Knitted Monads

A desperate attempt to talk about Haskell at Maths Jam

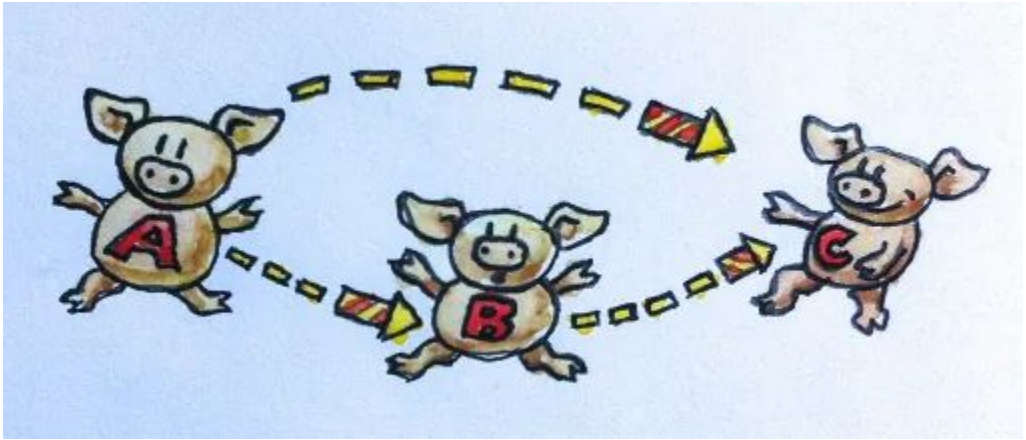
Stitch diagram

Why Haskell?

- Purely Functional language
- No-side effects
- Referentially transparent
- Type algebra
- Category Theory

Category Theory

A category consists of objects and arrows (morphisms) that go between them



Category Theory terms turn out to be great abstractions for programming - particularly concepts like the “monad”

“a monad is a monoid in the category of endofunctors, what's the problem?”

James Iry, incorrectly attributing a quote to Philip Wadler in *A Brief, Incomplete, and Mostly Wrong History of Programming Languages*

18.2 Sorry – a monad is not a burrito

Well, then what the heck is a monad?[1]

As we said above, a monad is an applicative functor with some unique features that make it a bit more powerful than either alone. A functor maps a function over some structure; an applicative maps a

Burritos for the Hungry Mathematician

Ed Morehouse

April 1, 2015

Abstract

The advent of fast-casual Mexican-style dining establishments, such as Chipotle and Qdoba, has greatly improved the productivity of research mathematicians and theoretical computer scientists in recent years. Still, many experience confusion upon encountering burritos for the first time.

Numerous burrito tutorials (of varying quality) are to be found on the Internet. Some describe a burrito as the image of a crêpe under the action of the new-world functor. But such characterizations merely serve to reindex the confusion contravariantly. Others insist that the only way to really understand burritos is to eat many different kinds of burrito, until the common underlying concept becomes apparent.

It has been recently remarked by Yorgey [9] that a burrito can be regarded as an instance of a universally-understood concept, namely, that of monad. It is this characterization that we intend to explicate here. To wit, *a burrito is just a strong monad in the symmetric monoidal category of food, what's the problem?*

- The Category of Food
- The Tortilla Endofunctor
- The Burrito Monad

Steps

1. Generate a 2-3 colour “palette”
2. Divide the image into rectangles
3. Get the average colour for each rectangle
4. Select the closest palette colour to each rectangle’s average

Interesting Maths No.0 - Quantization

- Used for displaying images with lots of colours on devices that can only display a few.
- Most popular algorithm: “median cut”

