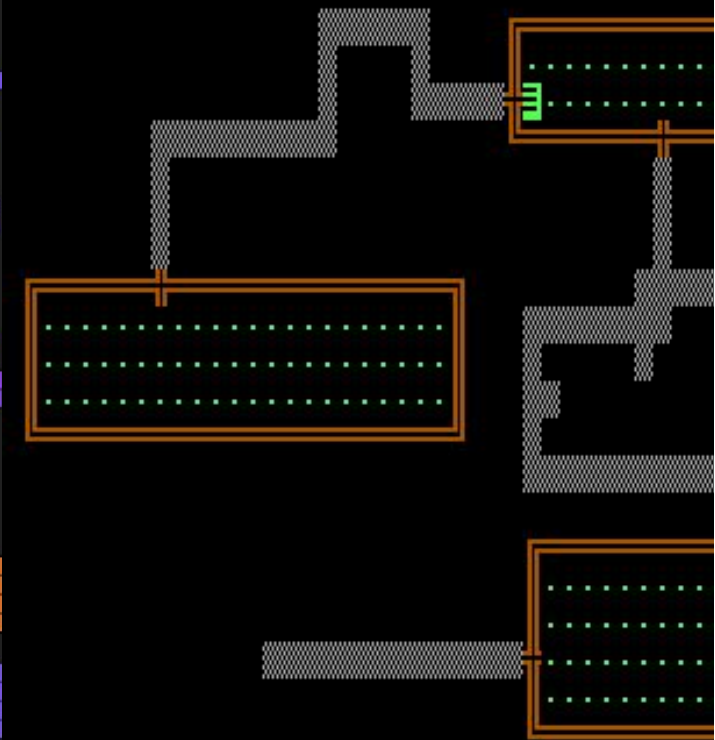
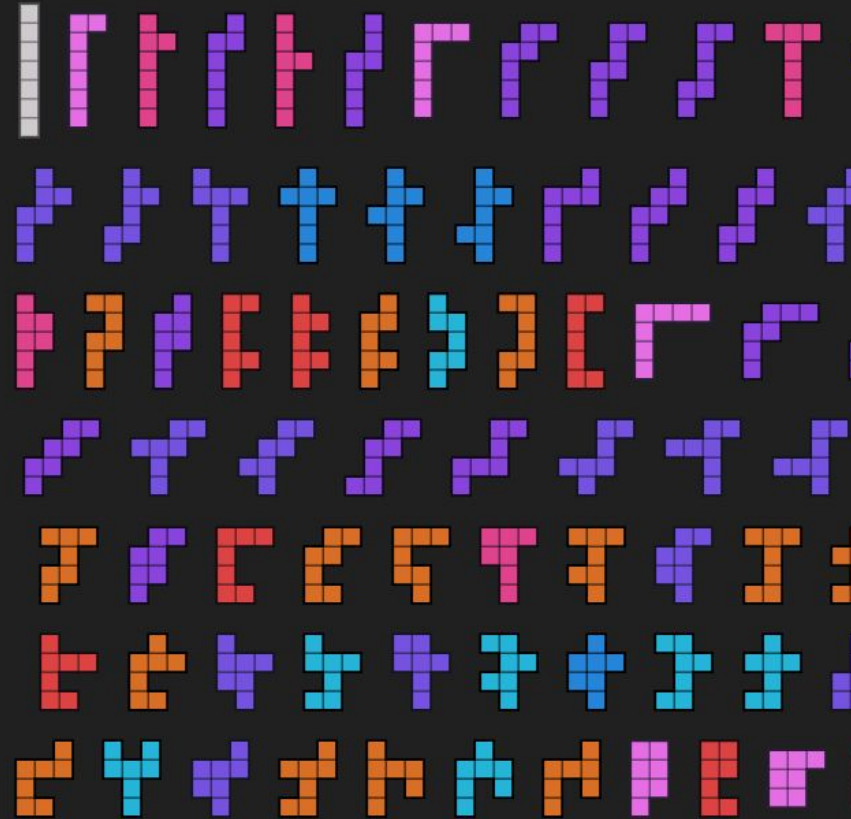


color	size	rule
yellow	100"	105
yellow, red, blue		

LeWitt 1985

heptominoes ( $n=7$ )



Level:1 Hits:12(12) Str:16(16)

# Sol LeWitt, Combinatorial Enumeration, and Rogue

Sol LeWitt, "All One, Two, & Three-Part Combinations of Three Transparent Colors", 1985

Labyrinth of Polyominoes, heptominoes

Rogue (PC port)

Mark Gitter (novice Artificer)  
Roguelike Celebration  
October, 2020

# Sol LeWitt



20th Century  
Conceptual Artist,  
1928-2007

- wall drawings
- minimalism
- lots of cubes!

← “Incomplete Open  
Cubes”, 1974

Photo by the speaker, at San  
Francisco Museum of Modern  
Art, 2018



# Wall drawings

Sol LeWitt, "Wall Drawing 289", 1976

A 6-inch (15 cm) grid covering each of the four black walls. White lines to points on the grids. Fourth wall: twenty-four lines from the center, twelve lines from the midpoint of each of the sides, twelve lines from each corner.

