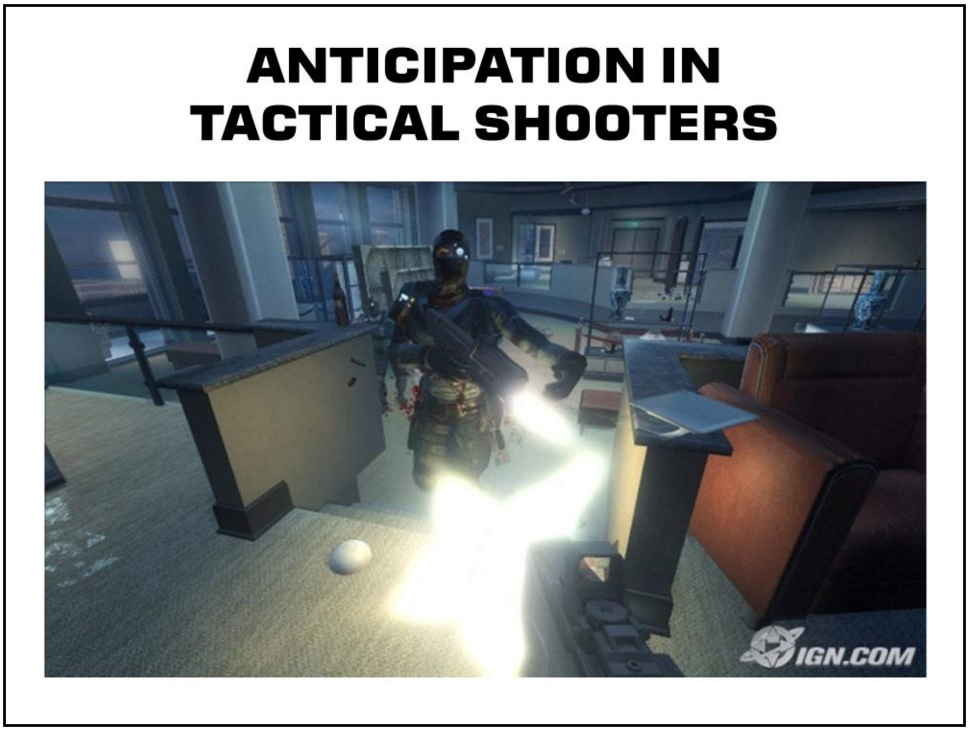
Donald draws back with raised leg in aanticipation of the dash he will make out of the scenet.

One of the 12 principles of Disney animation is anticipation – having a character convey what they're about to do so the viewer understands the action.

Anticipatory behavior must be a first level behavior in order for the player to understand how to work with AI.

http://www.epichuynh.com/2011/01/animation-notes-12-principles-of.html

Steve Gargolinski's talk at GDC 2011 on using anticipation as part of an AI framework for believable characters: http://gdcvault.com/play/1012484/Suspending-Disbelief-Bringing-Your-Characters



ANTICIPATION IN TACTICAL SHOOTERS

Screenshot from F.E.A.R 2: Project Origin.

Enemy behaviors in a tactical shooter you'd want to anticipate:

- Location/distance of enemies in adjacent tactical areas.
- Enemies flanking you.
- Reinforcements arriving.
- Enemies reloading or using suppressing fire to cover their teammates.
- Enemies knocking down desks or shelves to create cover for themselves.
- Enemies grabbing other weapons in the environment.

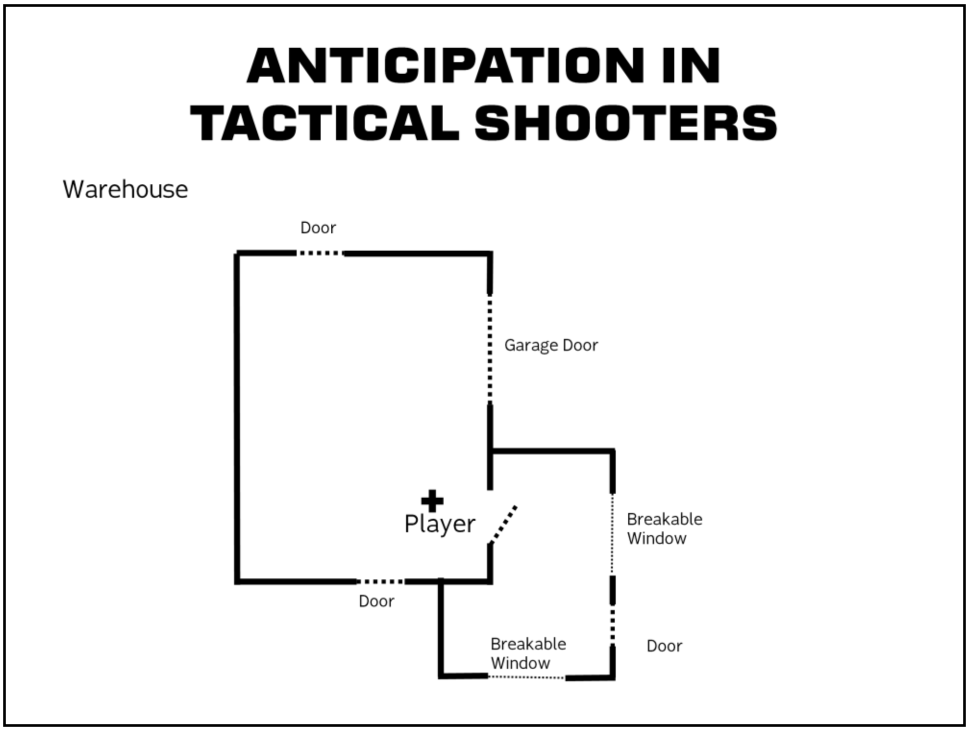
Enemies perform all those with animations explicitly made to make them stand out, with explicit dialog clips and sound effects that the player is meant to hear.

Office levels in this type of sub-genre typically have lots of visibility:

- Open working areas
- Split levels like this room
- Windows into offices/meeting rooms

This is specifically so you can actually SEE what the AI is planning, and interact with it.

The level layout & visibility isn't AI behavior, yet it's also crucially necessary to convey the anticipatory beats, along with actual behavior (dialog, animation, etc.). You still need to be considering those other elements as to how they impact your AI & its communication with the player.

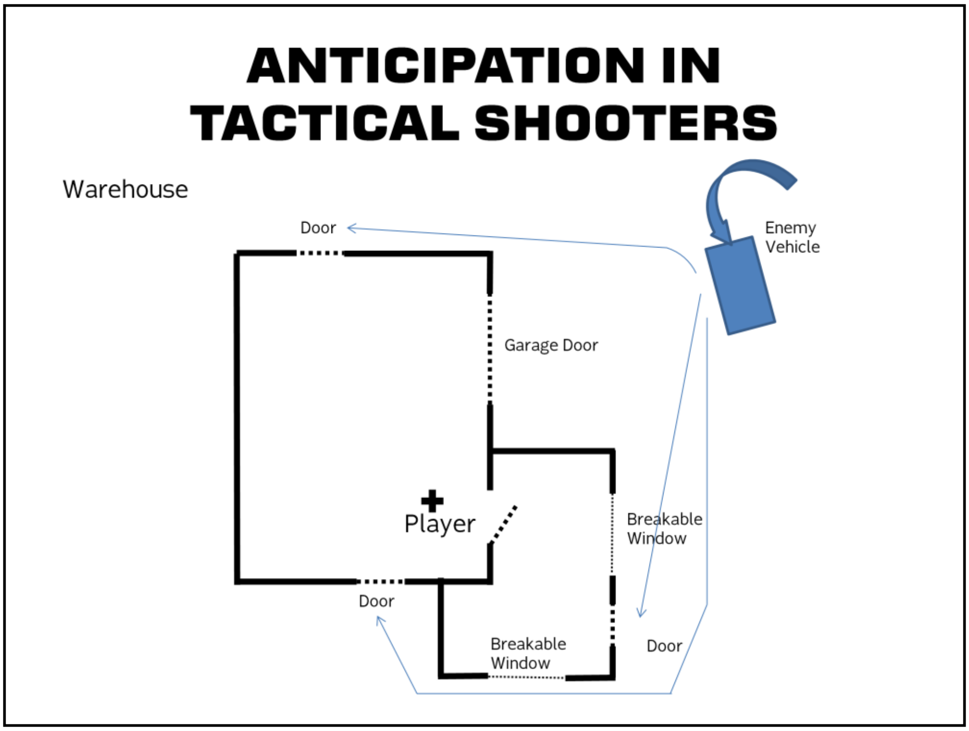


ANTICIPATION IN TACTICAL SHOOTERS

Warehouse

Here we have a simple warehouse layout for a tactical shooter. There’s some doors, a loading bay, and some windows the player can break (besides seeing through).

