

When Spreadsheets Aren't Enough: A Framework for Approaching System Visualization

Participants:

Alexei Pepers
Ichiro Lambe
Osama Dorias

Brent Ellison
Melissa Ramsden

Crystin Cox
Michael Austin

Additional Contributions:

Alexander Youngblood

Cat Manning

Overview

Video game systems can be difficult to understand, and communicating them effectively can have a big impact on quality. Visualizations are an incredibly helpful tool for managing complexity and presenting data in a way that is easily understandable.

Creating a truly effective visualization is often more difficult than simply following a few technical steps. There are many choices with complicated trade-offs between accuracy, conciseness, readability, usability, and maintainability.

To address this challenge, we propose the PAUSE Framework, designed to help designers create effective visualizations that are crafted to best meet the needs of their intended audience.

The PAUSE Framework

The PAUSE Framework uses five core questions to assist in choosing the best visualization for a given problem.

- **Purpose** - Why are you creating this visualization?
- **Audience** - Who is the audience for your visualization?
- **Usage** - How will this visualization be used?
- **Scope** - What is the minimum amount of information that needs to be shared?
- **Emphasis** - What aspects of your problem are the most important?

Purpose

Why are you creating this visualization?

To provide or obtain clear answers to questions, it is important to be clear about what those questions are. A visualization that serves one purpose, such as helping to fine-tune an economy, won't be as effective for another purpose, like explaining that economy to a publisher. Attempting to use the same visualization for both purposes is often unsuccessful.

It's easiest to think about your purpose in terms of a goal verb. Here are some examples:

Communicate	Plan	Iterate	Understand
<ul style="list-style-type: none">• Convince• Inform• Direct• Explain	<ul style="list-style-type: none">• Predict• React• List• Categorize	<ul style="list-style-type: none">• Explore• Create• Experiment• Tune	<ul style="list-style-type: none">• Confirm• Model• Test• Analyze

Audience

Who is the audience for your visualization?

Effective communication is a two-way process. It's important to consider the expectations, expertise, and goals of the viewer; the content you create should be tailored to the specific audience you are trying to reach. For example, the needs of your development team will differ from those of your players or investors. Creating a visualization that is relevant and valuable to one group won't be as effective for a different group.

Here are questions to ask about your audience:

- What are their goals for this visualization?
- How much domain knowledge do they have?
- What types of communication are they comfortable with?
- What accessibility requirements do they have?

Usage

How will your visualization be used?

Knowing how your visualization will be used will allow you to tailor it to the context. If it is for a single-use presentation, perfect accuracy is seldom important. If it will go into permanent documentation, it's important to be as accurate as possible. If it's in a presentation, it will need to be simpler with fewer details, but if it's as part of an app that people will be directly using on their computers, it can be more dense.

Here are questions to ask to determine the usage of your visualization:

- What is the lifetime? Is this single-use, or will it need to be kept up-to-date over time?
- Will it be interactive and allow for iteration, or will it be static?
- At what distance and in what space will it be viewed?
- How much time will your audience spend with the visualization?

Scope

What is the minimum amount of information that needs to be shared?

Effective communication relies on simplicity. To find the right balance between providing enough information and avoiding overwhelming your audience, focus on what is most important for them to know.

Here are some questions to ask that will help define scope:

- What must be, should be, and could be communicated? Prioritize in that order.
- Is it important to be complete, showing every detail or category?
- Is it important to be accurate and avoid over-simplification?
- Is it important to reveal every edge case?

Emphasis

Which aspects of your problem are the most important?

Direct your audience's attention to the most important part of the data you are sharing, establishing a clear hierarchy of importance. If you do not provide the lens for your audience, they will bring their own; what you emphasize will show what you think is important.

Here are questions that determine emphasis:

- What do you want your audience to focus on?
- How can the data be naturally subdivided or categorized?
- What relationships are most important to show?
- What relationships are less important to show?

Example Walkthrough

To see how the PAUSE framework can help focus your visualization, let's go through an example.