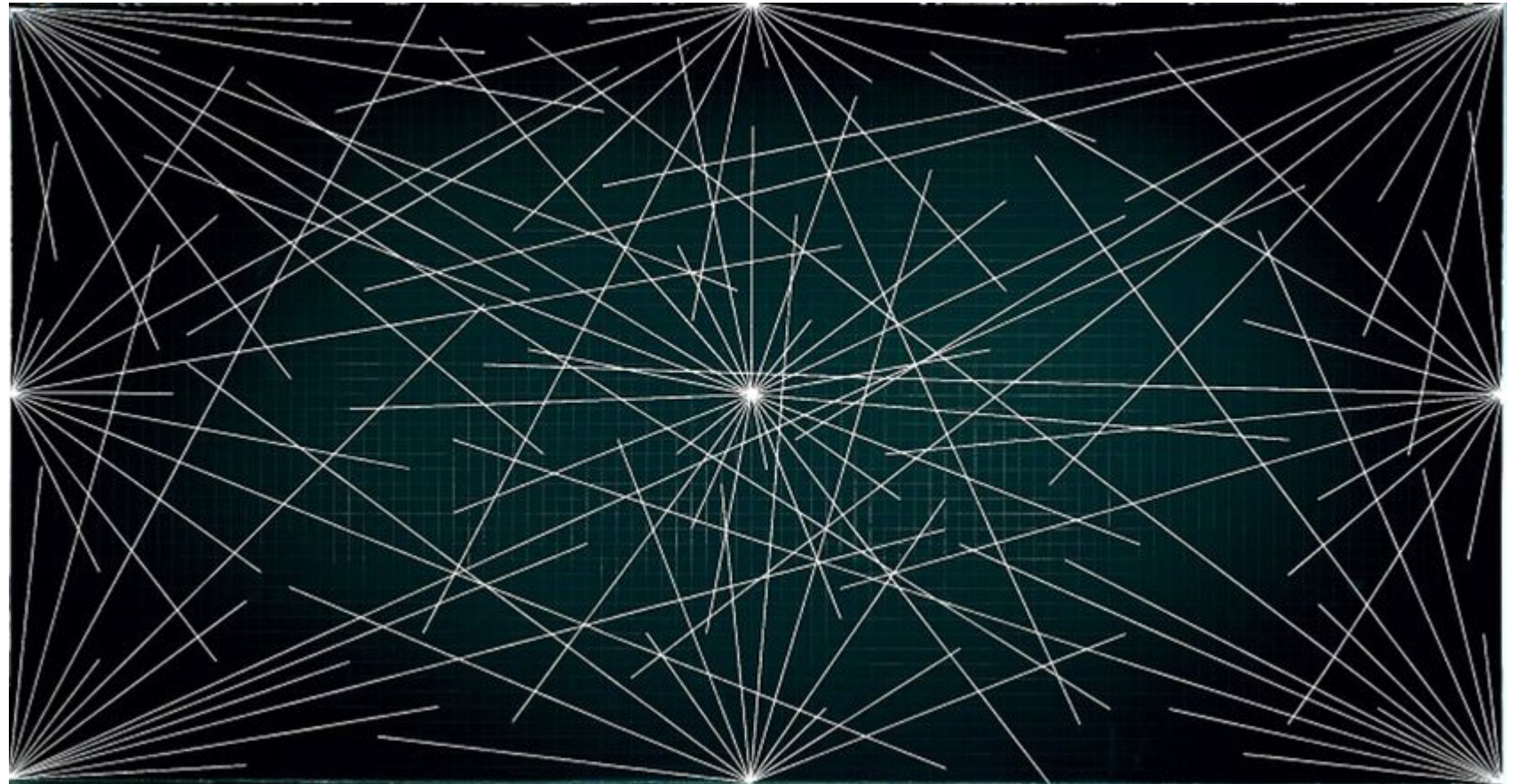


Image from <https://whitney.org/collection/works/25530>

# Wall drawings (2)

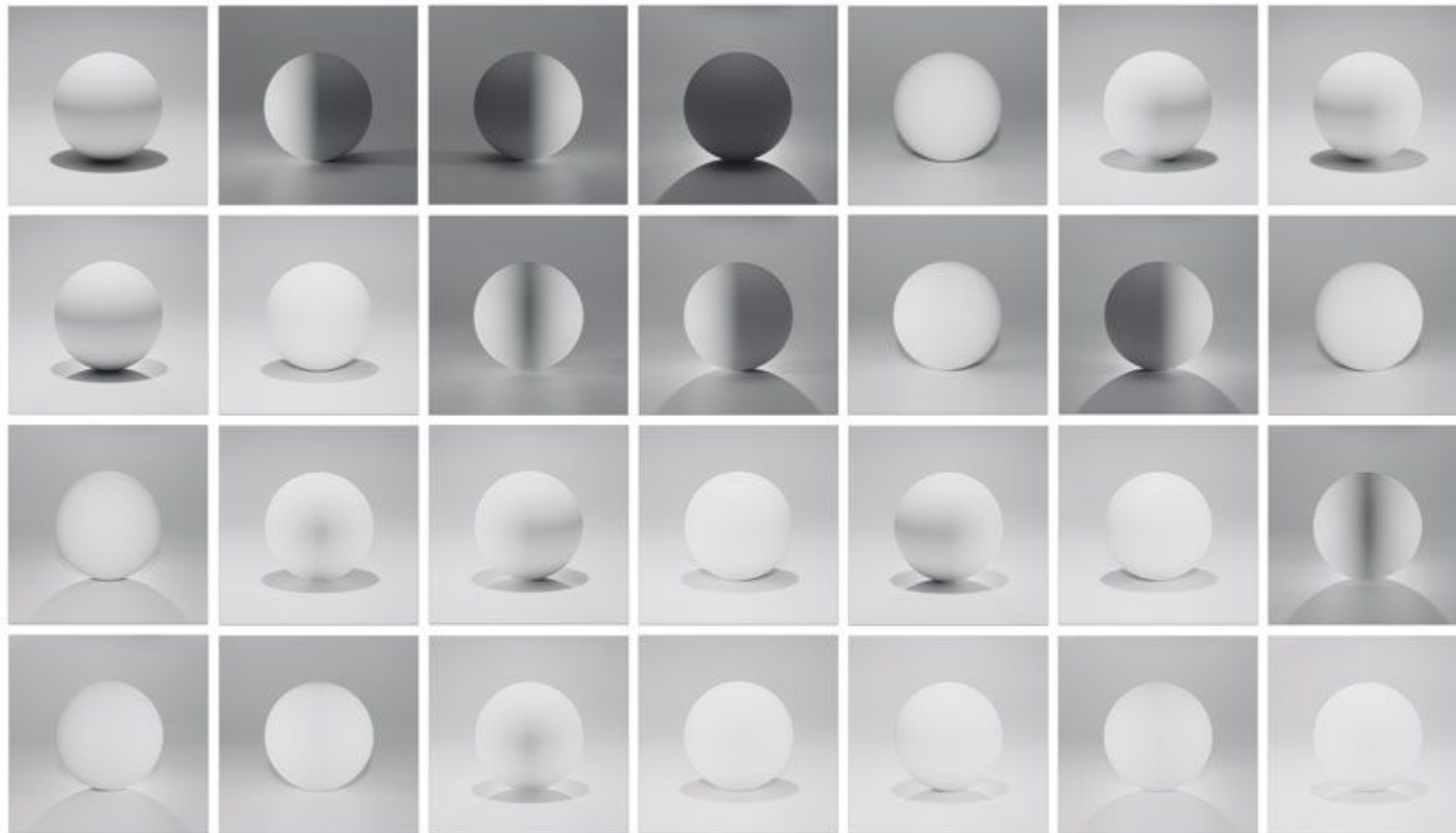
Wall drawings are generative art! They are given as **instructions** rather than as an image. (\*)

*“The only permanent, concrete form of Wall Drawing #289 is a set of typewritten guidelines and a certificate of authenticity signed by the artist... The exact angle and length of the lines are determined by those who draw them, and the work’s precise configuration and scale may be adapted to fit a variety of architectural contexts. Consequently, the wall drawing can differ significantly with each realization.”* – Whitney Museum of American Art,

<https://whitney.org/collection/works/25530>

(\*) this separation is ruined a bit by having to hire a draftsman from the LeWitt Studio to execute an “official” drawing.

# Serial Art



Sol LeWitt, "A sphere lit from the top, four sides, and all their combinations", 2004

Image from  
[sollewittprints.org](http://sollewittprints.org)

# Serial Art (2)

This takes a further step backwards into abstraction. The artwork is a description of a condition or axiom or law, which is then carried out to produce a tangible result. The artwork typically juxtaposes all of these results together as a finished work.

*"The serial artist does not attempt to produce a beautiful or mysterious object but functions merely as a clerk cataloguing the results of his premise."* – Sol LeWitt

(I disagree, I think these are beautiful and mysterious – Mark)

# Serial Art (3)

*“Three basic operating assumptions separate serially ordered works from multiple variants:*

*1. The derivation of the terms or interior divisions of the work is by means of a numerical or otherwise systematically predetermined process (permutation, progression, rotation, reversal).*

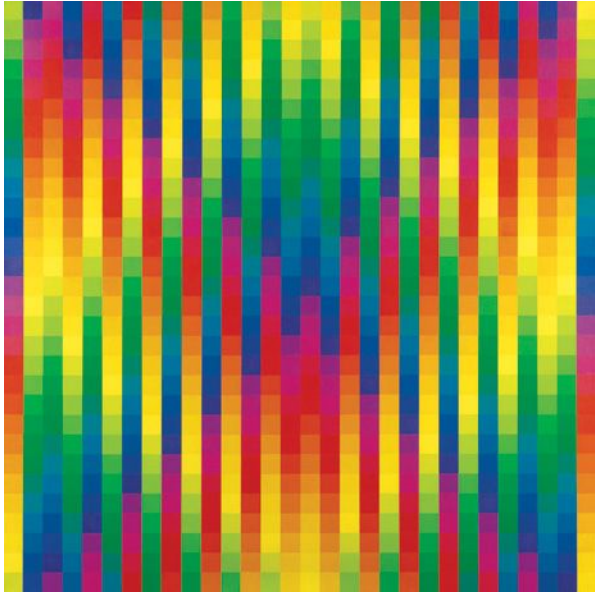
*2. The order takes precedence over the execution.*

*3. The completed work is fundamentally parsimonious and systematically self exhausting.”*

– Mel Bochner, “The Serial Attitude”, Artforum, December 1967



Josef Albers, "Homage to the Square". Photo by Selena N. B. H. (cc-by-2.0) at the Tate Modern.

