

Silly reasons to not make a puzzle game

Cerise Talis
(she/they)

Hi folks! I'm excited to speak to you all today.

Background

- OPLAYER
- List of things I used to believe
- This talk will make your life harder



OPLAYER (2025)

This is a talk that arose out of the development of OPLAYER, a team project for which I designed a bunch of puzzles and did the 3D art.

OPLAYER is a static image puzzle game: its entire world is on your screen right now! And to make that possible, we had to stick to a basic set of mechanics that could reasonably be inferred just by looking. So it's all sliding tile puzzles where you connect wires to transmit power.

As development progressed, we began to wonder why we hadn't seen any similar games in the thinky realm. Nothing identical, of course: we were imagining all the games we could've made if we hadn't constrained ourselves to a static image. Sliding tiles with wires is well-trodden ground in games overall, but rarely are those puzzles aspirational to any significant degree.

There was a sinking feeling when we realized what this meant: we weren't exploring unexplored systems. Others had been here but had turned back, silently, for reasons unknown.

This talk is a catalogue of reasons you might turn back from making a puzzle game. I've used all these reasons to justify not making a project at some point in my life. But with time and experience they've all proven silly.

If the creative process is about pruning ideas, by the end of this talk you'll have less justifications to use when pruning, which means this talk will make your life harder. My apologies!

Background

Luckily, I worked on the game long enough that I had time to come up with something better.

– Lucas Pope, *Obra Dinn Modeling Timelapse*

Lucas Pope recently said in an Obra Dinn modeling timelapse...

[Read quote]

My hope with this talk is that developers will give themselves more time to solve problems which are worth solving.

So! For the first reason.

“Impossible to extract”

Extraction: Puzzlehunt term for *puzzle output*

- Impossible for computers, possible for players
- Post-modern ≠ Stopping too soon

“I’m not pursuing these puzzle systems because they’re impossible to extract from.”

For folks unfamiliar, “extraction” is puzzlehunt terminology for plucking the answer out of a puzzle in order to put it into an answer-checker. When you’re making a puzzle for a puzzlehunt, you need to design it in such a way that an answer pops out, typically a word or a series of numbers.

Rather frustratingly, most of the world isn’t *about* words and numbers. So it’s all too easy to make a puzzle that boils down to shuffling symbols around until an answer appears.

On a wider scale, game design has this problem where it conflates what a computer is able to evaluate with what a player is able to value. So you’ll discover these beautiful systems which communicate ideas, but they’ll seem less worth exploring because you can’t make an answer pop out.

Now exploring this territory of unextractable systems can feel like wishy-washy unfulfilling post-modern territory. But if you have a sense for how much work goes into a large-scale piece of art, then you can use this sense to avoid “stopping too soon”. You can feel reassured that you’re not just being lazy.

“Impossible to extract”



Opus Magnum (2017)



Half-Life: Alyx (2020)

- Opus Magnum -

So in Opus Magnum, there's a simple goal of product-in-product out, but that's easy to achieve in a nonsense way. Those additional goals in the bottom-right were only suggestions, and I didn't follow them. Instead, in the moment-to-moment, I was guided by my own sense of beauty.

- Half-Life: Alyx -

In Half-Life: Alyx, the most cautious strategies are the most reliable but the least interesting. There's a potential version of the game where cautious behavior is punished, but it'd be dramatically less fun than what we've got.

Players can indeed optimize the fun out of your systems, especially if they're unsure whether or not your game is good. But that doesn't mean those systems aren't worth exploring with full confidence.

“Solutions are non-unique”

- Pen-and-paper habits (forward setting)
- Boxing up ambiguities
- Make smaller puzzles

Next up is a reason that may sound familiar to the pen-and-paper puzzlers in the audience.

“I’m not pursuing these puzzle systems because the answers are non-unique, or the solutions that arrive at those answers are non-unique.”

This reason shows up when you’re making puzzles and there’s no way to force the player to do something specific. There’s a small set of equally-valid solutions. Oftentimes it’s a matter of parity, or of the very last steps of a solution.

If your game is based on tight chains of logic, non-uniqueness problems can be really upsetting. Oftentimes, players can recognize and “box up” ambiguities they come across, as long as those ambiguities are classifiable and constrained. If uniqueness is important to your game, and the player can’t do this boxing up, consider reapproaching the scope of individual puzzles.

It’s all too easy to fall for the allure of forward setting, where you place down more and more clues into an empty grid until the solution is unique. Forward setting is great as a designer to explore your own systems, but without careful discretion, the puzzles that result are too large and unfocused to communicate anything clearly.

So if you’re overscoping your puzzles, don’t blame the systems!

