Maths for Programming, Fun, and Creativity

Dave Gurnell @davegurnell
Noel Welsh @noelwelsh

The School of Athens, © 1509-1511 Raphael, Public Domain, https://en.wikipedia.org/wiki/The_School_of_Athens

The School of Athens, © 1509-1511 Raphael, Public Domain, https://en.wikipedia.org/wiki/The_School_of_Athens

The School of Athens, © 1509-1511 Raphael, Public Domain, https://en.wikipedia.org/wiki/The_School_of_Athens

Medieval woodcut showing Pythagoras with bells and other instruments, Unknown, Gallica Digital Library, Public Domain, https://en.wikipedia.org/wiki/Pythagoras

octave = 2/1

fifth = 3/2

2/1	octave
3/2	fifth
4/3	fourth
5/4	major third
6/5	minor third

Number theory

Music

Geometry

Astronomy

What is maths?

IANAM

Counting, numbers, ratios?

Equations?

Proving things?

Structured thought

Abstraction

How many students are there?

Art 🙂😎🙂

Biology 🙂🙂😀😀😎😸

Computing 😎😀😎😎😀😎😎😀😎

History 🙂😀😸🙂

Maths 🙂😬😬🙂😀😬

Music 😬🙂😎😀🙂😀😸🙂

🙂😎🙂🙂🙂😀😀😎😸😎

😀😎😎😀😎😎😀😎🙂😀

😸🙂🙂😬😬🙂😀😬😬🙂

😎😀🙂😀😸🙂

Art 🙂😎🙂

Biology 🙂🙂😀😀😎😸

Computing 😎😀😎😎😀😎😎😀😎

History 🙂😀😸🙂

Maths 🙂😬😬🙂😀😬

Music 😬🙂😎😀🙂😀😸🙂

Art	🙂😎🙂	= 3
Biology	🙂🙂😀😀😎😸	= 6
Computing	😎😀😎😎😀😎😎😀😎	= 9
History	🙂😀😸🙂	= 4
Maths	🙂😬😬🙂😀😬	= 6
Music	😬🙂😎😀🙂😀😸🙂	= 8

$$= 3 + 6 + 9 + 4 + 6 + 8$$

$$= 3 + 6 + 9 + 4 + 6 + 8$$

$$= 36$$

Numbers are
an abstraction

36

36😎

36🏛️

36 🍺

Abstractions let us...

Identify common properties

Irrelevant differences

Adding is
an abstraction

1+1

1😎+1😎

1🍺+1🍺

1😎+1🍺

$$= 3 + 6 + 9 + 4 + 6 + 8$$

$$= 36$$

Directions of abstraction...

Students and counting

Numbers and addition

Abstract
up

Students and counting

Numbers and addition

Abstract up

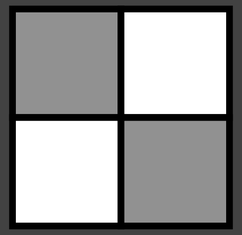
Abstract across

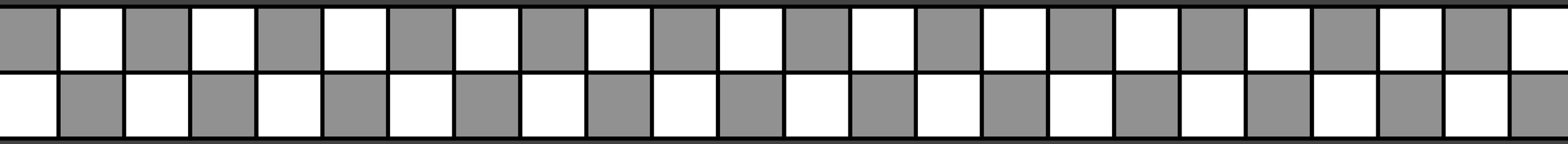
Students and counting

???

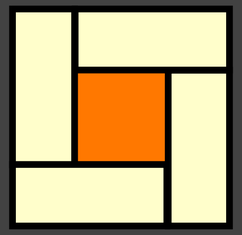
What else can we add?

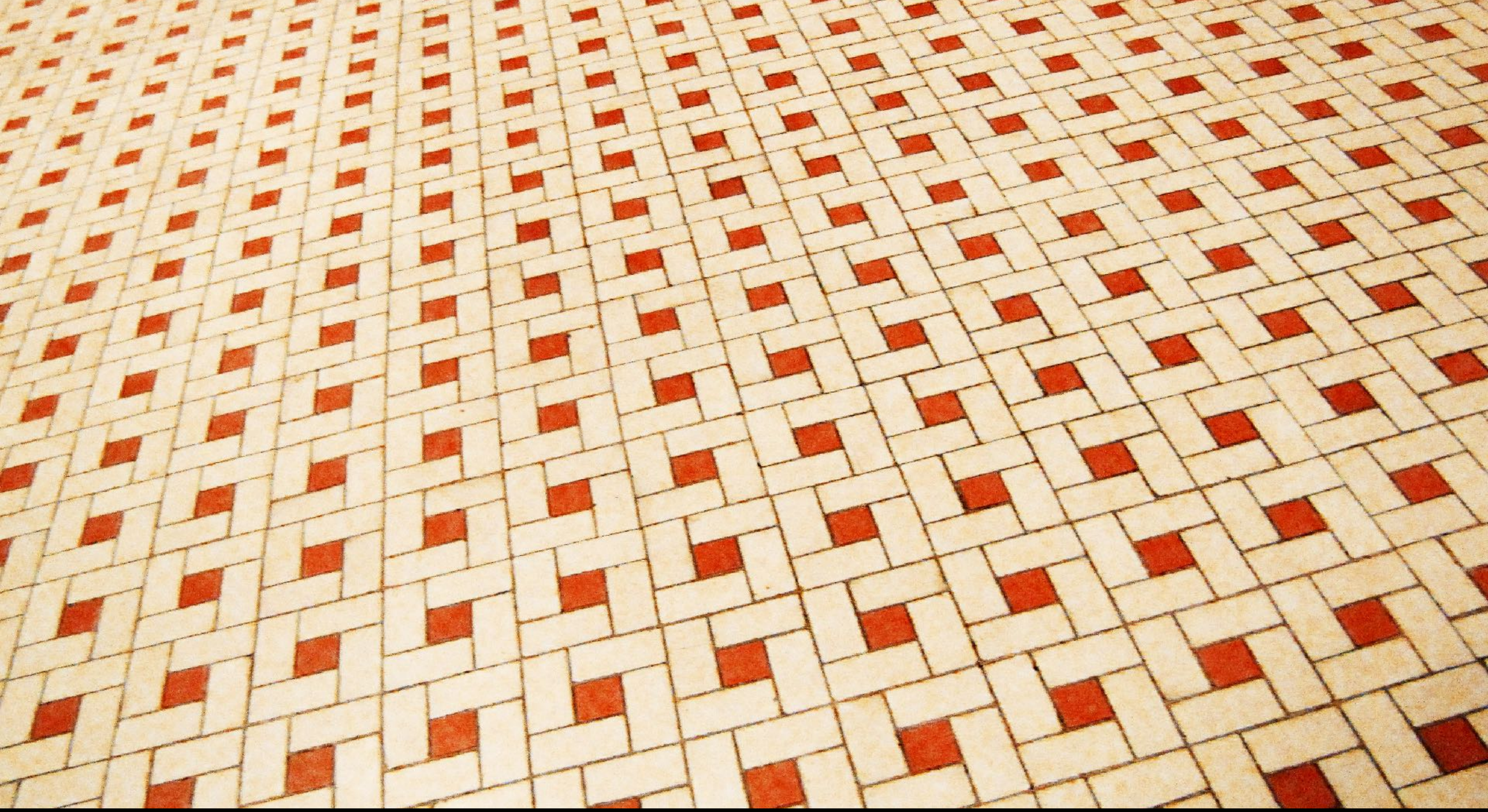
Pictures





Tile Floor, © shaireproductions.com, CC-BY-NC 2.0, https://flic.kr/p/7xtNyL





Tile Floor, © Brian Yap, CC-BY-NC 2.0, https://flic.kr/p/eDr3N

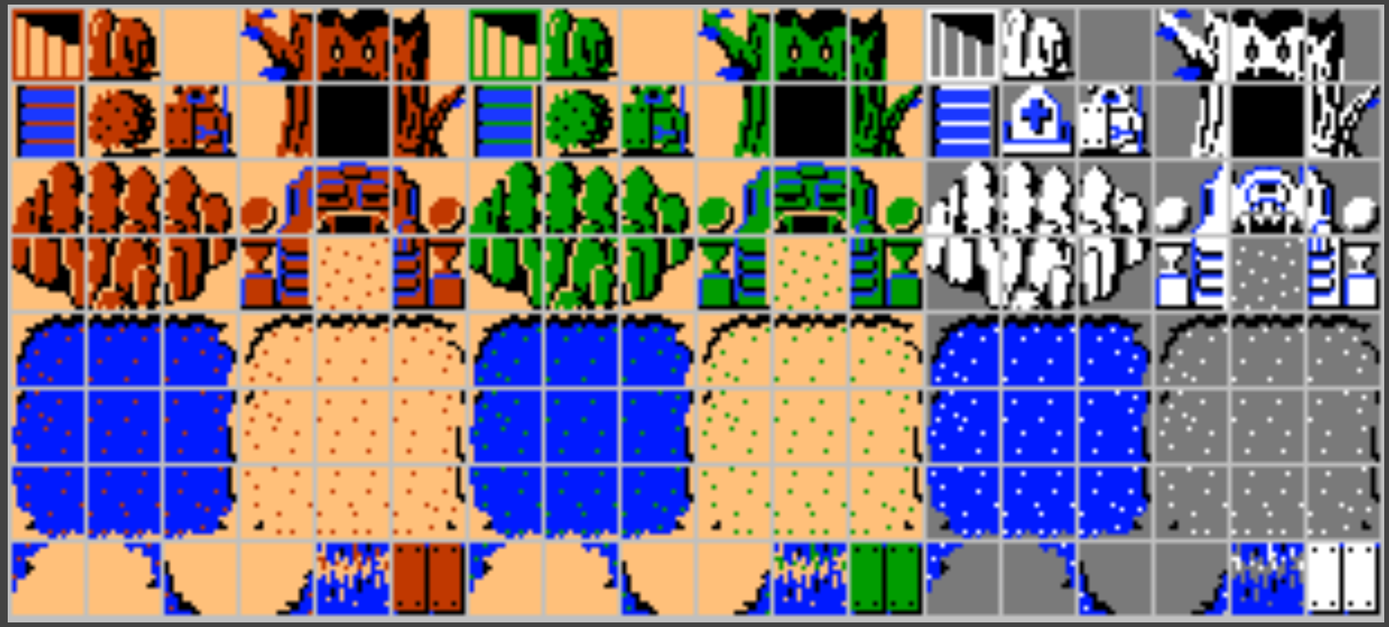
We’ve made an abstraction

Choose base image(s)

