



Puzzle Game Magic Secrets

Brett Taylor
Founder, My Dog Zorro

GAME DEVELOPERS CONFERENCE

MARCH 18–22, 2019 | #GDC19

ME

- Brett Taylor
- Masters in Cognitive Studies from Columbia
- Developing games (2008 - present)
- Founded My Dog Zorro in 2015
- Made Linelight!

LINELIGHT

- My Dog Zorro's first release!
- Puzzle game
- 2017
- PS4, Steam, iOS, Android
- Earned about \$100,000 in 2017 and 2018 after taxes





Metascore

Universal acclaim
based on 4 Critics

What's this?

Summary: Linelight is an elegant puzzle game set in a world of lines.

REVIEWS

[Review Policy](#)

4.9



10,747 total



Ratings and Reviews

4.9

out of 5

422 Ratings



LINELIGHT TRAILER



Overview

- 1. Working Memory**
- 2. Puzzle Characteristics**
- 3. 7 Lessons from Linelight**
- 4. How I Make Puzzles**

BEFORE WE GET STARTED...

- This talk prioritizes
 - Fun
 - Non-random puzzle games
- You can subvert these rules for alternate effect
- Sooo much I don't have time to cover here

Q: What makes a puzzle *fun*?

A: **Exhibiting mastery, skill, or wit.**

“MECHANICS” EXAMPLES

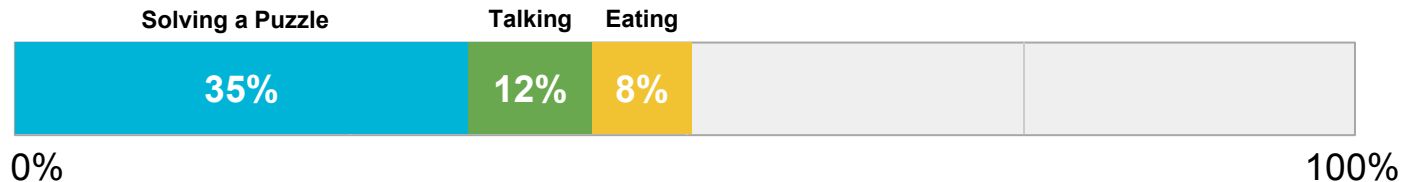
- Boxes
- Enemies
- Gravity-flipping switches
- Time-rewinding immunity
- Orange gel that makes you reeeal fast



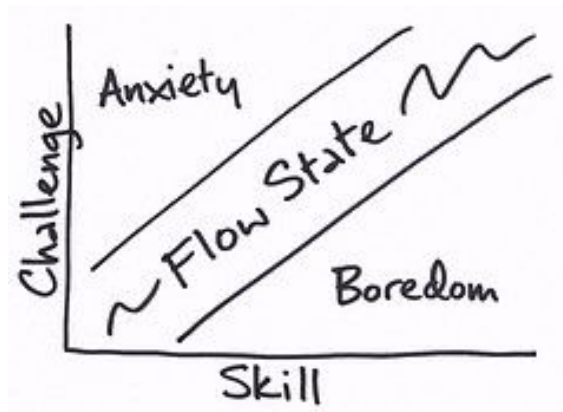
- 1. Working Memory**
- 2. Puzzle Characteristics**
- 3. 8 Lessons from Linelight**
- 4. How I Make Puzzles**

WORKING MEMORY?

- Human RAM.

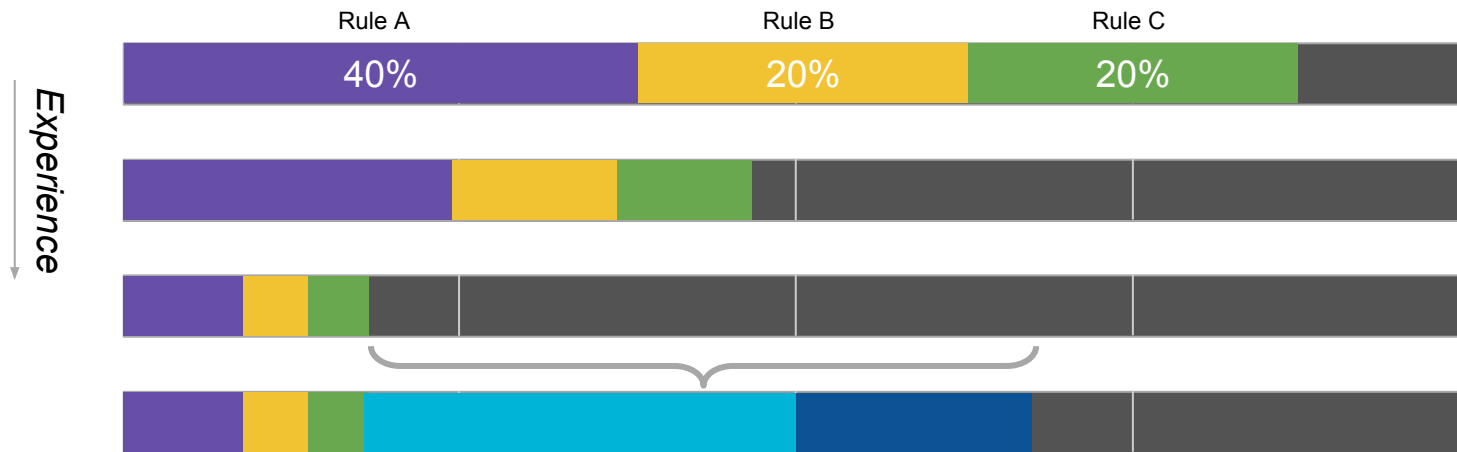


WORKING MEMORY & FLOW

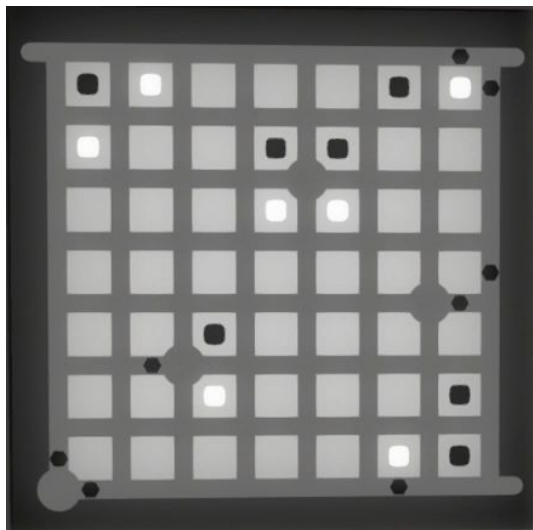


CONCEPTS COMPACT

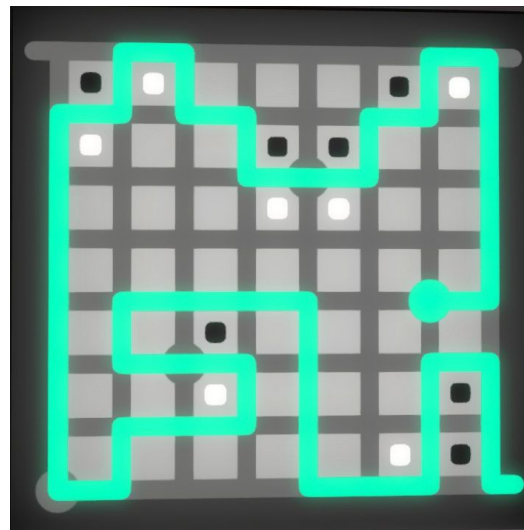
After repetition, concepts take up less space :D