The doors may be able to be locked by the player from the inside, or blocked with furniture.



ANTICIPATION IN TACTICAL SHOOTERS

Enemies arrive on the scene.

The player can anticipate the tactical situation via several methods:

- Environmental sounds of the car arriving, screeching to a halt, possibly sirens, car doors opening/slamming closed.
- The enemies get out of the car, hold position momentarily, talk between themselves as to where player is and possible approaches inside.
- Enemies run to possible entry points.
- One or two might stop at windows to peer in to see you.
- Enemies will notify others about state of the doors (openable, locked, blocked)
- Enemies follow tactical procedures for breaching doors.
- As entry options are disallowed, enemies will group at the available options (if all the doors are blocked, they may prepare to break windows).
- Player may have access to technological or supernatural heads-up-display style info as to enemy position & state.

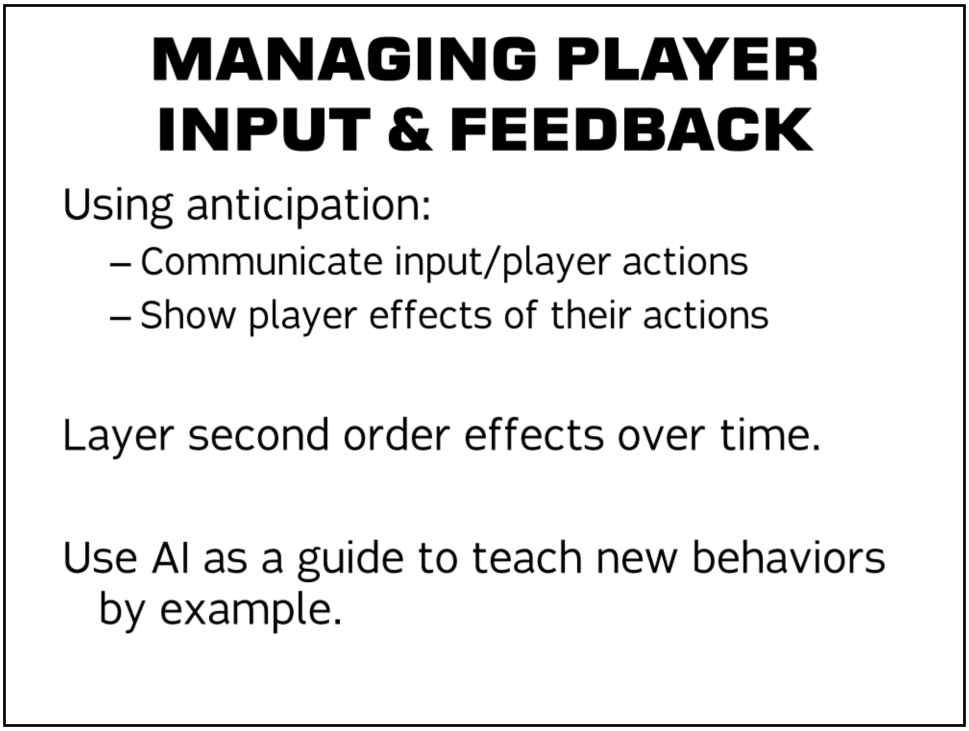
The player uses that info to form a plan among many options:

- Lock or bar doors preventing NPCs
- Get to a not-easily discoverable location in the warehouse.
- Surprise enemies by attacking though windows – possibly jump out of the building at the same time to prepare to flee.
- Open garage loading bay door, surprising enemies who have left their vehicle unattended, use it to flee.

All those tactical options provide interesting gameplay to the player, and most of them result from the AI (and also the level design). If the player can’t anticipate the AI’s behavior, they can’t learn, adapt and react properly.

The AI can still surprise them – but the anticipation has to be meaningfully built so the player expects one thing, and then is surprised with another.

Take away the AI and anticipatory beats and the player is basically staring at a warehouse wall until enemies show up and attack.



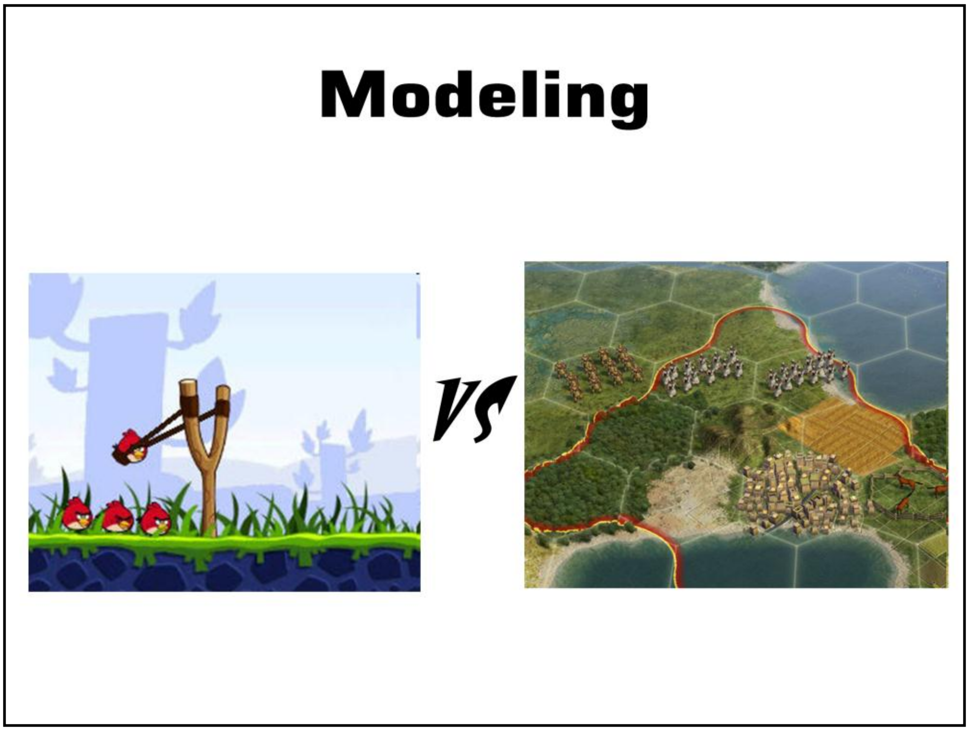
MANAGING PLAYER INPUT & FEEDBACK

Using anticipation:

- Communicate input/player actions
- Show player effects of their actions

Layer second order effects over time.

Use AI as a guide to teach new behaviors by example.



Modeling

VS

How we model our simulation is crucial in defining what opportunities we have to communicate to the player. It doesn’t just define possible player inputs & their effects, it defines the methods we have to communicate about it.

A model that strips out useful information that the player needs through state changes can prevent us from ever being able to communicate information the player needs to play the game.

Discrete vs. Analog Simulation/Modeling: Common design wisdom here suggests:

- Players deal better with discrete states
- Eg. Stealth genres reduce enemy alertness to 3 levels

The reality is more complex – players often balk at the discretely modeled hex-based war games, but flock (har har) to physical simulations as casual gameplay (reaching hundreds of millions of players).

Even dogs can make sense of physics after all.

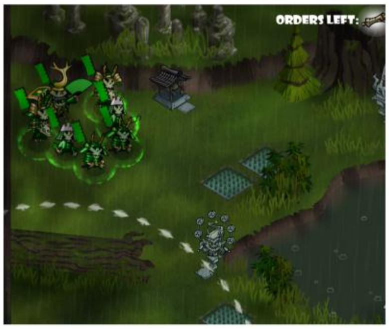
There are some useful parallels to draw from in deciding how complex we can be in our simulations.

People can deal with estimating factors affected by 2nd order derivatives (acceleration in this case), IF they have constant feedback to the current state.

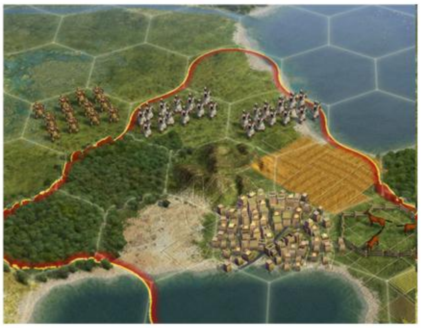
All trademarks and other intellectual property are owned by Société des Produits Nestlé S.A., Vevey, Switzerland
Photo by See Spot Run Photography

If you’re going to have one dog jumping for Frisbee slide, you might as well have two. That’s some pretty complex physics going on there.