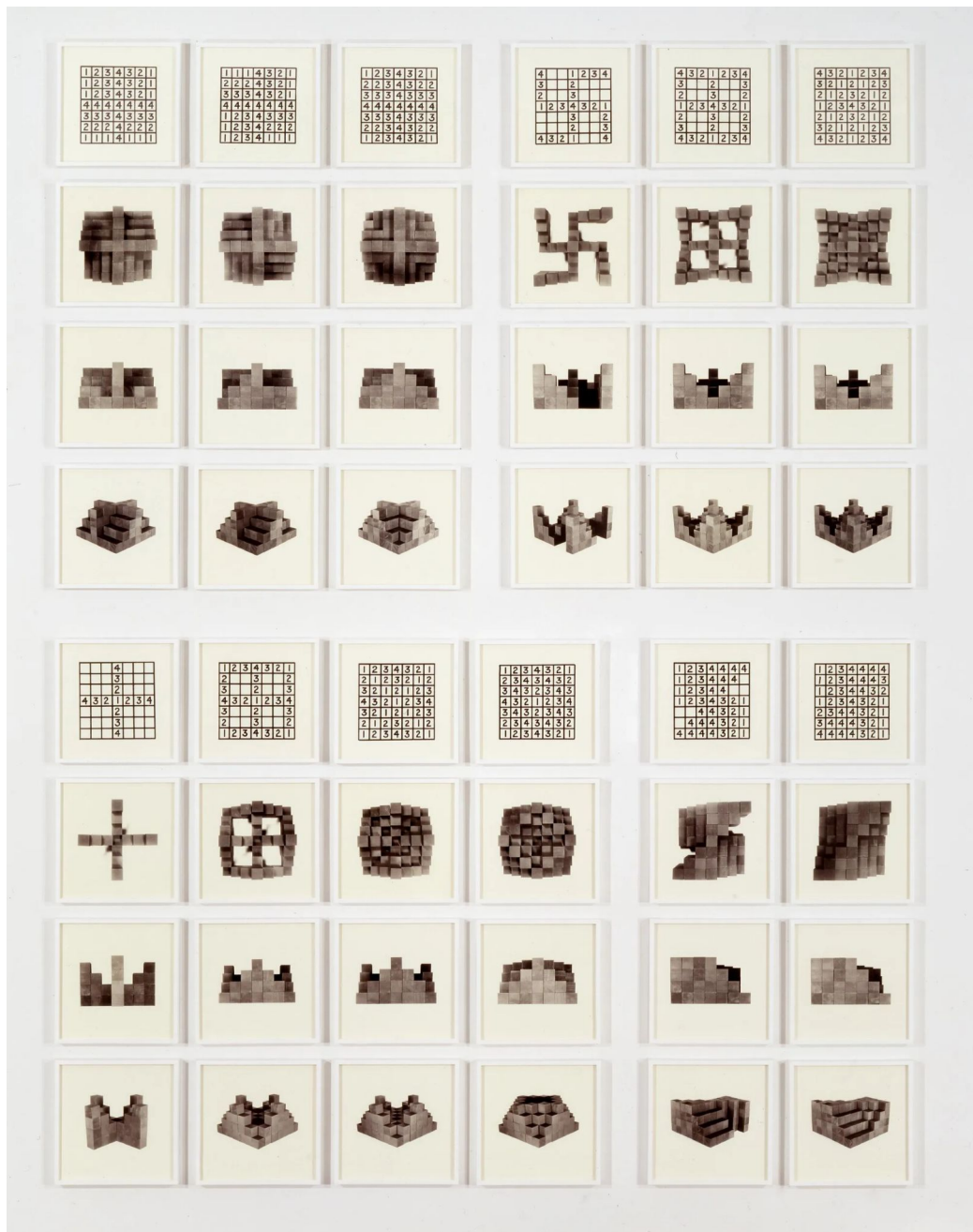


Richard Paul Lohse,  
"Dreissig vertikale  
systematische  
Farbreihen in gelber  
Rautenform",  
1943/1970  
from  
[https://www.lohse.ch/works\\_paintings3\\_e.html](https://www.lohse.ch/works_paintings3_e.html)



Yayoi Kusama, "Infinity  
Mirrored Room -  
Aftermath of  
Obliteration of  
Eternity", 2009  
from  
<https://hirshhorn.si.edu/kusama/infinity-rooms/#aftermath>





Mel Bochner, “36 Photographs and 12 Diagrams”, 1966,  
<https://www.glenstone.org/artworks/36-photographs-and-12-diagrams>

Hanne Darboven, (Süd-) Koreanischer Kalender, 1991  
<https://spruethmagers.com/artists/hanne-darboven/>

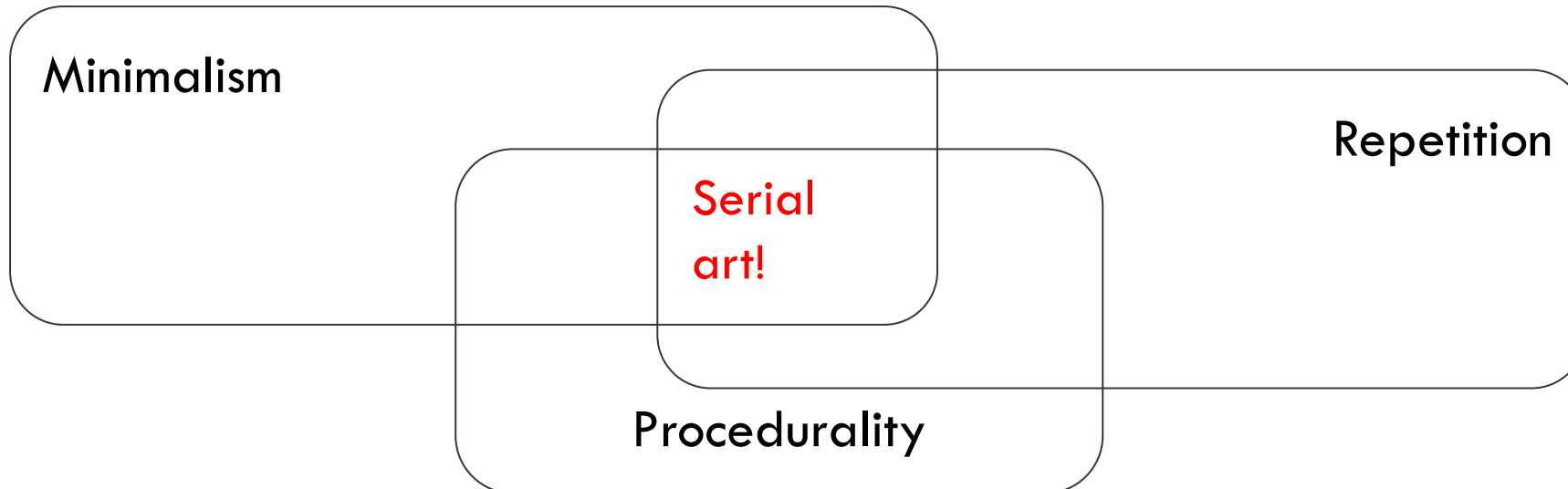


# Other Serial Art

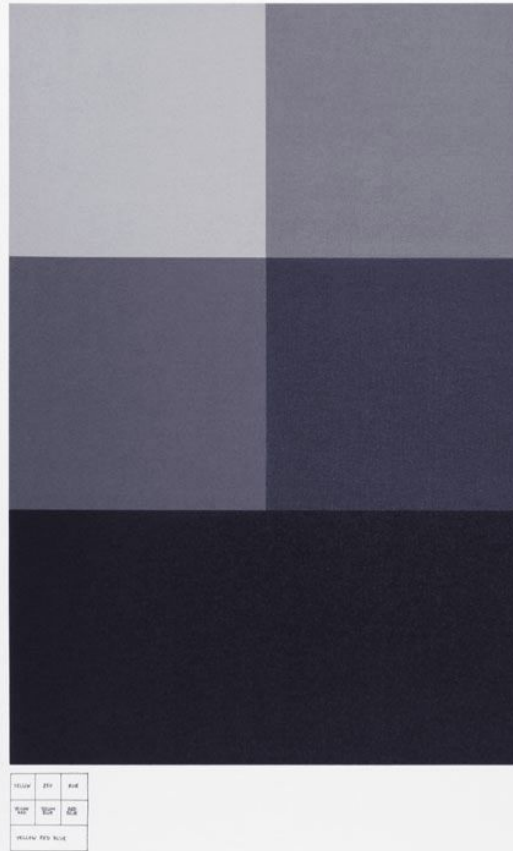
- “Change Ringing” – playing all permutations of a set of bells

# The idea becomes a machine that makes the art.

– Sol LeWitt, “Paragraphs on Conceptual Art”, 1967

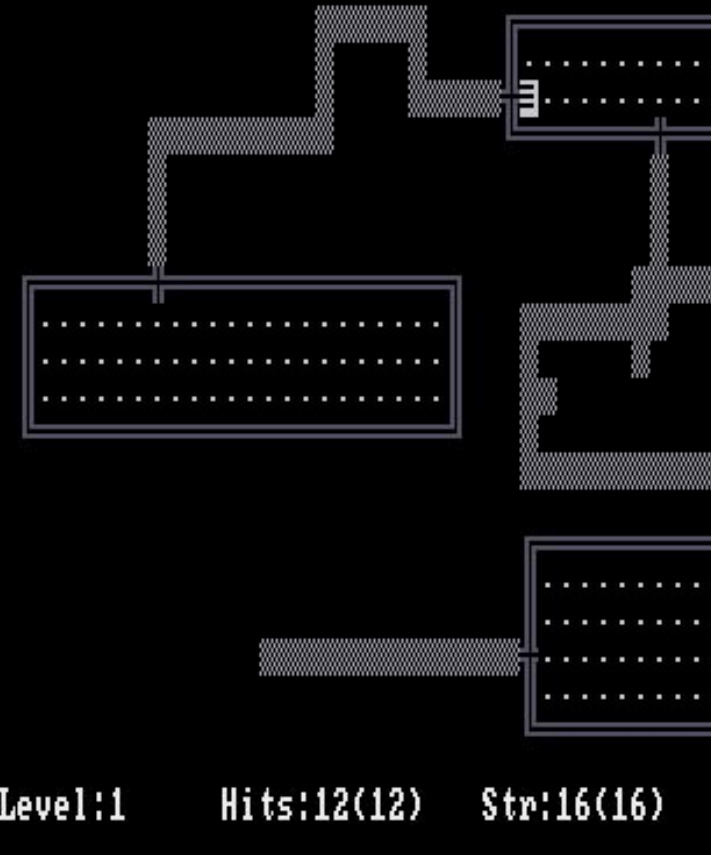






heptominoes ( $n = 7$ )