CONCEPTS COMPACT



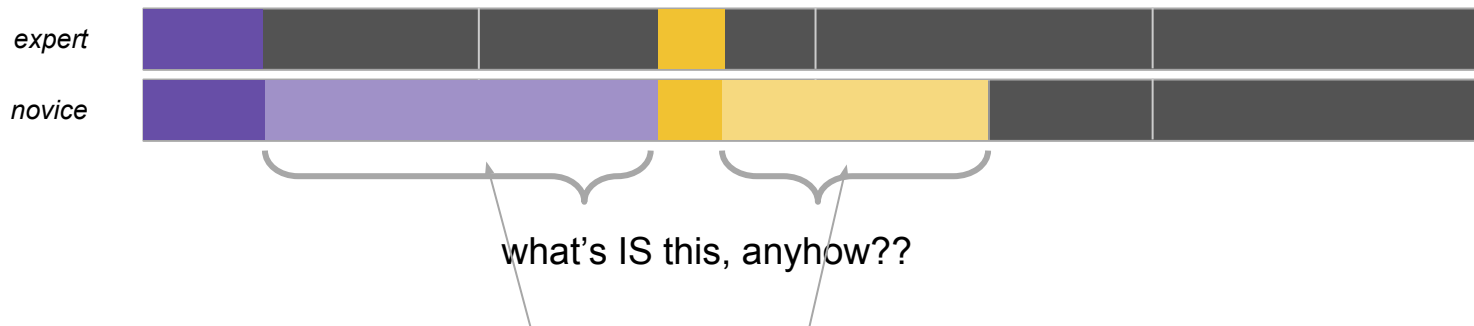
for beginner



for expert



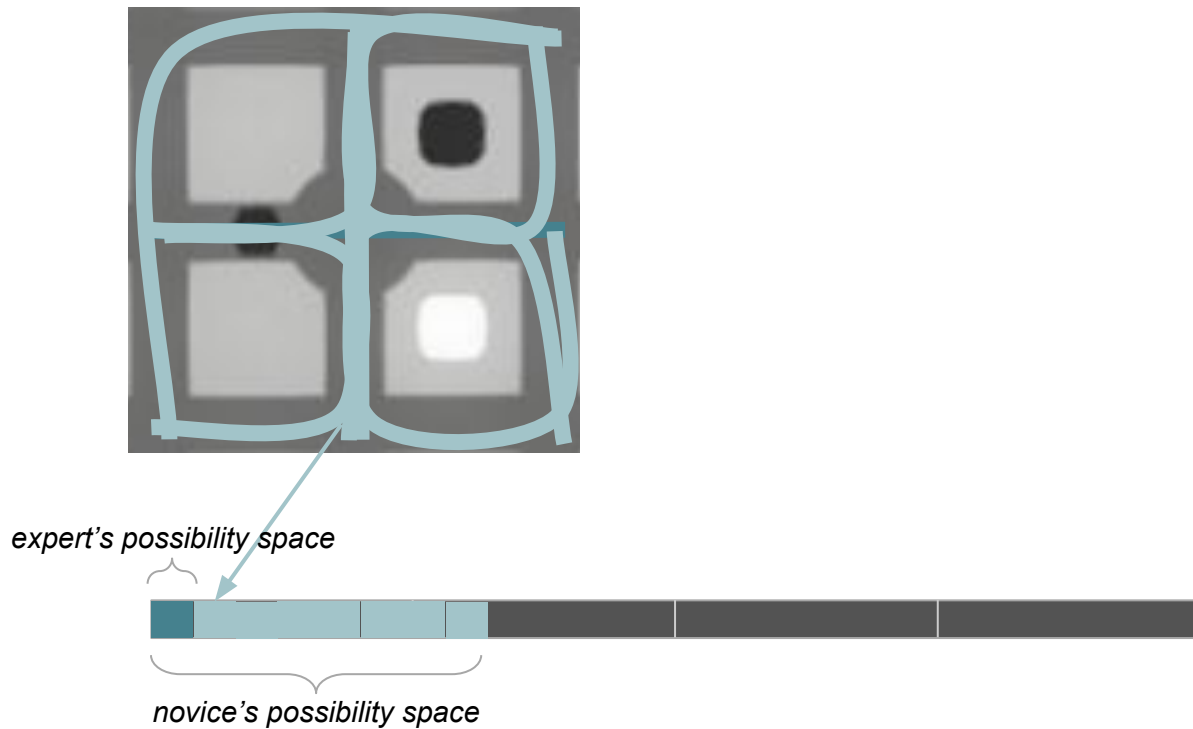
POSSIBILITY SPACE



Possibility Space!

Possibility space is reduced
as experience grows.

POSSIBILITY SPACE



1. Working Memory
- 2. Puzzle Characteristics**
3. 7 Lessons from Linelight
4. How I Make Puzzles

PUZZLE TRAITS



Braid

5	3			7				
6			1	9	5			
		8				6		
8				6				3
4		8			3			1
				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Sudoku

HANDCRAFTED

finite possible puzzles
few skills repeated
more novel solutions
authored content
fresh, varied



this
talk



RUSH HOUR



22

	7	9	6	2	4		
9			1				2
1	8	5	3		6		1
5		4	7	9			
		8					
4		3	2	1			7
9		2	4	8		5	
6			3				8
	8	6	9	5	1		



NOISE

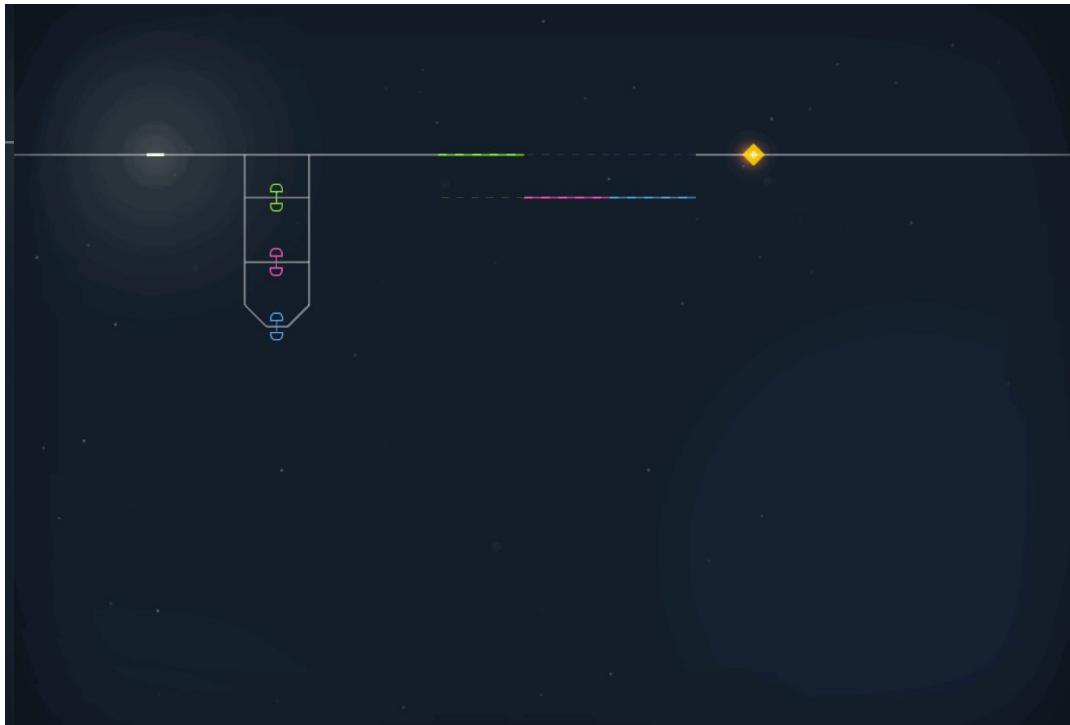
- **Anything that occupies working memory but doesn't contribute to the puzzle.**
- Steals working memory from player

“Flippers” and “Moving Streets”



NOISE

“Get 3 flippers in right setting to complete a path.”



EXAMPLE



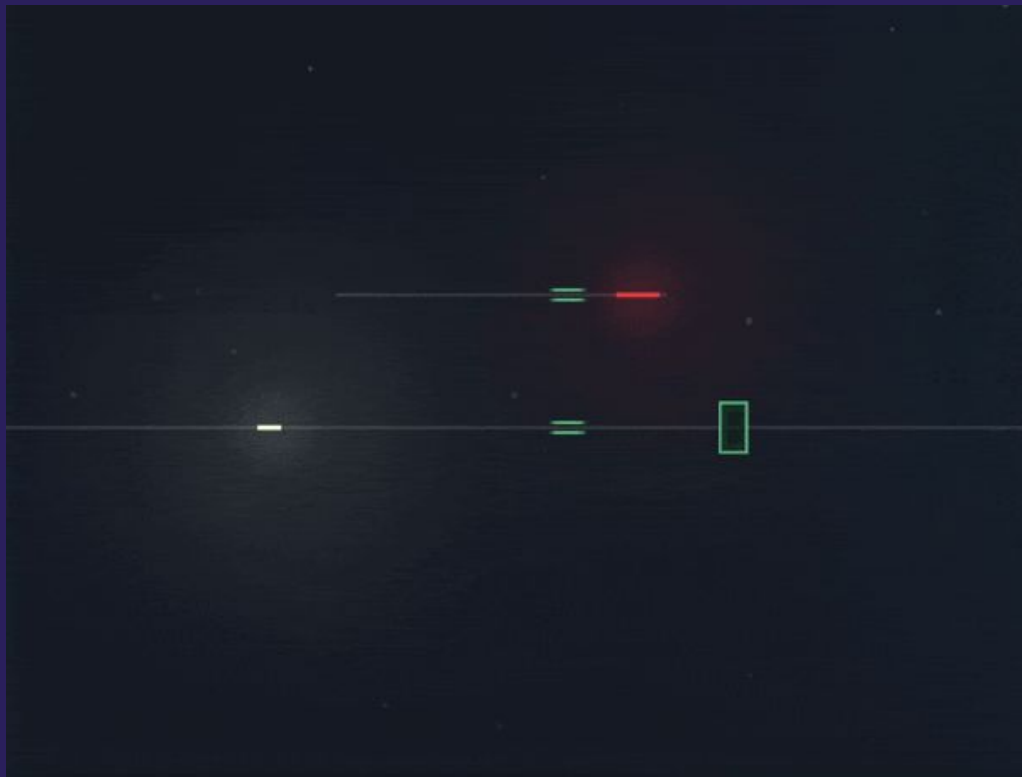
as is



noise removed



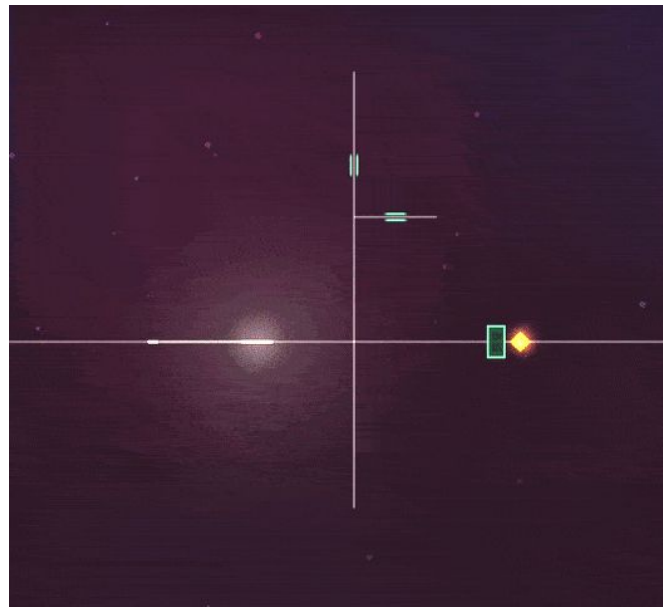
“Coverups”



