

# KNITTED MONADS

A desperate attempt to talk about Haskell at Maths Jam







# STITCH DIAGRAM



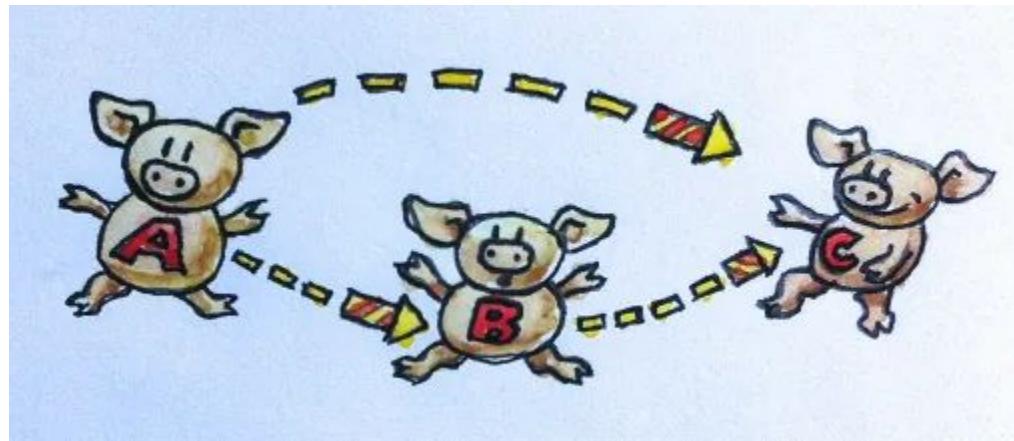
# WHY HASKELL?



- 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*

# Haskell Programming

from first principles

Christopher Allen  
Julie Moronuki

## 18.2 Sorry – a monad is not a burrito

Well, then what the heck is a monad?<sup>1</sup>

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



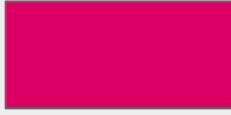
R: 80, G: 80, B:80



R:0, G: 0, B: 255



R: 255, G:0, B:0



R: 255, G:0, B:100



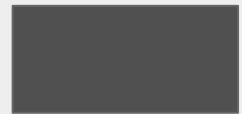
R: 220, G:150, B:0



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: 244, G:87, B:58



R: 0, G:255, B:106



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

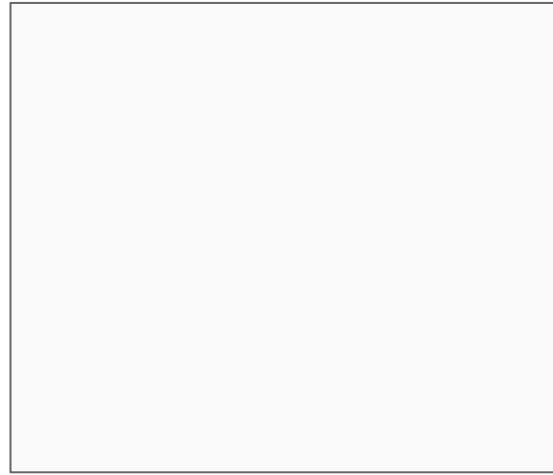
$$\sqrt{\frac{\Sigma r^2}{p}}, \sqrt{\frac{\Sigma g^2}{p}}, \sqrt{\frac{\Sigma b^2}{p}}$$

# WHY?

R: 125, G: 125, B: 125



R: 250, G: 250, B: 250



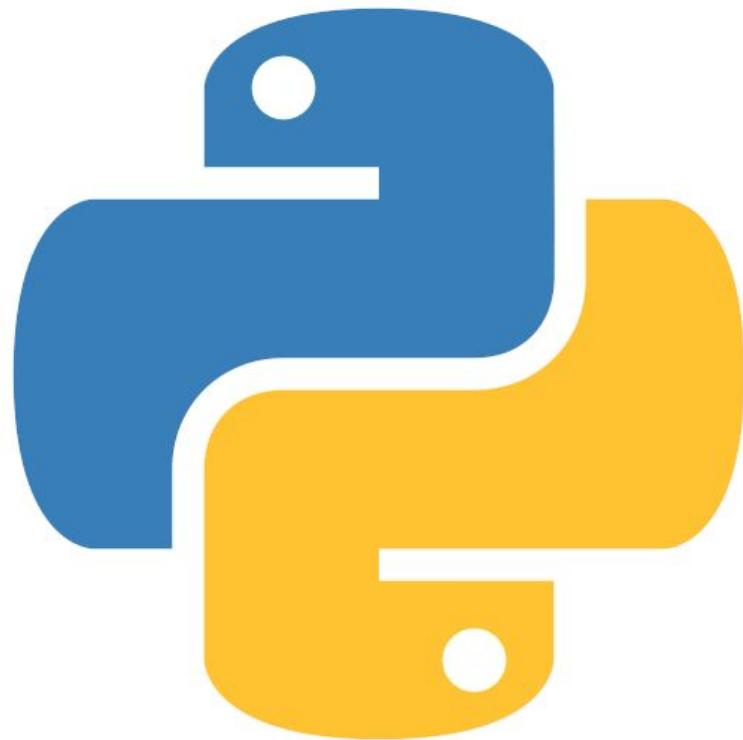
This is 4x brighter (not 2x)