R: 255, G:0, B:0

R: 220, G:150, B:0

R: 255, G:0, B:100

R:0, G: 255, B: 0

R:0, G:255, B:150

R: 0, G: 0, B: 255

R: 80, G: 80, B:80

R: 255, G:0, B:0

R: 255, G:0, B:100

R: 220, G:150, B:0

R: 80, G: 80, B:80

R:0, G: 255, B: 0

R:0, G:255, B:150

R:0, G: 0, B: 255

R: 255, G:0, B:0

R: 255, G:0, B:100

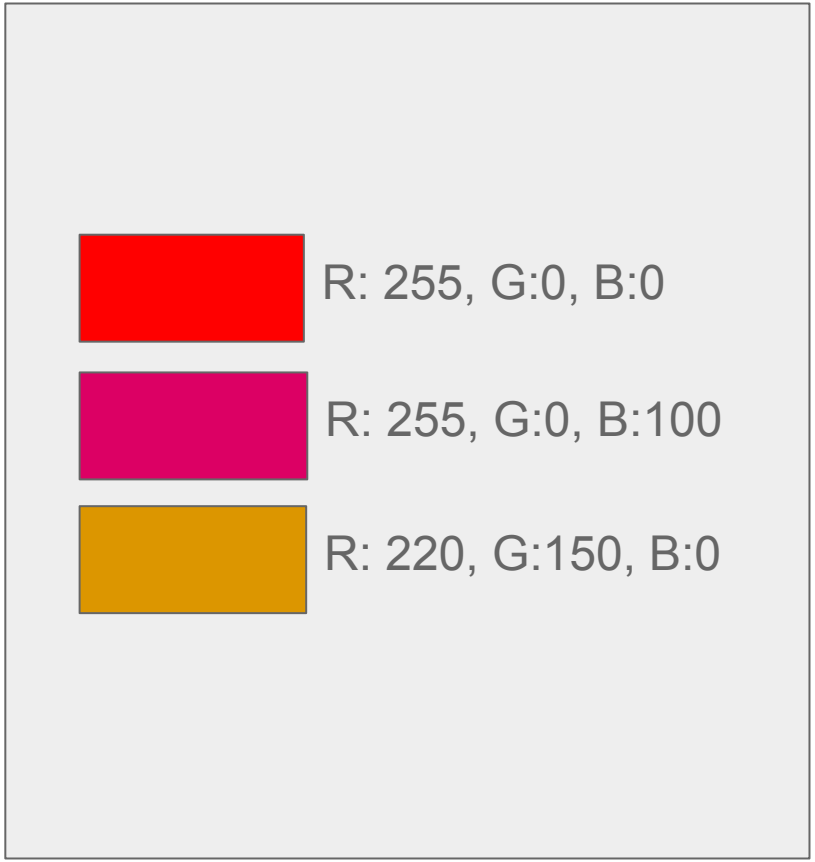
R: 220, G:150, B:0

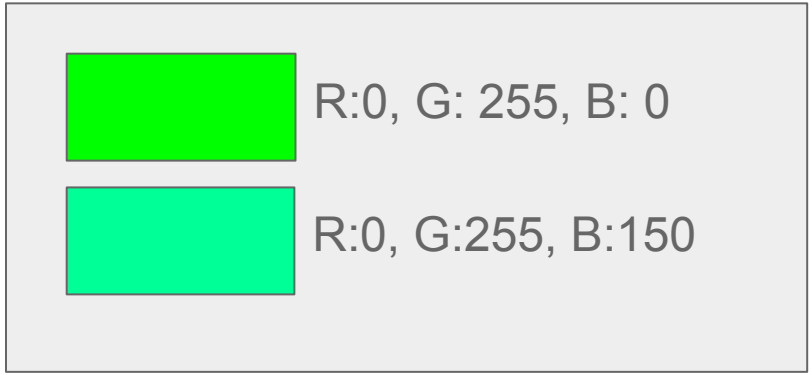
R:0, G: 255, B: 0

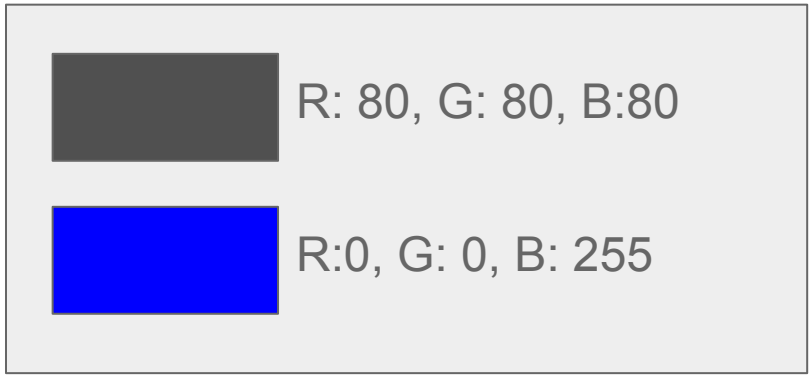
R:0, G:255, B:150

R: 80, G: 80, B:80

R:0, G: 0, B: 255


R: 255, G:0, B:0
R: 255, G:0, B:100