Imagine you are working on a visualization for the crafting system in a survival game, and want to share it with your team.

You can immediately imagine several ways to visualize this design.

- A spreadsheet of every craftable item, listing out links to all other associated content such as input resources and required tools.
- A high-level diagram of the crafting loop and how it connects to the larger game.
- A state diagram showing the steps involved in taking the inputs of a crafting recipe and resources, and generating a crafted item.

- A node diagram of crafting recipes, connecting each recipe to what systems are involved in gathering the resources to craft it, and what the resulting crafted item is then used for.
- A filterable graph that shows the relative number of craftable items when sliced along different dimensions, such as utility to the player, or resource cost..

To know the right approach, it's important to first identify your purpose. Do you want to show how the crafting system connects to the rest of the game? Do you want to be able to visually verify that every recipe is reachable? Do you want to balance the economy?

For this one, let's say you want to see how the crafting system connects to the rest of the game. Spreadsheets and graphs list content instead of showing connections, so we can rule them out. The state diagram is internal to crafting so also less effective, which leaves us the high-level or node diagram. Already, we've cut down the possibilities!

Purpose: *Communicate how the crafting system connects to the rest of the game*

Additional Insight: It can be tempting to look for a visualization solution that can meet all the needs of all audiences, but some purposes are inherently incompatible. If you find yourself frustrated trying to satisfy multiple purposes, consider whether multiple visualizations would be better.

Now, say that the audience you're sharing with is fellow designers. Whatever your own visualizations preferences are, you now must consider your team. Have they used an interactive node diagram app before? Do they go straight for whiteboard pens every time?

Let's imagine you're a tight-knit tech-savvy team. Your shared context frees you to use shorthand terminology in sketching the shape of major systems, spending your energy on a node diagram app that shows specific details. You trust that your team already knows that inventory and crafting will be connected, but they need to know through which specific subsystems. That context is the razor to let you cut unnecessary detail as you build your diagram.

Audience: *The design team, who has a good context for the high level function of crafting and is comfortable with a variety of visualization tools.*

Additional Insight: This approach will likely go over great with the team, but what if you are asked to also share this design with leadership? Tempting as it is to just send what you have, the shared context you used to focus on the most relevant bits isn't going to land for leadership. They're expecting a slide deck with a few diagrams, and sending a link to an interactive node diagram isn't going to work. It isn't starting from scratch, but revising the

assumptions about your audience isn't just replacing some terminology, it can mean rethinking your whole approach!

Continuing with the crafting system example, it's worth considering how often you've been asked to visualize this system and how often it's expected to change. Maybe this is your fourth major iteration on crafting, and you suspect it won't be the last. If you're using it to communicate changes in the updated system, why start from scratch every time?

An editable and flexible visualization tool is going to take more time to set up than sketching a diagram on a whiteboard, but if your team wants to keep referencing it while making changes, it's time well spent. If they also want very specific nitty-gritty details, consider how much of your time will go into maintaining that much detail, weighed against the time it would take an engineer to build you an automated update pipeline.

Even if the purpose is just communication, if effective communication relies on up-to-date details over a long period of time, it's worth investing in finding or building the right visualization tool to meet your needs. An up-to-date view that designers can reference on their own may save you having to answer questions frequently over Slack.

Usage: *Consulted throughout the project's lifetime, used periodically but not daily, expect to be edited as it's kept up-to-date with planned iteration on crafting.*

Additional Insight: Sometimes a visualization will only be used once, such as for a presentation or pitch. Creating a one-off visualization is still worth the time as long as you consider the usage and choose the right visualization for the job.

In a cohesive survival crafting system, if you exhaustively mapped every crafting connection of any kind, you'd be diagramming the whole game. Obviously that won't be readable for anyone, so the question you need to ask is: what's the minimal scope that meets your purpose?