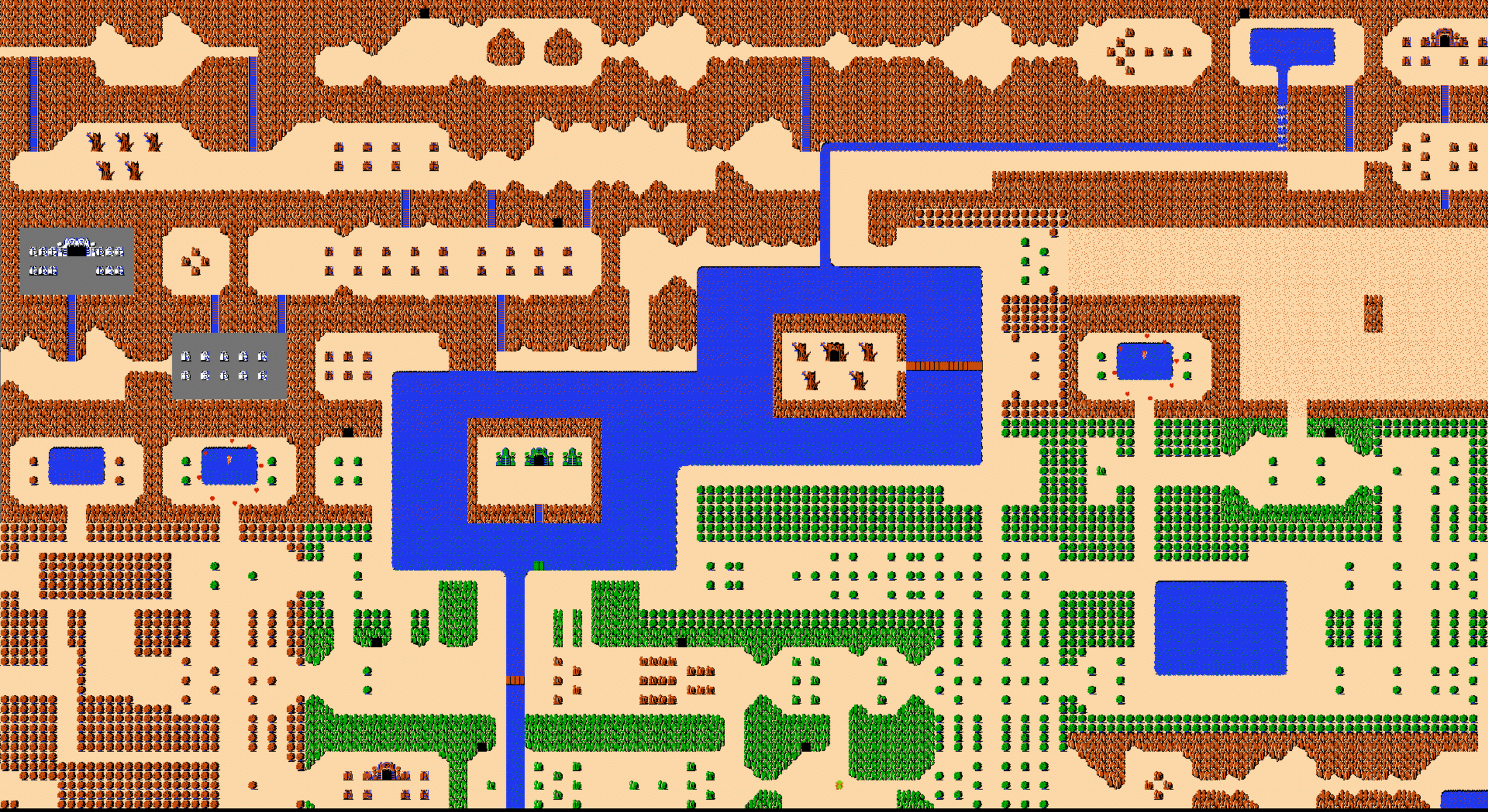
Combine along x-axis

Combine along y-axis

Exploration

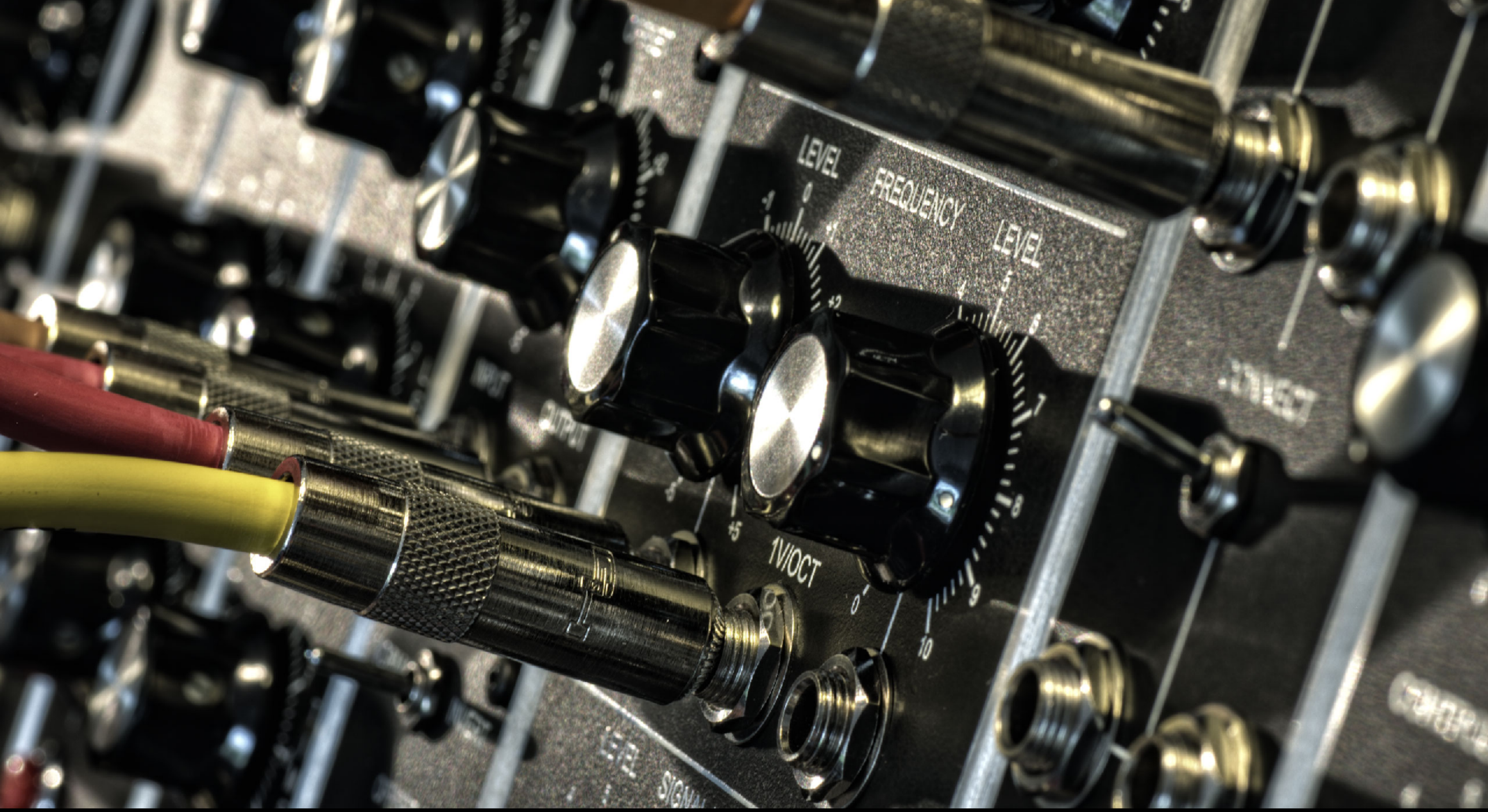


The Legend of Zelda, © 1986 Nintendo. Tileset extracted by Zephiel87

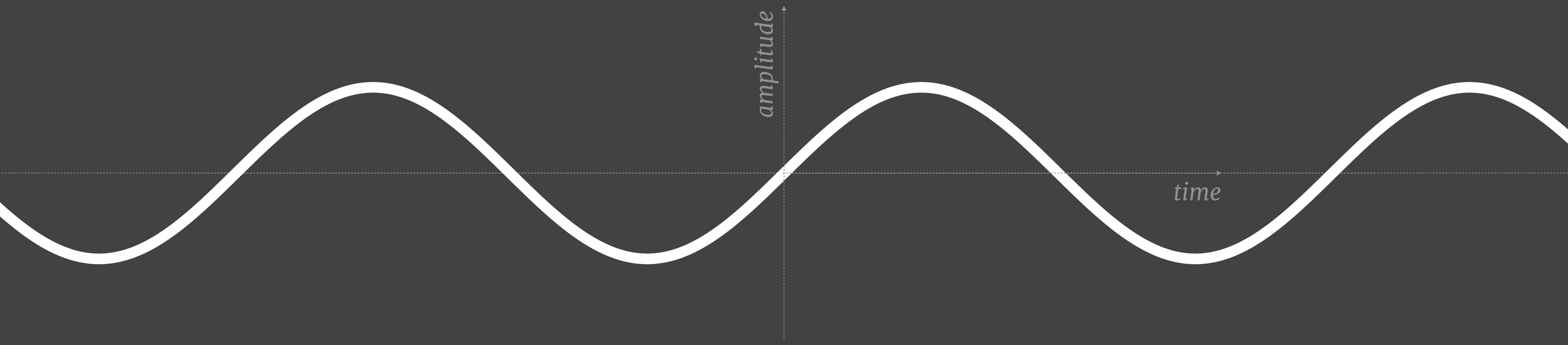


The Legend of Zelda, © 1986 Nintendo

Sound



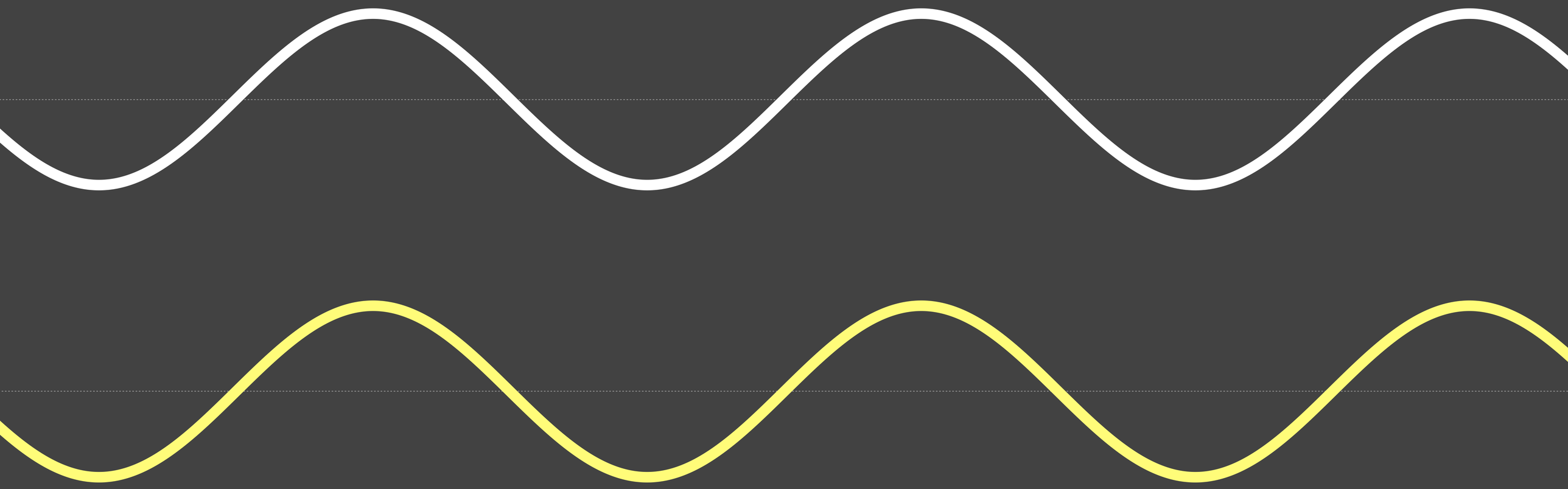
HDR Modular Synthesizer Macro, © Peter Gorges, CC-BY-NC-SA 2.0, https://flic.kr/p/561Mn9


amplitude
time

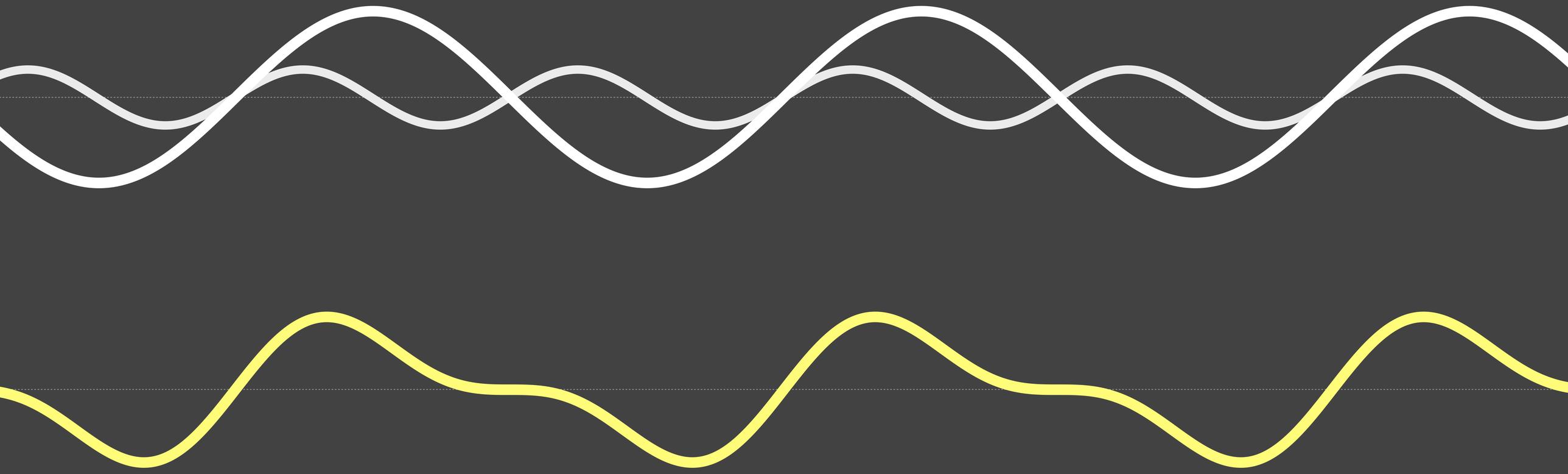
Speaker, © Rowen Atkinson, CC-BY-SA 2.0, https://flic.kr/p/348Tnx

Sawtooth Wave

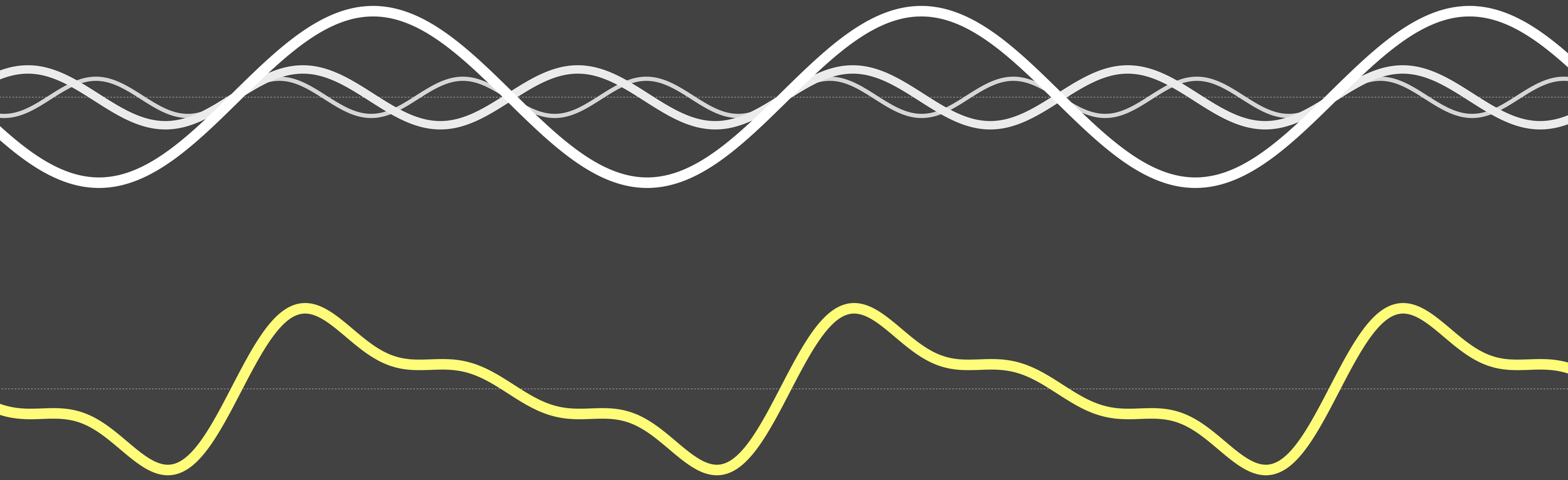
y=sin(x)