EXAMPLE

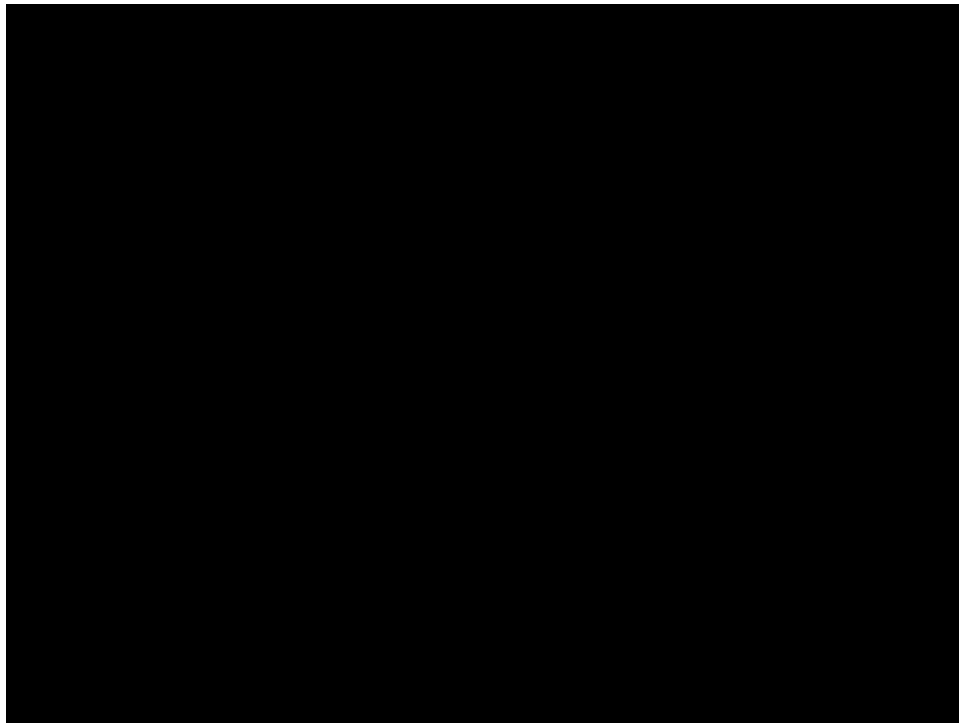


*w/ noise
(nobody solves accidentally!)*



*no noise
(everyone solved accidentally)*

EXAMPLE



NOISE

- Makes puzzles artificially harder
- Requires diligence to identify!
- If left unchecked, noise will pollute your game
- Useful if used intentionally
 - Can deter solving accidentally
 - Satisfying to use skill to see through noise (ala bullet hell)

Let's give our brains a quick break.

10-second stretch!

1. Working Memory
2. Puzzle Characteristics
3. **7 Lessons from Linelight**
4. How I Make Puzzles

1. SIMPLIFY, SIMPLIFY, SIMPLIFY.

Less noise → Cleaner, tighter puzzles

1. SIMPLIFY, SIMPLIFY, SIMPLIFY.

- *What:* Cleaner, tighter puzzles
- *How:*
 - Describe the solution.
 - **Remove anything you didn't mention.**
- #CutTheNoise
- Gets easier with practice

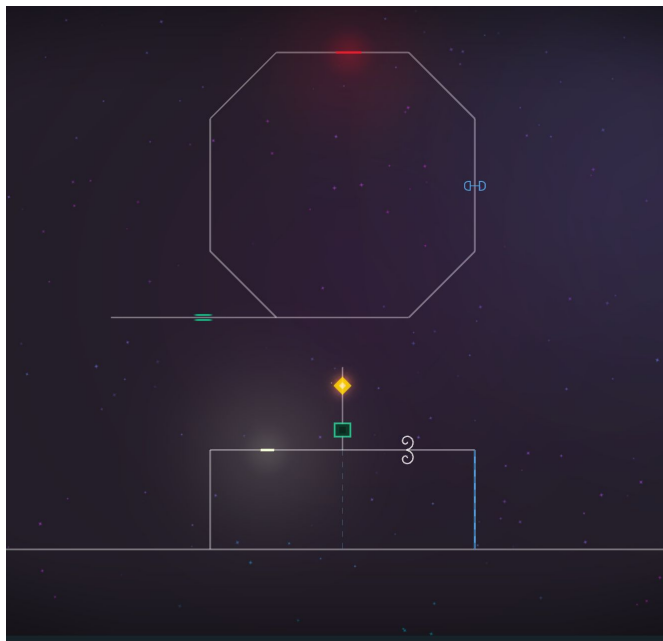
1. SIMPLIFY, SIMPLIFY, SIMPLIFY.

- I hacked every Linelight level down to the BONE
- Then KEPT removing extraneous elements
- Looking back, could have simplified much more

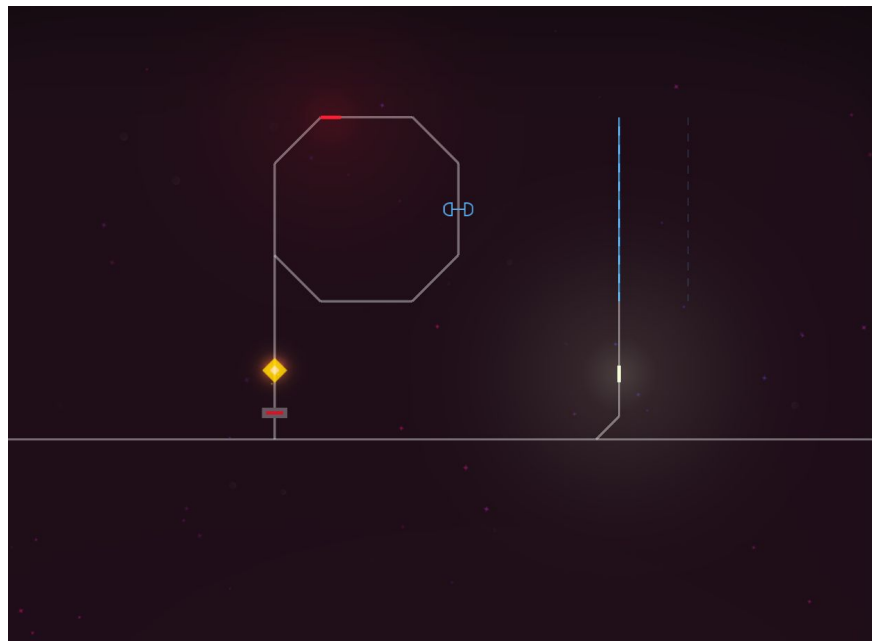
1. SIMPLIFY, SIMPLIFY, SIMPLIFY.

- Vet every level at least 2x
- I vetted Linelight's levels ~4x each
- Doesn't belong? Cut it.
- Very satisfying :)
- Want noise? Start with none; *add* it.
 - Noise can enhance experience, prevent accidental solving, change of pace, etc.