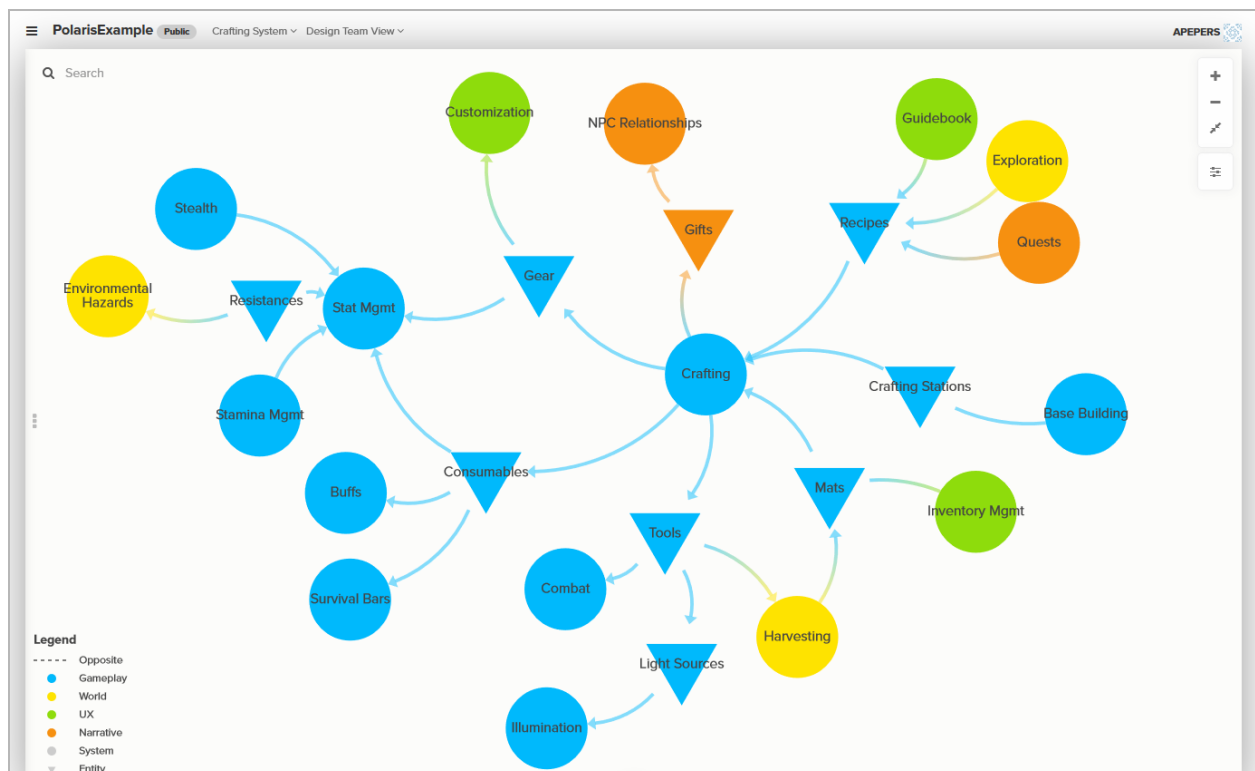
We established earlier that we care about system connections, not the specific content within crafting. We also know our audience has good context for the role of crafting, and just needs to understand the specific details they can work with. That's already cut down your scope a great deal!

We can narrow it down further with follow-up questions. Do they want specific technical details to inform their implementation? Or do they just want to know what mechanics exist within crafting? Do you already know that some details like UI connections are so crafting-specific they won't be applicable to this exercise, or is the point to be totally exhaustive and find the creative edge cases?

If you have an engineering team handling implementation, your scope will be more abstract than the technical specifics. However, it does need to be comprehensive in representing all the system pieces at play. Part of why you're communicating the system to your team is for effective collaboration - there needs to be enough detail to help them spot problems and opportunities you missed. Cutting off the low end of tech specs as well as the high end of simplified big-picture connections leaves you a workable middle - that big-picture is what you'll refocus on when you prepare that deck for leadership.