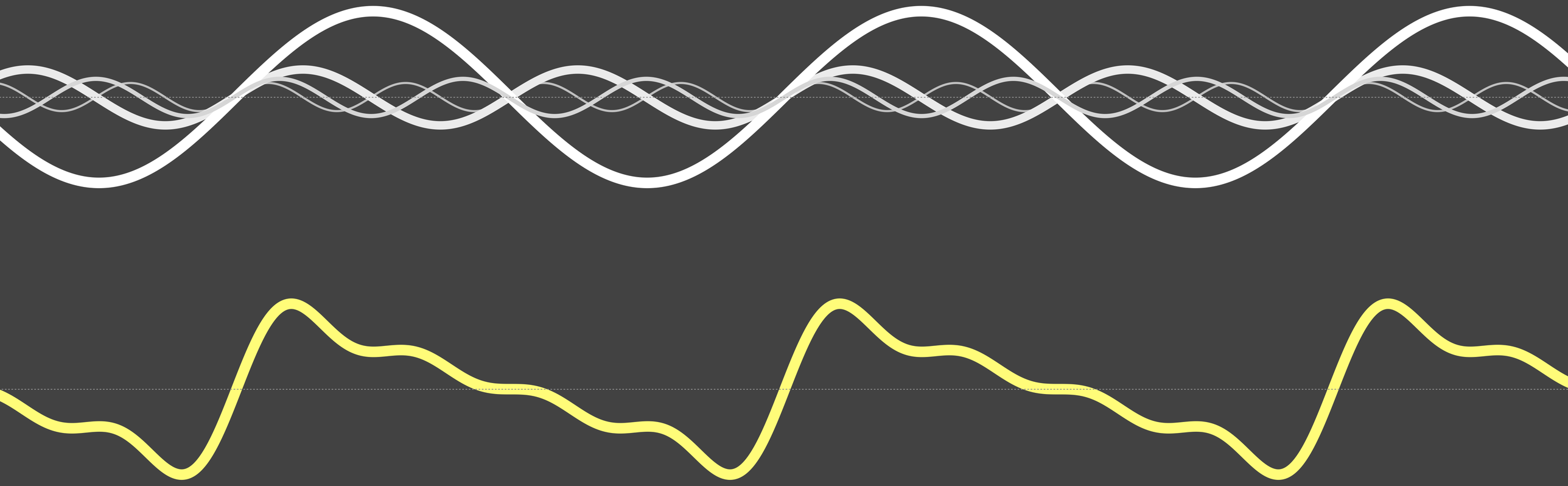
y=sin(x)+sin(2x)/2



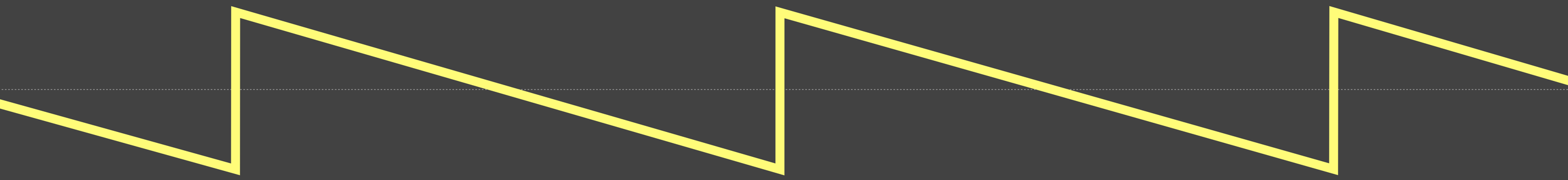
$$y=\sin(x)+\sin(2x)/2+\sin(3x)/3$$



$$y=\sin(x)+\sin(2x)/2+\sin(3x)/3+\sin(4x)/4$$

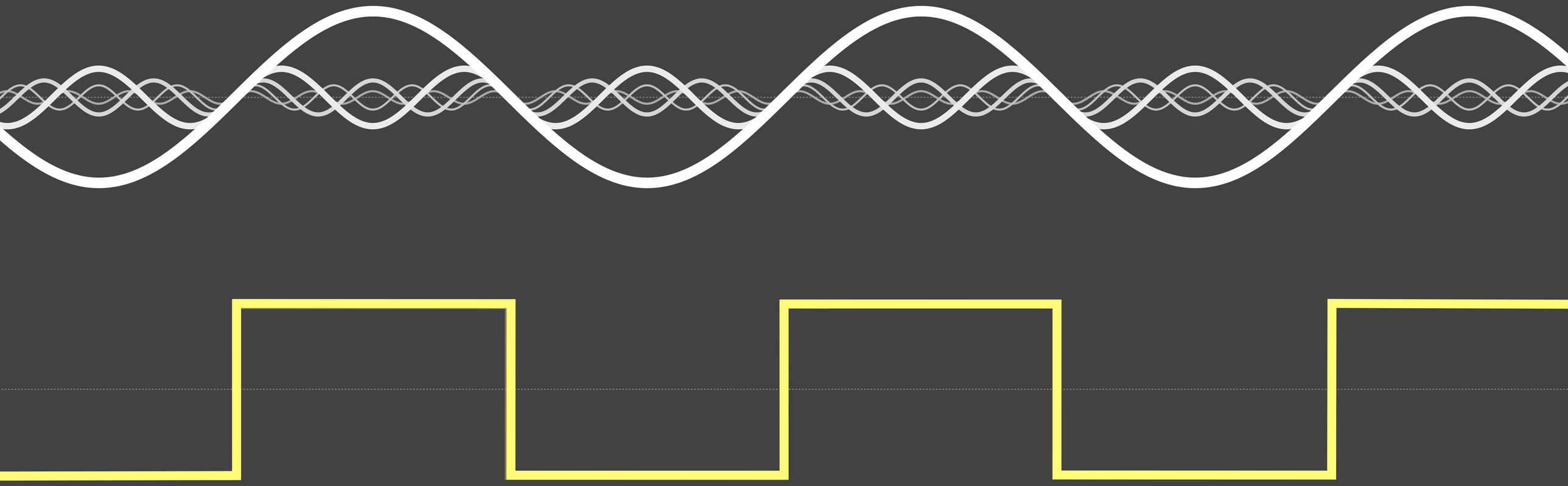


$$y=\sin(x)+\sin(2x)/2+\sin(3x)/3+\sin(4x)/4+\ldots$$



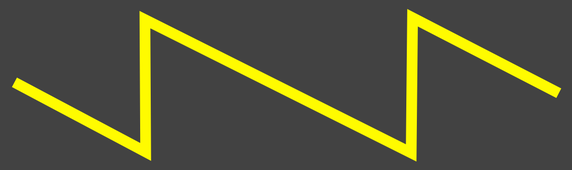
Square Wave

$$y=\sin(x)+\sin(3x)/3+\sin(5x)/5+\sin(7x)/7+\ldots$$

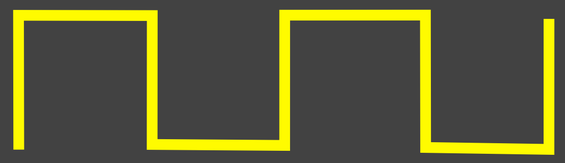


Exploration

Sawtooth



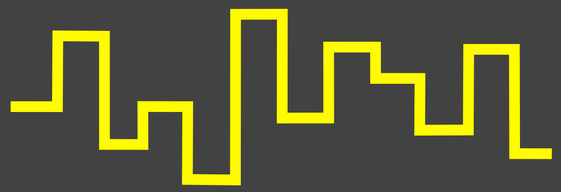
Square



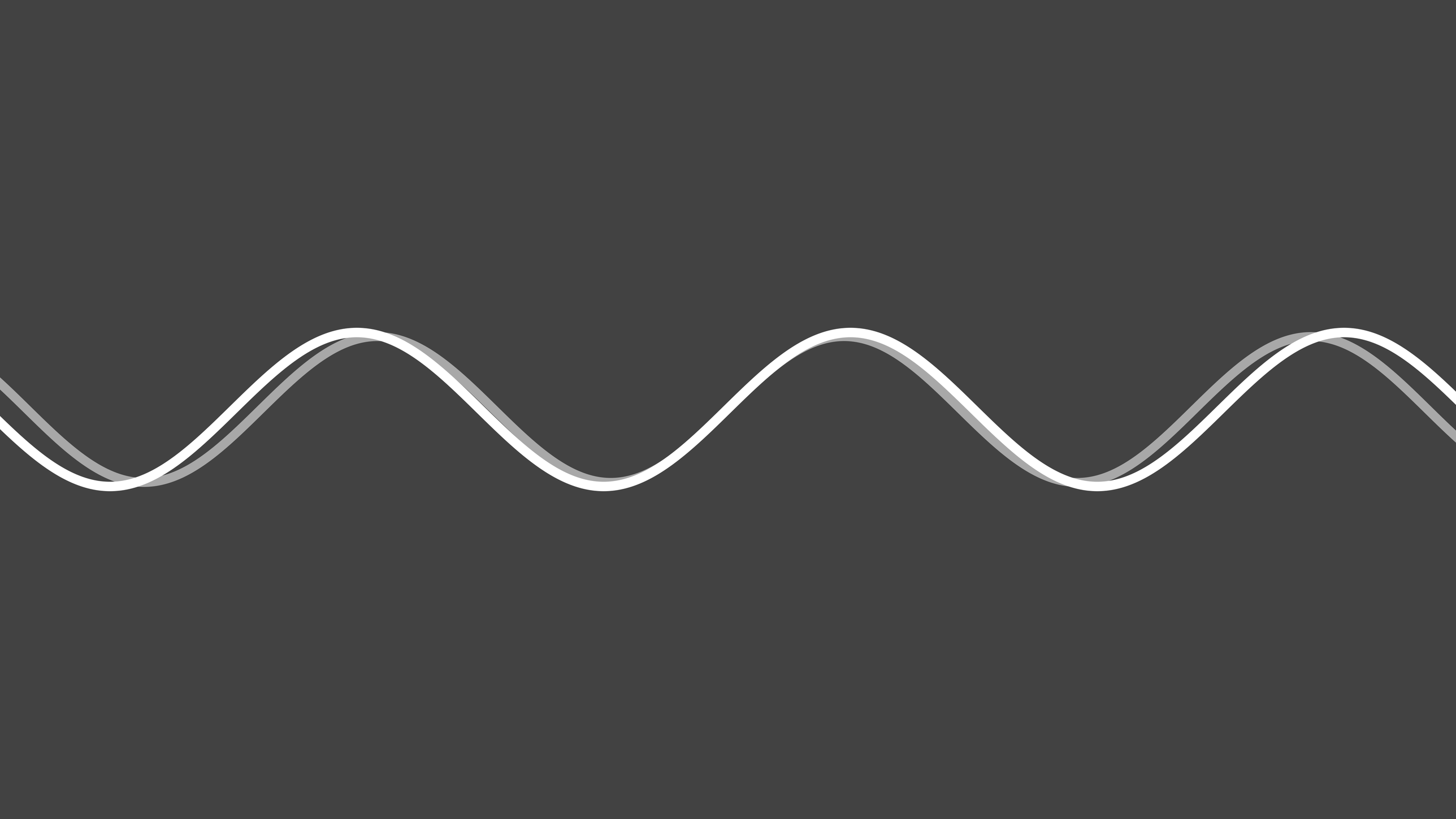
Triangle

Pulse

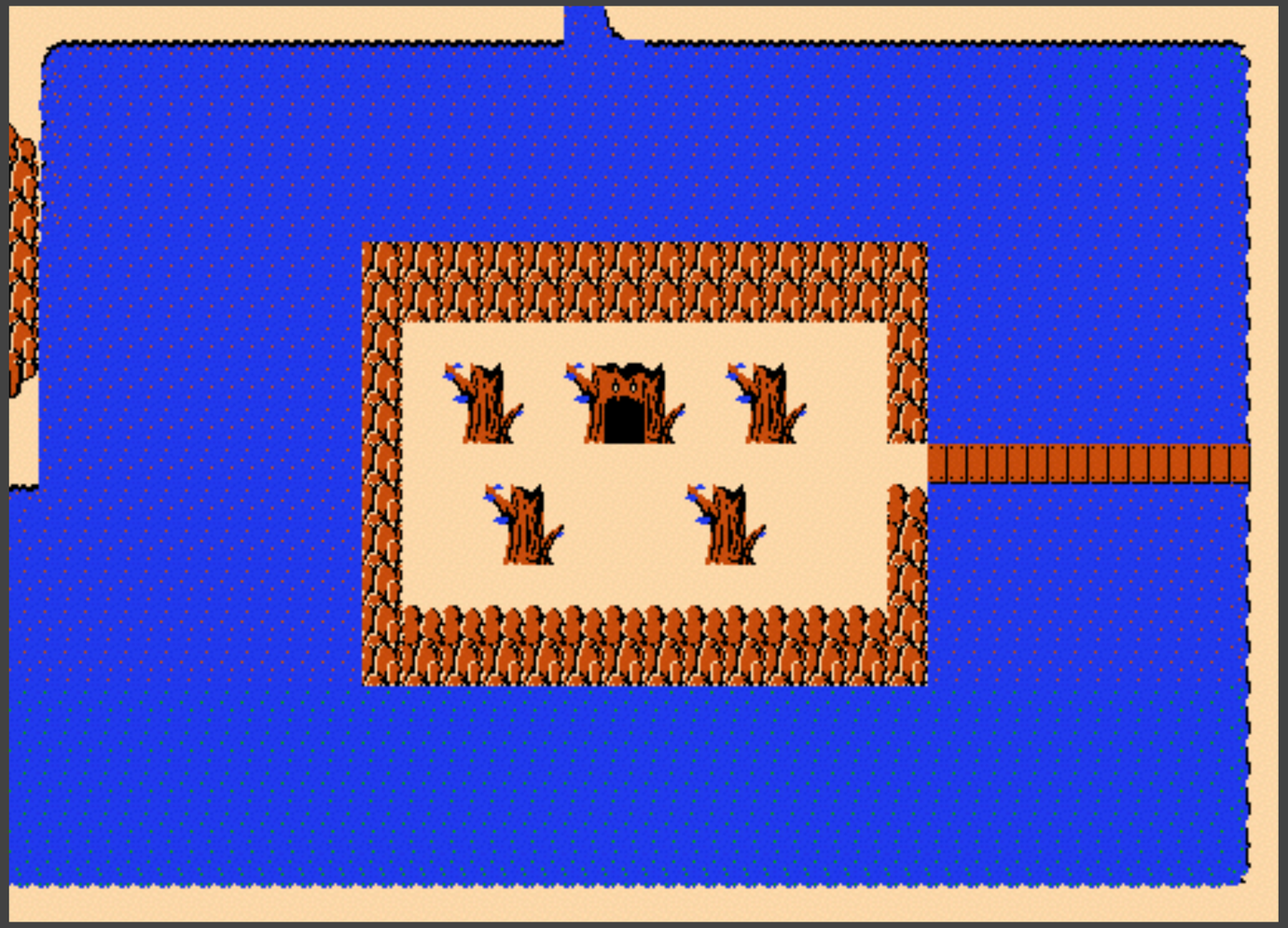
Random



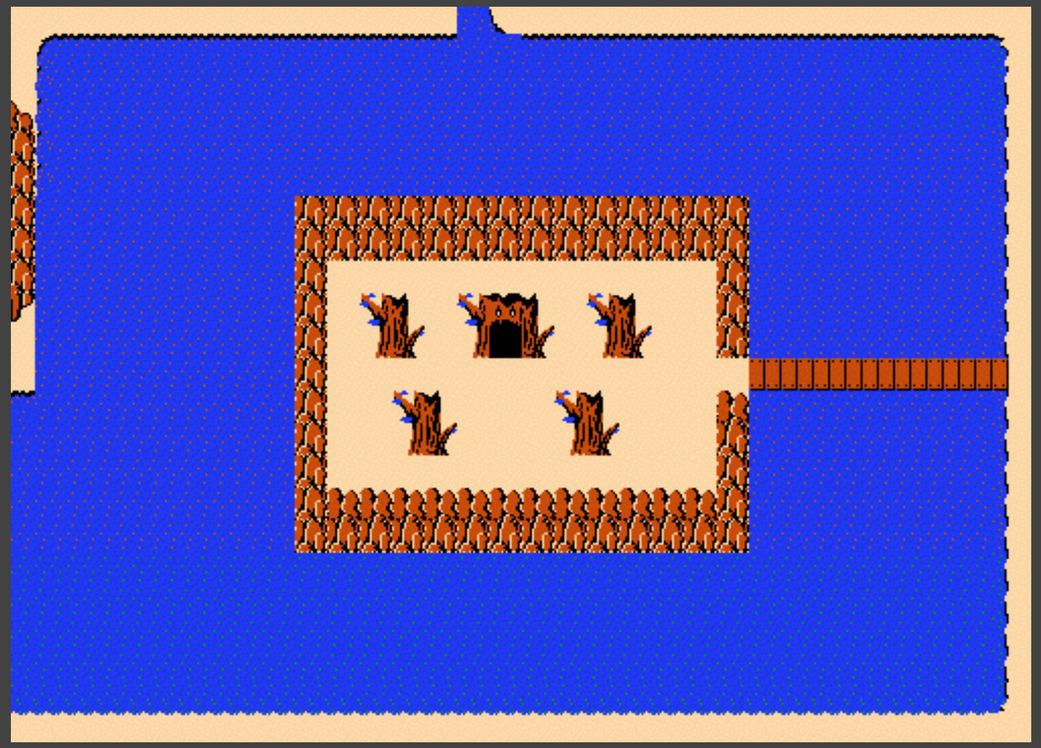
Sine + Noise

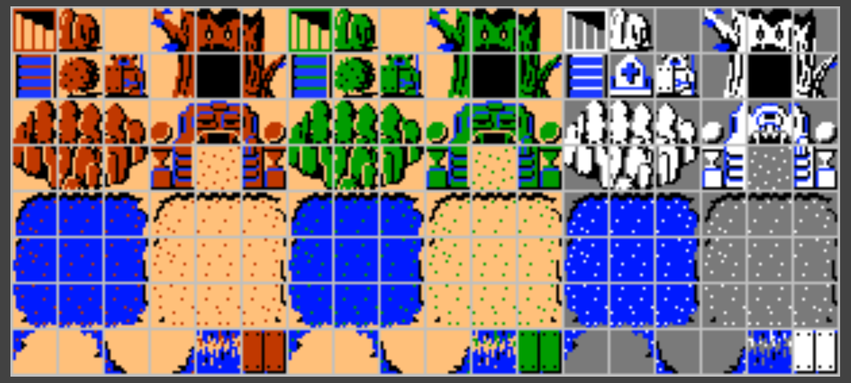


Abstractions
inspire creativity



The Legend of Zelda, © 1986 Nintendo





The Legend of Zelda, © 1986 Nintendo. Tileset extracted by Zephiel87

Korg microKORG Synthesizer, © Korg Inc, http://www.korg.com/uk/products/synthesizers/microkorg/

Jordan Rudess Custom Synthesizer, © Synthesizers.com, https://www.synthesizers.com

What about programming?

What can we add in code?

Numbers

Strings

Lists and Sets

Pictures

Oscillators

HTTP Routes

Monoid

Values

$0, 1, 2, 3, \ldots$

Values

$$0, 1, 2, 3, \ldots$$

Binary operation

$$x + y = z$$

Values

$$0, 1, 2, 3, \ldots$$

Binary operation

$$x + y = z$$

Empty value

$$0$$

```scala
trait Monoid[A] {
  def combine(a: A, b: A): A

  def empty: A
}
```