“Wouldn’t quiz the player enough”

- Making puzzles easier is the bulk of the work
- A friend wouldn’t quiz you
- You’re not obligated to grade the player



Storyteller (2023)

Next: “These puzzles systems don’t give the player enough of a challenge.”

The day-to-day of making puzzles is making puzzles easier, and being proud when you get your points across. Most of my puzzles stop only an inch short of telling the player what to do... and some of these are still difficult puzzles!

Lots of beautiful learning is robbed of its beauty when turned into quizzes, with all the apparatus a quiz entails. For reference, see primary school.

- Storyteller -

Storyteller is an easy game that explores its design space well. I would not have been convinced of the communicative power of Storyteller’s systems if the game was difficult.

“Lacks a big twist”

- Quizzing the player
- Don’t undermine your game’s confidence
- You’re not obligated to gaslight the player

Up next: “This puzzle game needs a big twist ending, and I can’t figure out what that twist would be.”

Surprisingly, this line of reasoning usually stems from wanting to quiz the player. It’s tempting to want your game to throw players for a loop: it’s an exercising of your position of power as a designer.

But twists in good games rarely trick the player into thinking one thing and then pull the rug. That’s gaslighting, and it’s usually not very fun!

Instead, the twist arises from the assumptions a player enters the game with. If you concern yourself with the chain of communication your puzzles live inside, you can begin to assess the priors of your players, and design twists accordingly.

“No musical bird”

- Sokoban is not the lowest level your system exists in
- When you paint the bird, you begin to spite the bird
- Sokoban is a thick brush

“No musical bird” is in reference to a quote from Harry Partch that I’ll show shortly.

When contemplating a system you’d like to explore in your game, it’s easy to forget that your game’s existing apparatus isn’t the most expressive form of that system. In fact, your apparatus represents a limited palette by which you still, **miraculously**, get your points across.

Moreover, the apparatus of a sokoban means that it’s exceedingly rare to find complementary systems. If you take the apparatus for granted and forget its weight, then the ideas you’d like to explore will lose definition.

The difficulty of notating a vocal line that preserves the integrity of spoken words is not a small one, but neither is it insurmountable... and the compensations are large.

Few orthodox musicians would call such vocal lines "musical," and Combarieu declares that the songs of birds are not "musical" either, because they are "very difficult to take down in notation."

Will some divine power please create a "musical" bird to sing the Air for G String in exact Equal Temperament for M. Combarieu?

– Harry Partch, *Genesis of a Music* (2nd ed.), 45-46

So here's that Harry Partch quote. Partch was a musician who made his own instruments, and he wrote a book about it called *Genesis of a Music*, which says...

[Read quote]

COOM-bah-ee-yuh

Apparatus of a Sokoban

1. Square grid
2. Player character
3. Solutions are sequences of directional inputs
4. Player pushes crates
5. Walls block player and crates
6. Multipush (or lack thereof)
7. Interactions take place in a world of physics
8. Linear, discrete time, with dead-end states
9. Undo stack management
10. Level ends immediately once solved

In games, the inflexible “notation” that we take for granted is the first prototype we make of a game. Sokobans are particularly inflexible because they come with lots of **sub-systems** that all have to be coded at once.

[Gesture: lower hand from top to bottom, palm up]

Which isn’t to say you shouldn’t make a sokoban! I love sokobans! But the tradition of sokobans can make good ideas seem less worthy of exploration.

“Doesn’t work without words”

- Your game is not pure maths
- Subtle prerequisites
- Using natural terms → Hesitation minimized

So, next up: “This game needs words or tutorials, and I was hoping it’d be wordless.”

As artists, we hold a belligerent hope that our art is “pure” or “raw” in some way. In reality, of course, art concerns itself with its environment.

This hopeful drive to make something pure gets especially prickly when we’re making sokobans, because sokobans *feel* like pure maths. But actually they’re this tradition with an apparatus we take for granted.

In practice, what this means is that we contort our puzzle games to seem more pure than they actually are, and the way we talk to the player feels stifled and unnatural. If we are to explore our design spaces unhesitatingly, we need to communicate unhesitatingly, which often means flat-out telling the player what to do.

“Doesn’t work without words”



Dr. Robotnik's Ring Racers (2024)



Vividlope (2023)

Here are two games from genres that are typically tutorial-free.

- Ring Racers -

Dr. Robotnik's Ring Racers is a kart racer made by fighting game freaks. You've got a ring meter, but it functions way differently than what you'd expect in a kart racer. You can, for example, go into debt. Go into too much debt, and you're viable to explode into bits.

The game needs its tutorials not just to make the basic mechanics clear, but to make clear the game's reasons for existing.

- Vividlope -

Vividlope is an arcade-style game where you run around stepping on tiles to change their color.

It's got a lotta PSX charm, and in the tradition of PSX arcade games, Vividlope introduces new mechanics through text tutorials. It's also got a bestiary where you can spawn in enemies and see text descriptions of their behavior.