Scope: *Complete in representing all mechanics within crafting and their existing connections to other mechanics, but not detailed to the point of providing implementation details. Does not include the specifics of content within crafting.*

Additional Insight: Scope is the step in the PAUSE framework that most benefits from an outside perspective. Talk to your audience about what they need from the visualization, how they envision engaging with it, and what decisions they might make with it. Try to get specifics so you can focus on the minimal scope needed to do the job.



At this point, you know your visualization approach, who will use it, how they'll use it, and what information it will contain. You've got everything you need to start building it - but even the most

concise visualization is information-dense, and how you organize that information will emphasize different aspects.

You could make every system an identically shaped node, connected by identically drawn lines, but is that an accurate representation? Crafting and inventory are deeply connected, with many interactive mechanics, whereas crafting and lighting may have only a cursory connection via a handful of light-creating items. Do you want to emphasize those more entwined systems so that your team focuses on the most impactful areas, or do you want to draw attention to the sparse areas in the hope of filling them out?

This intentional organization is the last trick that helps your visualization lead the audience towards fulfilling the intended purpose. If node sizes correspond to the amount of effort allocated to a system, your team will be drawn to making conclusions they might have missed before about where you're relying on risky systems. The same diagram but emphasizing how much content has been created for each system would lead to useful but different observations, so keep in mind which you're looking for, and never be afraid to provide more than one visualization to support multiple goals!

Emphasis: *Representing how much developer effort is allocated to systems via node size, to emphasize how well-supported connections between systems are likely to be.*

Additional Insight: In many ways, Emphasis represents the visual part of the visualization. Tackling Emphasis last is a great way to ensure your visualizations do more than just make a lot of data easier to read. Because you know so much about the purpose, audience, usage, and scope at this point, you can make targeted decisions about how data is represented.

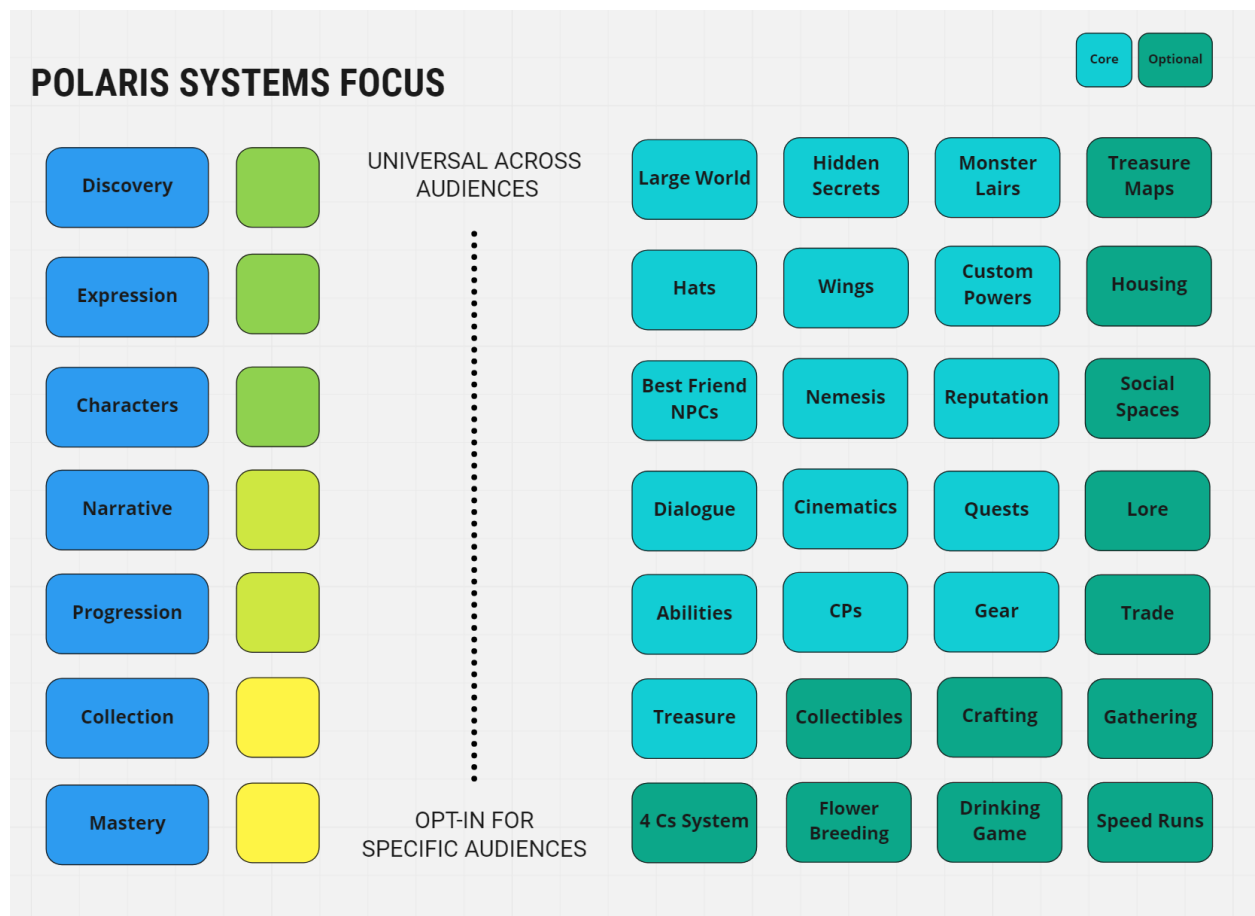
Appendix: Example Visualization Techniques