```scala
implicit object intMonoid extends Monoid[Int] {
  def combine(a: Int, b: Int): Int =
    a + b

  def empty: Int =
    0
}
```

1 |+| 2 |+| 3

6

```
"Hello" |+| "Lambda" |+| "World"
```

```
"HelloLambdaWorld"
```

💻 |+| ☕ |+| 👍

💻☕👍

```
def add3[A: Monoid](x: A, y: A, z: A): A =
  x |+| y |+| z
```

```
add3(💻, ☕, 👍)
```

Abstracting across

CRDTs

{G,PN}-Counters
{G,2P}-Sets
Treedoc
Woot

Approximate data structures

Bloom filters
HyperLogLog
Count-min sketch

Monoid laws

Identity

$$0 + x = x + 0 = x$$

Identity

↔ 😎 = 😎

Identity

Associativity

$$(x + y) + z = x + (y + z)$$

Associativity

(😎↔🏛️)↔🍺=😎↔(🏛️↔🍺)

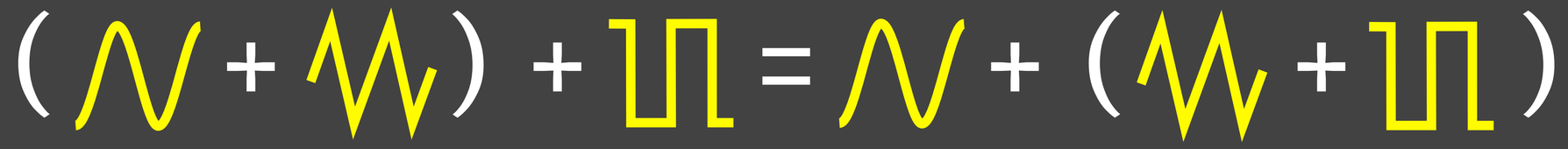
Associativity

😎🏛️ ↔ 🍺 = 😎 ↔ 🏛️🍺

Associativity

😎🏛️🍺 = 😎🏛️🍺

Associativity



$$= 3 + 6 + 9 + 4 + 6 + 8$$

$$= 36$$

Composing abstractions

Monoids |+| Sequences

foldLeft

foldRight

```
(1 to 5).combineAll
```

15

```
(1 to 5).foldMap(_.toString)
```

```
"12345"
```

```
(1 to 100).foldMap(n => sine(n) / n)
```

The story so far...

Abstractions are useful
for creative thinking

Abstractions are useful
for problem solving

Abstractions are composable
(provided they are “lawful”)

And now for
something totally different

Event sourcing

state0
evt1
evt2
evt3

(,)
state0
evt1
evt2
evt3

state1
evt2
evt3

state1
evt2
evt3

state2
evt3

state2
evt3

state3

(psst... this is just folding)

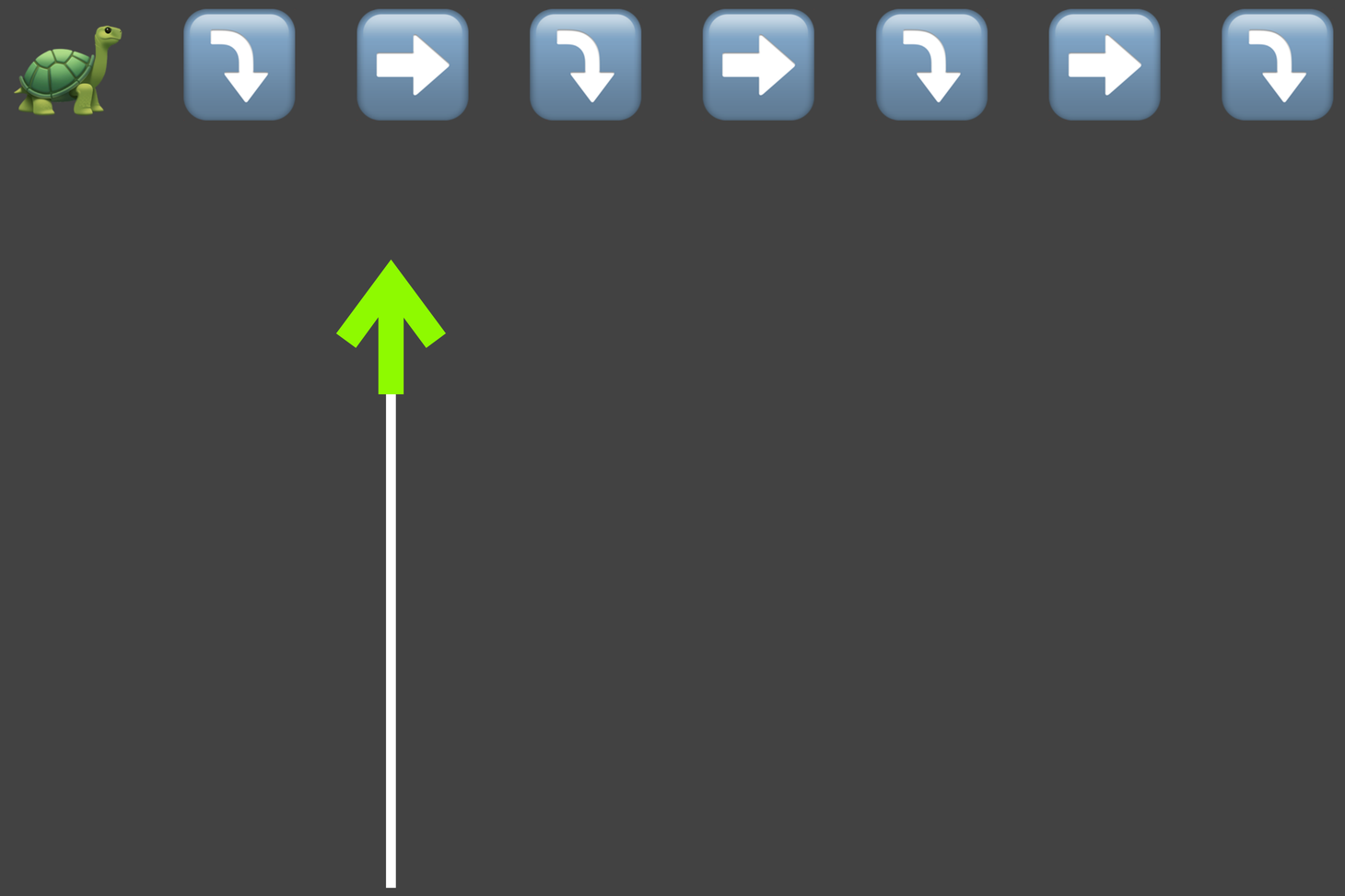
What else can we event source?