1. SIMPLIFY (Remove Extraneous Stuff)

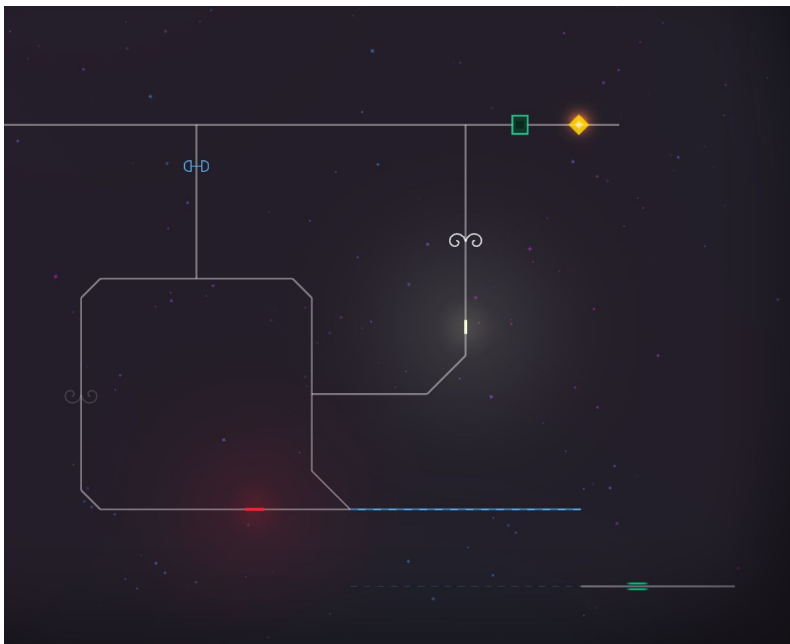


before

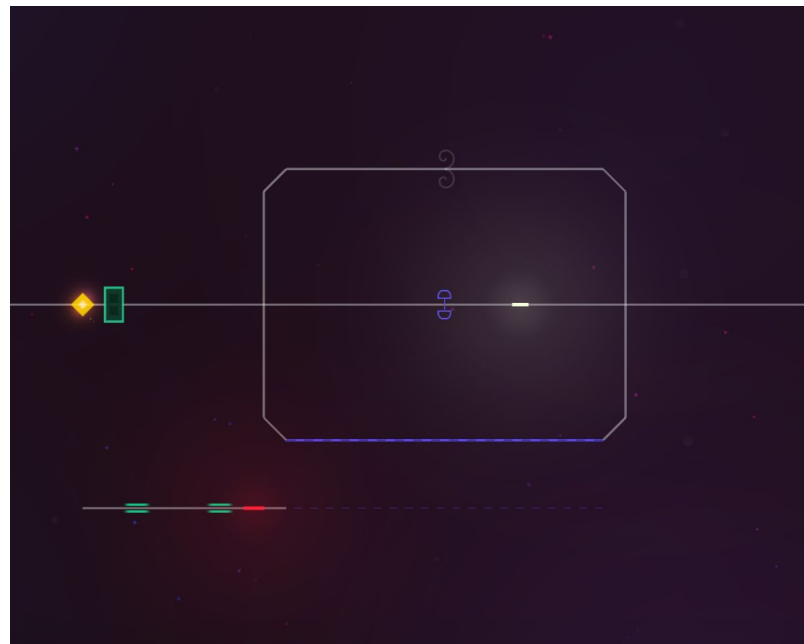


after

1. SIMPLIFY (Symmetry)

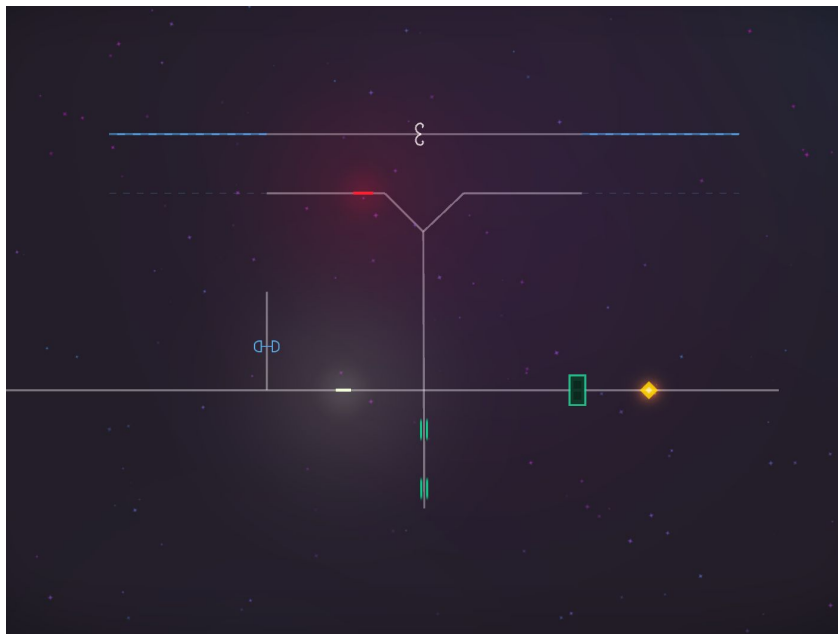


before

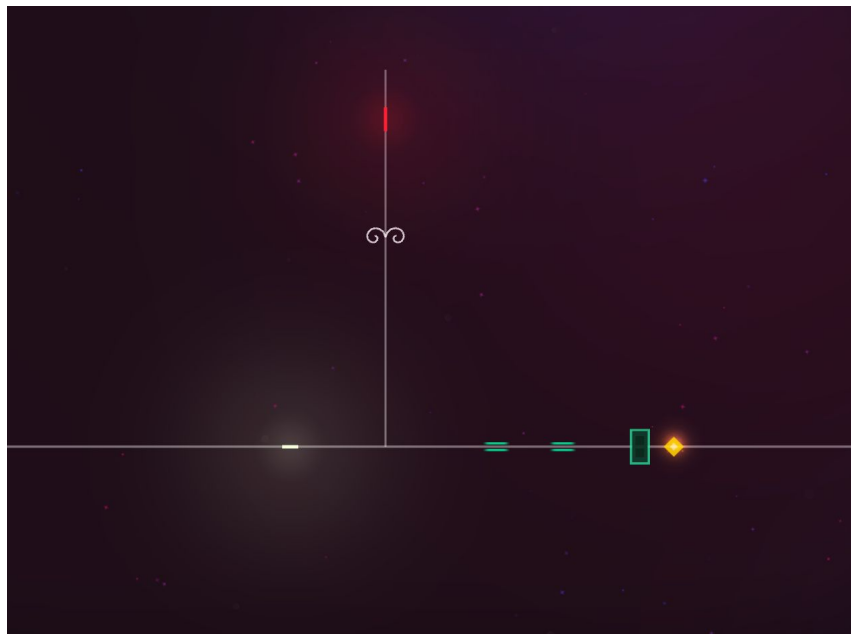


after

1. SIMPLIFY (Remove Extraneous Stuff)



before



after

2. CUT POINTLESS LEVELS

- They're there. Accept it.
- For each level, ask "What's its purpose?"
- Can't answer? Road to mediocrity.
- Possible level purposes:
 - Teaches something
 - Reinforce a skill
 - Has unique identifiable moment (*we'll explore this later*)
 - Palette cleanser, or to break up pacing

side-note on “mediocrity”

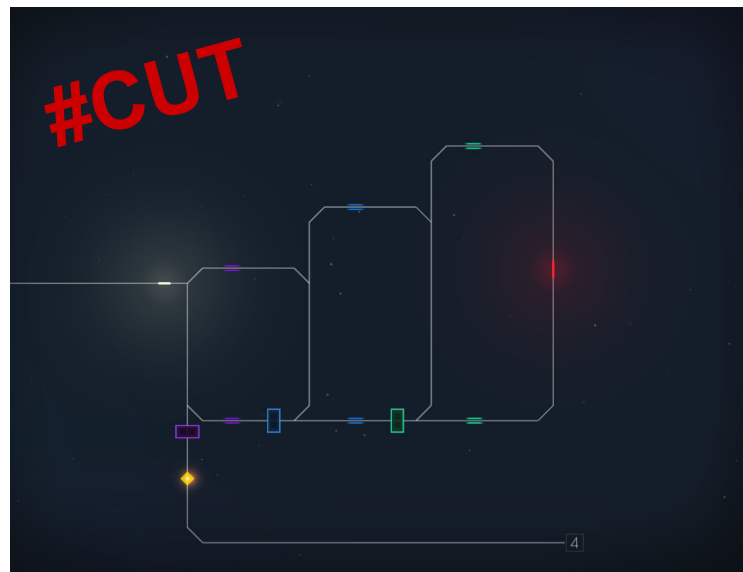
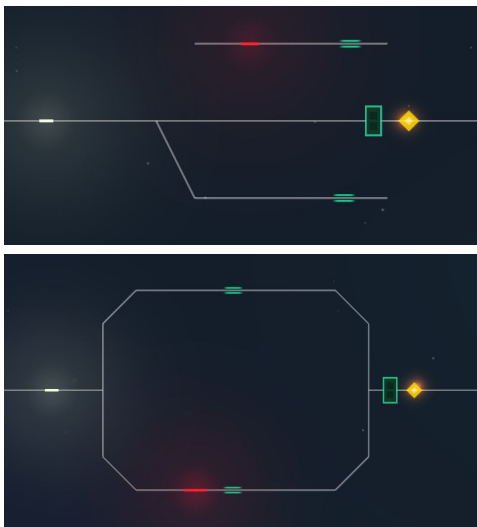
- **You are good enough. Always. I promise.**
- Don't fear not being good enough.
- Make things anyway.

2. CUT POINTLESS LEVELS

- Can be hard to admit a level is pointless
Especially if nothing better to replace it.
- Vet every level. No freeloaders!

EXAMPLE

- Not educational, interesting, fun, or unique
- Preceded by these...



3. YOU'RE THE EXPERT PLAYER

- Complaints about "timing" being "too hard."
eye roll
- Even when they had a *full 1.5-second window*

but...

Knowing solution and not being able to execute
suuucks

3. YOU'RE THE EXPERT PLAYER

- People complaining? Listen to them.
- Play through levels with your feet
- Can't solve it with your feet? Make it easier.

Locksteps



EXAMPLE



Me: “I’ve logged hundreds of hours playing this game...
I know what an average player can handle.”

3. YOU'RE THE EXPERT PLAYER

Lesson:

People *still* complaining?

They're right. Keep changing it.

It's not them; it's you. Serve your audience.

4. KEEP ACTION AND PUZZLES SEPARATE

- For *single-solution* puzzles.
- Action OR Puzzles → (=
- Action AND Puzzles →]=<
- Don't serve **tuna** and **ice cream** on same plate
- Don't serve **action** and **puzzle-solving** in same moment

4. KEEP ACTION AND PUZZLES SEPARATE

- **Action** adds randomness & distractions
- **Puzzle-solving** requires patterns & consistency
- Player needs an idea of what they don't yet know

4. KEEP ACTION AND PUZZLES SEPARATE

- Player should know if level requires action *or* puzzle skills
- ~10% of players complained. Too many!
- Suggestion: Make action levels optional

4. KEEP ACTION AND PUZZLES SEPARATE

don't have levels in this murky area

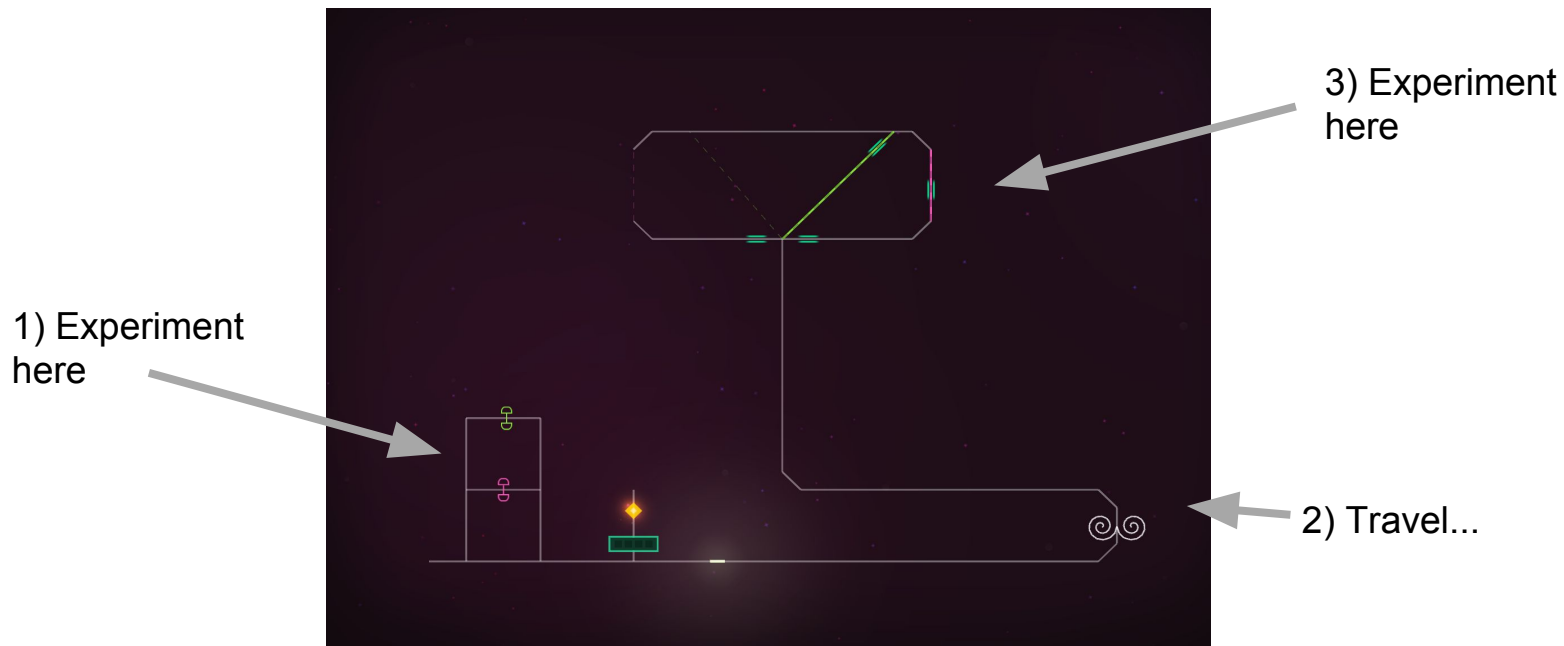


5. MAKE SOLUTION UNAMBIGUOUS

- Red flags:
 - "Was that right?..." (solving didn't feel right)
 - "Oh. I solved it?" (didn't understand what they did)
 - Players attempt impossible action (it's not obviously impossible)
- Damages player's trust in the game