Here's an appendix of specific visualization techniques you can add to your toolbox. This includes discussion of very well known techniques like spreadsheets and flow charts, where we give an especially strong example of the technique and talk about what makes it exemplary. We also include some of the more novel visualization techniques our team shared, not just as inspiration for more people to use the same techniques, but as an argument for the benefits in figuring out a domain-specific visualization technique that works for your project.

Systems Motivation and Scope Diagram

The purpose of this visualization is to demonstrate what systems are part of the game, the motivations they are targeted to, and the importance of that audience segment to the game.

It's very easy to go too complex here or try to be perfectly accurate and mess up the clarity of the communication. For this type of diagram, half of the purpose is to show the intention and thought that went into the design. In many executive pitches, there is not enough demonstration of the 'why' of the decisions and instead the focus is on the 'what'.



Purpose - Identify core and optional systems, show intention and thought

Audience - Executives

Usage - Pitch Deck

Scope - Concise, High Level

Emphasis - Player motivations, broad systems

Object Size Standards Visualization

The purpose of this visualization is to demonstrate art standards for objects in an easy to understand way to the art team. Often, arbitrary constraints like “Objects must be bigger than 0.5m in size” or “Don’t make things 3.5m tall” are hard to remember; and they're confusing, since there’s no perceived justification.

Showing relationships between classes of objects next to each other is particularly effective, as it helps the viewer build an intuition for the broader rules that are being demonstrated.