108



# Enumerating All the Things

# Are these serial art?

- Labyrinth of Polyominoes: <https://minos.tesseract.li/>
- Genji-ko: <https://www.oranlooney.com/post/genji-ko/>
- Damien Riehl and Noah Rubin, <https://allthemusic.info/faqs/>
- Every 5x5 nonogram: <https://pixelogic.app/every-5x5-nonogram>
- Every UUID: <https://eieio.games/blog/writing-down-every-uuid/>
- Every small category: [Smallcats.info](https://smallcats.info)

# The Labyrinth of Polyominoes

By Roguelike  
Celebration  
2024 speaker  
@tesseractis!

All polyominoes  
(up to  $n=8$ ) here  
arranged by  
“genealogy”.

<https://minos.tessera.li/>





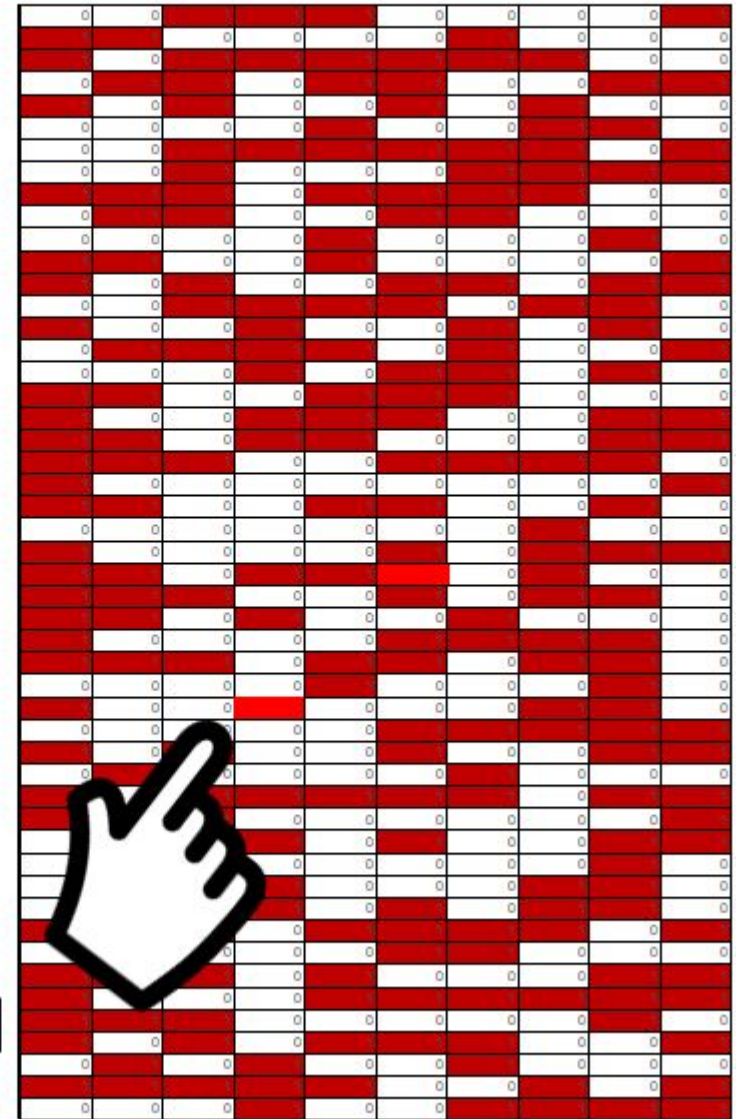
# All the Music

Damien Riehl and Noah Rubin,  
<https://allthemusic.info/>

- Generate every possible melody, up to 10 or 12 notes longs.
- Copyright them all.
- Stop “substantially similar” music lawsuits forever?
- Argument: either Riehl has copyrighted every unused melody, and they are free to use, or melodies aren’t copyrightable.



Finite Melodic Dataset

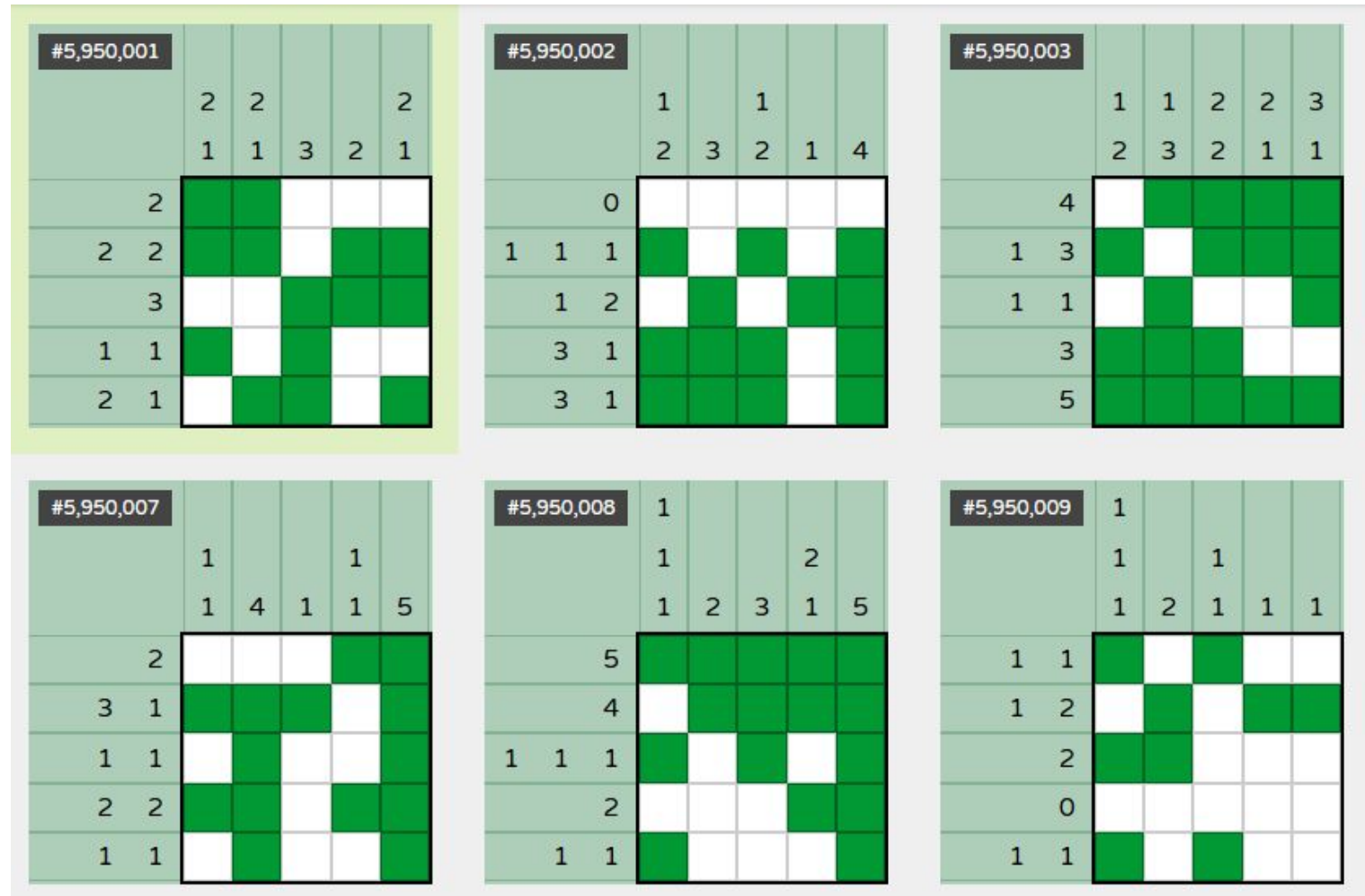


P.S. in the age of AI you should be able to find the unfortunate flaw in this argument.

# Every 5x5 Nonogram

Okayest Studio,  
<https://pixellogic.app/every-5x5-nonogram>

24,976,511  
possibilities, all solved  
by humans via a  
collaborative effort.