R: 220, G:150, B:0


R:0, G: 255, B: 0
R:0, G:255, B:150


R: 80, G: 80, B:80
R:0, G: 0, B: 255

R: 0, G:255, B:106

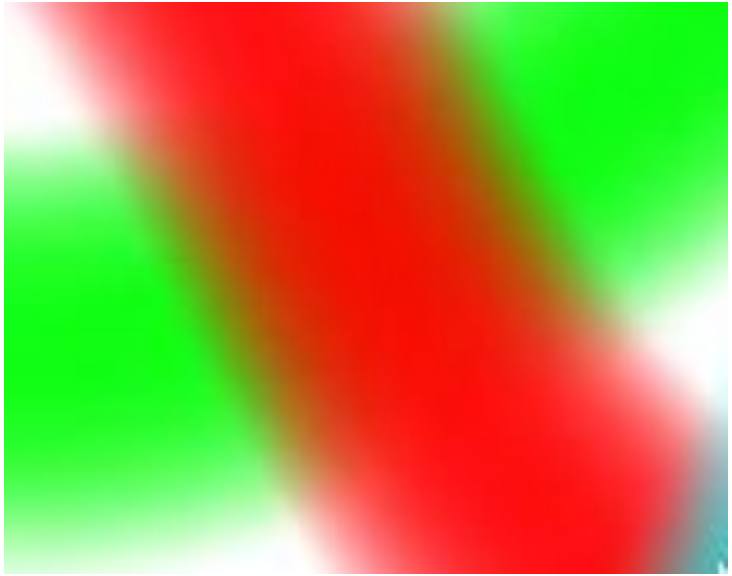
R: 244, G:87, B:58

R: 57, G:57, B:189

Maths no.1 - how do you average colour?

This is wrong!

$$\frac{\Sigma r}{p}, \frac{\Sigma g}{p}, \frac{\Sigma b}{p}$$



The right way to average colour

1. Sum the squares of the Red, Green and Blue values
2. Divide by the number of pixels
3. Square root back the result

$$\sqrt{\frac{\Sigma r^2}{p}}, \sqrt{\frac{\Sigma g^2}{p}}, \sqrt{\frac{\Sigma b^2}{p}}$$

Why?