Purpose - Define range of art standards

Audience - Art and design team

Usage - Design spec, Maintained

Scope - Exemplars, Accurate

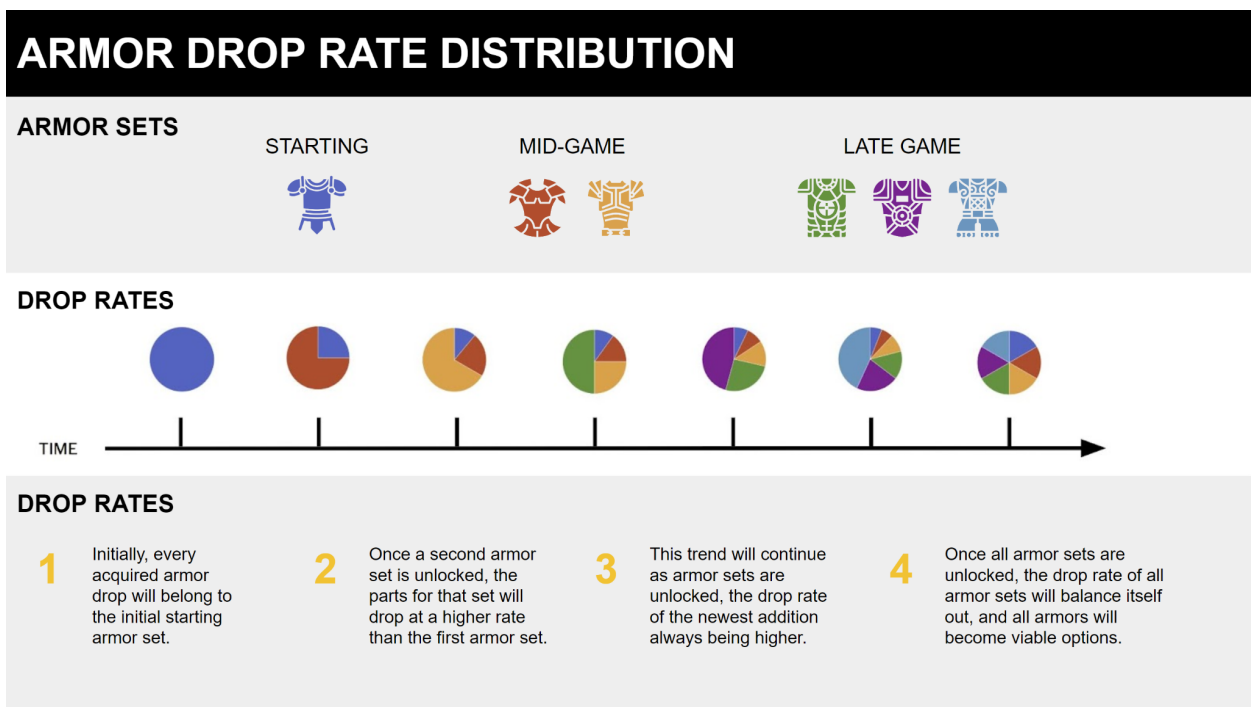
Emphasis - Size metrics

Visualization of Loot Drop Rate Over Time

The purpose of this visualization is to illustrate the intended drop ratios of armor types as more are added over time.

This visualization is paired with a step-by-step text explanation to ensure that the reader understands the context, and so ensure the understanding of the visuals.

Adding specific numbers/percentages or values on the timeline would distract from the message and mire the reader in the details.



Purpose - Demonstrate intended armor loot drop ratios over time

Audience - Design team

Usage - Design intentions, starting point

Scope - Concise, high level

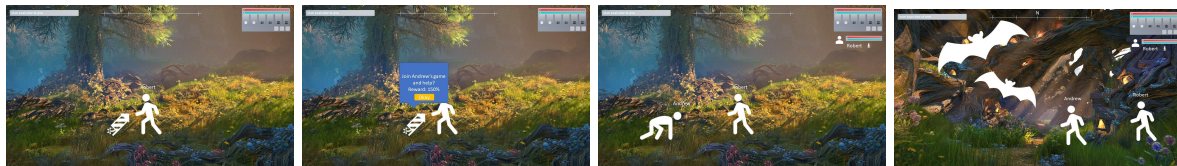
Emphasis - Economy systems

Emphasis - Economy systems, gameplay modes

Storyboarding for Systems Visualization

Often, when communicating a system design to others on the development team, what is needed is an understanding of how players will experience the design more than how the design functions or how it is implemented. Simple storyboards that show, step-by-step, how a player will engage with the system can communicate both the design intent of the system as well as direction around UX and interaction design needed to support the system. Creating this type of visualization also has the added benefit of forcing the system designer to consider player experience concretely at a relatively early point in design.

This type of visualization works best when it can be done quickly and with little concern about aesthetics. It is usually created early in the design process to create alignment with others on the development team.