Turle, © Asinka Photography, CC-BY 2.0, https://flic.kr/p/S42wM4

➡️ ⤵️ ➡️ ⤵️ ➡️ ⤵️ ➡️ ⤵️

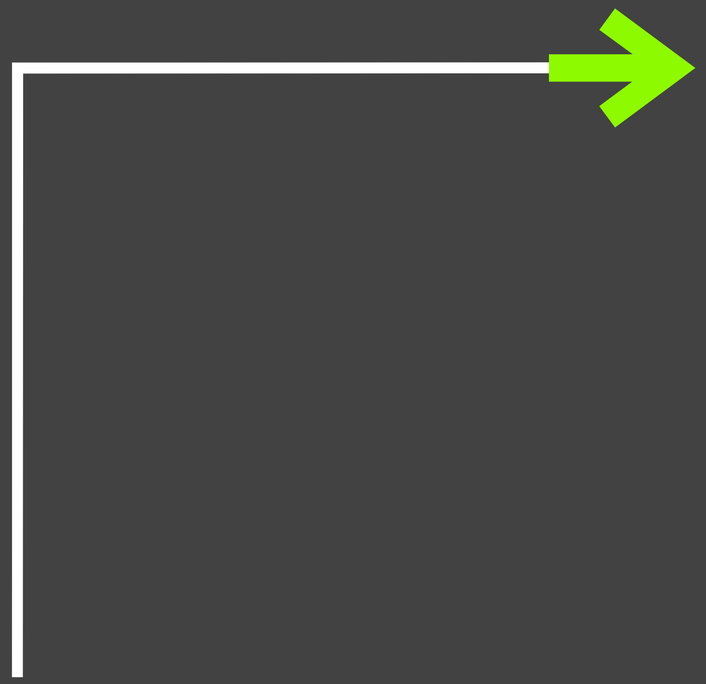
⬆️

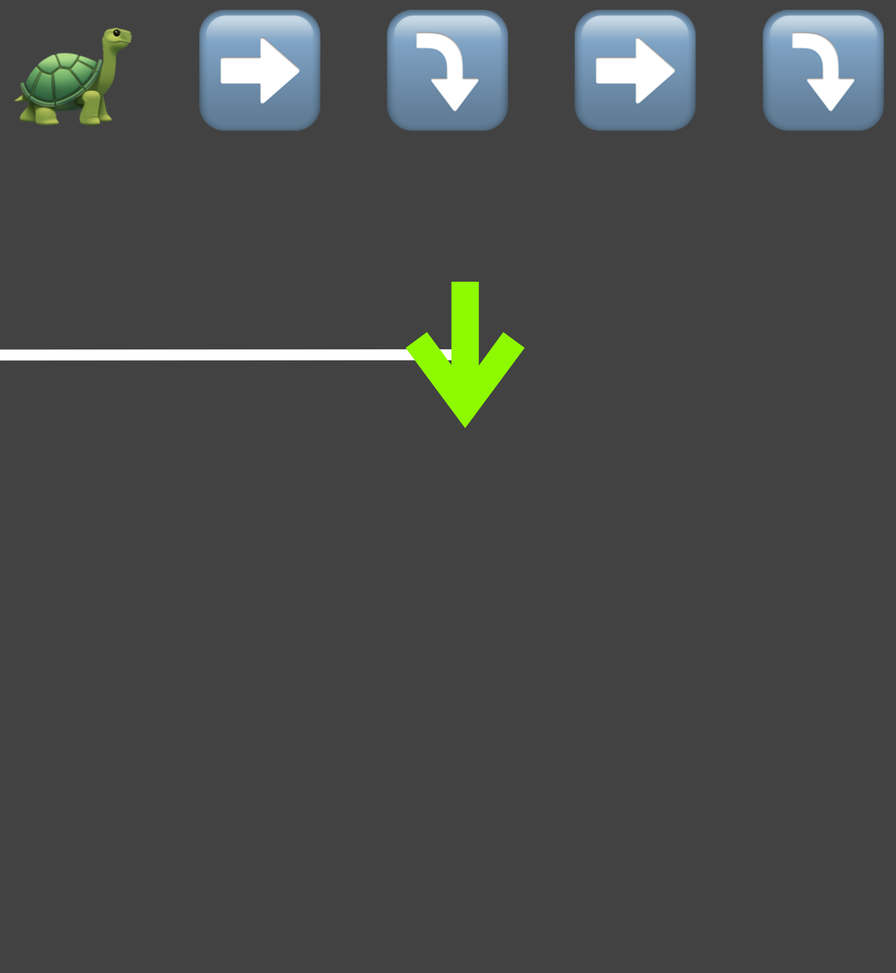
(🐢,➡️)⤵️ ➡️ ⤵️ ➡️ ⤵️ ➡️ ⤵️



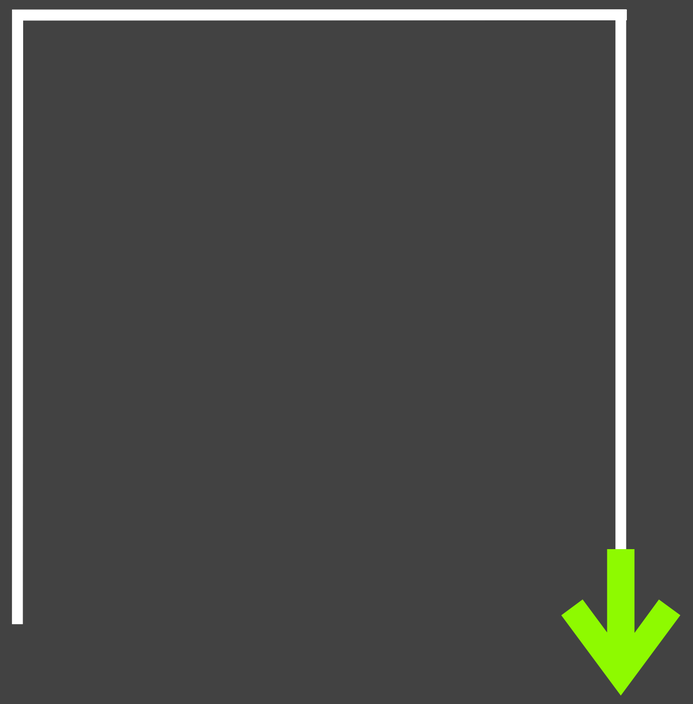
(🐢,⤵️)➡️ ⤵️ ➡️ ⤵️ ➡️ ⤵️

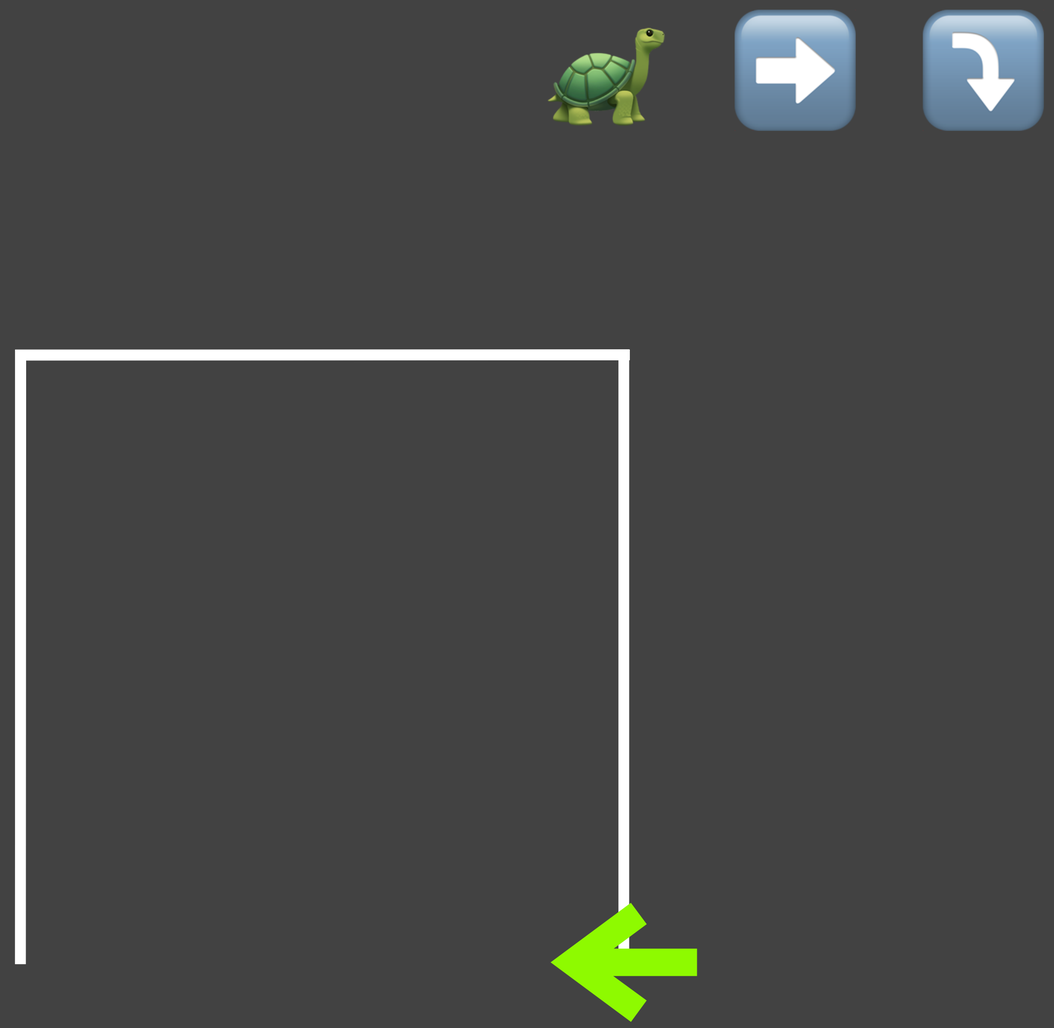
🐢 ⤵️ ➡️ ⤵️ ➡️ ⤵️

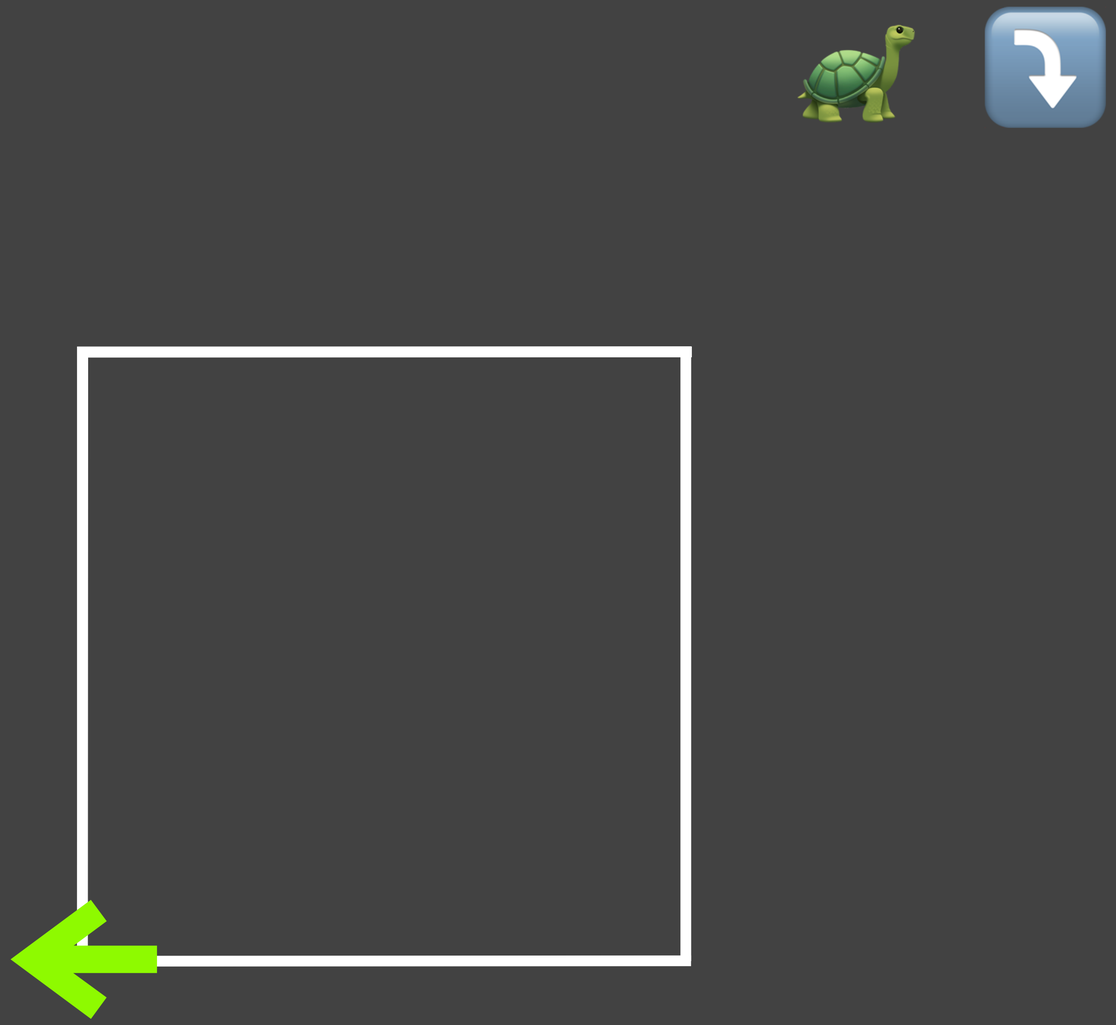


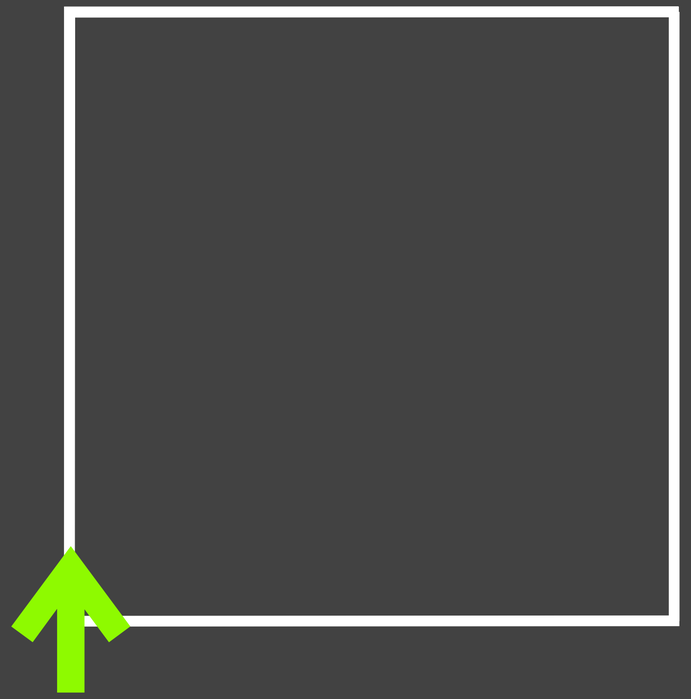


🐢 ⤵️ ➡️ ⤵️









Lindenmayer Systems

L-Systems

Alphabet

$$A, B$$

Axiom

$$A$$

Production rules

$$(A \Rightarrow AB), (B \Rightarrow A)$$

$$A$$

$$(A \Rightarrow AB), (B \Rightarrow A)$$

$$A$$

$$(A \Rightarrow AB), (B \Rightarrow A)$$

$$AB$$

$$(A \Rightarrow AB), (B \Rightarrow A)$$

$$AB$$

$$(A \Rightarrow AB), (B \Rightarrow A)$$

$$ABA$$