MODELING



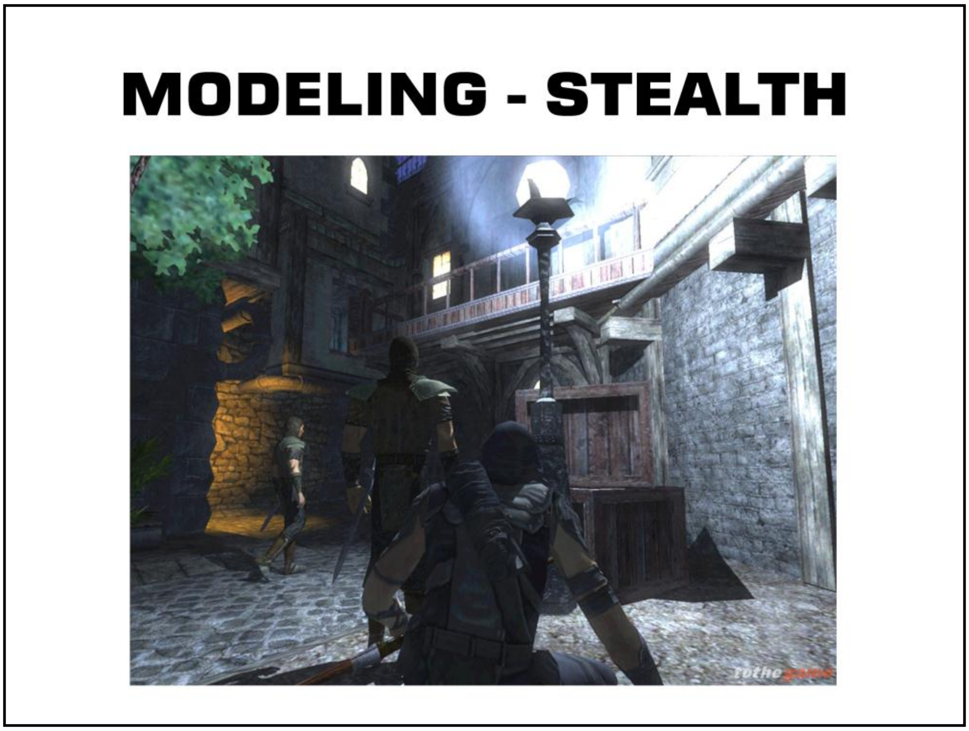
VS



In order to make strategy in Skulls of the Shogun more accessible, we removed the grid. Players react immediately to the wider space of options.

Position & viable path information is clearly visible (movement is radial based), so players have all the information they need.

They’re less concerned with the lowest level of detail in the modeling (the physical space and choosing the right spot/hex) and more concerned with higher-level tactics like flanking, feinting, and defending.



MODELING - STEALTH

Screenshot from Thief: Deadly Shadows

A lot of factors affect enemie’s confidence in knowing your location:

•Lighting/Shadow (can be very ambiguous visually)

•Sound/footsteps (can be based on environment/ground type

•Suspicious player behavior

•Dead bodies/environmental anomalies

This creates ambiguity:

•Highly realistic environments blur the line between safe/dark and unsafe/light locations.

•Realistic characters imply higher detail sensory modeling.

•Extreme failure conditions (being sighted can immediately cause you to fail).

The actual value of an enemies analog confidence & it’s impacts are hard to gauge with no exact feedback, so enemies are given a few basic high level states of alertness:

•Patrolling/Unaware

•Suspicious

•Investigating

•Alarmed/Alert/Chasing

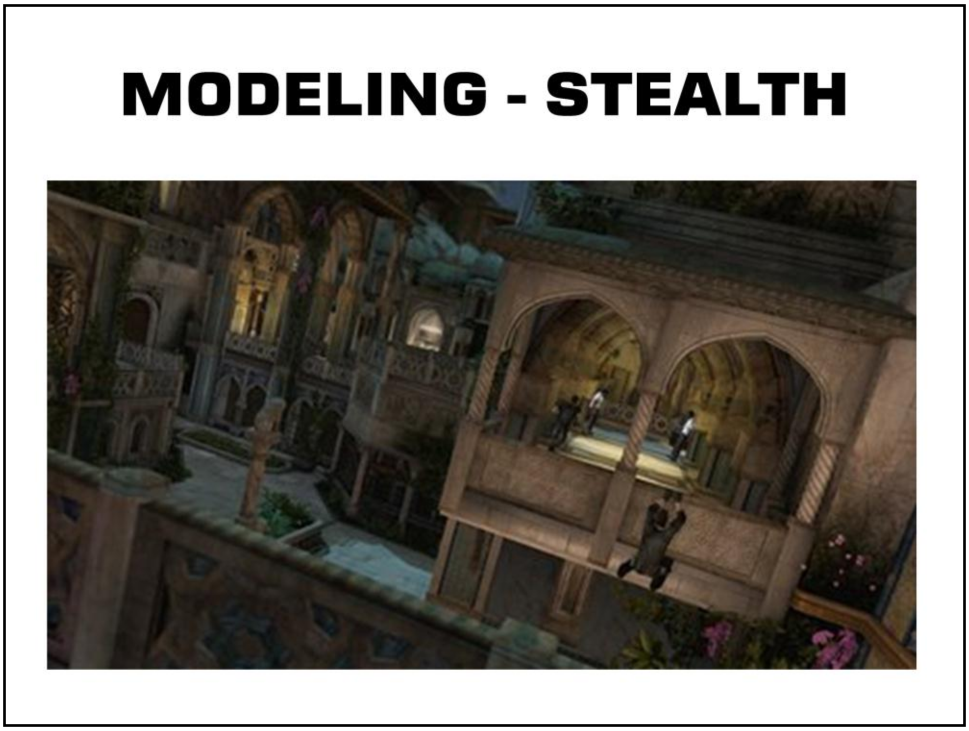
Instead some games trigger performances around the events that cause state changes, even if the internal model is much more analog.

Games like Batman: Arkham City, Manhunt make use of NPC dialog and animation to convey specific events that cause state changes:

* finding a body

* sighting character/losing track

* group state (last enemy in area, notifying reinforcements)



MODELING - STEALTH

Uncharted 2’s museum stealth level (early in the game).

This suffers from several problems dealing with these ambiguities. The environment is highly realistic, but NPC vision is very simpler (much simpler then Thief).

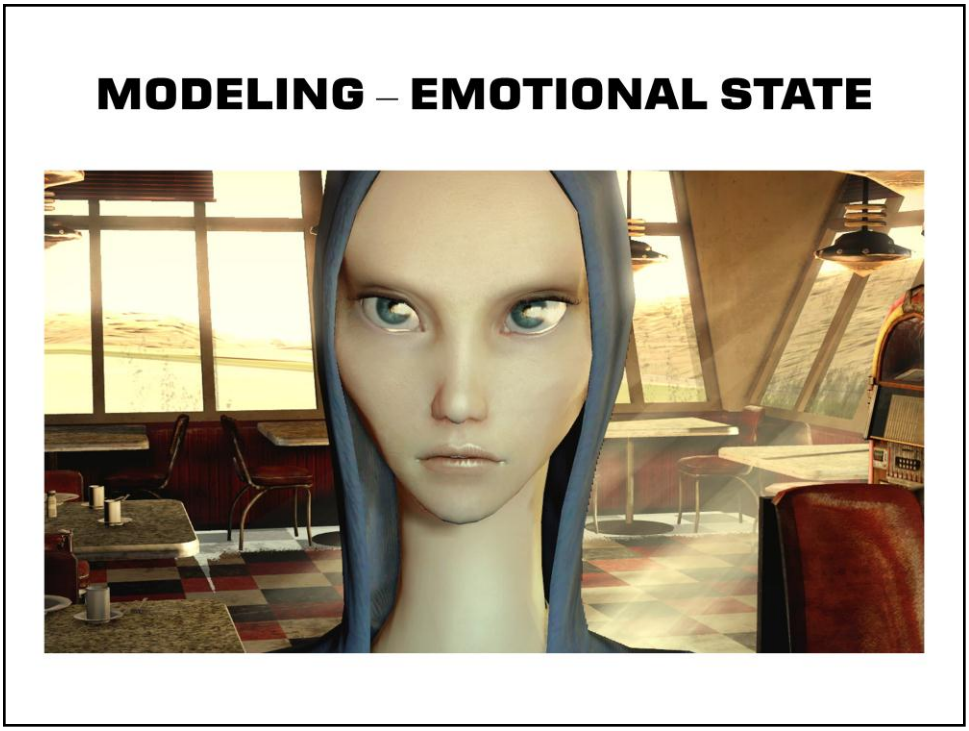
There’s large field of view in front of characters, no field of view behind characters, limited light/darkness visibility modeling. NPC sighting also causes hard failure, little to no soft failure communicated through NPC acting (like starting to look around in front of them if the player is actually in a further away blind spot).