# SmallCategories

Ben Spitz,

<https://smallcats.info/>

("in beta")

A "category" is a mathematical object consisting of dots with arrows in between them, obeying a couple simple rules.

164,975 categories each with properties listed, the "multiplication table" of composition, and a visualization.

## Quick Reference

**Web ID** 6ca55ed1-f425-4004-a199-29def9560838

**Morphisms** 9

**Objects** 3

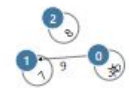
**Index** 59248

**Name** N/A

**Description** N/A

## Visualization

This feature is still janky! Please excuse the mess.



Morphisms 0 through 2 are identities, shown as objects in this visualization.

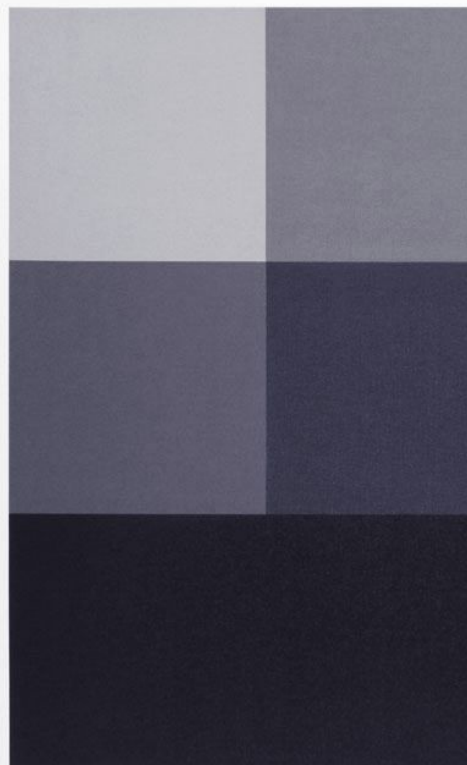
## Table

row ◦ col	0	1	2	3	4	5	6	7	8
0	0	/	/	3	4	5	/	/	/
1	/	1	/	/	/	/	6	7	/
2	/	/	2	/	/	/	/	/	8
3	3	/	/	3	4	5	/	/	/
4	4	/	/	3	4	5	/	/	/
5	5	/	/	3	4	5	/	/	/
6	6	/	/	6	6	6	/	/	/
7	/	7	/	/	/	/	6	7	/
8	/	/	8	/	/	/	/	/	2

## Facts

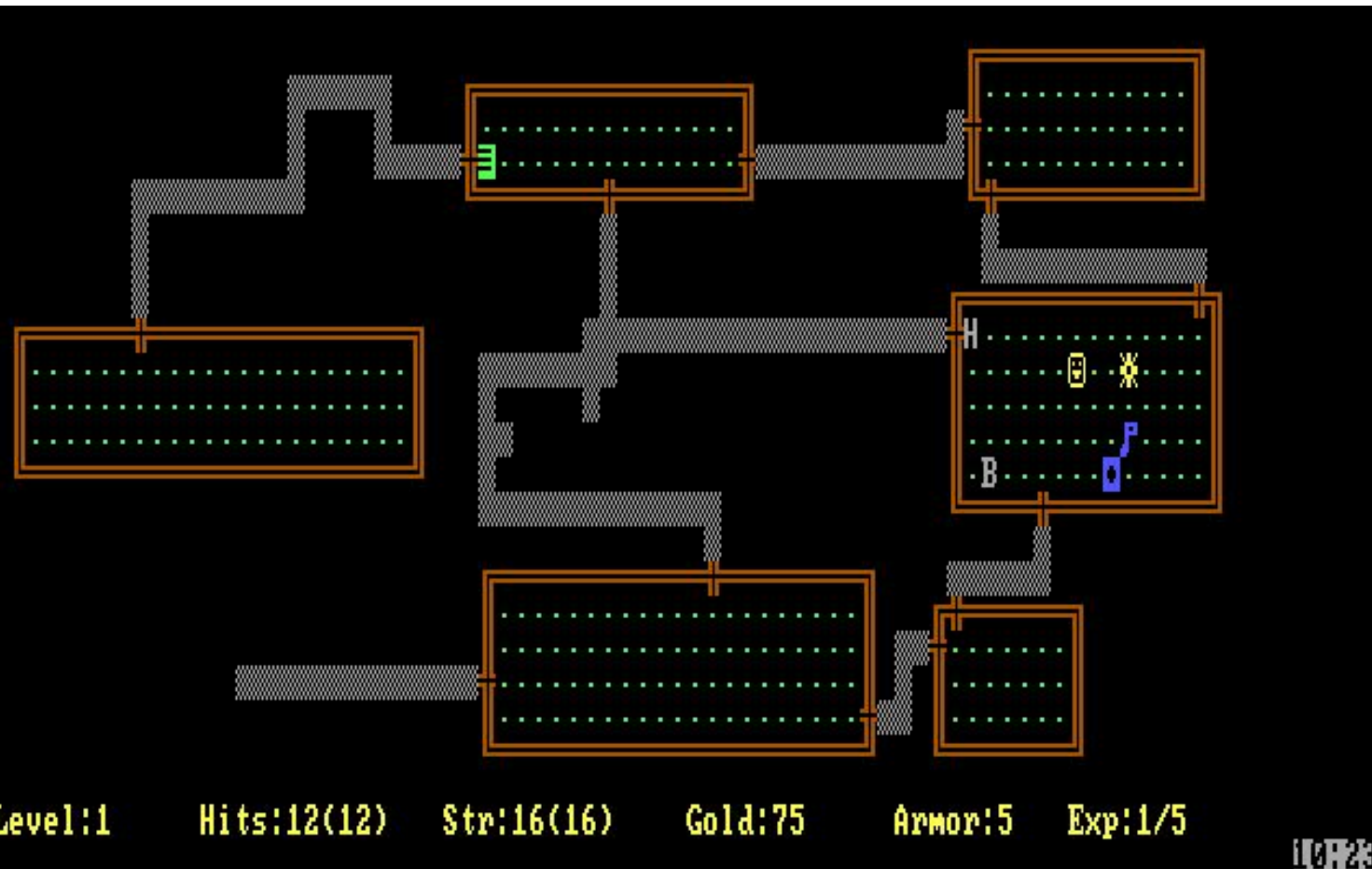
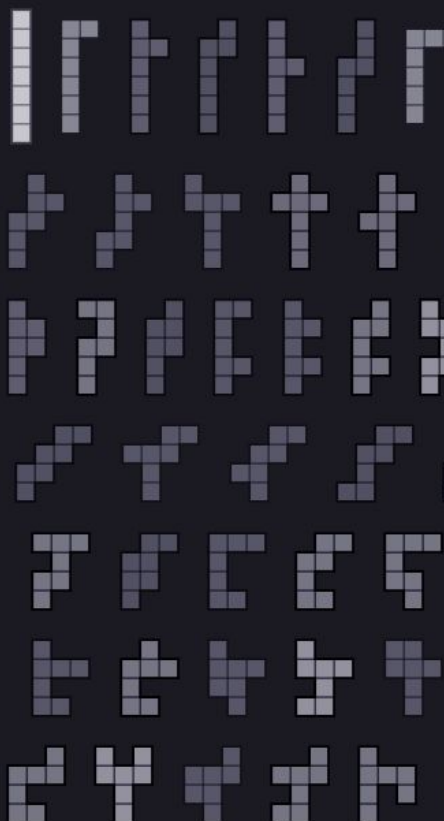
- ✗ has\_terminal\_object
- ✗ has\_binary\_products
- ✗ is\_discrete
- ✗ is\_terminal
- ✗ has\_initial\_object
- ✗ has\_binary\_coproducts
- ✗ is\_groupoid
- ✗ is\_monoid
- ✗ is\_connected
- ✗ has\_finite\_products
- ✗ is\_preorder
- ✓ is\_skeletal
- ✗ has\_equalizers
- ✗ has\_finite\_coproducts
- ✗ is\_complete
- ✗ is\_cocomplete
- ✗ has\_coequalizers
- ✗ is\_initial





NAME	EXP	HP
100	100	100
ORIGINAL REX NINE		

heptominoes ( $n = 7$ )