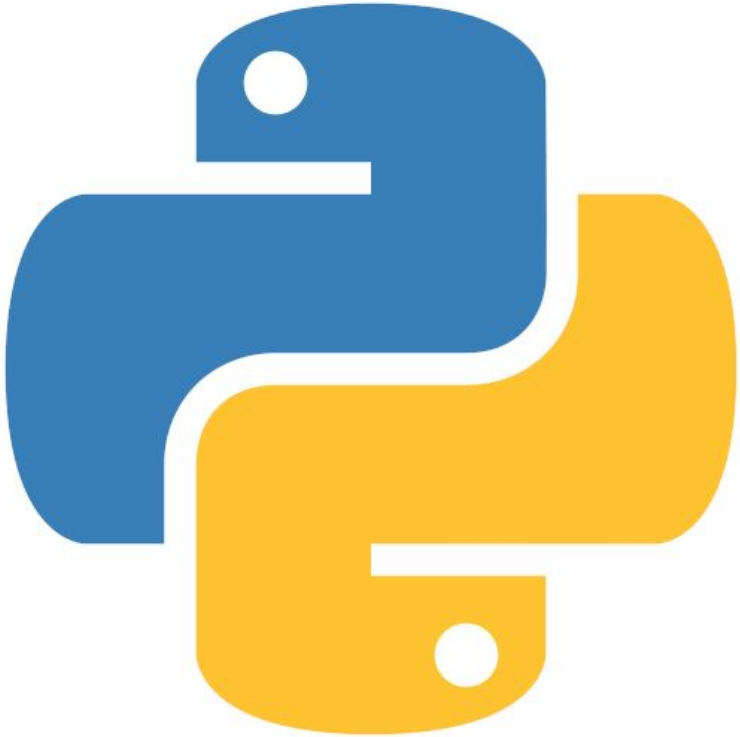
R: 125, G: 125, B: 125

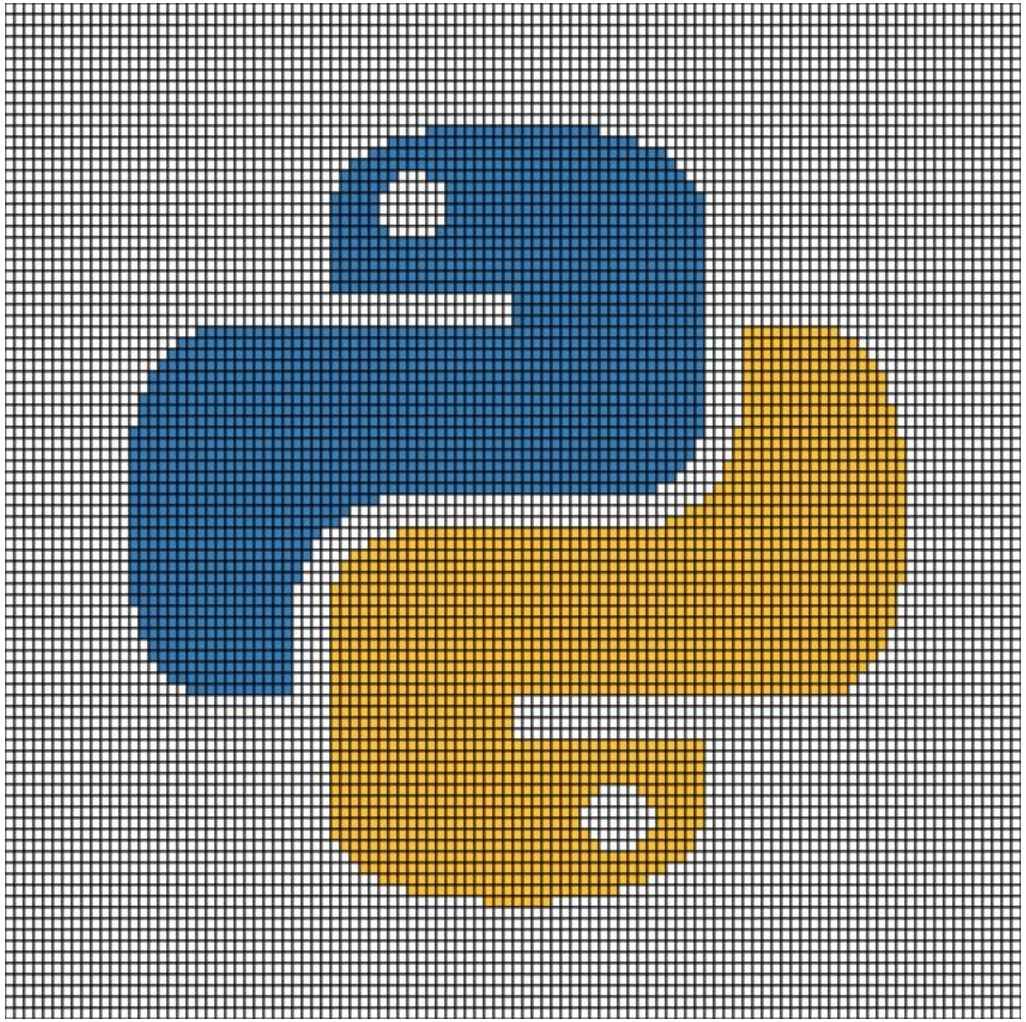