EXAMPLE



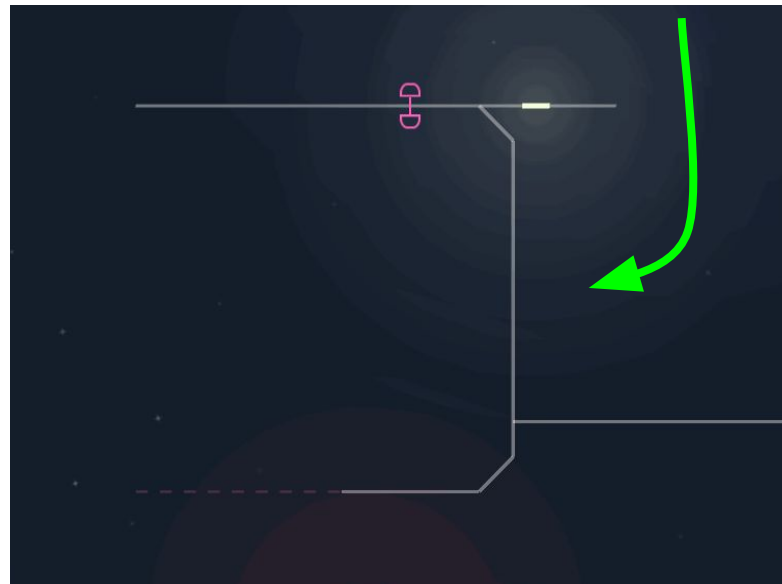
Mostly trial-and-error. No fun.

EXAMPLE

“Can I make that?”



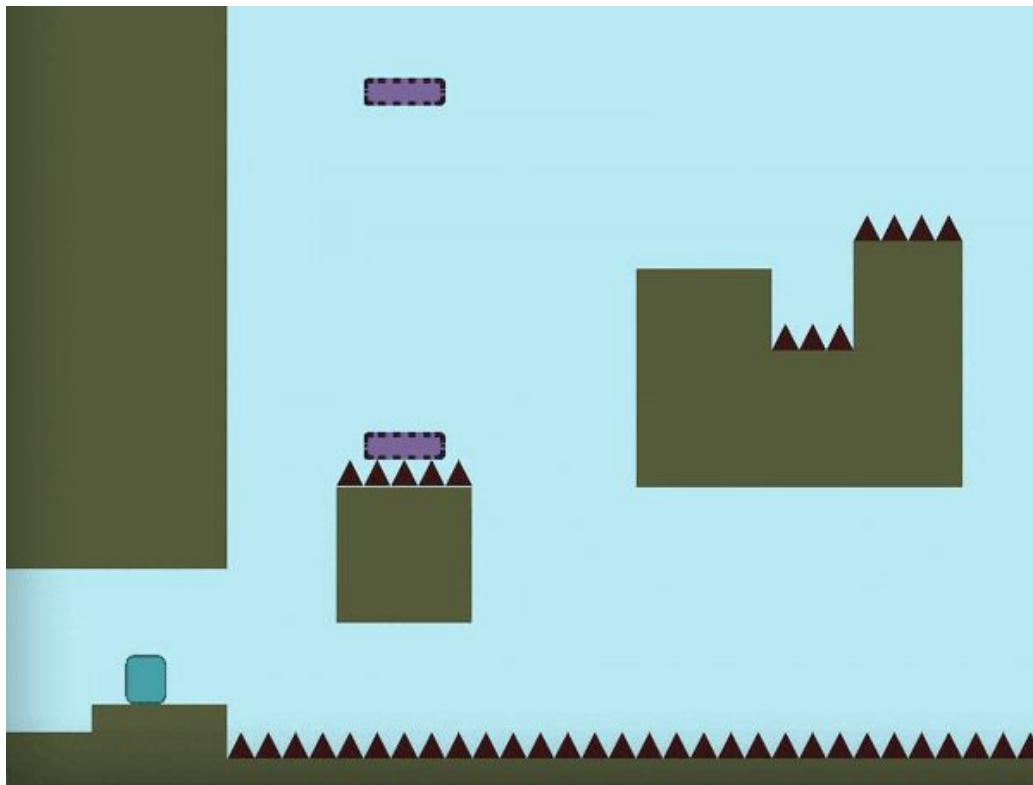
“I definitely can’t make that...”



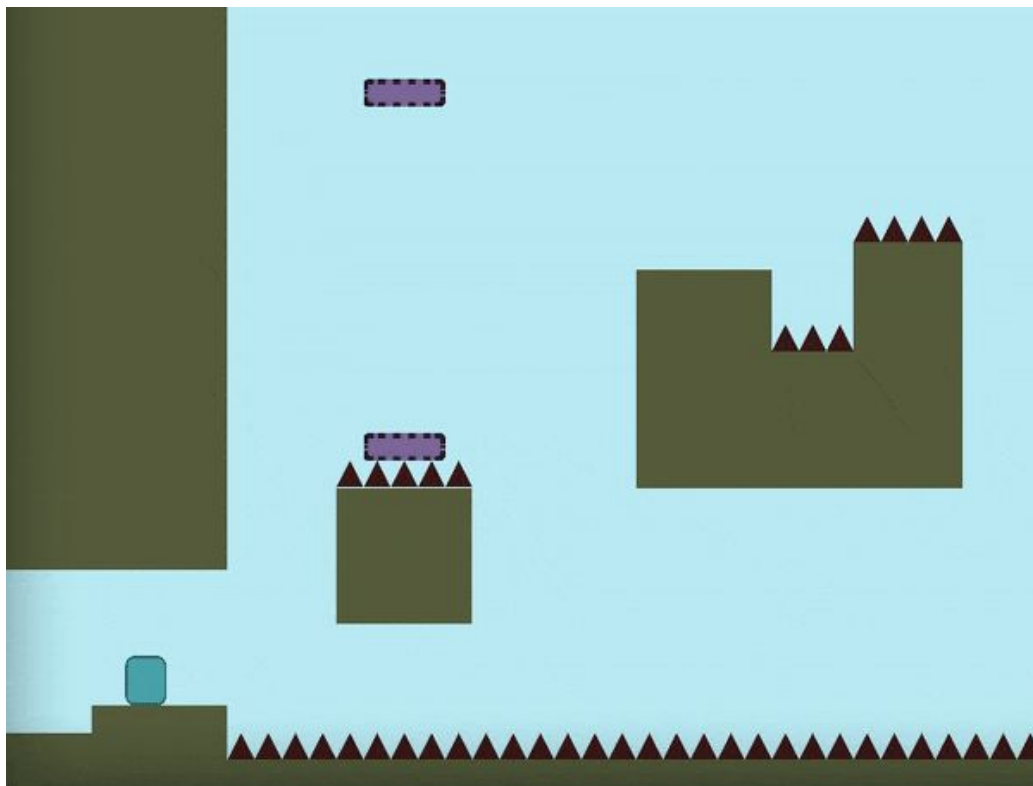
5. MAKE SOLUTION UNAMBIGUOUS

- Move elements farther apart
- Cut extraneous elements
- Platformer? Please. Make. All. Jumps. Obviously. Possible. Or. Impossible.
 - My pet peeve. Move your platforms farther, people!