... and Rogue

# Rogue Level Generator

9 rooms

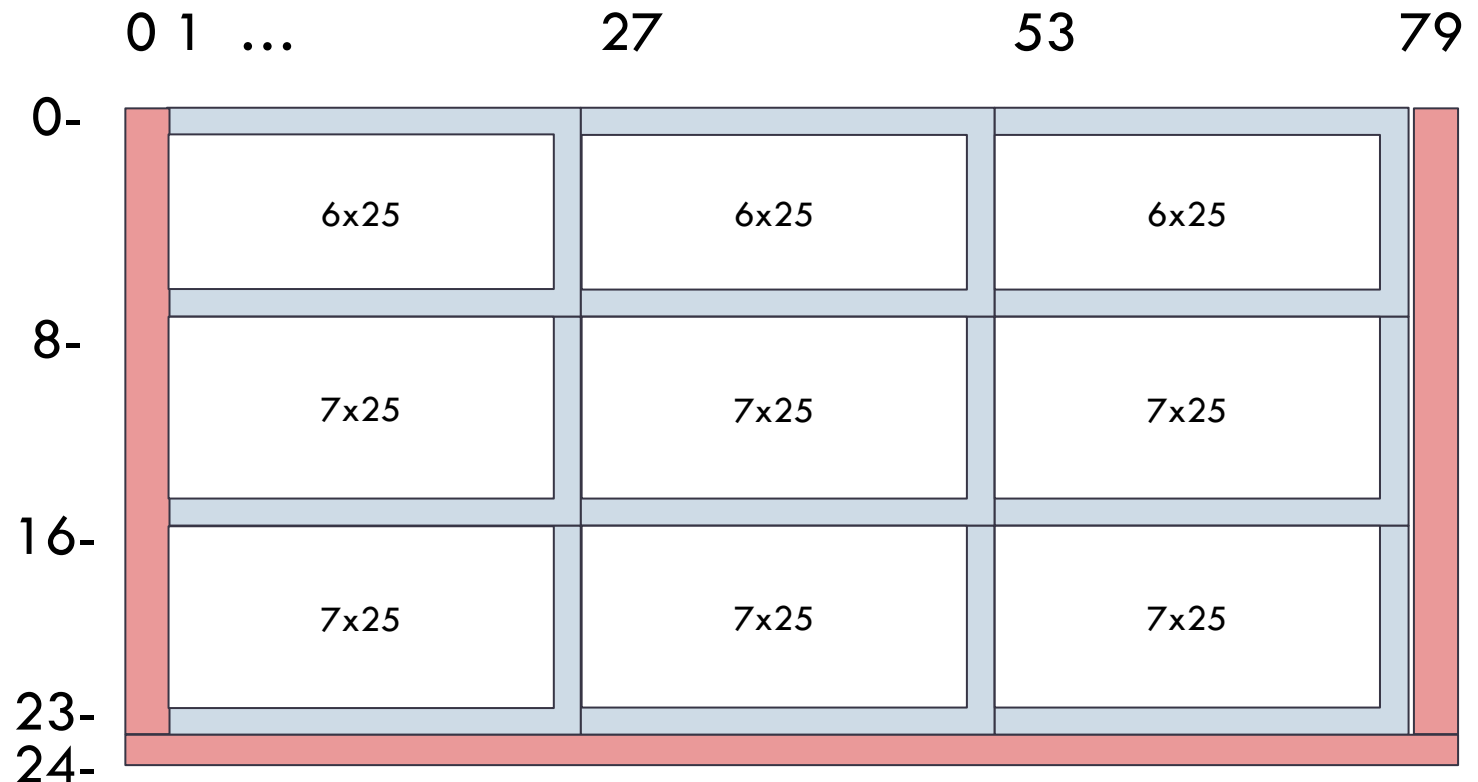
Up to 3 can be missing

For each room:

- Pick a width [4,25]
- Pick a height [4,7]

Pick a location within the “cell” that fits.

Start over if the room would be on the first row (so top rooms can only be 4-6.)



# Counting the Possibilities

Width 25	Width 24	....	Width 5	Width 4	
1 horizontal position	2 horizontal positions	...	21 horizontal positions	22 horizontal positions	=253 possibilities

Same calculation for height gives 6 (top row) or 10 (other two rows) possibilities.

So, there are about  $(253 \times 6)^3 \times (253 \times 10)^6$  possible combinations of room sizes, if no rooms are absent. That's about  $2^{(99.5)}$  possibilities.

- P.S. this number doesn't change much if we account for "gone" rooms



# Rogue as Serial Art

Rogue uses a 32-bit PRNG with a seed value.

There are only  $2^{32}$  possible versions of a level!

We could enumerate all of them.

- As an art project?
- To better understand the generator?
- I just thought it was cool that not only are there room configurations that are impossible, they *vastly* outnumber the ones that are valid.

# The Rogue RNG(s)

## Original rogue:

```
((seed = seed*11109+13849) >> 16) & 0xffff)
```

This is a standard Linear Congruential generator, but using bits 16-32 of the seed avoids some of its worst behaviors.

## MS-DOS port:

```
seed *= 125;  
seed -= (seed/2796203) * 2796203;  
return ((ran() + ran()) & 0x7fffffff1)
```

“This is adapted from the FORTRAN version in ‘Software Manual for the Elementary Functions’ by W.J. Cody, Jr and William Waite.”

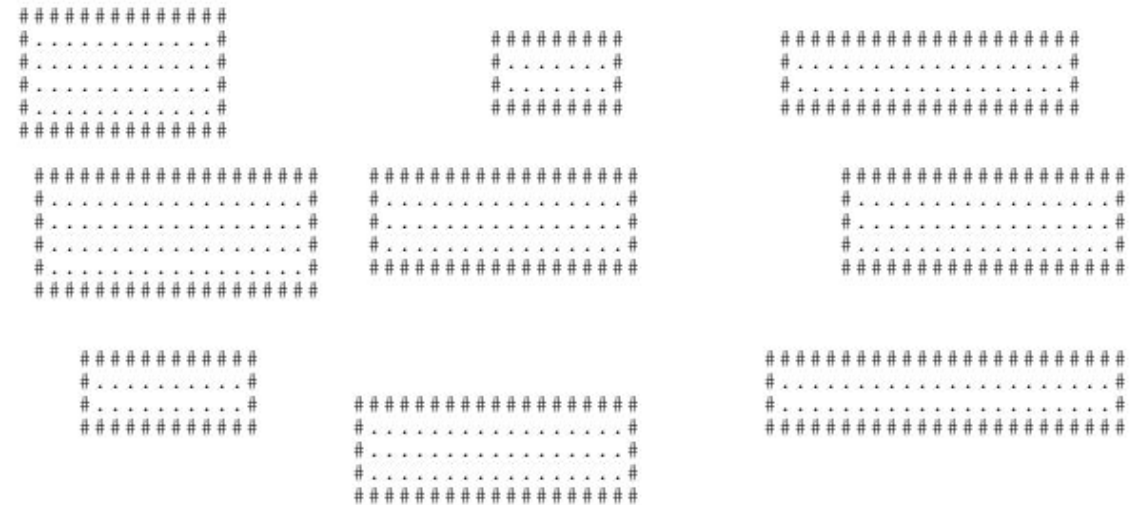
# It's not even hard!

I thought that this would take some nontrivial engineering work.

- Parallelize, implement on a GPU, ...?
- Implement an index for reverse lookup ...?

Nope, I copied the implementation into a C++ program (\*) and it only takes 11 minutes to run through every possible seed.

(\*) = turns out it is not possible to copy Rogue's room generation faithfully without also copying its monster and item generation, oops



first 100 seeds

# Postselection

What if the top-left room is 4x4?

This happens 63,579,000 times. The distribution of the remaining rooms is about even.

(This plot shows how what % of time the given square is occupied.)





[illegible]

A large grid of 100 small images, each showing a different combination of red and blue horizontal stripes, representing the possible combinations of the two colors in the stripes of the flag.

[illegible]

The figure displays a 10x10 grid of 100 small plots, each representing a time series of a variable over 100 time steps. The variables are labeled on the left and top of the grid. The plots show various patterns of change, including step functions, linear trends, and random fluctuations. The color scale for the y-axis ranges from -10 (blue) to 10 (red), with 0 being white.