$$(A \Rightarrow AB), (B \Rightarrow A)$$

ABA

$$(A \Rightarrow AB), (B \Rightarrow A)$$

ABAAB

$$(A \Rightarrow AB), (B \Rightarrow A)$$

n = 0 : A

n = 1 : AB

n = 2 : ABA

n = 3 : ABAAB

n = 4 : ABAABABA

n = 5 : ABAABABAABAAB

n = 6 : ABAABABAABAABABAABABA

n = 7 : ABAABABAABAABABAABABAABAABABAABAAB

Alphabet

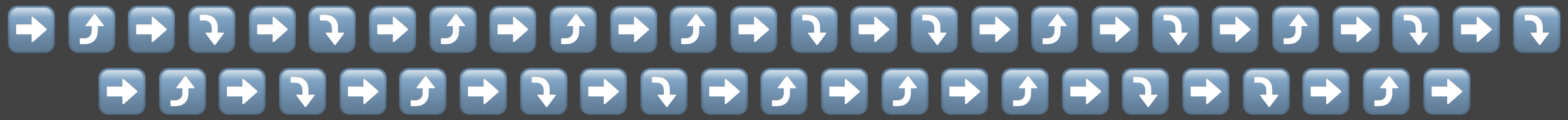
Axiom

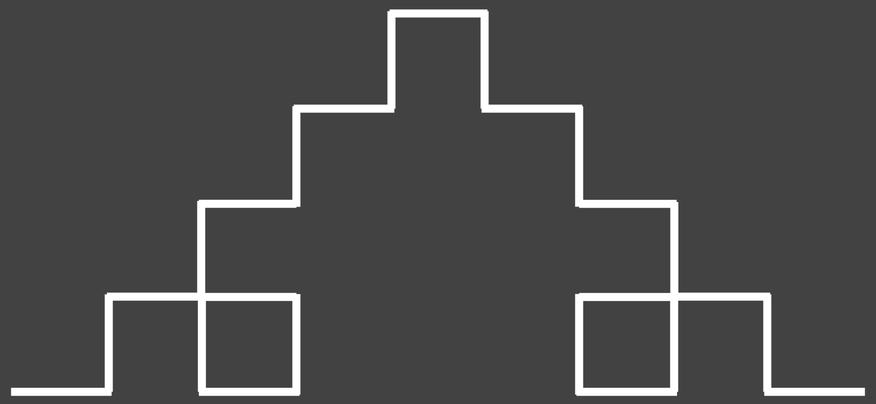
Production rules

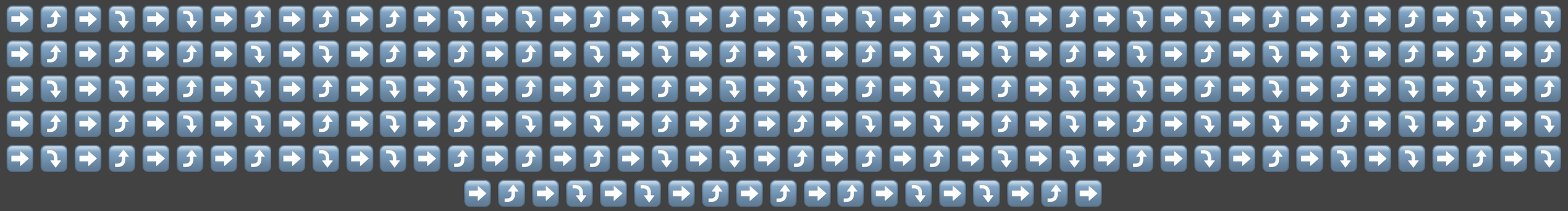
➡️

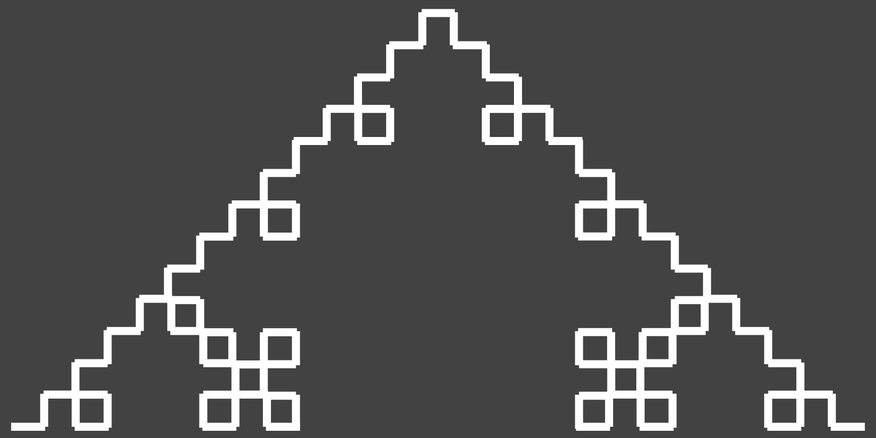
➡️

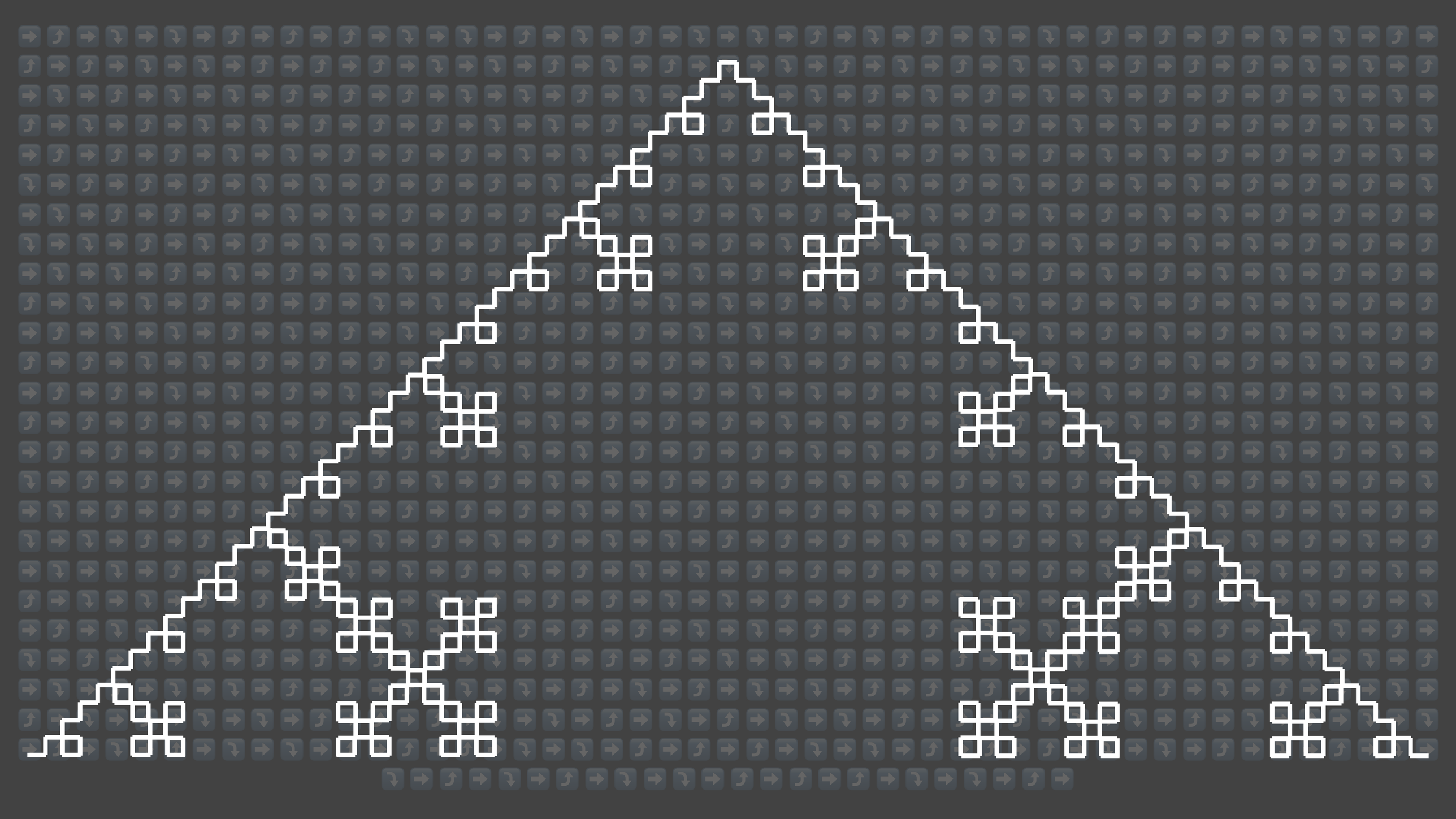
➡️ ⇒ ➡️ ⤴️ ➡️ ⤵️ ➡️ ⤵️ ➡️ ⤴️ ➡️







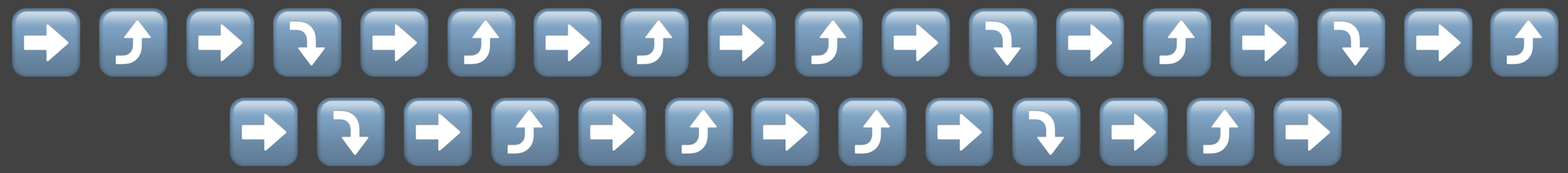


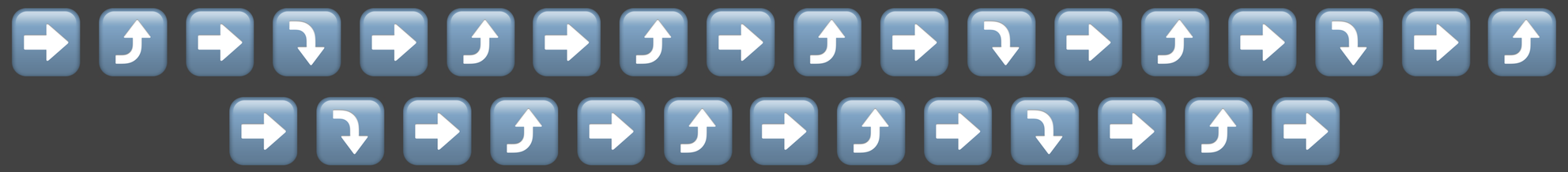


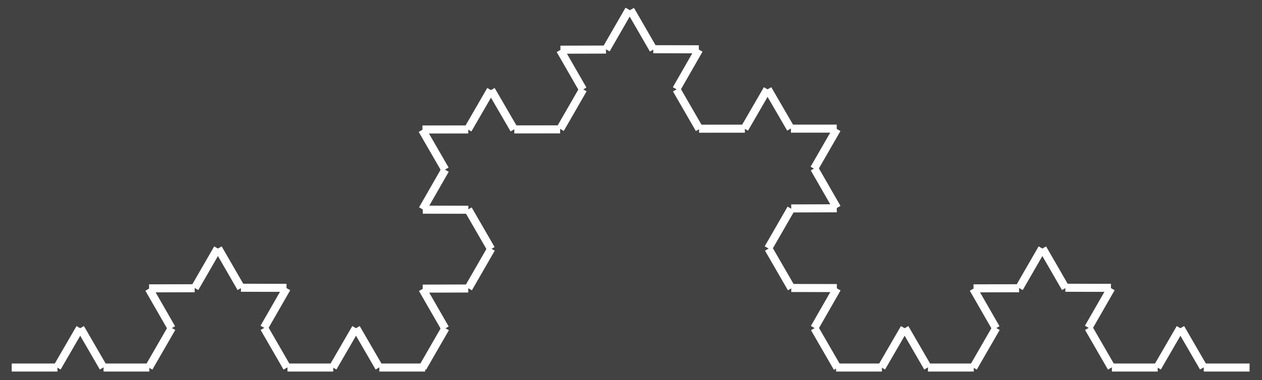
Exploration

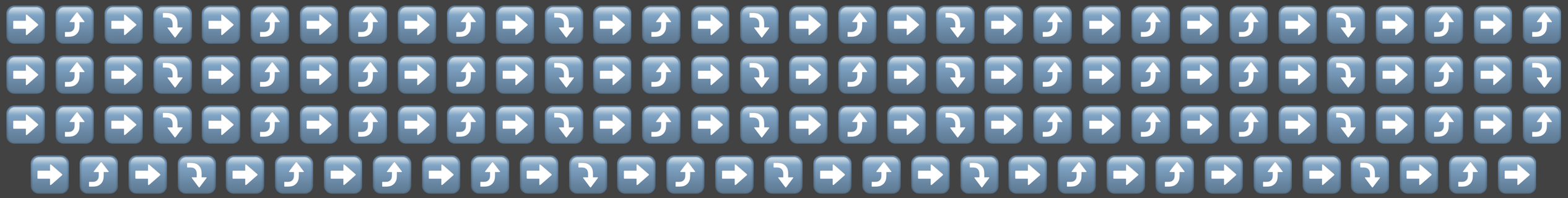