UC2: diagram of sensory radii around character, they can perform if player is approaching sensed area.

MODELING - STEALTH

- Too discrete:
  - Players become unclear what causes state changes
  - No soft failure opportunities

- Too analog:
  - Players can’t make sense of the stream of information coming at them
  - Abstract analog data can conflict w/fiction



MODELING – EMOTIONAL STATE

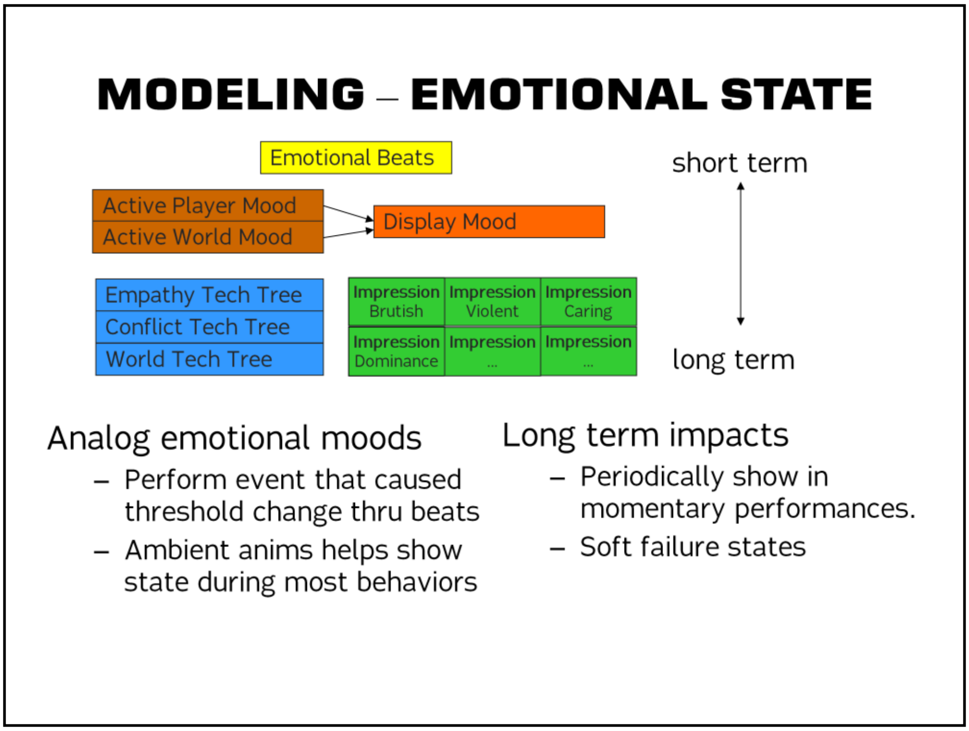
So between 2007 & 2009 I was working this sort of action/adventure-RPG FPS, and you had a partner character whose relationship with was supposed to change over time based on your actions in the game.

(There's a write up on 1Up about it on which I make officially no comment: http://www.1up.com/features/story-steven-spielberg-lmno as well as a video of the game's visual target: http://gamevideos.1up.com/video/id/32068)

We were trying to model her emotional state in a way that was more complex & ambitious than modern FPS games, but at the same time the player would be able to understand how she felt towards him (the player's character was ex-special ops essentially).

There's lots of elements that make up a person's emotional state, reactions to individual events, longer term moods, memories of specific repeated behaviors.

Now I don't mean to suggest we found a way to make all that understandable, but its something we were absolutely thinking about so it may be interesting to dive in to some of our thoughts.



MODELING – EMOTIONAL STATE

Emotional Beats

Active Player Mood
Active World Mood

Display Mood

short term

long term

Empathy Tech Tree
Conflict Tech Tree
World Tech Tree

Impression Brutish	Impression Violent	Impression Caring
Impression Dominance	Impression …	Impression …

Analog emotional moods

- Perform event that caused threshold change thru beats
- Ambient anims helps show state during most behaviors

Long term impacts

- Periodically show in momentary performances.
- Soft failure states

Emotional beats would be what she’s performing – these could include:

- Full body animation
- Short term AI behavior (1-2 min tops)
- Facial performance (audio/dialog)
- Typically reactionary

Moods represented her attitude towards either the player or the environment (which was kinda generalized for simplicities sake to be “everything that’s not the player”:

- Meant to be longer in length (more like 5 minutes)
- Defined allowable behavior
- Communicated by affecting any emotional beat, affect other animation elements like her posture, idles, default facial expression.
- Only communicate one at a time based on what around her is most pressing – streamlined to reduce the number of things she would have to communicate at once through animation.

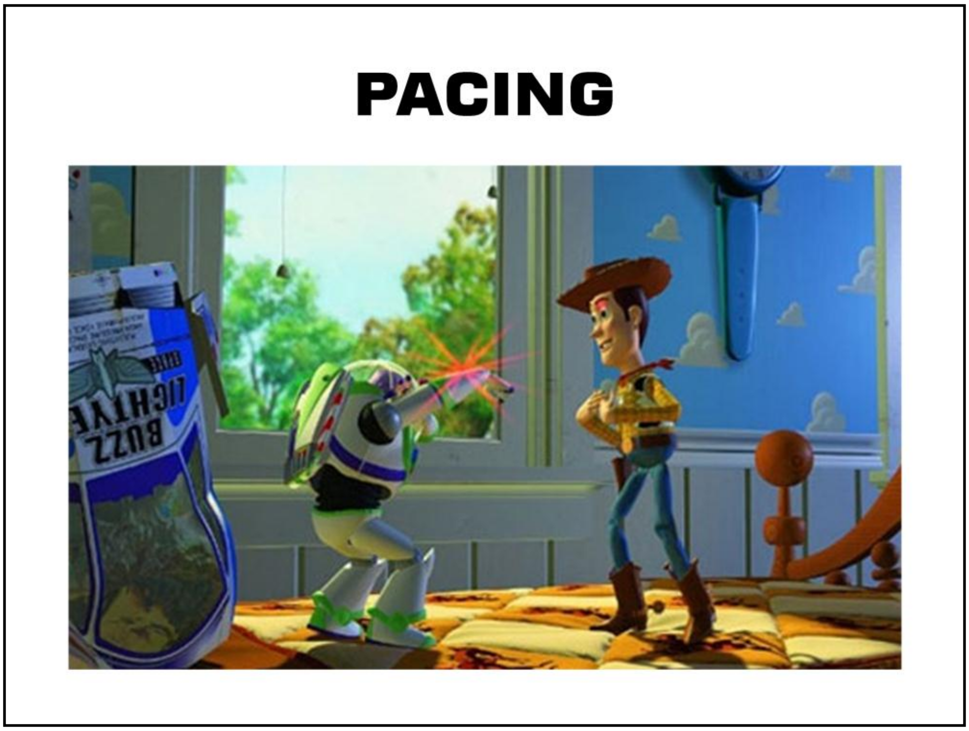
Long term changes:

- Tech tree unlocked new abilities or behaviors – many had game effects, some could be simply flavoring her behavior/personality
- Impressions could impact individual behaviors in a more narrow way – if you never saved her in combat she might flee when combat starts.
- Soft failure – impressions would require several instances of you performing the same behavior for her to form one, could slowly undo them.
- There also weren’t negative per se – if she though you were uncaring towards her that would make her more self-reliant in combat, which might be desiredable by several play-styles.

MODELING

Make sure model allows:

- Performances around events that cause threshold/state change.
- Anticipatory behavior preceding that.
- Soft failure states.
- Regular, consistent feedback that state is changing.



PACING

Pacing is crucial to any sort of entertainment trying to reach a lot of people.

In film, you have a movie like Toy Story, that weaves together lots of elements that will appeal to different people in different ways:

- Jokes geared towards kids like slapstick comedy
- Subtle jokes about pop culture or nostalgia that only parents or older audience members will pick up (like Star Trek & Star Wars references)
- Physical danger/action
- Emotional growth of the characters.

And all of this is balanced together over time.

With AI in AI games, w need to:

- Layer complexity presented to player.

- Provide time for player to learn behavior.

- Avoid creating barriers b/c of difficulty.

- Drama keeps player engaged through struggle/challenge.