**Variables (Left Column):**

- Variable 1
- Variable 2
- Variable 3
- Variable 4
- Variable 5
- Variable 6
- Variable 7
- Variable 8
- Variable 9
- Variable 10

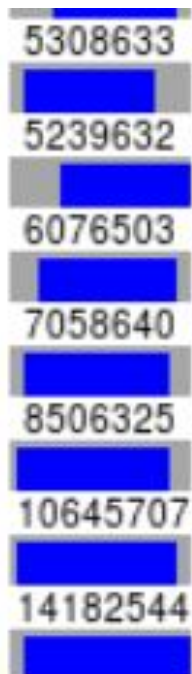
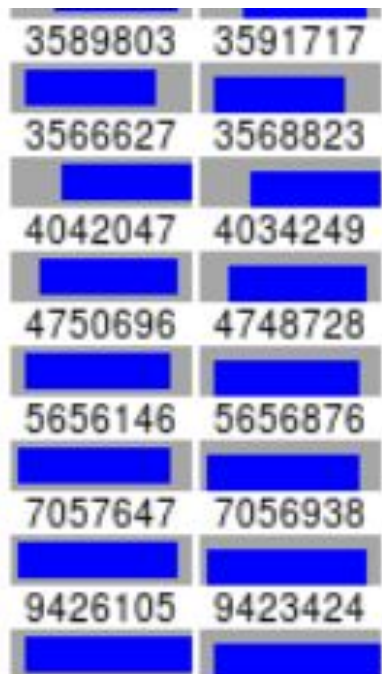
**Time Steps (Top Row):**

- Time Step 1
- Time Step 2
- Time Step 3
- Time Step 4
- Time Step 5
- Time Step 6
- Time Step 7
- Time Step 8
- Time Step 9
- Time Step 10

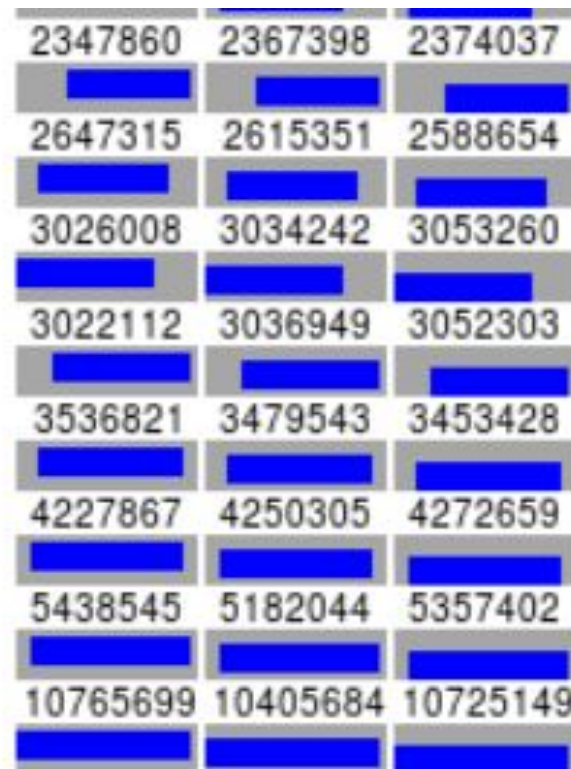
The plots show a variety of patterns, including step functions, linear trends, and random fluctuations. The color scale for the y-axis ranges from -10 (blue) to 10 (red), with 0 being white.

A large grid of 100 small images, each showing a different flag or pattern. The flags are arranged in a 10x10 grid. The flags are of various colors and designs, including solid colors, stripes, and patterns. Some flags are recognizable as national flags, while others are less so. The grid is a visual representation of a large dataset of flags.

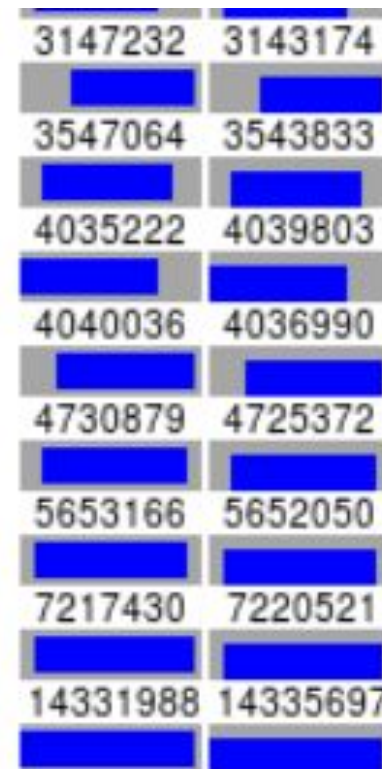




room 1

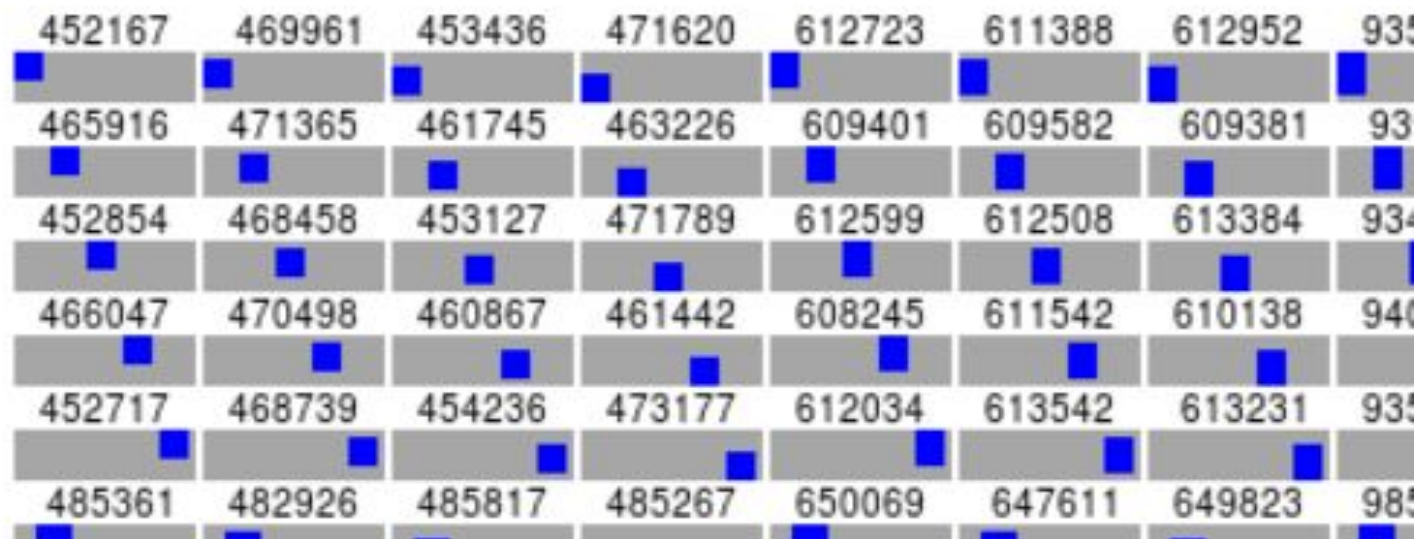
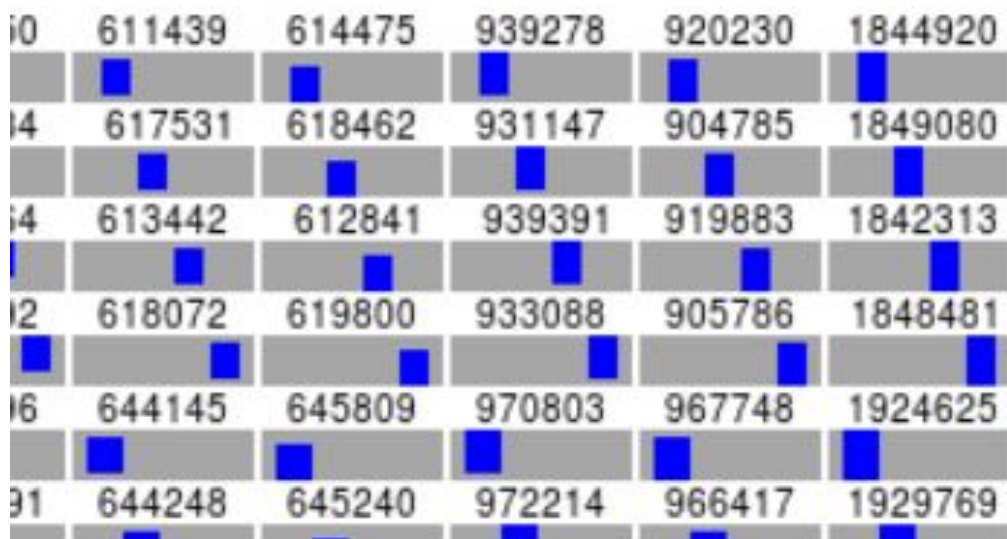