“Undo stack management too futzy”

Designing to reveal the nature of the undo stack. – Katelyn Δ

- You’re right to be concerned with the moment-to-moment
- If your game needs an undo stack... you have options

Next: “Navigating these systems creates an undo stack that’s too futzy to be helpful.”

Your head’s in the right place! My good friend Katelyn Delta once quipped that complicated sokobans are “designed to reveal the nature of the undo stack.”

Which is to say, undo stacks add a tremendous amount of overhead. The player needs to keep a running model of every past state of the puzzle. Experientially, playing a puzzle game can feel like rock climbing, where you’re hyper-aware of your limbs. You’re bifurcating up a cliff, and it’s stressful.

Thankfully, if your game does need an undo stack, there are ways to make it less of a nightmare.

“Undo stack management too futzy”



*Trifolium: The Adventures
of Gary Pretzelneck
(upcoming)*



Jack's Sokoban (2022)

- Trifolium -

In the demo for Trifolium, you can click on your own body to navigate to that point in the undo stack, then splice in new inputs while leaving the rest of the undo stack intact. I’m excited to see how this plays out in the full game.

- Jack’s Sokoban -

Jack’s Sokoban takes a maximalist approach: the undo stack is visualized, and you can reorder individual states.

Importantly, it implements a ruleset isomorphic to sokoban, which removes the need for a player character. Not having a player character means the undo stack is much tidier.

“Motivated by spite”

I am surprised that among so many rare minds who would have done this much better than I, there was no one who had the patience to unravel these things and that they almost all imitated those travelers who leaving the main road to take a crossroad remain lost in the brambles and precipices.

– René Descartes, *Meditations on First Philosophy*,
as translated by John Cottingham

So: “I’m only making this game out of spite.”

Making art out of spite feels petty and small, but best I can tell the experience is incredibly common. If you’re engaging at all with your contemporaries, and your game is something you’d like to see in the world, you’ll be frustrated that others didn’t do it sooner.

Descartes says:

[Read quote]

“Afraid of edge cases”

- What's useful to you is useful to your players
- Animations are helpful



Rivals of Aether (2017)

“I’m afraid that the abundance of edge cases in my system, situations where I have to make a choice for my game, will make the game feel too artificial.”

Resolving edge cases gracefully is one of the great joys of puzzle game design. Players are interested in how you as a designer chose to remedy ambiguous scenarios!

Oftentimes, solving an edge case involves making a whole new system for your game. As a player, it’s cool to explore the types of thorny systems which require “protection against infinite loops”, for example.

If you’re making an earnest attempt at exploring a design space, and you’re putting in the effort to let down your veil, these secondary systems won’t feel undeserved. You’re not ruining the magic.

- Rivals of Aether -

Rivals of Aether and other fighting games of its ilk provide a Training Mode where you can view hitboxes, mess with CPUs, and step through scenarios frame-by-frame.

Fighting games are primarily systems of hitboxes and vectors. But normal gameplay happens too fast to internalize these systems with clarity. By letting down the veil, secondary decisions that the developers make feel justified, or at the very least colorful.

“Induces analysis paralysis”

- Common in boardgame design
- Overly concerned with player discomfort
- Be a friend, but don't feel like a burden

“Analysis paralysis” is a phrase that boardgame designers love love love to use when justifying their design decisions.

When you're refining your systems, it can feel a lot like tailoring your game to maintain the player's comfort, so that they don't get overwhelmed. But often it's just good-old-fashioned subtractive design.

If your systems are refined enough and the communication clear enough, navigating a busy possibility space can be a lotta fun! So if you'd like to make a game all about massive possibility spaces, if you're friendly and generous with your visualizations, you don't gotta feel like a burden.

“Requires pre-existing trust”

- Right now is the first time your game could ever be made
- Be generous, be kind <3

Finally, one last reason: “This game requires the player to have pre-existing trust in the designer.”

If you’re trying to do something new, and you’re being honest, generous, and loving, don’t worry so much about deserving trust.

It can be hard to believe, but sometimes you’re doing something and this is the first time it could ever happen. So the trust you’d ideally want to have is impossible anyways.

If you’re looking for reasons for your game to exist that live outside your own capacity for care, don’t worry: you’re plenty enough!

- Impossible to extract
- Solutions are non-unique
- Wouldn't quiz the player enough
- Lacks a big twist
- No musical bird
- Doesn't work without words
- Undo stack management too futzy
- Induces analysis paralysis
- Motivated by spite
- Afraid of edge cases
- Requires pre-existing trust

Thanks to...
Abi
Autumn
Katelyn Δ
Seren☆

cerisetalis.com

If you're from the future and would like to chat, you can find me at cerisetalis.com.