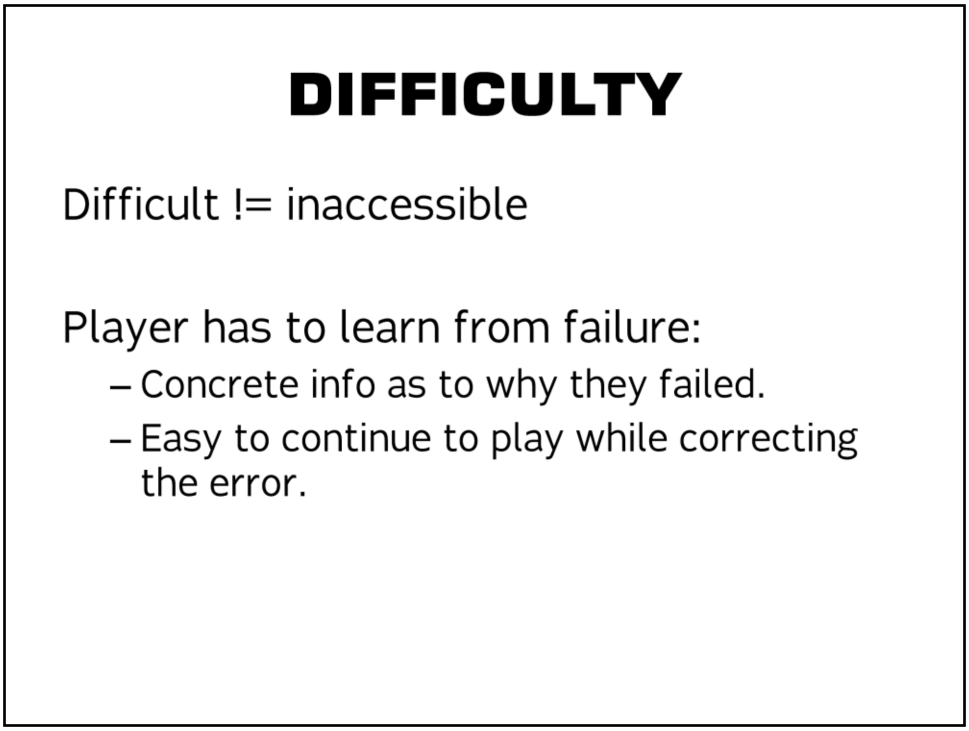
- Weave disparate elements for different groups of people to enjoy.

At the most basic level it provides room to make sure all of your audiences are learning what you need them too.

But most importantly, you must use it to manage many spinning plates in what your presenting to the player – different audiences will pick up on different aspects, and seek out those they think they will enjoy the most. That to me, is the hallmark of a work that’s truly accessible.

Pacing in games at a low level is effected by the communication of available player input & feedback on their actions, as we’ve been discussing.

At a high level though, pacing in games is built through a combination of difficulty and drama.



DIFFICULTY

Difficult != inaccessible

Player has to learn from failure:

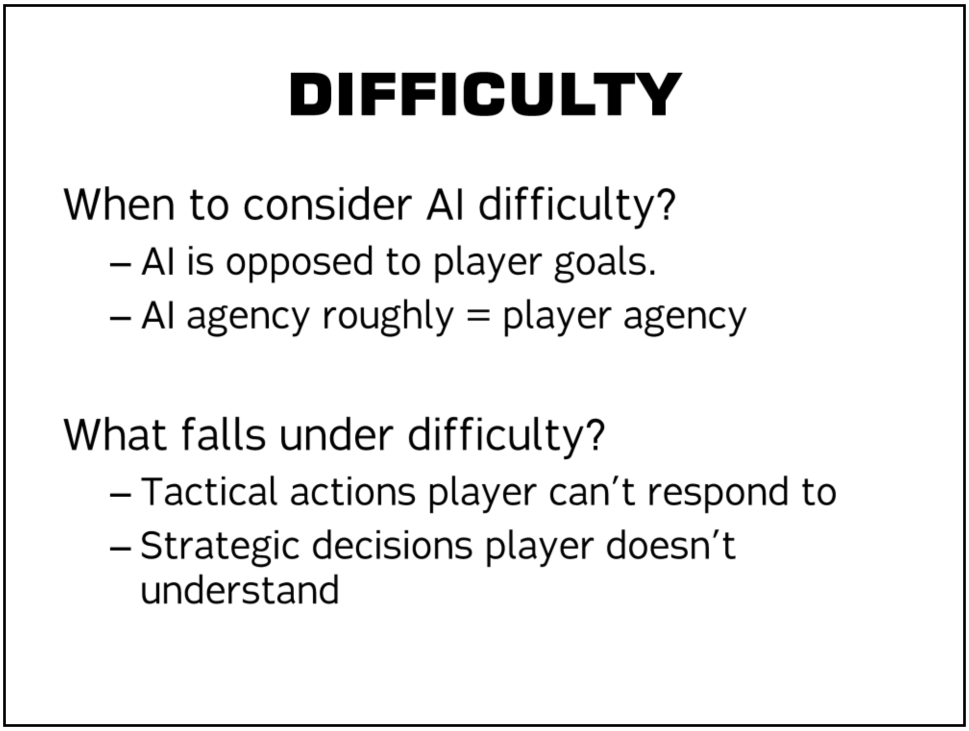
- Concrete info as to why they failed.
- Easy to continue to play while correcting the error.

Difficulty in games can be a very tricky thing in terms of accessibility – it can absolutely shut out people. At the same time though, having no challenge in achieving their goals can make the player feel unfocused or bored.

Often times AI can create challenge for the player when it’s not properly communicating it’s actions, but that’s a separate design problem as discussed earlier.

If the player is being given the information they need to make a decision, the reason they fail to respond is because:

- They physically can’t react to deal with it (tactical decisions) – this might be ok but the scope of the reaction time allowed has to be within reasonable margins for your audience. Not really AI but your AI will fail if you don’t consider it.
- They have not learned/processed higher level strategies or strategies that result from complex dynamics.
- They fall prey to human error - they forget tactical/strategic elements they have been made aware of and have learned. This is an acceptable



DIFFICULTY

When to consider AI difficulty?

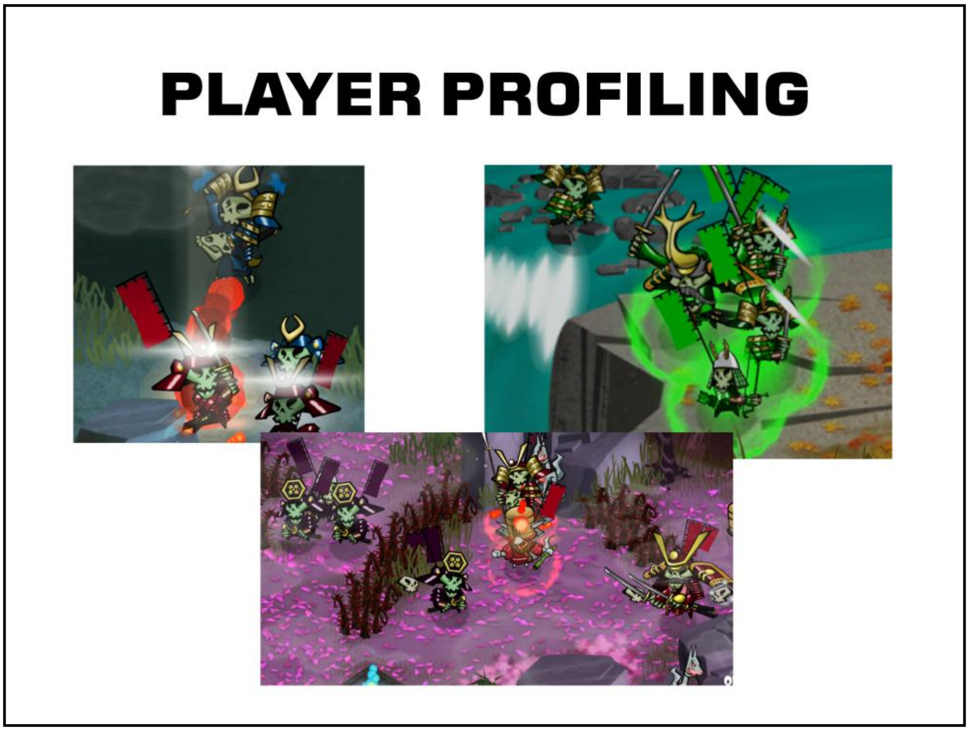
- AI is opposed to player goals.
- AI agency roughly = player agency

What falls under difficulty?

- Tactical actions player can’t respond to
- Strategic decisions player doesn’t understand

Let’s consider AI difficulty as when an AI is making decisions as an agent at a similar level to the player.

When AI is making decisions at a level higher than or separate from the player, that’s typically drama management in some form.



PLAYER PROFILING

In SotS, I profiled strategies by player experience level (novice/experienced). I didn’t do anything fancy, just watched a lot of players.