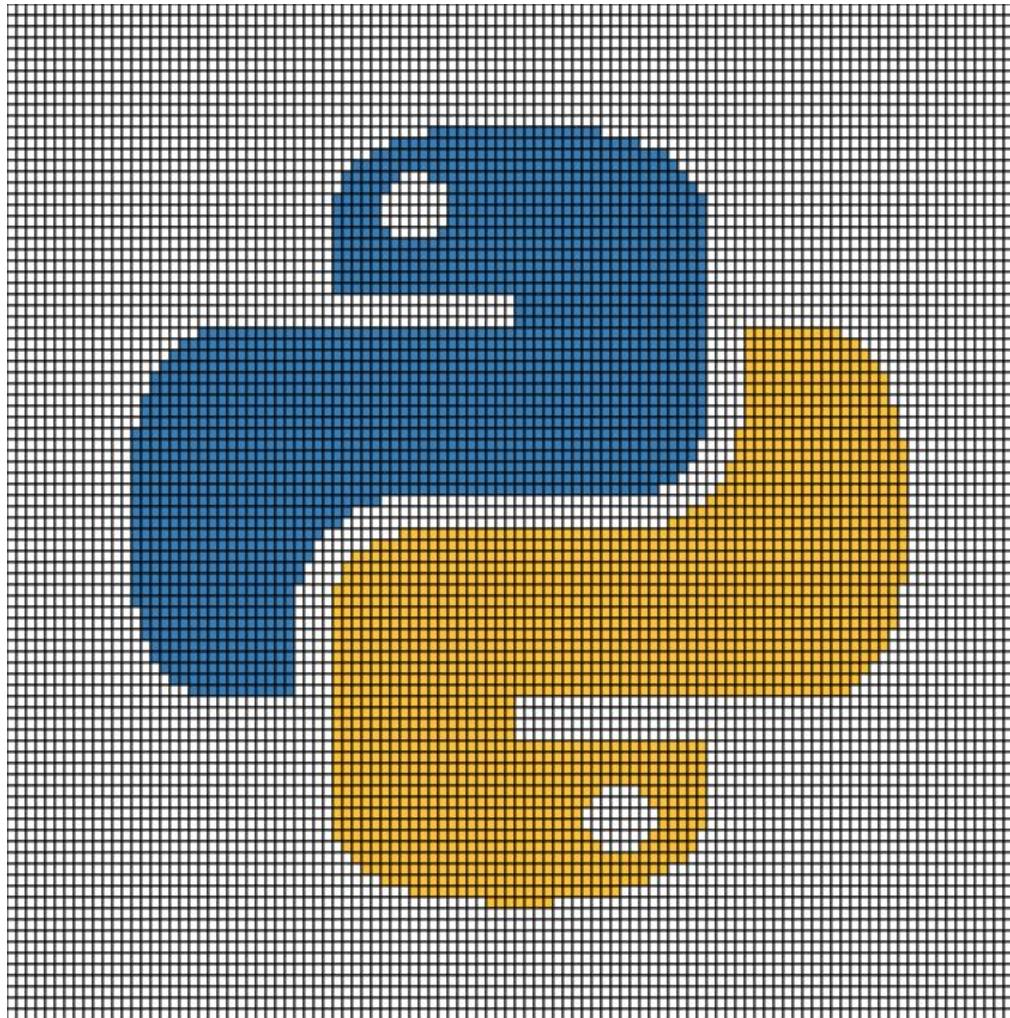
Starting with this

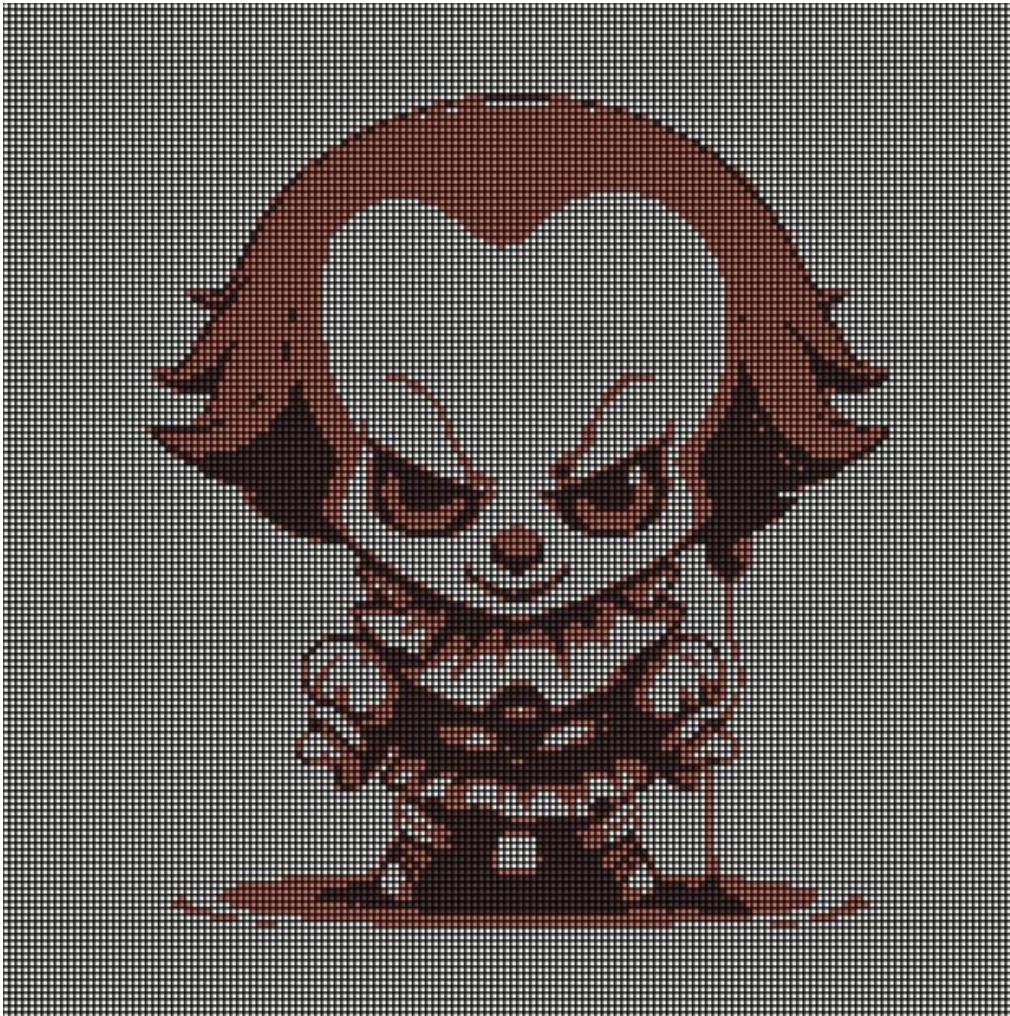