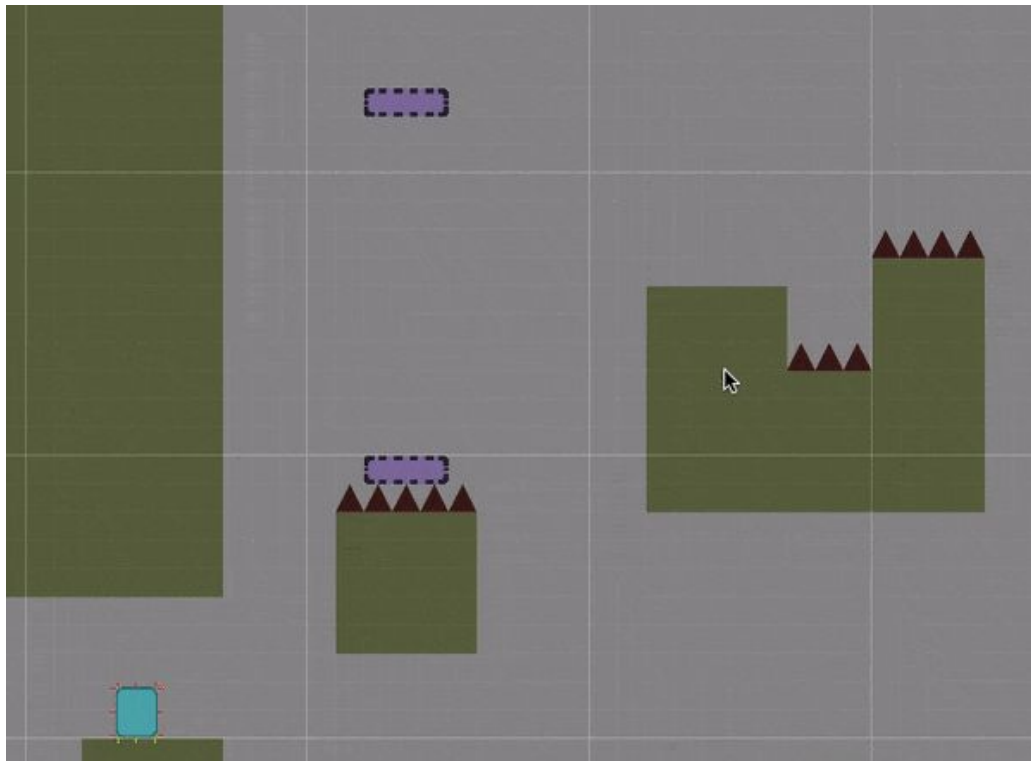
EXAMPLE



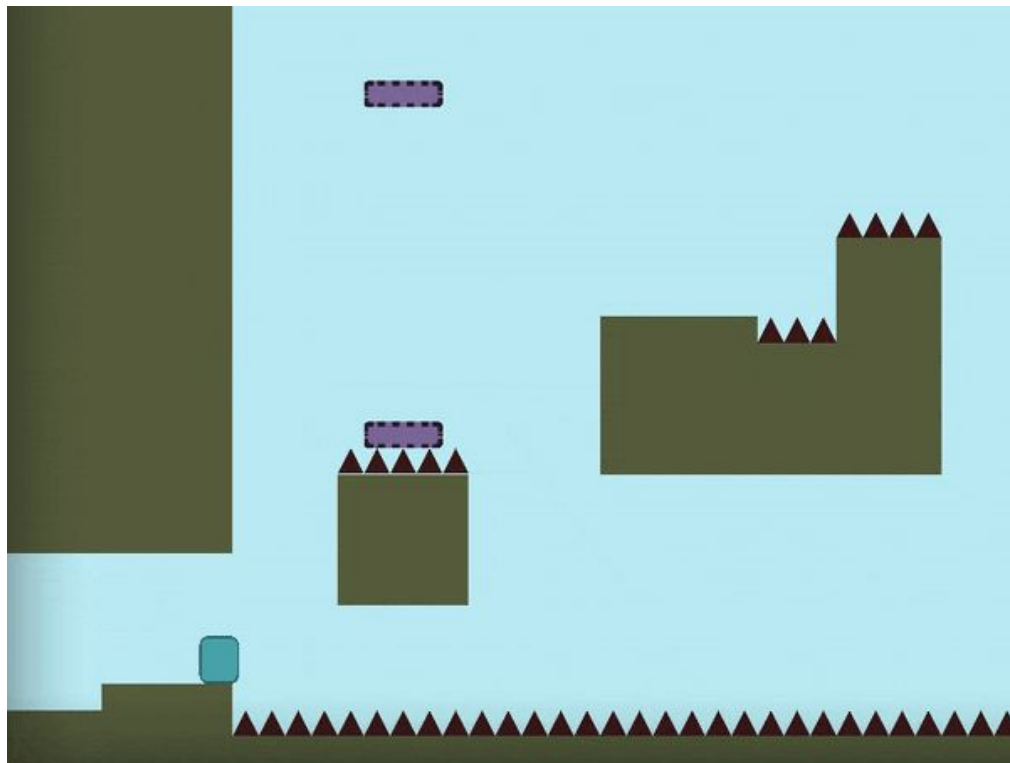
EXAMPLE



EXAMPLE



EXAMPLE



6. PLAYER TRUST

- Players assume no agility needed
- **Break this rule once → EVERY puzzle maybe requires reflexes**
- Linelight players assumed MANY puzzles required reflexes because SOME puzzles required reflexes
- 2 quick-timing puzzles → 15 *perceived* quick-timing puzzles
- Player perception is all that's real

6. PLAYER TRUST

- Player perception is all that's real
- Player perception is all that's real
- Player perception is all that's real
- Player perception is all that's real
- Player perception is all that's real

7. EXHAUSTIVE DESIGN vs. FUN

- Fun = Exhibiting mastery, skill, or wit
- Exhaustive Design = One of every possible thing
- Sometimes incompatible :(

7. EXHAUSTIVE DESIGN vs. FUN

I struggled with this for a year.



Me in 2016.

7. EXHAUSTIVE DESIGN vs. FUN

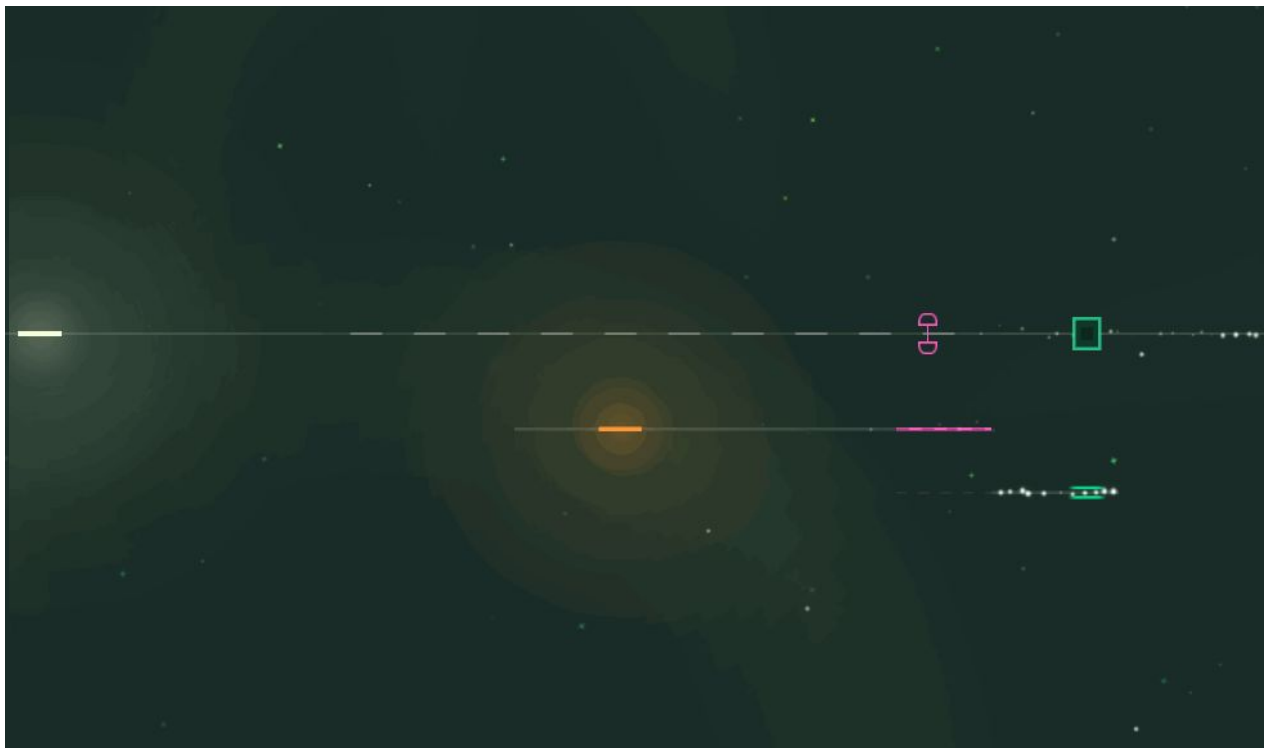
- Braid & The Witness = Exhaustive design to the extreme
- ...Ultimately, I went with player fun
- Fun for designer \neq Fun for player

EXAMPLE

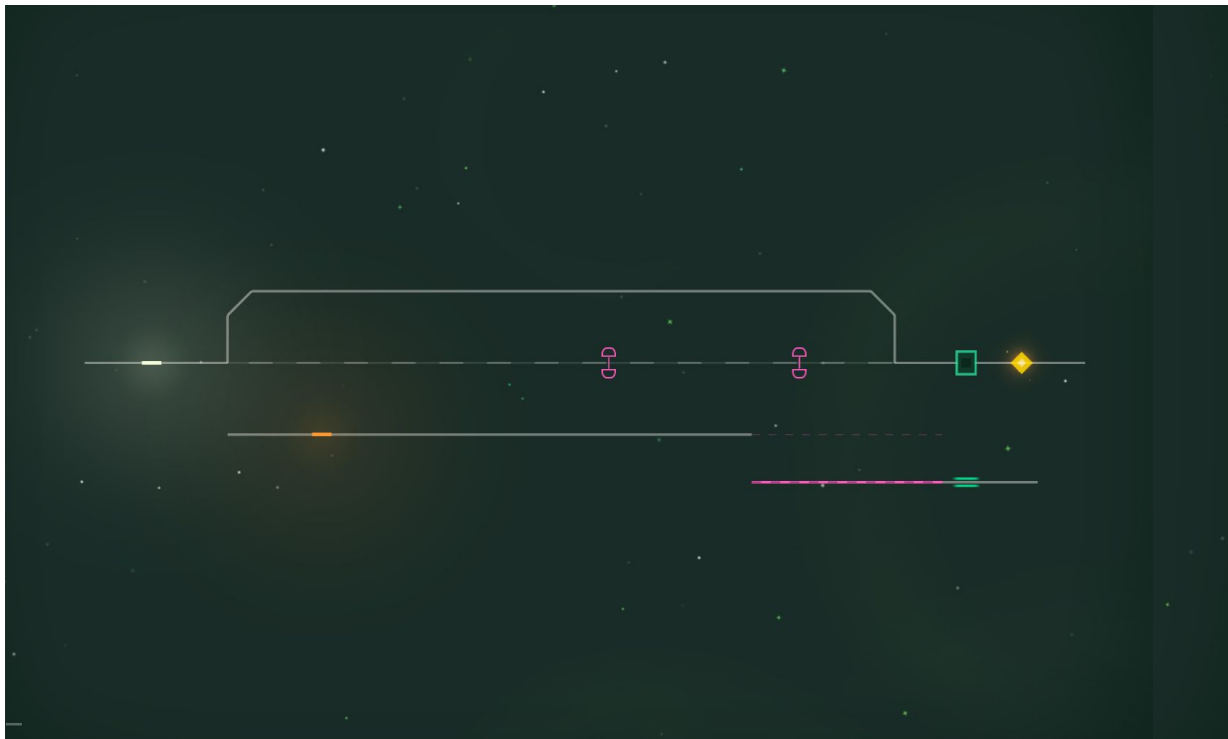


1. oneWayLockFlip1
2. oneWayLockFlip2
3. oneWayLockFlip3
4. oneWayLockFlip4
5. oneWayLockFlipDouble1
6. oneWayLockFlipDouble2
7. oneWayLockFlipDouble3
8. oneWayLockFlipDouble4