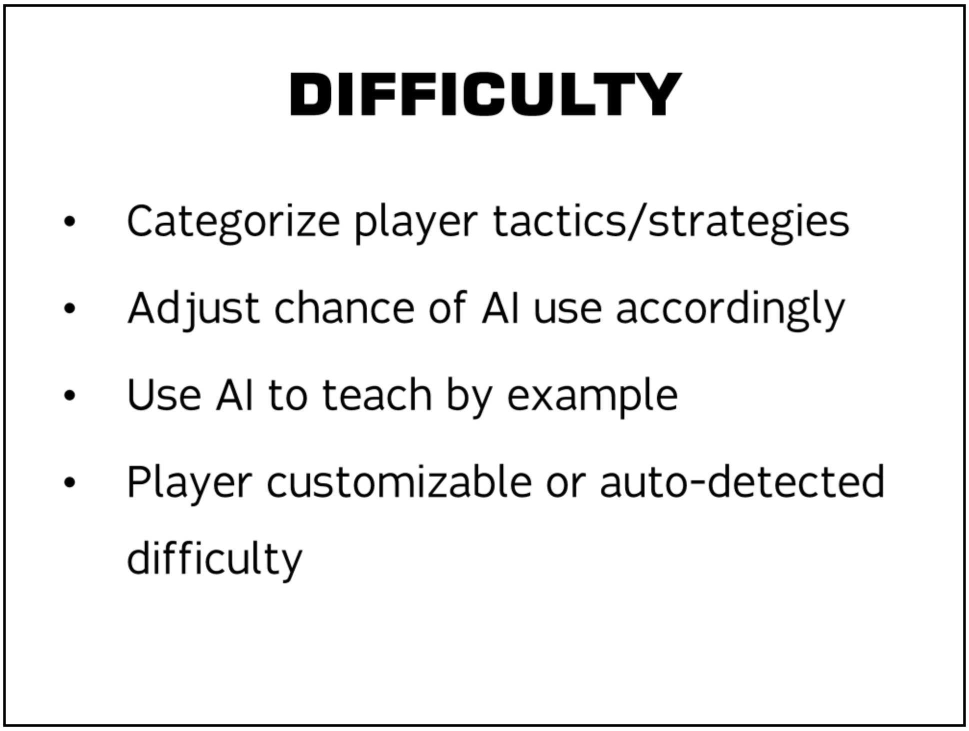
Strategies that novice players would forget/not learn very quickly:

- Maximizing unit movement - moving after using a unit’s action to flee combat, advance closer to a resource for the next turn.
- Forming protective barriers by grouping units together around weaker units.
- Knocking units off ledges using melee attacks (unit dies instantly).
- Leaving general undefended at inopportune times
- Advanced spells (for the characters that can unlock them).

So lower AI difficulty:

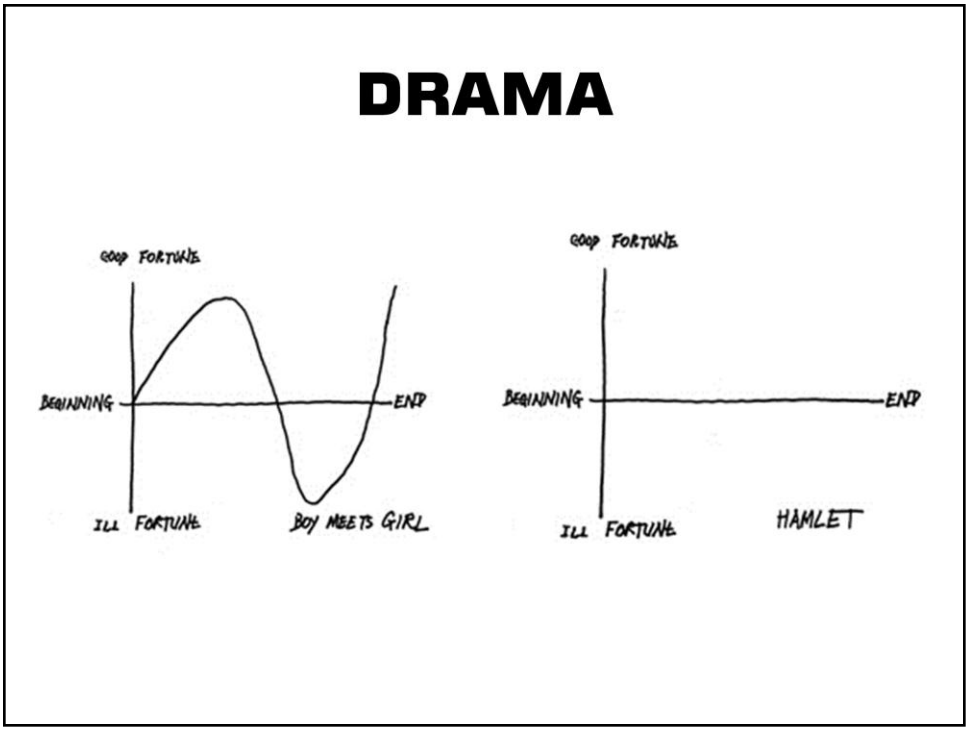
- Doesn’t use movement after acting.
- Doesn’t specifically try to maneuver into good positions to knock units off ledges (it still happens naturally though).
- Different priorities for defending their own general from attackers.
- Lower priority for targeting the enemy general when he is open.

- Less likely to use advanced spells (still will though).



DIFFICULTY

- Categorize player tactics/strategies
- Adjust chance of AI use accordingly
- Use AI to teach by example
- Player customizable or auto-detected difficulty



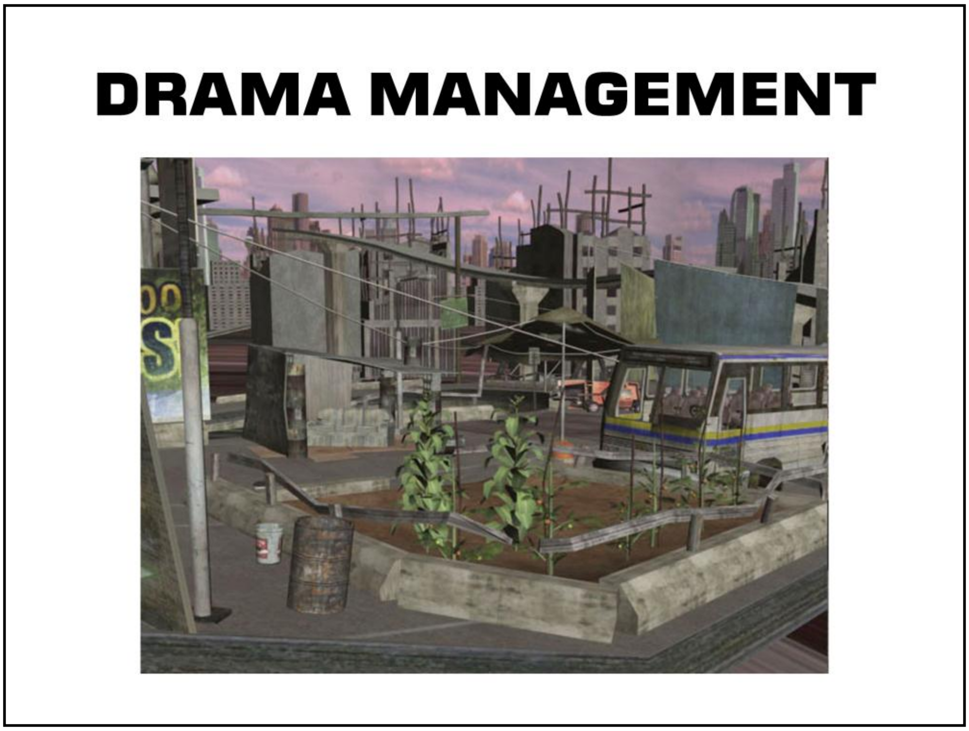
Kurt Vonnegut illustrates this point on drama very well, in an essay:

http://www.laphamsquarterly.org/voices-in-time/kurt-vonnegut-at-the-blackboard.php?page=all

Your prototypical story, boy meets girl, mapped out as the main character’s lot in life changes from ill fortune to good fortune (meets girl, loses girl, gets girl in the end).

Now you look at something like Hamlet and through all the events that go on in the play, it is difficult to know for sure whether something's’ good or bad. He see his fathers ghost & learns of his death – sound bad, but can we be sure that is in fact his father? Meanwhile he’s putting on this play to confirm who’s the killer, his fortune isn’t changing but we find out new information that Hamlet is trying to also figure out what it means. And then his death at the end, it’s ambiguous again (does he go to heaven? Or hell? Or is there no such thing? Is he better off simply for not having to live in that world with those people, or is his death sad?).