➡️

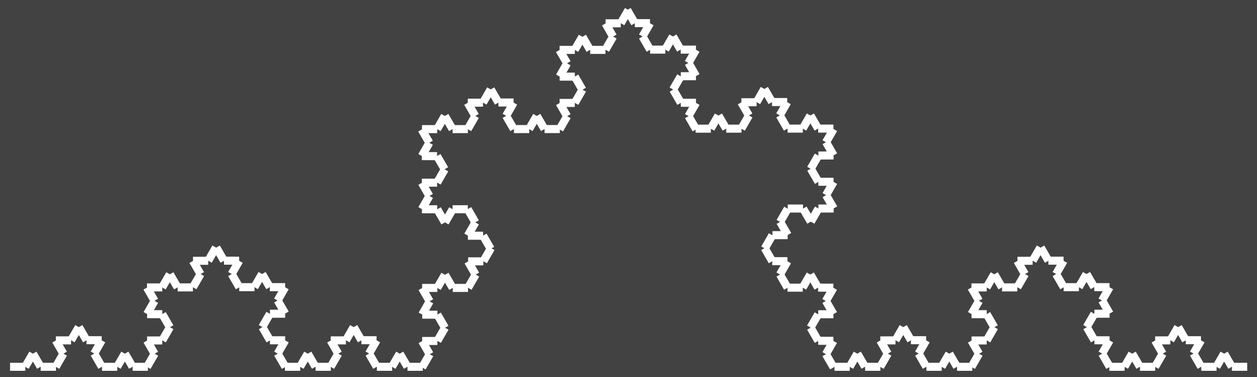
➡️ ⇒ ➡️ ⤴️ ➡️ ⤵️ ➡️ ⤴️ ➡️

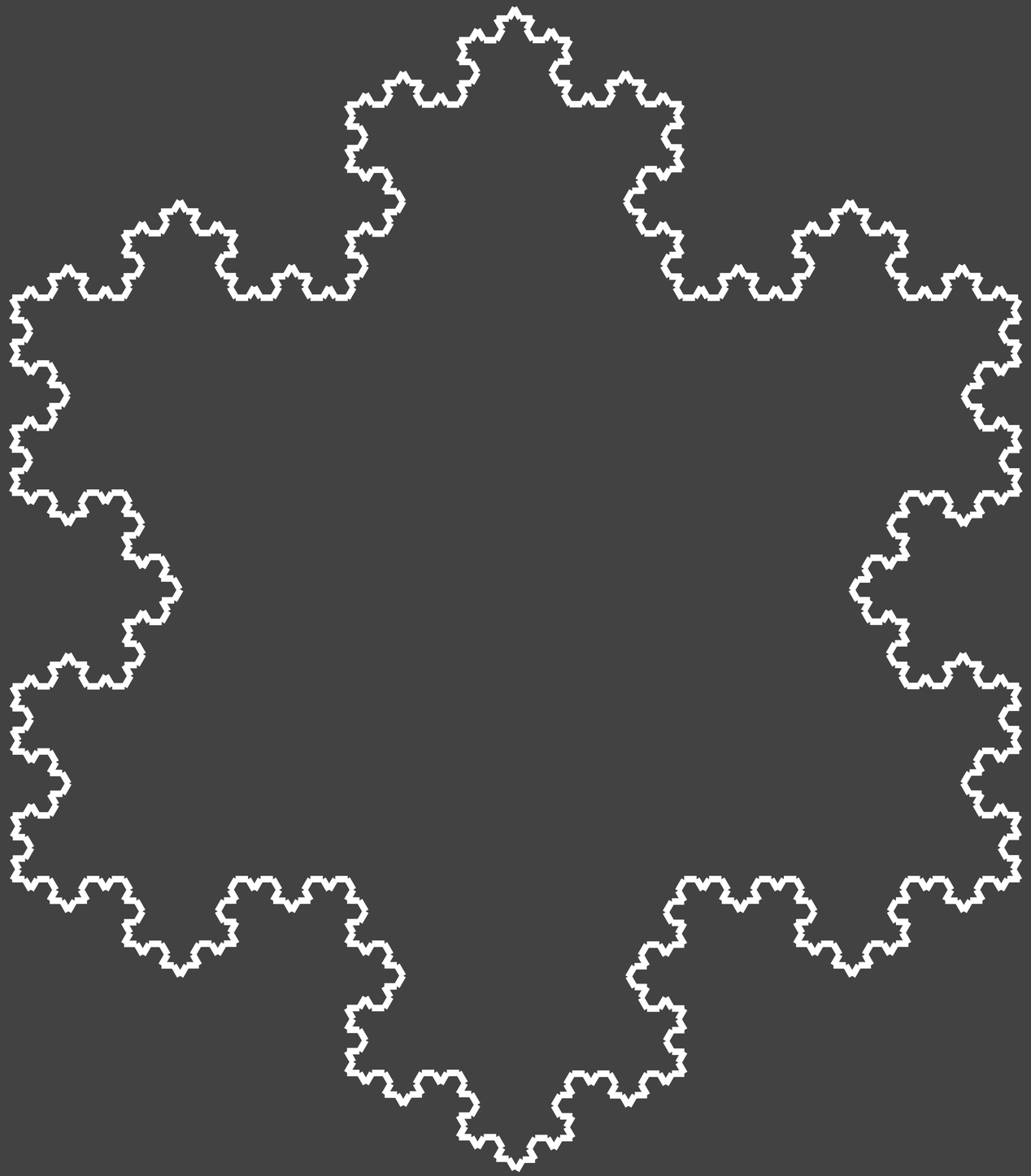












Alphabet

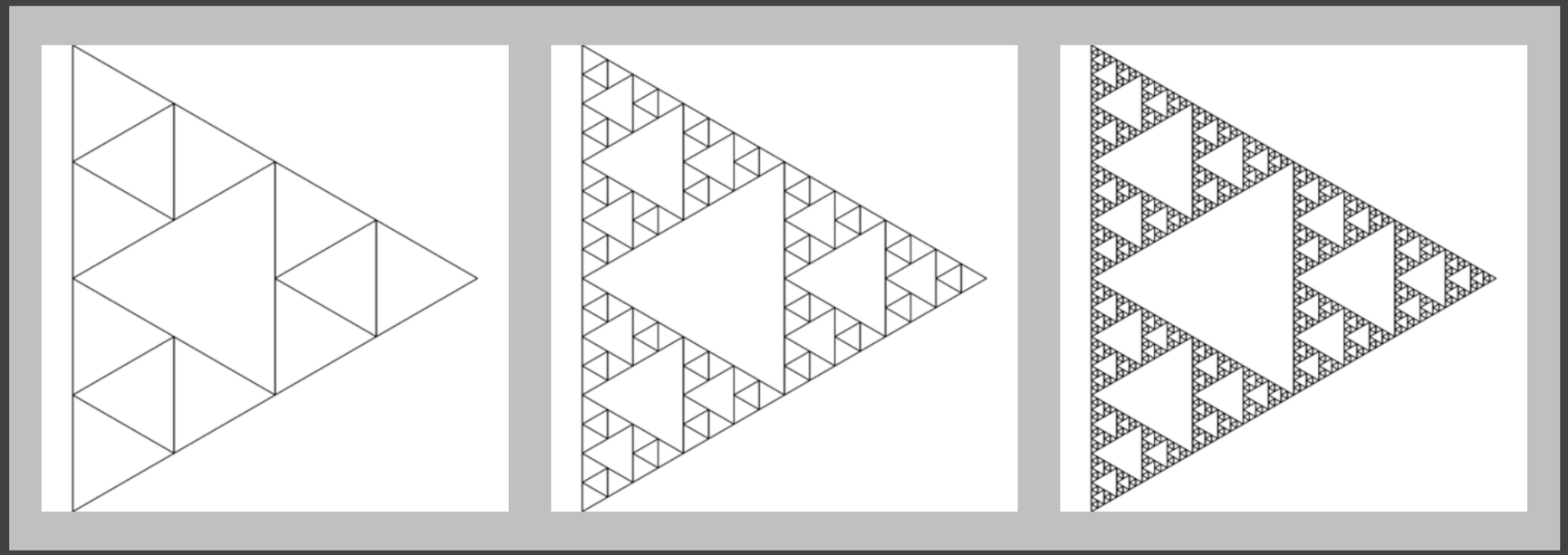
➡️ ▶️ ⤴️ ⤵️

Initiator

➡️ ⤴️ ▶️ ⤴️ ▶️

Production rules

(➡️ ⇒ ➡️ ⤴️ ▶️ ⤵️ ➡️ ⤵️ ▶️ ⤴️ ➡️) , (▶️ ⇒ ▶️ ▶️)



Sierlinski triangle, © Lingo Loco, CC-BY-SA 4.0, https://en.wikipedia.org/wiki/L-system

Alphabet

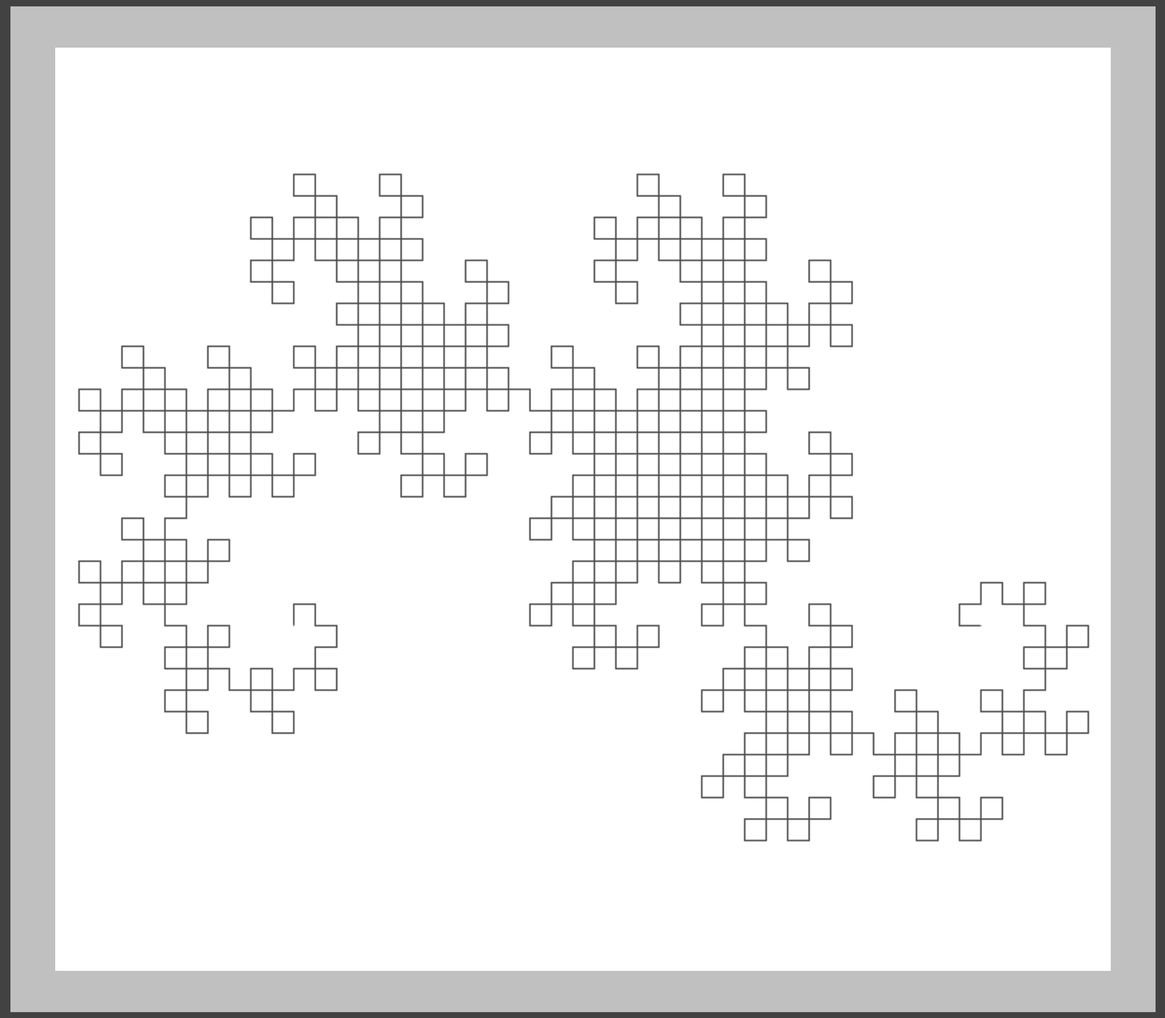
➡️ ⤴️ ⤵️ ❇️ ✴️

Initiator

➡️ ❇️

Production rules

(❇️ ⇒ ❇️ ⤴️ ✴️ ➡️ ⤴️) , (✴️ ⇒ ⤵️ ➡️ ❇️ ⤵️ ✴️)



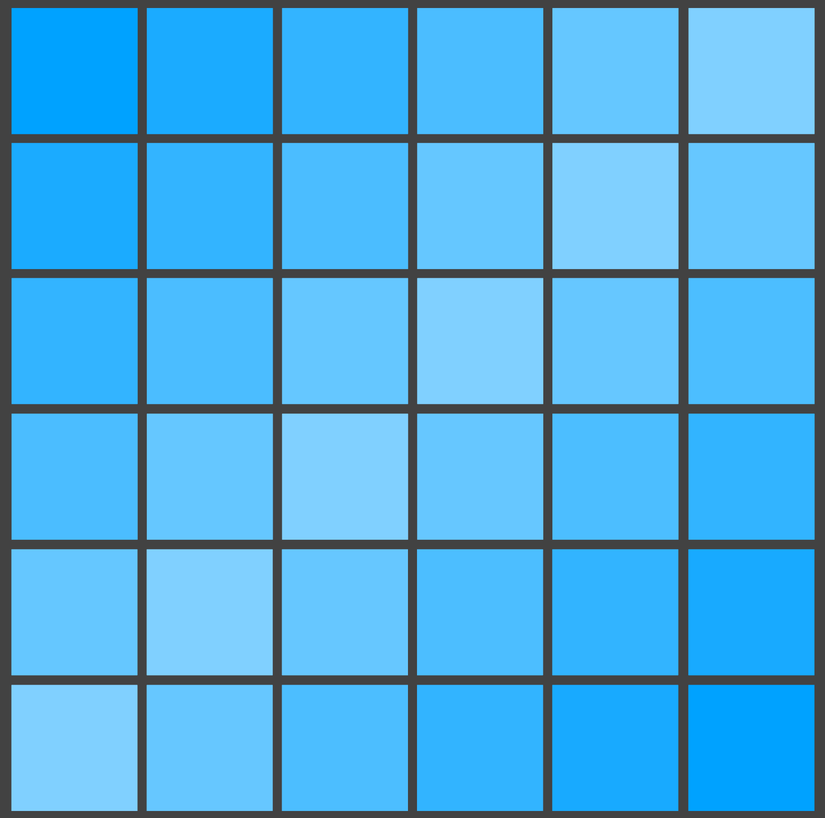
Dragon curve, António Miguel de Campos, Public domain, https://en.wikipedia.org/wiki/L-system

Fractal trees (drawn using dragon curves), Solkoll, Public domain, https://en.wikipedia.org/wiki/L-system