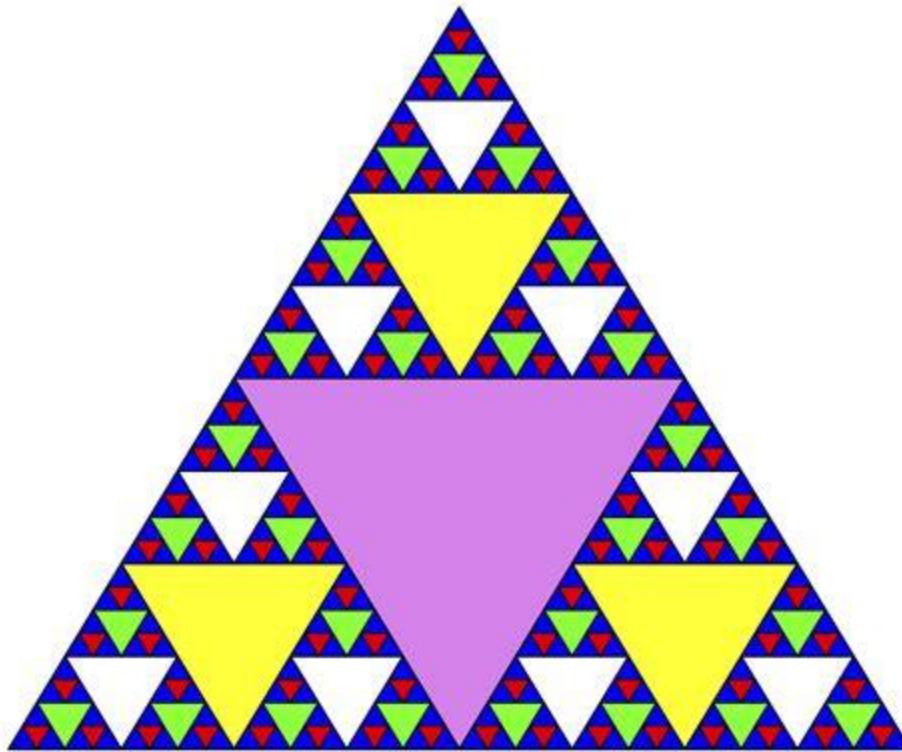


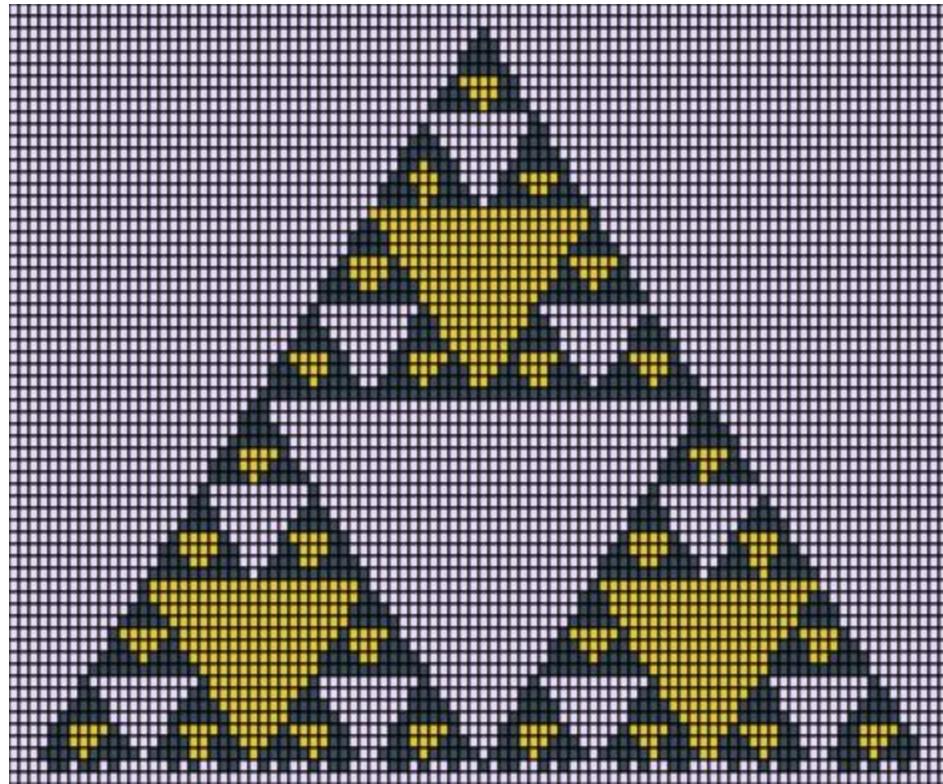
Python