room 2



room 4

room 5

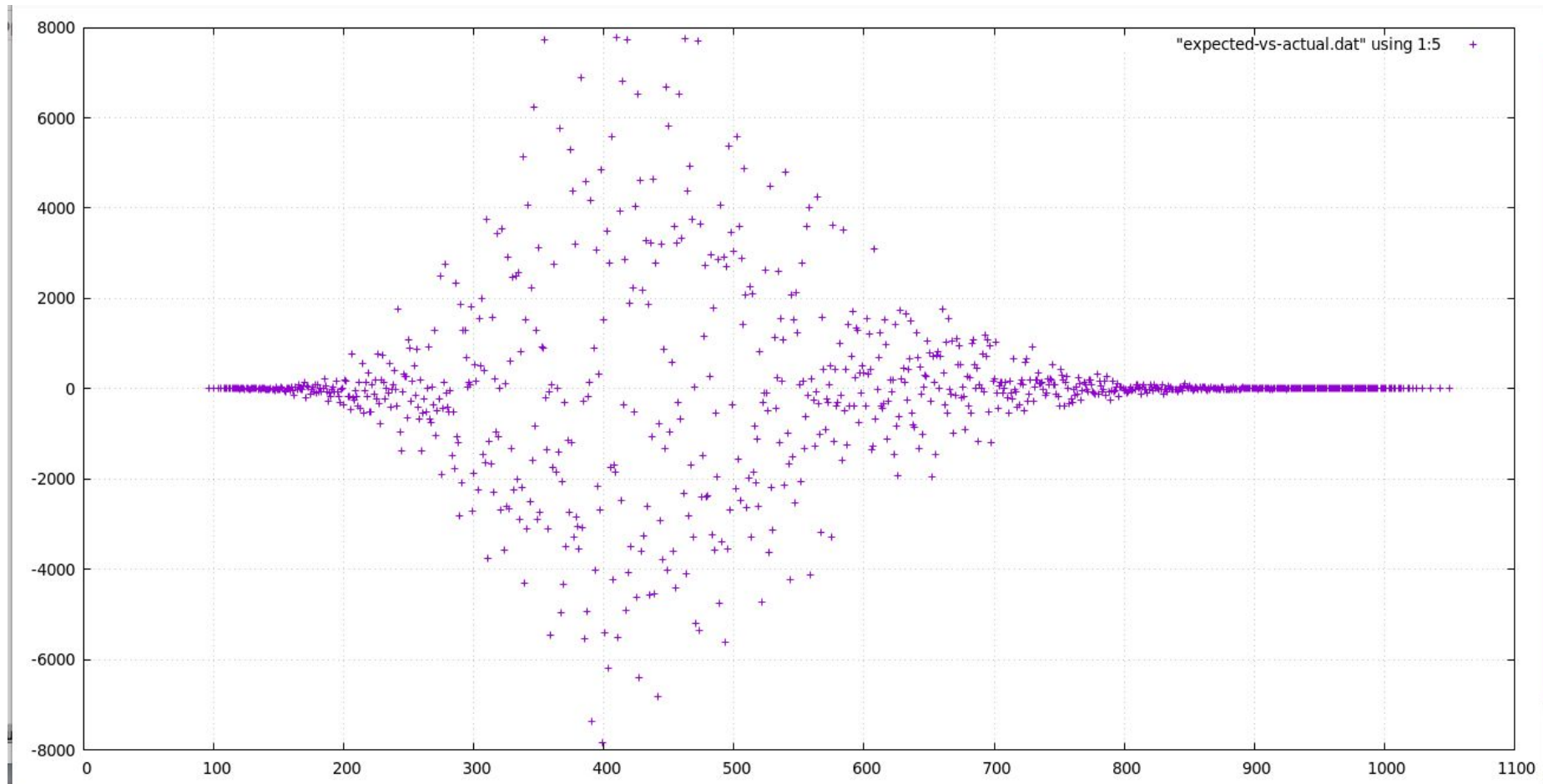


# Finding extremal seeds

total room area	actual / cumulative	expected cumulative
96-110	0	2
112	2	7
113	4 (6)	9
116	7 (13)	21
117	6 (19)	26
118	5 (24)	30
120	18 (42)	55
121	16 (58)	69
122	11 (69)	82

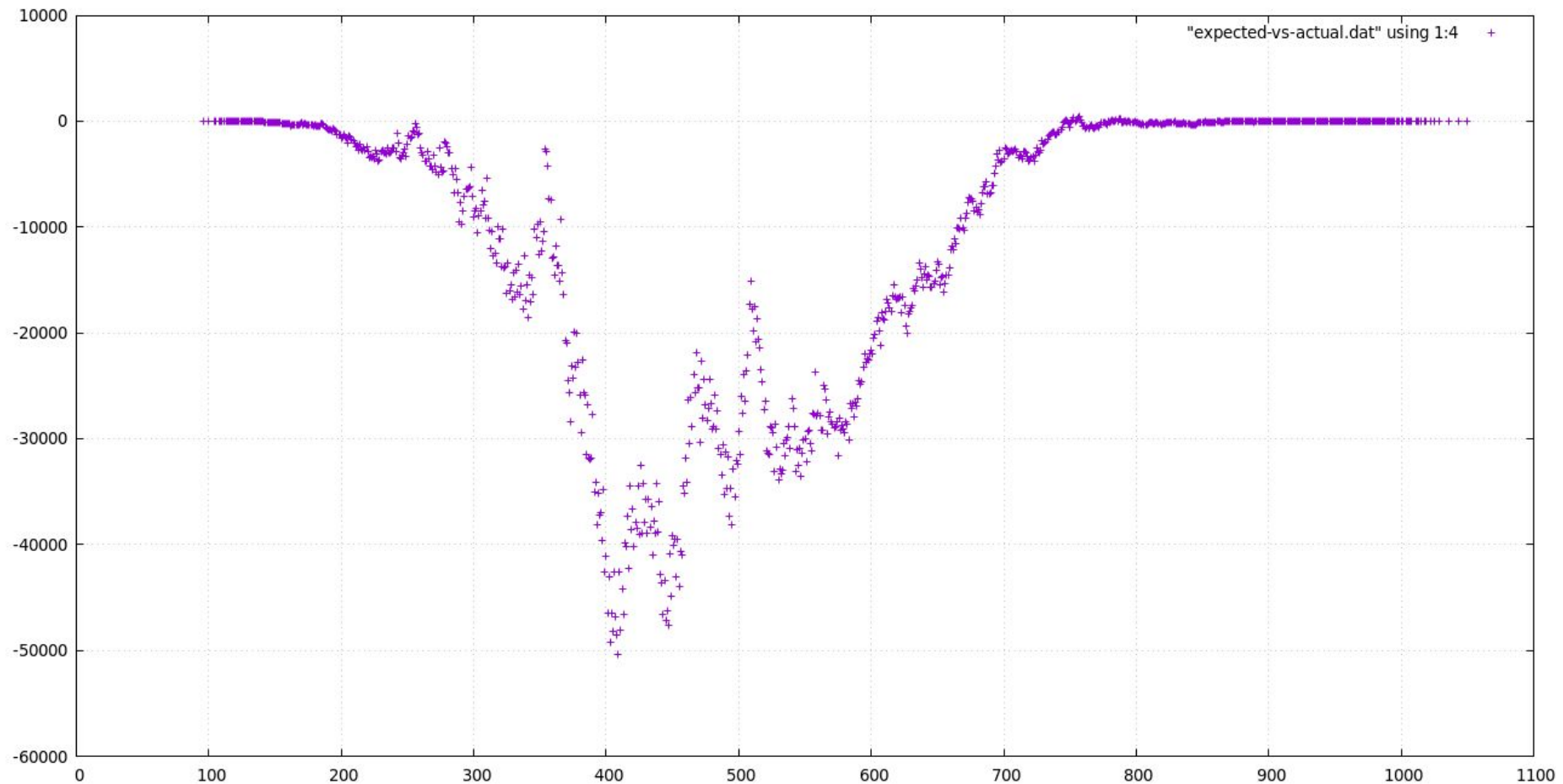
There seems to be a deficit in how often the smallest areas appear, compared to their computed distribution (in the case where three rooms are gone, so 25% of the sample space).

# Expected vs. actual (per size)





# Expected vs. actual (cumulative)



# What I'd like to find (but haven't)

- Some artifact of the room distribution that I can point to and say “this is because Rogue uses a linear congruential generator.”
- Some better visualization of the 36-dimensional space of room configurations

# What's the point?

Generators that use 32-bit seeds can be *fully* explored

- As science! As art! As CYA?!?

What are other connections between Serial Art and procedural generation?

- I'd love to hear an actual art historian weigh in!

Building a solver – could easily “reconstruct” the seed by mistake. :(

Middle ground? Games with seeds in the hundreds rather than in the billions?

# THANK YOU!



2020 Roguelike  
Celebration speaker

On Mastodon: @markgritter@mathstodon.xyz

On Bluesky: @markgritter.bsky.social

On Github:

`markgritter/rogue-room-generation`

In Real Life: Principal Engineer at [thirdlaw.io](https://thirdlaw.io)