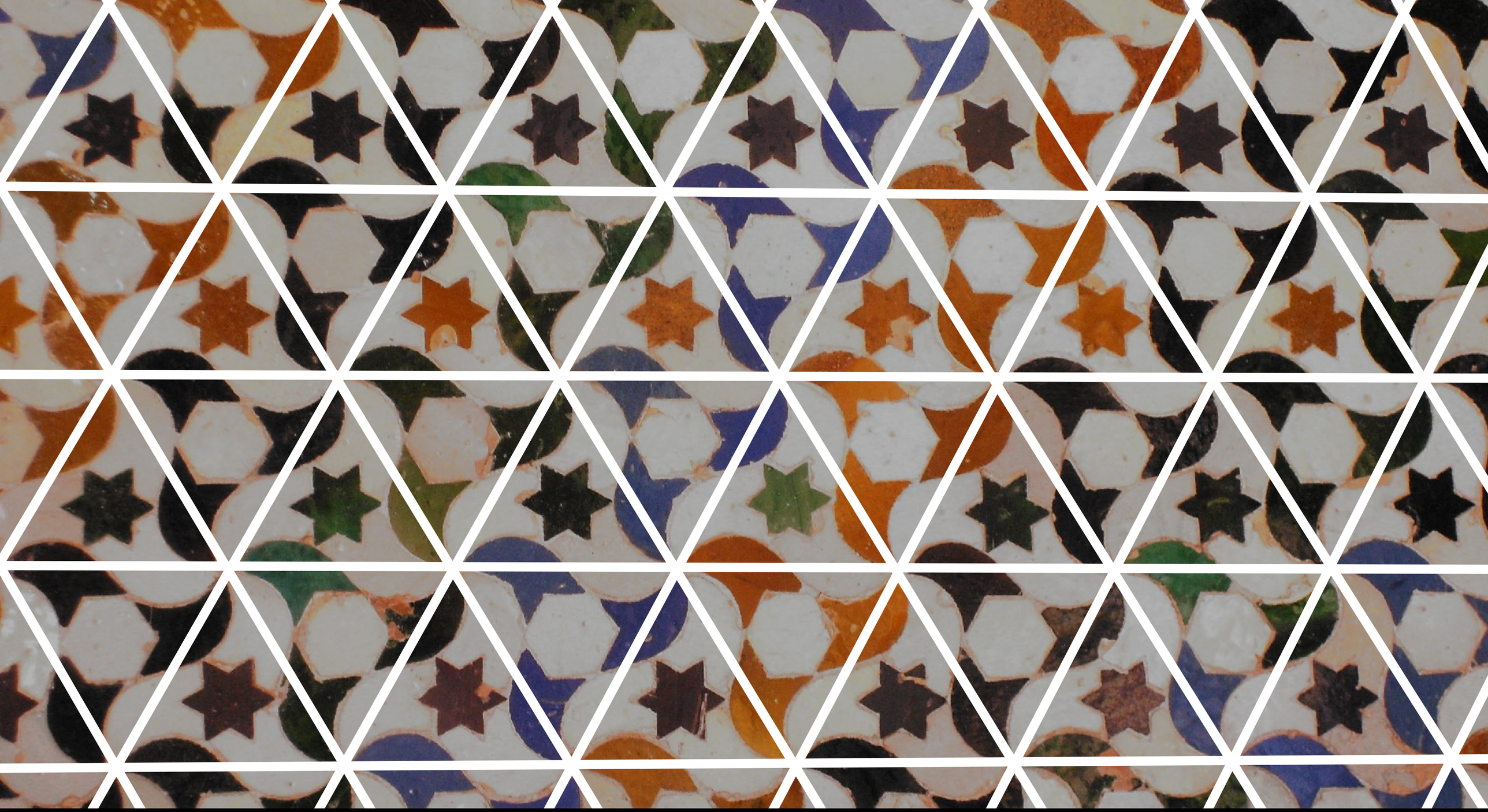
The power of abstraction

Mathematical power

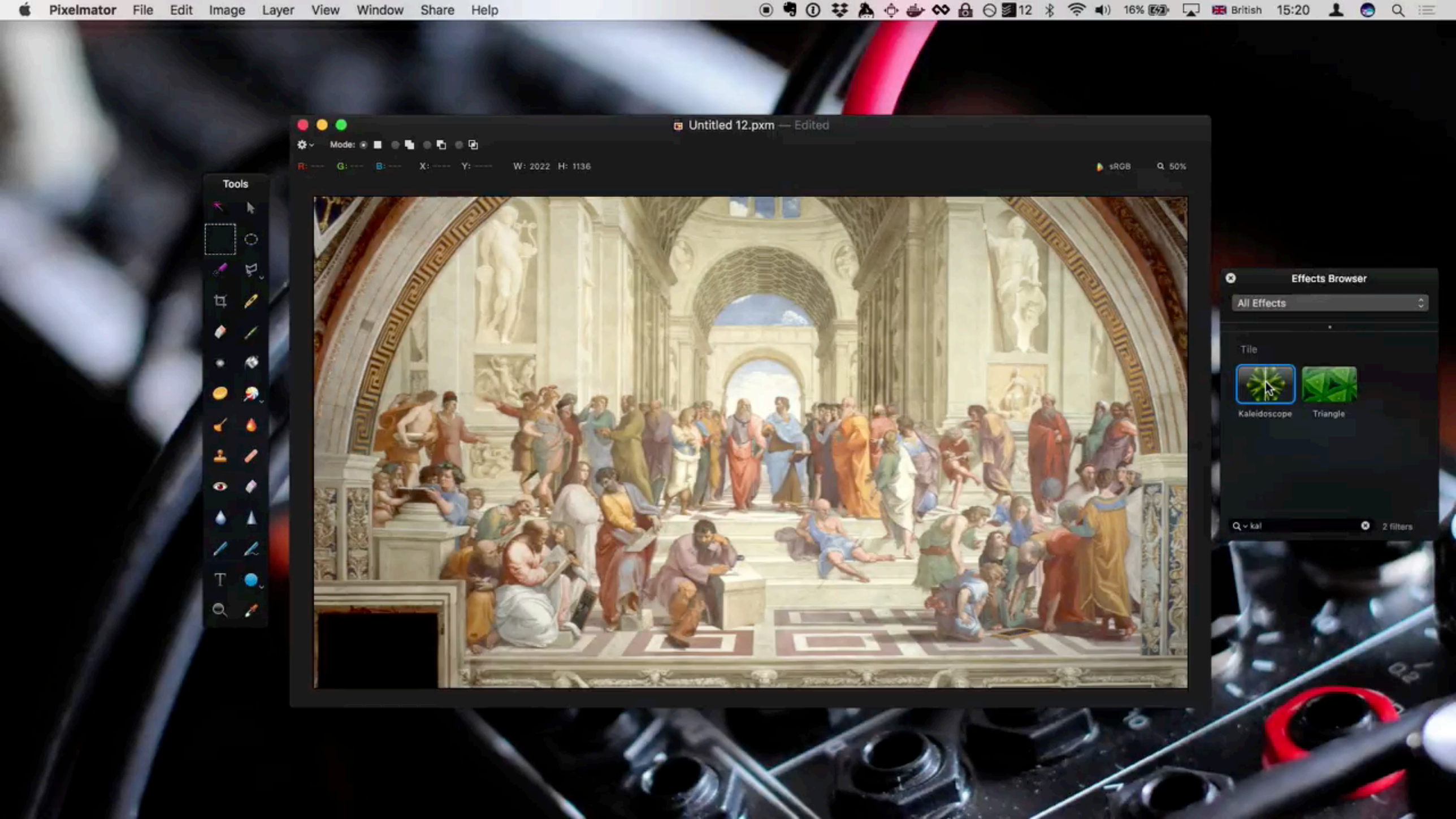
“Creative” power



13th Century Design, Alhambra, Grenada, © Rogiro, CC-BY-NC-SA 2.0, https://flic.kr/p/6JxRsQ



13th Century Design, Alhambra, Grenada, © Rogiro, CC-BY-NC-SA 2.0, https://flic.kr/p/6JxRsQ


Pixelmator File Edit Image Layer View Window Share Help
16% British 15:20
Untitled 12.pxm — Edited
Mode:
R: --- G: --- B: --- X: ---- Y: ----
W: 2022 H: 1136
sRGB
50%
Tools
Effects Browser
All Effects
Tile
Kaleidoscope
Triangle
kal
2 filters

Fractal trees (drawn using dragon curves), Solkoll, Public domain, https://en.wikipedia.org/wiki/L-system

Abstracting over this talk

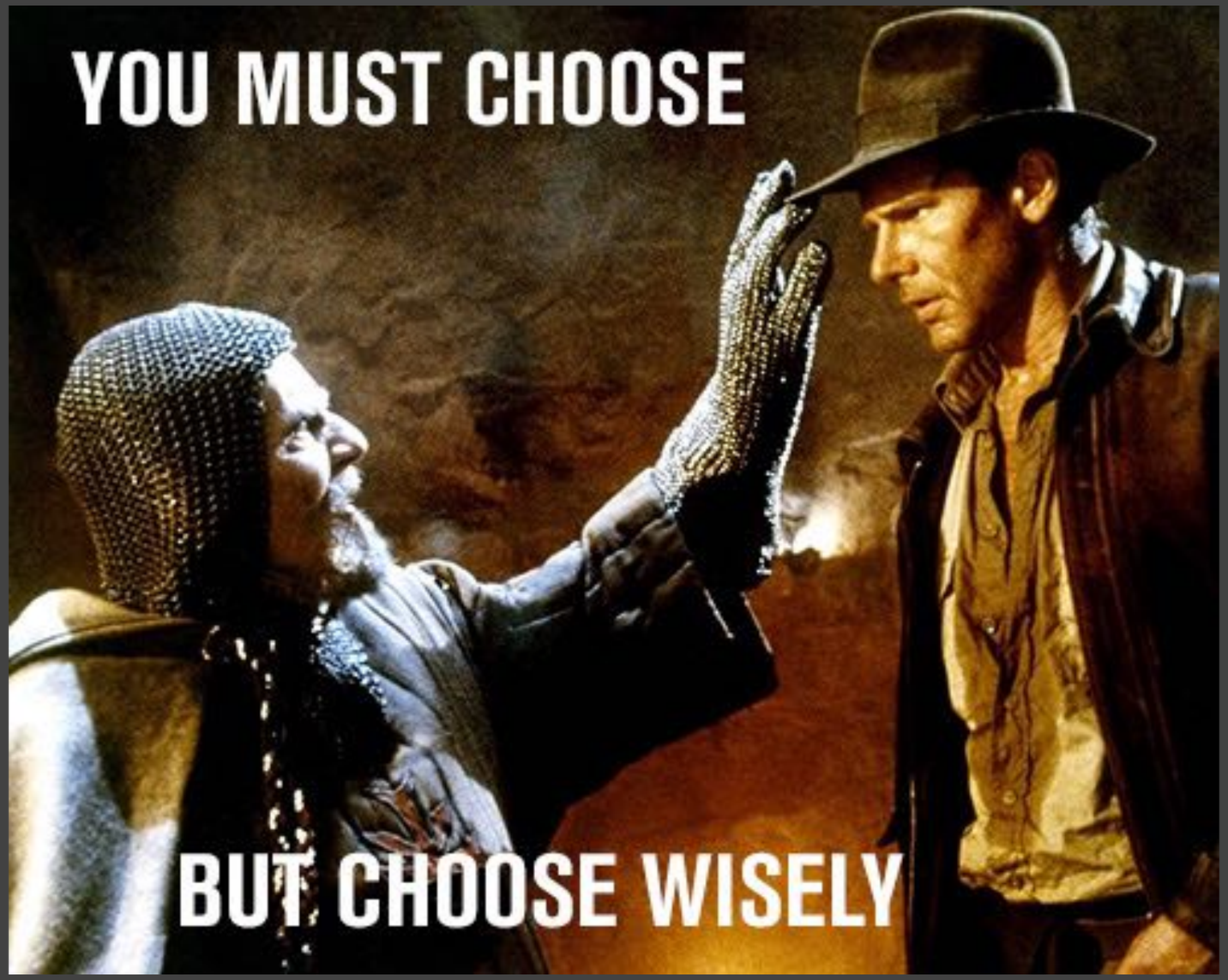
Maths is about abstraction

Find common properties
Remove irrelevant differences

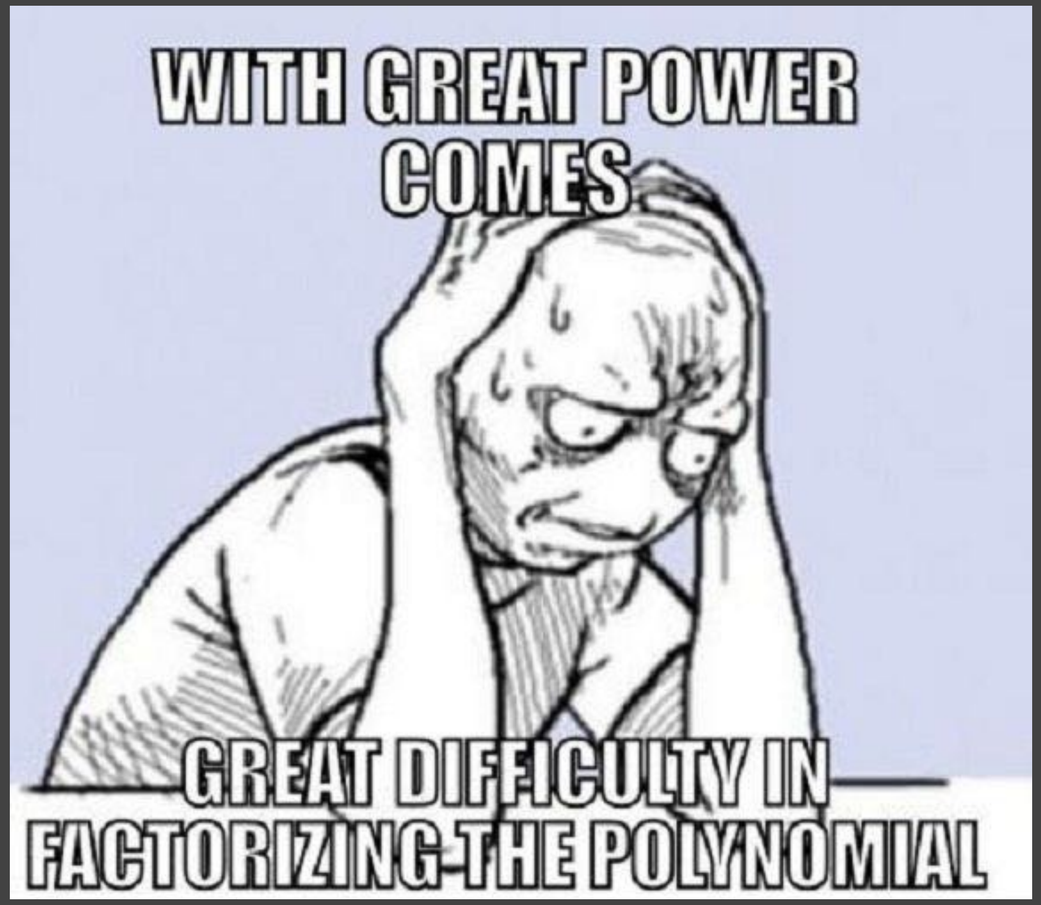
Abstractions impose restrictions

Restrictions simplify

Restrictions inspire



Indiana Jones and the Last Crusade, © 1989 Lucasfilm Ltd.


WITH GREAT POWER
COMES
GREAT DIFFICULTY IN
FACTORIZING THE POLYNOMIAL

The School of Athens, © 1509-1511 Raphael, Public Domain, https://en.wikipedia.org/wiki/The_School_of_Athens

Unknown artist, Cats on Syntheziers in Space, http://www.catsonsynthesizersinspace.com/image/142412968507

Thank you!

Dave Gurnell @davegurnell
Noel Welsh @noelwelsh