EXAMPLE



EXAMPLE

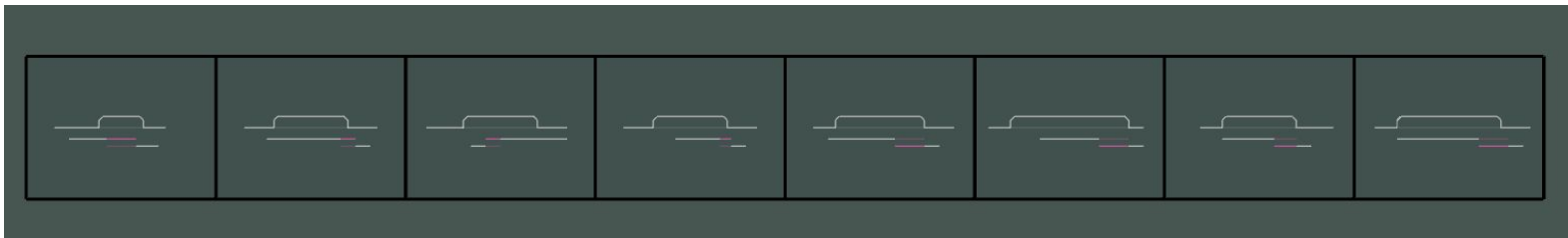


7. EXHAUSTIVE DESIGN vs. FUN (Example)

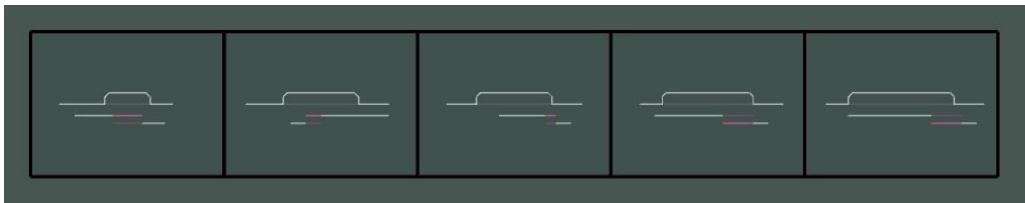
- Players got tired of the same layout
- Even though each level was unique concept!
- Boringness wasn't obvious to me at the time

EXAMPLE

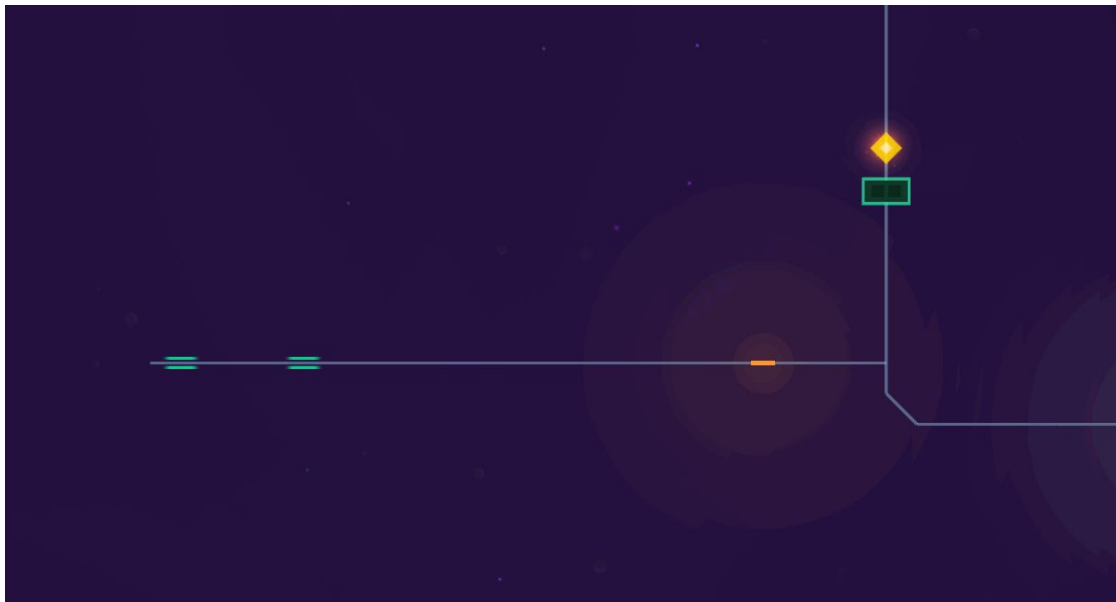
before



after



EXAMPLE



Slow Locksteps: As fun as speed-walking in Times Square

~ Brain Break Time ~

10-second stretch!

1. Working Memory
2. Puzzle Characteristics
3. 7 Lessons from Linelight
4. How I Make Puzzles

HOW I MAKE PUZZLES (Disclaimer)

- These are *my* methods
- I like games with **handcrafted** puzzles
- There're always more skills out there for all of us to learn!

“*Where* did you come up with this level??”

I didn't. The *mechanics* did.

MY STRATEGY

1. Create mechanics
2. Use mechanics together
3. Boom, puzzles!

HOW DO WE GET THE MECHANICS?

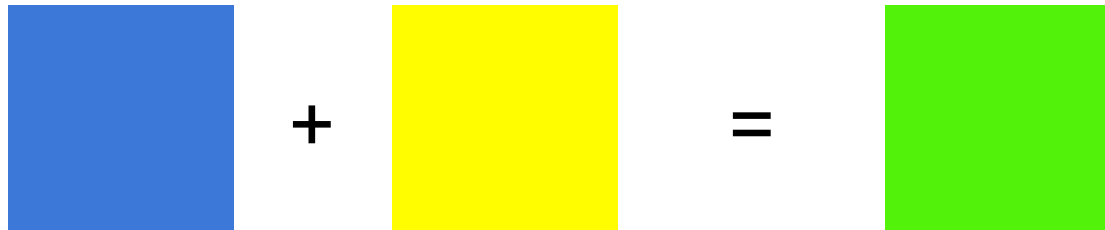
- Experience, intuition, and luck
- Create mechanics that...
 - are diverse
 - could interact in many ways
 - excite you
- Cut the duds. Don't force it.
- +Experimenting → +Great stuff!
- Have fun. =)



not diverse. meh.



diverse! woohoo!



MECHANICS

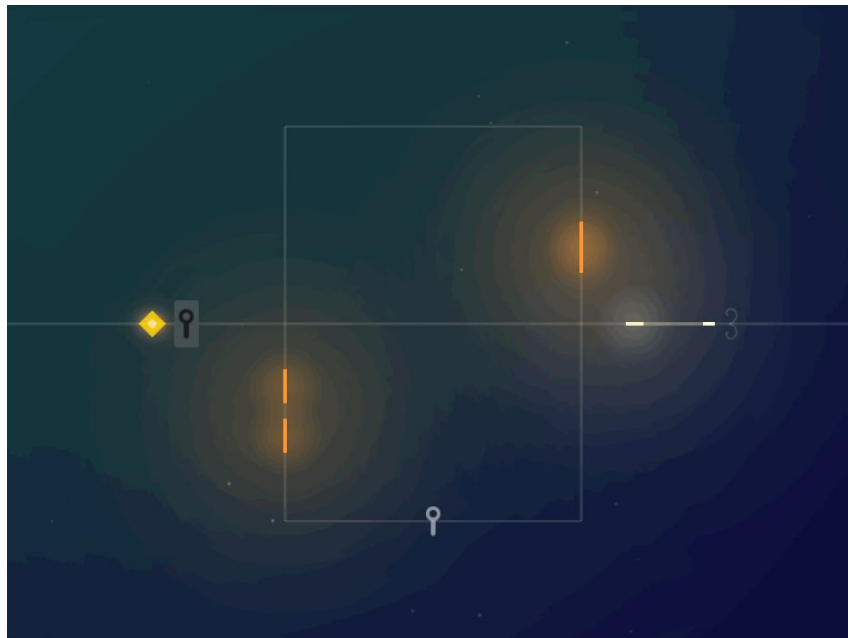
- Goal: Have **few mechanics** that yield **many puzzles**
- Start with a few mechanics and try to exhaust the possibilities
- Being surprised by your mechanics = good sign!

SOLUTION SENTENCES

- Linelight's best puzzles revolve around **one moment**
- My goal: Make simplest level with that concept as the solution

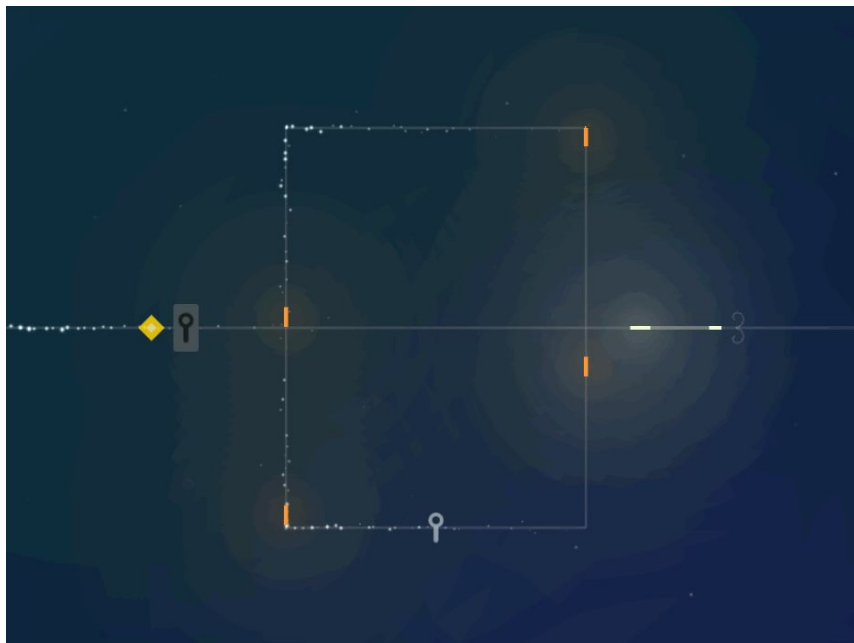
SOLUTION SENTENCES (EXAMPLES)

“Use length to get Locksteps moving clockwise”



SOLUTION SENTENCES (EXAMPLES)

“Use length to get Locksteps moving clockwise”



Solution Sentences Examples TODO

TODO: Gif

- Have to lose all length to fit inside circle of locksteps
-

SOLUTION SENTENCES

- Elegance is inevitable
 - Simplicity is intrinsic to elegance.
- Diversity in layout WILL happen naturally
- Each sentence is like a magic trick's secret