R: 250, G: 250, B: 250

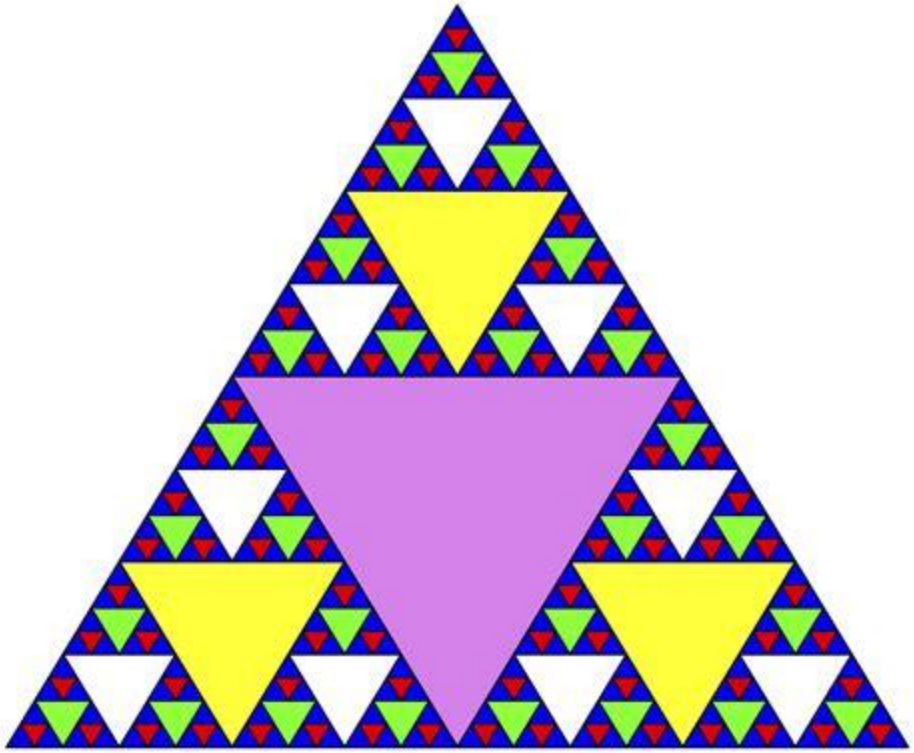
This is 4x brighter (not 2x)