Purpose - Show how players will experience a system

Audience - Developers

Usage - Design documents

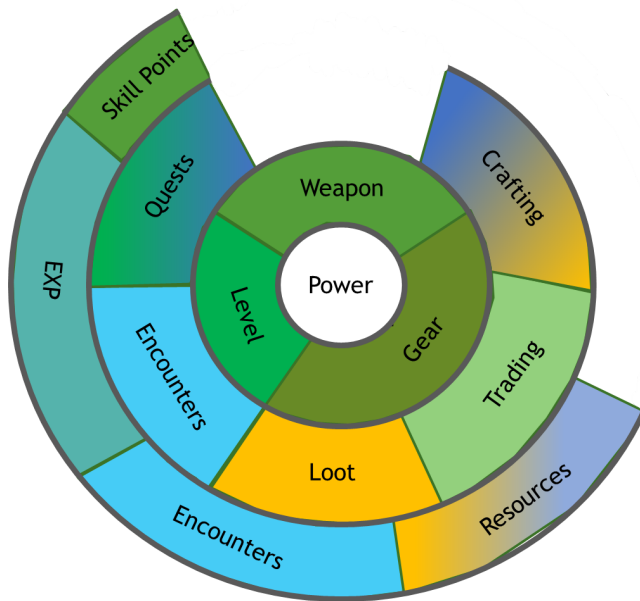
Scope - Comprehensive, low detail

Emphasis - Usability, player understanding

Systems Relationship Visualization

We may wish to communicate the relationships between various systems. Here, showing multiple layers of relationship is a challenge when using a more traditional flow chart, but understanding that two systems affect each other non-directly is sometimes vitally important. This example uses an expanding ring system to show that some systems affect a player's power directly (weapon, level, gear) while others affect it indirectly through one or more other systems. Colors are also used to emphasize how one system may influence multiple others.

System designers often use this type of visualization to help themselves gain a holistic view of how systems interact and get a sense for how the whole of the meta-system is balanced. In this example, for instance, weapons have relatively few connected systems, gear has a large number of directly connected systems, and level has the largest number of systems that influence it. Without this type of visualization, it can be easy for unintentional imbalances to emerge and not be discovered until well into play testing.



Purpose - Show how various systems affect a key player experience through multiple layers

Audience - Designers

Usage - Design documentation

Scope - High level, comprehensive

Emphasis - Systems relationship, non-direct system-to-system impact

Spreadsheets as Gut-Check

While spreadsheets sometimes become inscrutable to humans (for which the above systems visualization are potential cures), they remain popular tools for their relative accessibility - they often don't require a great deal of technical acumen to create or reference.

The below example defines a list of items and various resource costs for each item. Without any bespoke visualization tools, conditional formatting allows us to see which resources might currently be underutilized.

Tile Id	Resource Costs																	
	Food	Water	Work	Wood	Energy	Plastic	Unrest	Compl	Metal	Miner	Points	Seeds	Stone	Occup	Machi	Res15	Res16	Resou
Trail	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Bridge	0	0	0	50	0	0	0	0	10	0	0	0	50	0	0	0	0	0
Lookout Post	0	0	0	10	0	0	0	0	0	0	0	0	5	0	0	0	0	0
Mountain Lookout Post	0	0	0	20	0	0	0	0	0	0	0	0	10	0	0	0	0	0
Camp	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apartments	0	0	0	50	0	10	0	0	0	0	0	0	50	0	0	0	0	0
Highrise	0	0	0	0	0	20	0	0	20	20	0	0	100	0	0	0	0	0
Tilled Land	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
Farmland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Empty Farm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Farm Level 2	0	0	0	100	0	20	0	1000	20	0	0	0	0	0	0	0	0	0
Arboretum	0	0	0	100	0	20	0	1000	20	0	0	0	0	0	0	0	0	0
Renewable Forest	0	0	0	0	0	0	0	1000	0	0	0	100	0	0	0	0	0	0
Well	0	0	0	20	0	0	0	0	0	0	0	0	20	0	0	0	0	0
Pump Station	0	0	0	0	0	50	0	0	50	0	0	0	0	0	0	0	0	0
Water Tower	0	0	0	0	0	10	0	0	10	0	0	0	0	0	0	0	0	0

This should be familiar to any designer, but is occasionally overlooked as a tool available when engineering resources aren't!