So to make a long point very short, drama is basically just about information – the *complication* of events over time that keep the audience engaged and wondering what will happen. You have to give them enough information to get them interested in the characters, but you have to keep asking questions as to their fate to keep the audience on the hook.

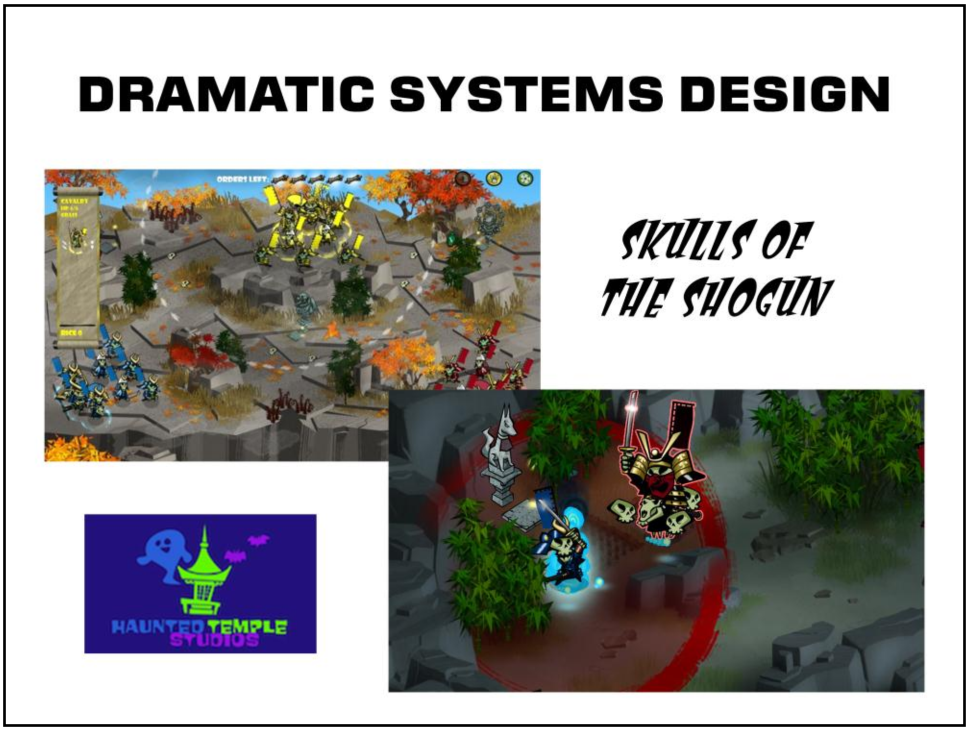


DRAMA MANAGEMENT

Applying these to principles is pretty interesting to me.

Back in 2001 to early 2003, I was also doing the independent games thing, for the first time, and we were working on a 3rd person shooter action RPG (in a cyberpunk world). One of the things I was experimenting with was an adaptive difficulty system that took into account the desired dramatic tension setting. So at given point, the enemies would spawn, say for a high tension area, as difficult enemies – but difficult according to the level of skill the player had shown. In that way it was a very simple progenitor to something like the system in Left 4 Dead.

Ultimately, after the project failed, I was dissatisfied with the overall system. I still believed drama management could improve the overall experience of a game for an average user, and definitely for in experienced users. But it felt like it was trying to apply a band aid to a problem that was deeper down. Why couldn’t the underlyijng systems create drama through their dynamics? Was drama management even necessary?



DRAMATIC SYSTEMS DESIGN

SKULLS OF THE SHOGUN

On Skulls of the Shogun, one of the goals was to make the underlying strategy very dramatic. In part to allow for new multiplayer gameplay, but also to fix issues we saw with the squad tactics genre, and strategy games in general (too slow, too much reliance on Rock, Paper Scissors mechanics, not enough player expressivity).

We do this a number of ways:

Rice paddies, which create rice every turn if you capture them, run out after 6 turns.

•Avoids stalemates where players have equal, perpetual resources

•Provides interesting choices in combat about prioritizing long-term gain versus short term safety (you won't counter attack)

•Rice paddies near players' starting positions get haunted early in game, providing players with the most rice (and therefore units) in mid game, building larger armies. As a match comes to a close, resources are used up and armies are whittled down.

Units power up in the game by eating the skulls of their fallen enemies:

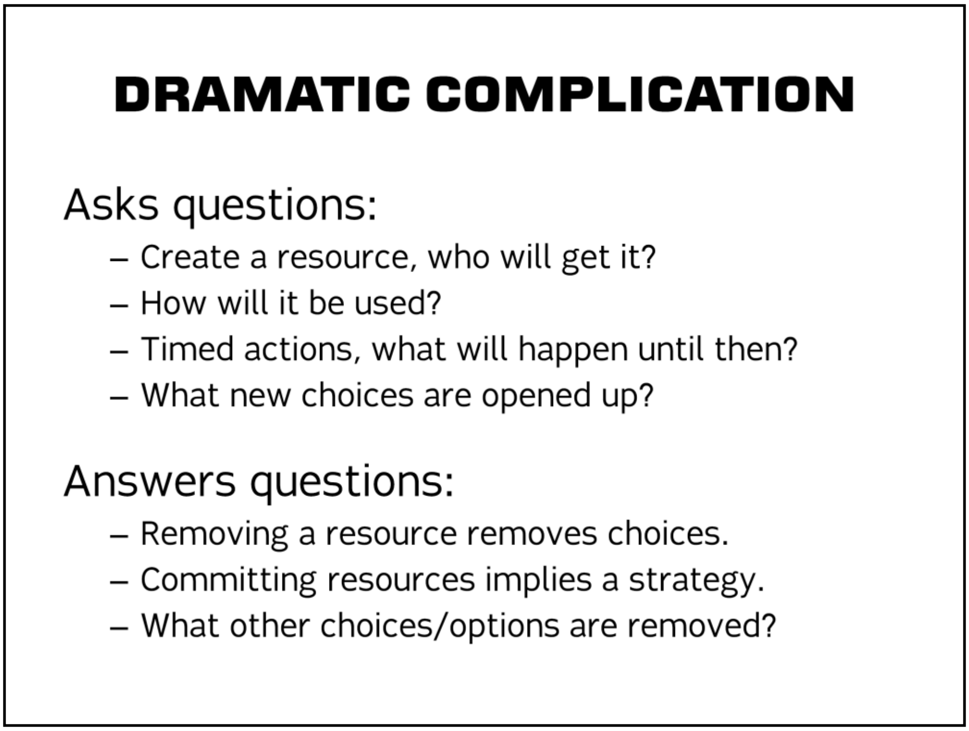
•Defeating an enemy removes choices from the opposing player, but creates a resource who's control is unknown at that point.

- Three skulls turns an enemy into a demon, who can act twice per turn. They also have double their max health.
- Players only have 5 units they can move per turn, making the second action a powerful advantage,
- An average unit dies in 2-3 attacks. A demon at max health can therefore still be killed in one round.
- When a demon dies all the eaten skulls are dropped and can be used by anyone.
- You can swing the tide of battle by focusing on one demon in your turn, and then grabbing/protecting the resulting skulls
- The number of units slowly diminishes, but skulls are never removed. Battles tend to end dramatically with a few very powered up units.

Increasing stakes is an easy way to create drama. By giving units very impactful advantages, but making them just as easily lost, creates a dynamic back and forth. Players rarely get into a long-term losing position (where they are force to wait until they lose).

Many factors play into the dramatic effect a set of mechanics has:

- Visual design of the demons makes them look very impressive – heightens the drama of their perceived advantage. Down side – can make players feel like they can't win, when all they need to do is change strategies.
- Perception & psychology of numbers – an army of 10 units vs. an army of 5 is much closer to the same strength given the limit of 5 moves per turn, but that can make for dramatic comebacks.
- Random numbers can make some elements more unpredictable (if you use this to your advantage you can make unpredictable=dramatic).