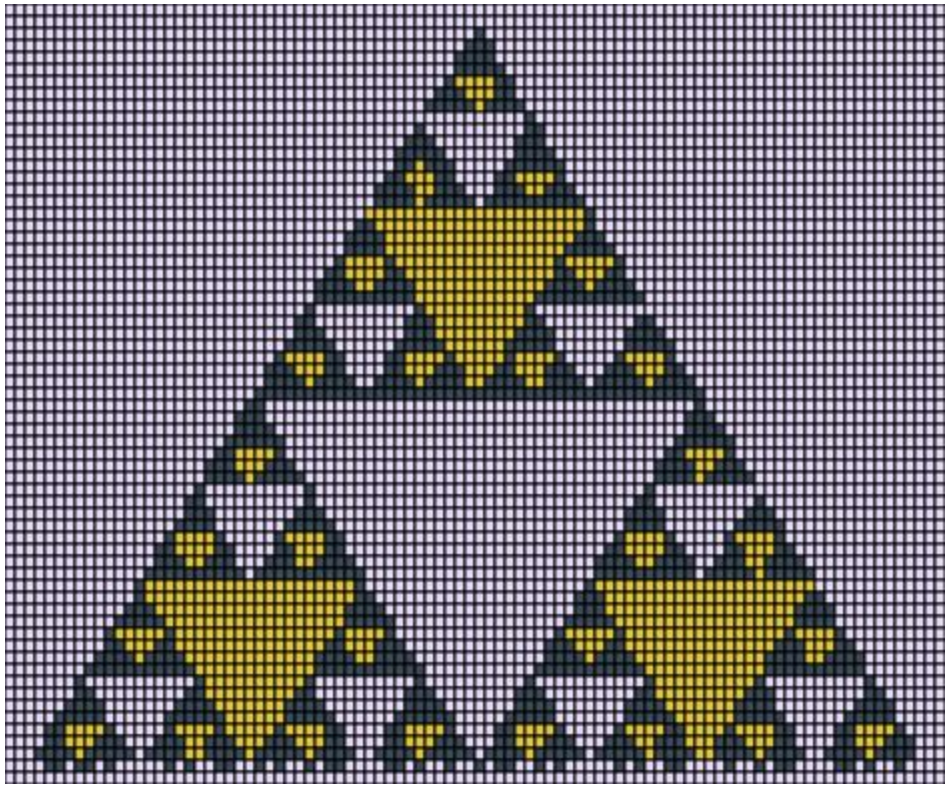
Starting with this

Python

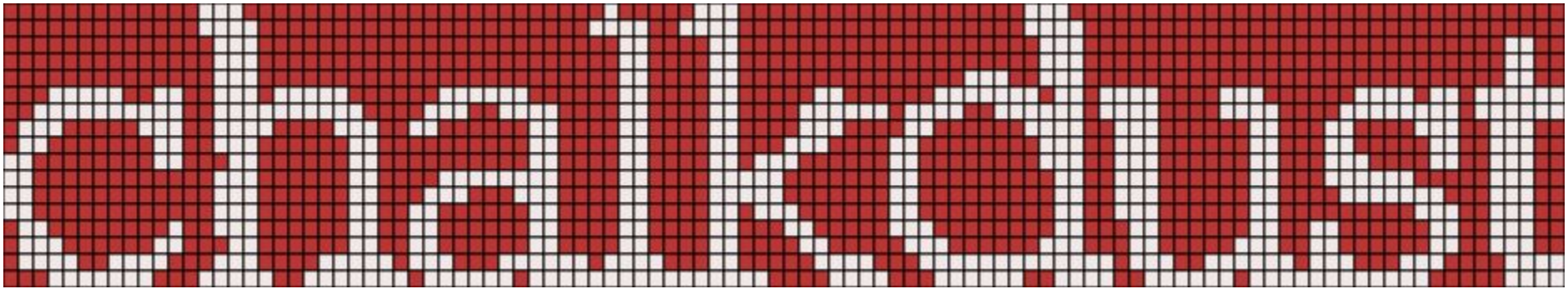


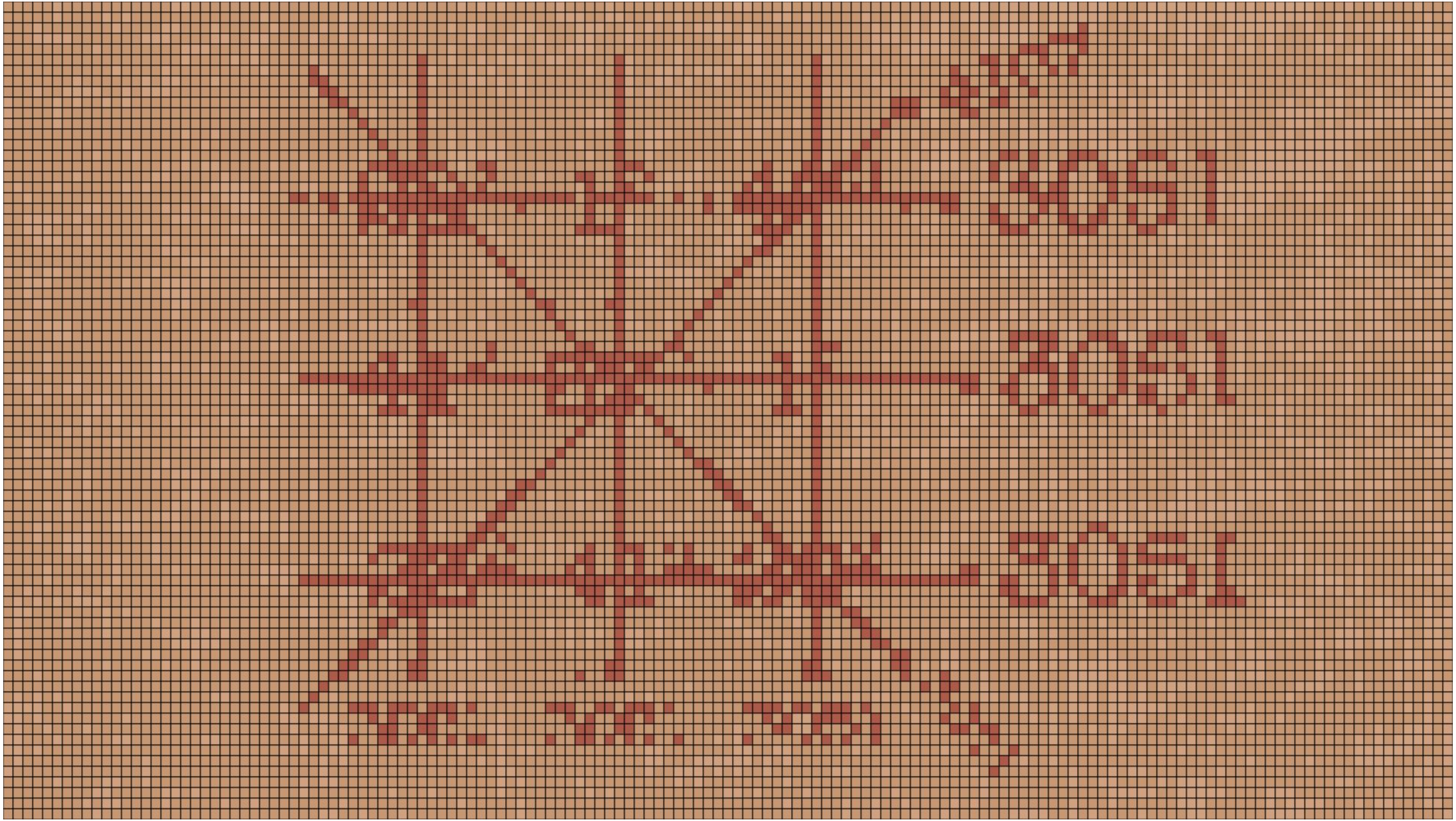







chalkdust


chalkdust



My code is on github

https://github.com/RaphaelColman/hintarsia

References

Category Theory for Programmers, Bartosz Milewski:
https://bartoszmilewski.com/2014/10/28/category-theory-for-programmers-the-preface/

Haskell Programming from first principles, Christopher Allen,

Julie Moronuki

Minute physics: Computer color is broken:
https://www.youtube.com/watch?v=LKnqECcg6Gw