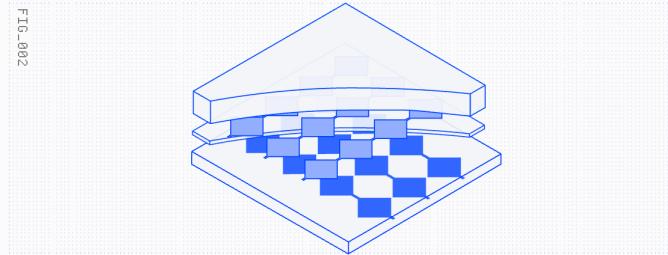


# MAKING SOFTWARE

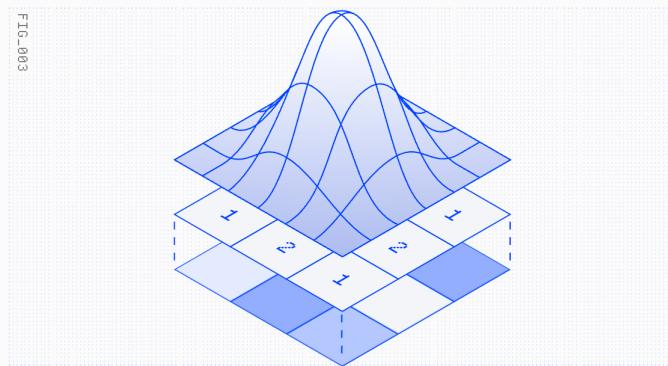
A reference manual for people who design and build software.  
Written and illustrated by Dan Hollick.

**H**ave you ever wondered how a touch screen knows you are touching it? Well, it has these layers of transparent metal electrodes embedded in the display. When your finger gets close to the screen it causes a disturbance in the magnetic field that the electrodes sense.



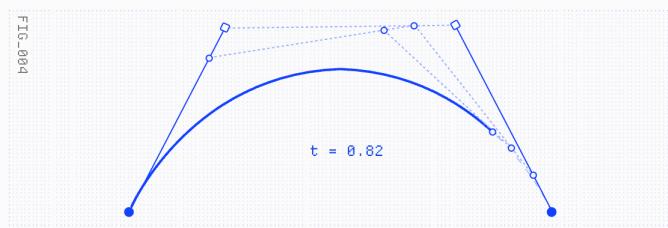
Because the electrodes are laid out on a grid, they can report back the x and y co-ordinates of the disturbance to the operating system. Pretty neat.

Or maybe you've wondered why we call it a Gaussian blur? When we blur an image, we look at all the neighbouring pixels and multiply them by a matrix of weights called a kernel.

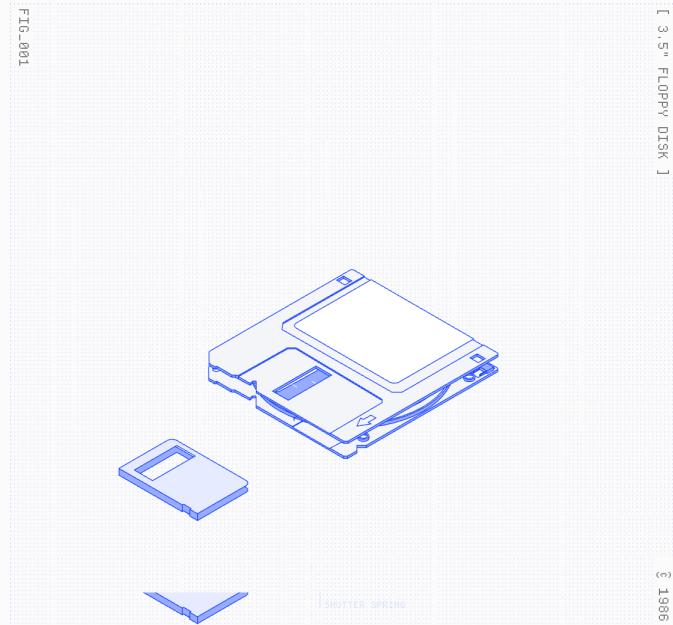


The most common type of kernel has a gaussian distribution, meaning it gets stronger towards the middle and weaker at the edges. This produces a more realistic blur without being too computationally expensive.

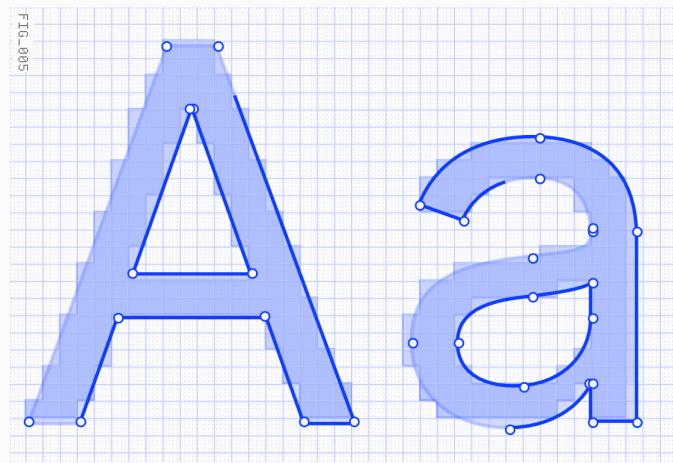
Maybe you've always wanted to know how the pen tool works in Figma and what those handles actually do when you move them.