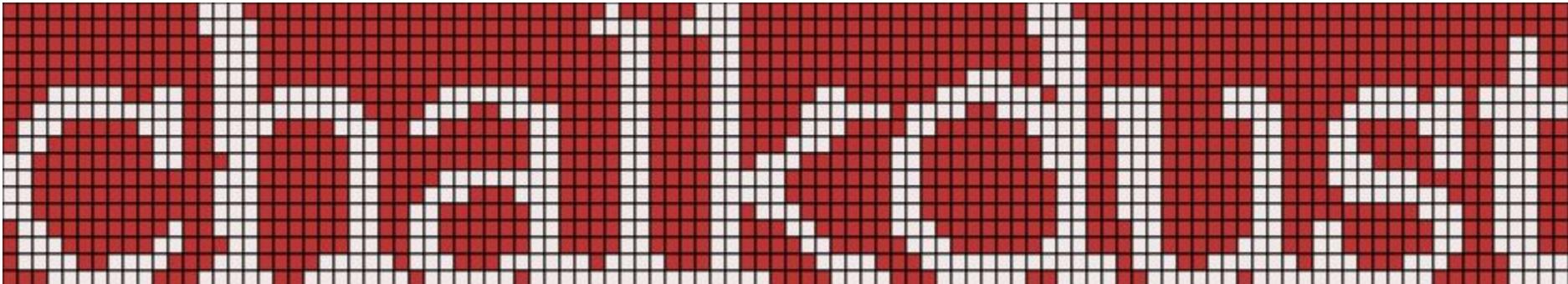








# chalkdust



THE DUE DATES ARE AS FOLLOWS:

LEVEL 1: 2001-06-20

LEVEL 2: 2001-07-15

LEVEL 3: 2001-08-15

LEVEL 4: 2001-09-15

LEVEL 5: 2001-10-15

LEVEL 6: 2001-11-15

LEVEL 7: 2001-12-15

LEVEL 8: 2002-01-15

LEVEL 9: 2002-02-15

LEVEL 10: 2002-03-15

LEVEL 11: 2002-04-15

LEVEL 12: 2002-05-15

LEVEL 13: 2002-06-15

LEVEL 14: 2002-07-15

LEVEL 15: 2002-08-15

LEVEL 16: 2002-09-15

LEVEL 17: 2002-10-15

LEVEL 18: 2002-11-15

LEVEL 19: 2002-12-15

LEVEL 20: 2003-01-15

LEVEL 21: 2003-02-15

LEVEL 22: 2003-03-15

LEVEL 23: 2003-04-15

LEVEL 24: 2003-05-15

LEVEL 25: 2003-06-15

LEVEL 26: 2003-07-15

LEVEL 27: 2003-08-15

LEVEL 28: 2003-09-15

LEVEL 29: 2003-10-15

LEVEL 30: 2003-11-15

LEVEL 31: 2003-12-15

LEVEL 32: 2004-01-15

LEVEL 33: 2004-02-15

LEVEL 34: 2004-03-15

LEVEL 35: 2004-04-15

LEVEL 36: 2004-05-15

LEVEL 37: 2004-06-15

LEVEL 38: 2004-07-15

LEVEL 39: 2004-08-15

LEVEL 40: 2004-09-15

LEVEL 41: 2004-10-15

LEVEL 42: 2004-11-15

LEVEL 43: 2004-12-15

LEVEL 44: 2005-01-15

LEVEL 45: 2005-02-15

LEVEL 46: 2005-03-15

LEVEL 47: 2005-04-15

LEVEL 48: 2005-05-15

LEVEL 49: 2005-06-15

LEVEL 50: 2005-07-15

LEVEL 51: 2005-08-15

LEVEL 52: 2005-09-15

LEVEL 53: 2005-10-15

LEVEL 54: 2005-11-15

LEVEL 55: 2005-12-15

LEVEL 56: 2006-01-15

LEVEL 57: 2006-02-15

LEVEL 58: 2006-03-15

LEVEL 59: 2006-04-15

LEVEL 60: 2006-05-15

LEVEL 61: 2006-06-15

LEVEL 62: 2006-07-15

LEVEL 63: 2006-08-15

LEVEL 64: 2006-09-15

LEVEL 65: 2006-10-15

LEVEL 66: 2006-11-15

LEVEL 67: 2006-12-15

LEVEL 68: 2007-01-15

LEVEL 69: 2007-02-15

LEVEL 70: 2007-03-15

LEVEL 71: 2007-04-15

LEVEL 72: 2007-05-15

LEVEL 73: 2007-06-15

LEVEL 74: 2007-07-15

LEVEL 75: 2007-08-15

LEVEL 76: 2007-09-15

LEVEL 77: 2007-10-15

LEVEL 78: 2007-11-15

LEVEL 79: 2007-12-15

LEVEL 80: 2008-01-15

LEVEL 81: 2008-02-15

LEVEL 82: 2008-03-15

LEVEL 83: 2008-04-15

LEVEL 84: 2008-05-15

LEVEL 85: 2008-06-15

LEVEL 86: 2008-07-15

LEVEL 87: 2008-08-15

LEVEL 88: 2008-09-15

LEVEL 89: 2008-10-15

LEVEL 90: 2008-11-15

LEVEL 91: 2008-12-15

LEVEL 92: 2009-01-15

LEVEL 93: 2009-02-15

LEVEL 94: 2009-03-15

LEVEL 95: 2009-04-15

LEVEL 96: 2009-05-15

LEVEL 97: 2009-06-15

LEVEL 98: 2009-07-15

LEVEL 99: 2009-08-15

LEVEL 100: 2009-09-15

My code is on github

[https://github.com/RaphaelColman/  
hintarsia](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>