SOLUTION SENTENCES (EXAMPLES)

- Fast enemy switch places with slow enemy
- Locksteps + length
- Have to wait as echo for an enemy to pass over so echo doesn't die
- Echo with length block Flipper
- Have your Echo turn off a CartEchoSpawnerSwitch :D
- Use ONE repeater spawner to get TWO Echoes :D
- You and your Echo pull an Obedient double fast :)
- Enemy blocking an intersection
- Have to enter level tail-first or lockstep will break out of its coverup cage and eat ya
- Flip MovingStreet with Lengther to make your Echo *take* that length
- Use Echo to pull an Obedient into the Spawner
- Fill up more Lengther so Echo can block *two* enemies
- Bring Obedient from a secret area somewhere else! :D
- Block Obedient with your Echo's length
- Give Obedient a MovingStreet for some wind-up room
- Lockstep on self-flipping street. You have to hop on to get it off

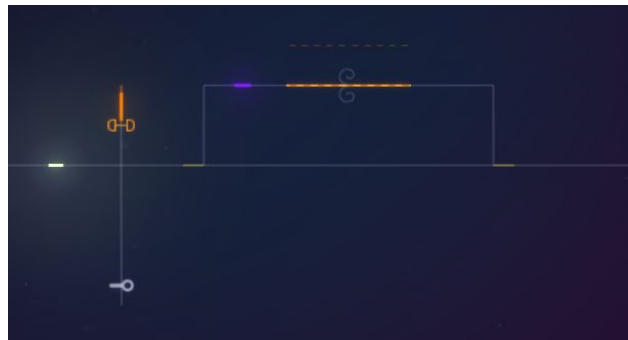
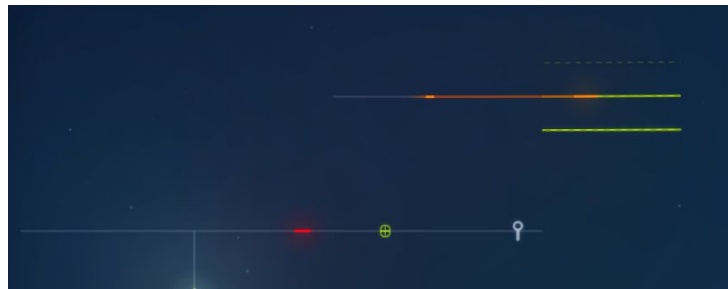
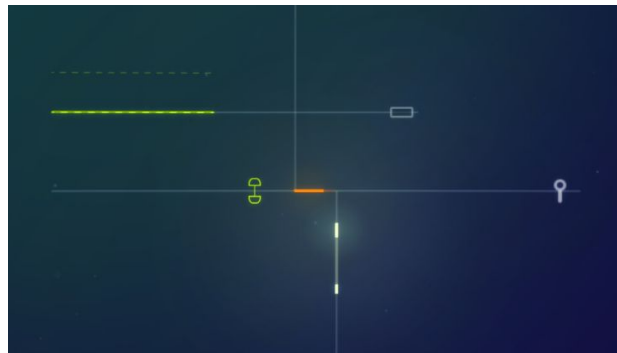
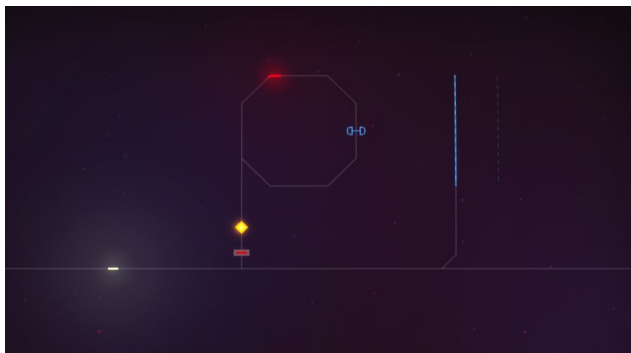
SOLUTION SENTENCES (EXAMPLES)

- "Dump" all your repeating echos into an enemy-only street
- Send your echo off somewhere you can't-- and never will-- see
- Use echo from previous level
- Exclusive MovingStreet
- Bring in Obedient Enemy from another level :)
- Obedients that push!
- Have to lose all length to fit inside circle of locksteps! Then, have to lose some length to fit in another circle :)
- Give lockstep length so it gets out of sync with another lockstep
- Must flip enemy with length so that its head doesn't hit a flipper
- OneWayStreets with coverups on them: It's about *when* you enter the One-Ways.
- Need to clock-block Enemy to get it going in right length dir :)
- Leave some length in a lengthier: give enemy some and you some
- Echo of an Echo!!
- Block your Echo with your length
- Block Flipper to stop Echo
- Enemy kills your Echo

SOLUTION SENTENCES “CHEAT”

- **Echo** with length block flipper
 - **Player** with length block flipper
 - **Enemy** with length block flipper
 - **Lockstep** with length block flipper
 - **Obedient** with length block flipper
-
- Each level still unique challenge! (yah, way!)
 - They're scattered across whole game
 - Small changes in sentence = significant changes in layout

“_____ with length block flipper”



HOW DO YOU GET THOSE SENTENCES?

- Force mechanics to play together and see what happens
- See something (almost) happen-- idea!
- Ask "What if _____?"
- **Don't try too hard. Let the system do the work for you.**
- Great puzzles occur naturally.

HOW DO YOU GET THOSE SENTENCES?

	Enemy	Flipper	OneWay	Padkey	Coverup	Lockstep	Exclusive	Length	Lengther	Echo	Chase	Obedient		
12 Enemy	0												0	Don't explore
6 Flipper	0	0											1	Perhaps explore
15 OneWay	1	1	1										2	Definitely explore!!
1 Padkey	0	0	0	0										
9 Coverup	1	1	1	0	0									
10 Lockstep	1	1	0	0	0	0								
12 Exclusive	1	0	1	0	1	0	1							
19 Length	2	1	2	0	2	2	2	1						
14 Lengther	1	-	2	0	1	2	2	-	-					
17 Echo	2	1	2	1	1	2	0	2	2	0				
16 Chase	1	1	1	0	1	2	1	2	2	2	1			
11 Obedient	2	0	2	0	0	0	0	2	2	2	1	0		

Systematic approach. Often works well!

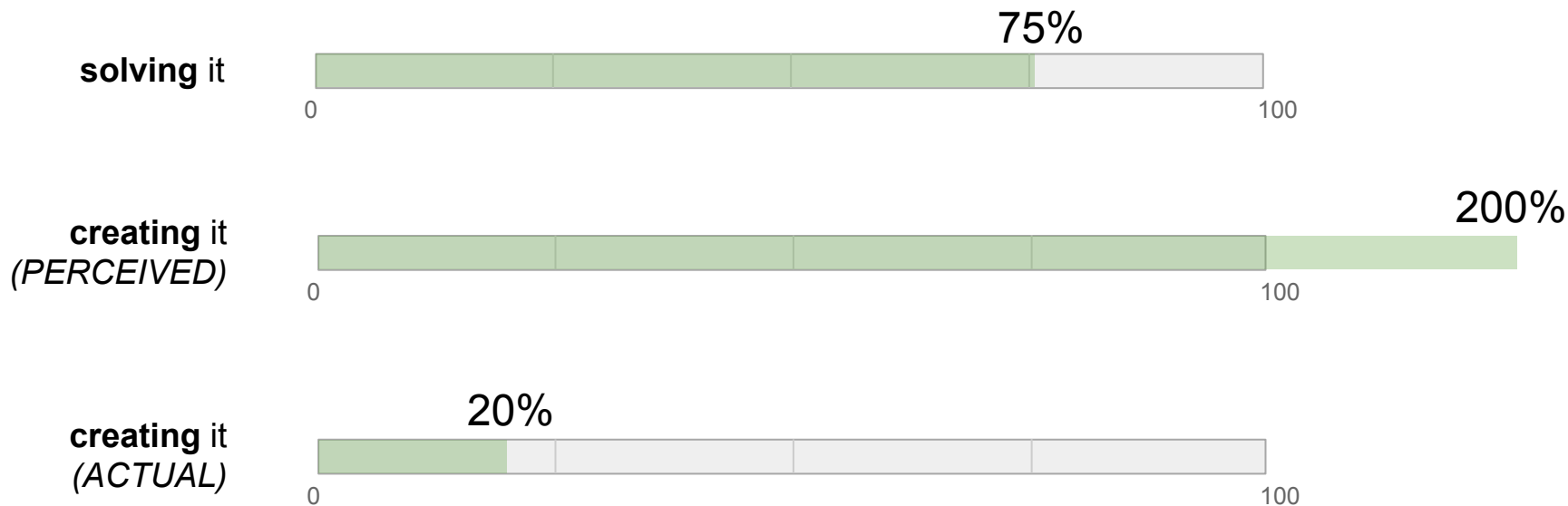
Story time.

FOR HANDMADE PUZZLE GAMES...

- Nobody's brain is big enough
- The *system* the designer created "thought" the puzzles up
- Designers don't give birth to the puzzles;
Designers *deliver* the puzzles.



MAKING PUZZLES & WORKING MEMORY



RECAP

RECAP

WORKING MEMORY

- Human RAM
- Frees up with experience



RECAP

HANDCRAFTED vs. PROCEDURAL

finite # puzzles
novel solutions
fresh, varied

∞ puzzles
~~novel solutions~~
predictable experience



		7	9	6	2	4		
9				1				2
	1		8	5	3		6	
5			4	7	9			1
				8				
4			3	2	1			7
	9		2	4	8		5	
6				3				8
		8	6	9	5	1		

RECAP

NOISE

- Occupies Working Memory, but doesn't contribute to puzzle
- Will pollute your game if not used mindfully
- Practice finding/eliminating it
- OK in moderation

RECAP

METHODS to CREATE PUZZLES

- Combine mechanics systematically/organically
- Good mechanics → Great puzzles
- System births puzzles; designer *delivers* them
- Have fuuuuun!!

THINKING,
FAST AND SLOW



DANIEL
KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS

Brett Taylor
brett@mydogzorro.com
@batzerk

linelightgame.com

These slides:

bit.ly/gdc2019puzzletalk