Purpose - Visualization when bespoke tools are unavailable

Audience - Designers

Usage - A quick gut-check

Scope - High-level

Emphasis - Systems data

Interactive Possibility Space

Procedural content generation often uses layers of loosely coupled systems, combining many sources of content. The possibility space of possible combinations quickly becomes impossible to mentally map, but it's hugely important for the team to know what end results to expect.

If a tool can be set up to simulate running the generation rulesets on the current game content, then it can dump the full results into a machine-readable format and surface them with a user-friendly visualization interface. In the below example, data from the game project was converted into JSON, and fed into a simple HTML frontend that presented the combined content in a form that could be filtered based on individual priorities.

Initially created with designers in mind as the audience, it became invaluable for QA who had a daily need to verify the current expectations for content within a procedural world. The time saved in no longer having to chase down answers to “where in the game would I find X?” more than made up for the effort spent creating a bespoke tool!

Room Summaries

☒ Show Items ☒ Show Resources ☒ Show Schematics ☒ Show Encounters
Filter to only rooms with: ☐ Items ☐ Resources ☐ Schematics ☐ Encounters

<div><div>DungeonRoom(6)</div><div>Large Tile:</div><div><ul style="list-style-type: none">PillarRoom<ul style="list-style-type: none">Items<ul style="list-style-type: none">DaggerResources<ul style="list-style-type: none">StoneCoinsEncounters<ul style="list-style-type: none">GoblinAmbushDragonLair<ul style="list-style-type: none">Items<ul style="list-style-type: none">EnchantedDaggerCloakResources<ul style="list-style-type: none">CoinsGemsDragonToothEncounters<ul style="list-style-type: none">GoblinAmbushDragonBattleRoamingThiefDefaultSpawnRoom<ul style="list-style-type: none">Encounters<ul style="list-style-type: none">Tutorial</div><div>Small Tile:</div><div><ul style="list-style-type: none">SmallTrap<ul style="list-style-type: none">Items<ul style="list-style-type: none">Mechanism</div></div>	<div><div>ForestRuins(3)</div><div>Large Tile:</div><div><ul style="list-style-type: none">PillarRoom<ul style="list-style-type: none">Items<ul style="list-style-type: none">DaggerResources<ul style="list-style-type: none">StoneCoinsEncounters<ul style="list-style-type: none">GoblinAmbush</div><div>Small Tile:</div><div><ul style="list-style-type: none">SmallTrap<ul style="list-style-type: none">Items<ul style="list-style-type: none">MechanismResources<ul style="list-style-type: none">StoneWoodSchematics<ul style="list-style-type: none">BasicTrapTreasureCache<ul style="list-style-type: none">Items<ul style="list-style-type: none">EnchantedDaggerWandResources<ul style="list-style-type: none">CoinsGemsSchematics<ul style="list-style-type: none">ChestEncounters<ul style="list-style-type: none">RoamingThief</div></div>	<div><div>DungeonHall(3)</div><div>Large Tile:</div><div><ul style="list-style-type: none">DefaultSpawnRoom<ul style="list-style-type: none">Encounters<ul style="list-style-type: none">Tutorial</div><div>Small Tile:</div><div><ul style="list-style-type: none">SmallTrap<ul style="list-style-type: none">Items<ul style="list-style-type: none">MechanismResources<ul style="list-style-type: none">StoneWoodSchematics<ul style="list-style-type: none">BasicTrapSmallPoisonTrap<ul style="list-style-type: none">Items<ul style="list-style-type: none">MechanismPoisonVialResources<ul style="list-style-type: none">NightshadeStoneSchematics<ul style="list-style-type: none">AdvancedTrap</div></div>
---	--	---

Purpose - Show the complete possibility space generated by many sources of content

Audience - Designers and QA

Usage - The up-to-date source of truth to consult frequently, used to check results of changes

Scope - Comprehensive, Accurate

Emphasis - Dungeon rooms as the organizing container for content