They control the points on a bezier curve which is a cool piece of math we use to draw curves in vector graphics, like fonts and SVGs.



But of course, our screens are made of pixels and struggle to display smooth curves. So we have to take these curves and figure out how to display them so that they represent the shapes as accurately as possible.



This is called rasterisation but it's not as simple as it seems and we need a whole bunch of clever tricks like, anti-aliasing to trick our eyes into thinking we are looking at straight lines.

If you've ever wondered about any of these things or if they've sparked your curiosity, then this is for you.

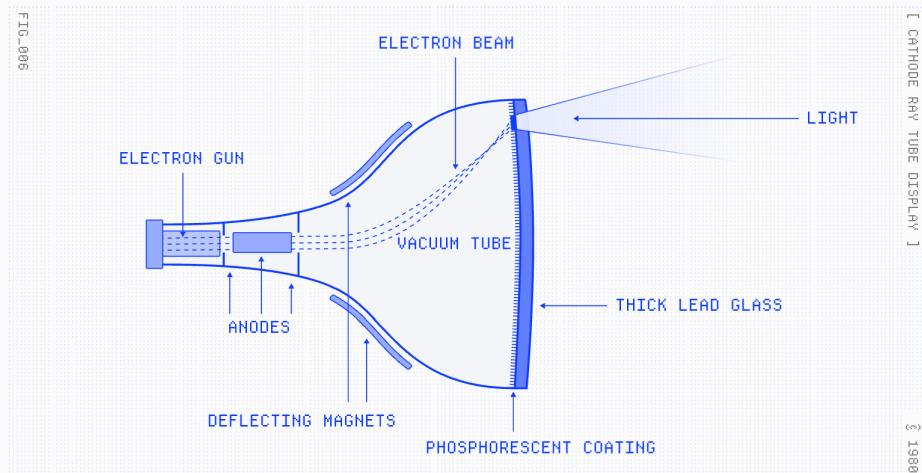
This book won't teach you how to actually make software - it's not a tutorial or a guide but rather something more interesting than that. It's a manual that explains how the things you use everyday actually work.

As everything around us has become more complicated our understanding of technology has diminished. It used to be that we needed to understand our tools deeply but today we understand them in a shallow, abstracted way.

It won't make you a better designer or programmer tomorrow - there's nothing actionable in here. But knowing how things work comes in handy when you find yourself out of your depth. Or at the very least, you can pretend to be smart in front of your friends.

You don't need to be technical to read this - there are a lot of pictures and diagrams to do the heavy lifting. You just need to be curious.

MAKING SOFTWARE



#### SIGNUP FOR UPDATES:

I promise to only email you cool shit. Draft chapters, progress updates, sneak peaks at illustrations I'm working on. Stuff like that.

ENTER YOUR EMAIL

JOIN MAILING LIST

TABLE OF CONTENTS [v1.0]

[ SECTIONS: 8 · CHAPTERS: 36 ]

## 1. PIXELS AND COLOR

- How does a screen work? ..... 3.6K WORDS
- What is a color space? ..... 6.2K WORDS
- Color contrast. ....

## 5. DATA AND COMPRESSION

- Bits, bytes and binary. ....
- Entropy and compression. ....
- Image compression. ....

- Blending modes.
- Digital images.
- Touch screens.

## 2. FONTS AND VECTORS

- Drawing curves.
- How to make a font.
- Rasterisation and anti-aliasing.
- Scalable Vector Graphics.
- Boolean operations.

## 3. 3D AND GRAPHICS

- How does a GPU work?
- Shaders. 4.5K WORDS
- Rays and SDFs.
- Blurs, noise and other effects.
- 3D projection.

## 4. AI AND ML

- Neural nets and transformers.
- Gradient descent and backpropagation.
- Embeddings and attention.
- Generating images.

- Cryptography.
- How is data stored?

## 6. NETWORKING AND THE WEB

- Sending and receiving data.
- How the internet works.
- Servers and the cloud.
- What is a browser?

## 7. COMPILERS AND INTERPRETERS

- What is code?
- What is an interpreter?
- What is a compiler?
- What makes code fast?

## 8. MISC

- Regular Expressions.
- QR and barcodes.
- Quantum computing.



↳ [View all 100 questions >>>](#)

↳ [How do you make the illustrations?](#)

## COMMON QUESTIONS

When will it launch?

Is this a physical book?

↑ ↓

IN: IS THIS A PHYSICAL BOOK?

How much will it cost?

OUT: No, it's a digital book *for now*.

↳ [What does it cost?](#)

↳ [What does it cost?](#)

Once it's fully ready, I'm going to do a print run but there's a ton of work involved in making a physical book and I want to do it properly.

I want it to be like a *really fucking nice* coffee table style book. People who buy the advanced digital edition will get first dibs on pre-ordering any print editions, so if that's something you're interested in, make sure you get that one.

## SIGNUP FOR UPDATES:

*I promise to only email you cool shit. Draft chapters, progress updates, sneak peaks at illustrations I'm working on. Stuff like that.*

ENTER YOUR EMAIL

JOIN MAILING LIST

CURRENT PROGRESS: 

\*\*\*

| MAKING SOFTWARE |

A REFERENCE MANUAL FOR PEOPLE WHO DESIGN AND BUILD SOFTWARE.  
WRITTEN AND ILLUSTRATED BY DAN HOLICK

| © 1990 |