DRAMATIC COMPLICATION

Asks questions:

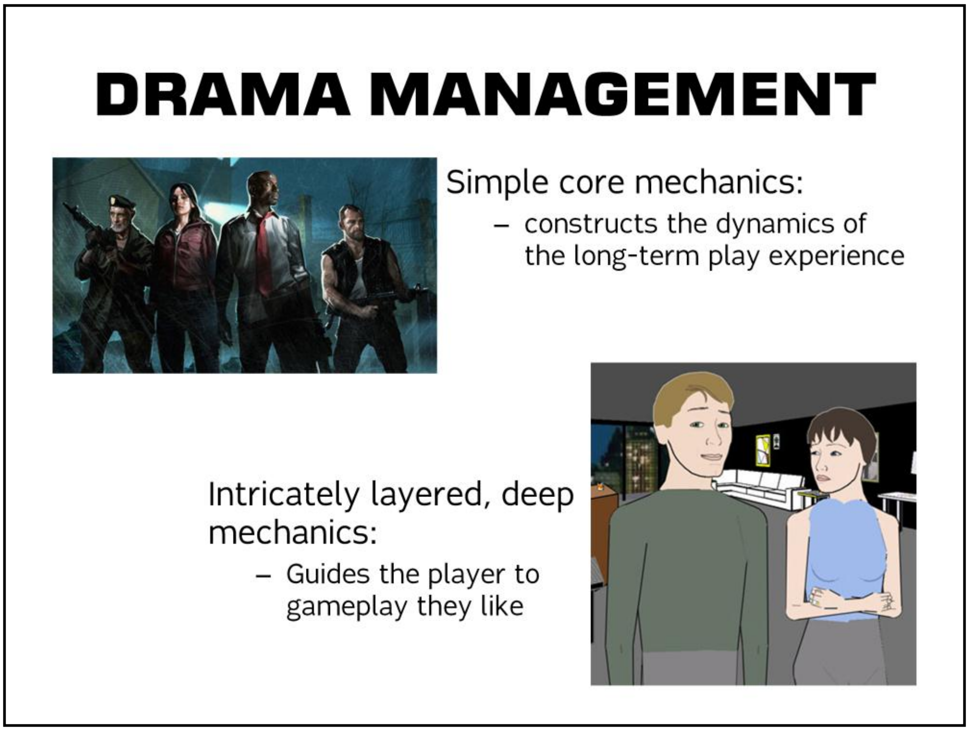
- Create a resource, who will get it?
- How will it be used?
- Timed actions, what will happen until then?
- What new choices are opened up?

Answers questions:

- Removing a resource removes choices.
- Committing resources implies a strategy.
- What other choices/options are removed?

A resource here can be anything the player has to use and is limited. It could even be simply what action they choose to do – performing one action at a given time automatically implies they're not performing many others.

So for our lowest level mechanics, and the dynamics they create, we have to ask ourselves how they complicate the play scenario. And then, how are we adjusting this complication over time, to resolve questions and open up new ones?



DRAMA MANAGEMENT

Simple core mechanics:

- constructs the dynamics of the long-term play experience

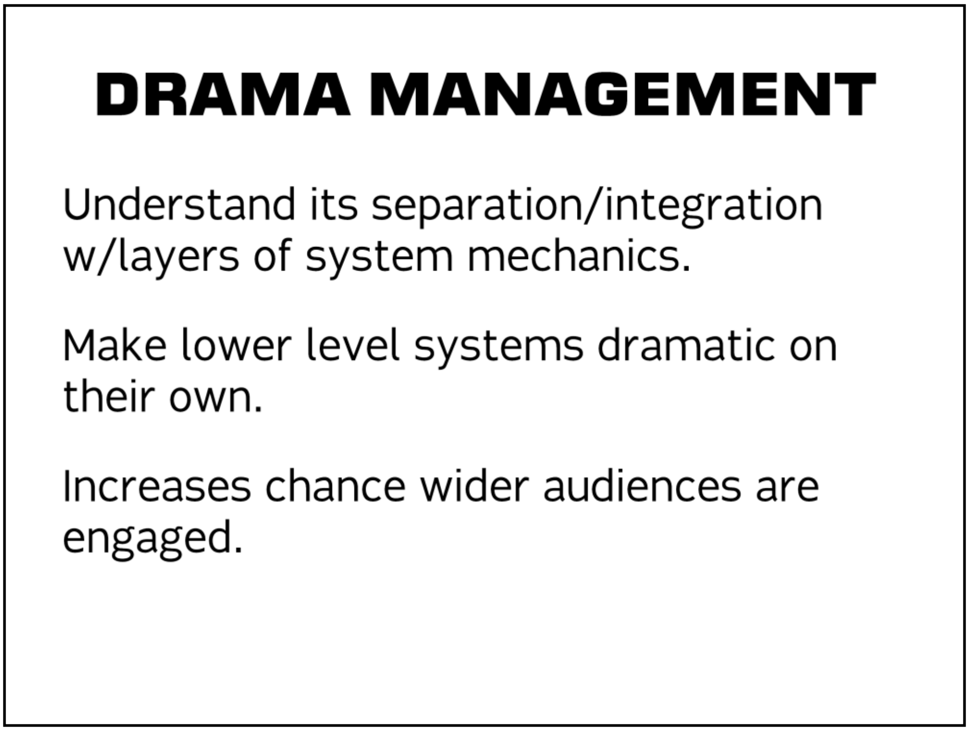
Intricately layered, deep mechanics:

- Guides the player to gameplay they like

So if we’re focused on creating drama in our lowest level mechanics, what role should drama management play?

There’s certainly a spectrum, on the one hand you have Left 4 Dead, whose core shooting mechanics are simple in the sense they’re well understood. There’s a handful of NPC types, each with their own rules but not that many. Those custom NPC rules are designed to be dramatic, like pulling friends away from the fight to be rescued. Drama management on top of that is like a mixer, remixing situations to keep them fresh and interesting, and make sure the simplest low-level rules, are combined together in a good way. The NPC rules would be less effective drama-wise if you kept experiencing the same one over and over.

But if we have an extremely complex set of mechanics (like the character interactions in Façade), that same style of drama management would interferes with other goals of the systems (like making characters real & believable), and so it’s more about directing the player to gameplay they like, finding opportunities for drama on the players existing path, that kind of thing.



DRAMA MANAGEMENT

Understand its separation/integration w/layers of system mechanics.

Make lower level systems dramatic on their own.

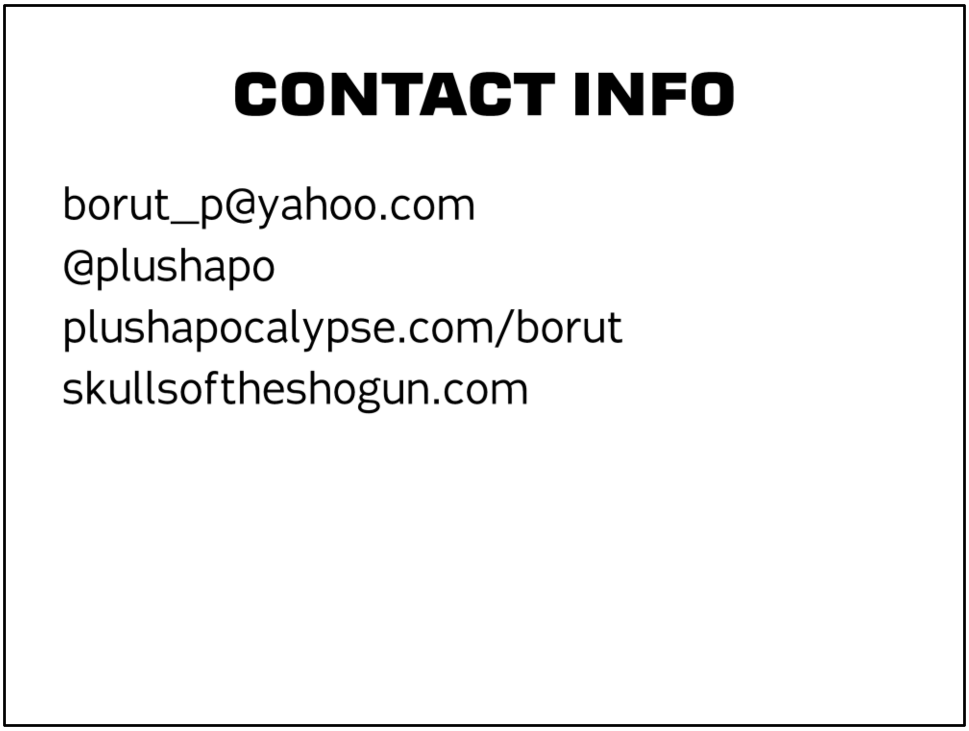
Increases chance wider audiences are engaged.

We just have to be careful to straddle the line between the usefulness and increased accessibility of drama management, and over complication – it's seem like a straightforward distinction but it's easy to cross that line.

But on the whole, it increases the likelihood that audiences of different skill levels and interests are engaged. Especially if you take explicitly into account their skill level and model what they've learned to do in the game and use that to help guide players.

ACCESSIBILITY

- Expressing complex things simply
- Layering info given to the player
- Making model allow for communication
- Timing anticipatory beats
- Pacing those moments (for clarity/learning)
- Guiding the player by introducing new behaviors through AI



CONTACT INFO

borut_p@yahoo.com
@plushapo
plushapocalypse.com/borut
skullsoftheshogun.com