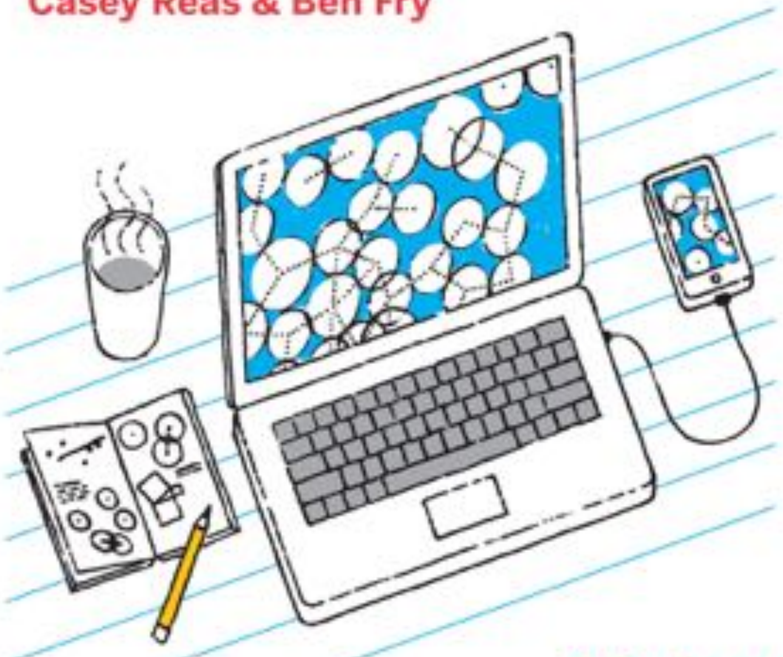
all code illustrations taken from the processing.org website +

Make: PROJECTS

A HANDS-ON INTRODUCTION TO MAKING INTERACTIVE GRAPHICS

Getting Started with Processing

Casey Reas & Ben Fry



O'REILLY

Make: makezine.com

Open Source/Graphics Programming

Make: PROJECTS

Getting Started with Processing

Learn computer programming the easy way with Processing, a simple language that lets you use code to create drawings, animation, and interactive graphics. Programming courses usually start with theory, but this book lets you jump right into creative and fun projects. It's ideal for anyone who wants to learn programming, and serves as a simple introduction to graphics for people who already have some programming skills.

Written by the founders of Processing, this book takes you through the learning process one step at a time to help you grasp core programming concepts. Join the thousands of hobbyists, students, and professionals who have discovered this free and educational community platform.

- Quickly learn programming basics, from variables to objects
- Understand the fundamentals of computer graphics
- Get acquainted with the Processing software environment
- Create interactive graphics with easy-to-follow projects
- Use the Arduino open source prototyping platform to control your Processing graphics

Casey Reas is a professor in the Department of Design Media Arts at UCLA and a graduate of the MIT Media Laboratory. Reas's software has been featured in numerous solo and group exhibitions in the U.S., Europe, and Asia.

Ben Fry, a designer, programmer, and author based in Cambridge, Massachusetts, received his doctoral degree from the MIT Media Laboratory. He worked with Casey Reas to develop Processing, which won a Golden Nica from the Prix Ars Electronica in 2005.

Make: makezine.com

O'REILLY

US $19.99 CAN $24.99
ISBN: 978-1-449-37980-3
51999
9 781449 379803

6/Media

Processing is capable of drawing more than simple lines and shapes. It's time to learn how to load raster images, vector files, and fonts into our programs to extend the visual possibilities to photography, detailed diagrams, and diverse typefaces.

Images

There are three steps to follow before you can draw an image to the screen:

1. Add the image to the sketch's *data* folder (instructions given previously).
2. Create a *PImage* variable to store the image.
3. Load the image into the variable with *loadImage()*.

Load an image into your sketch and draw it to the screen

Load an image into your sketch and draw it to the screen

Using mouseX and mouseY, move the image around

Use and *embedded for loop* to draw a grid of 2 separate images to the screen

Fonts

Processing can display text in many fonts other than the default. Before you display text in a different typeface, you need to convert one of the fonts on your computer to the VLW format, which stores each letter as a small image. To do this, select Create Font from the Tools menu to open the dialog box (Figure 6-1). Specify the font you want to convert, as well as the size and whether you want it to be smooth (anti-aliased).

Figure 6-2. Typography coordinates.

When you click the OK button in the Create Font tool, the VLW font is created and placed in the sketch's *data* folder. Now it's possible to load the font and add words to a sketch. This part is similar to working with images, but there's one extra step:

1. Add the font to the sketch's *data* folder (instructions given previously).
2. Create a *PFont* variable to store the font.
3. Load the font into the variable with *loadFont()*.
4. Use the *textFont()* command to set the current font.

Reproduce this haiku in processing:

The lamp once out
Cool stars enter
The window frame.

- Natsume Soseki

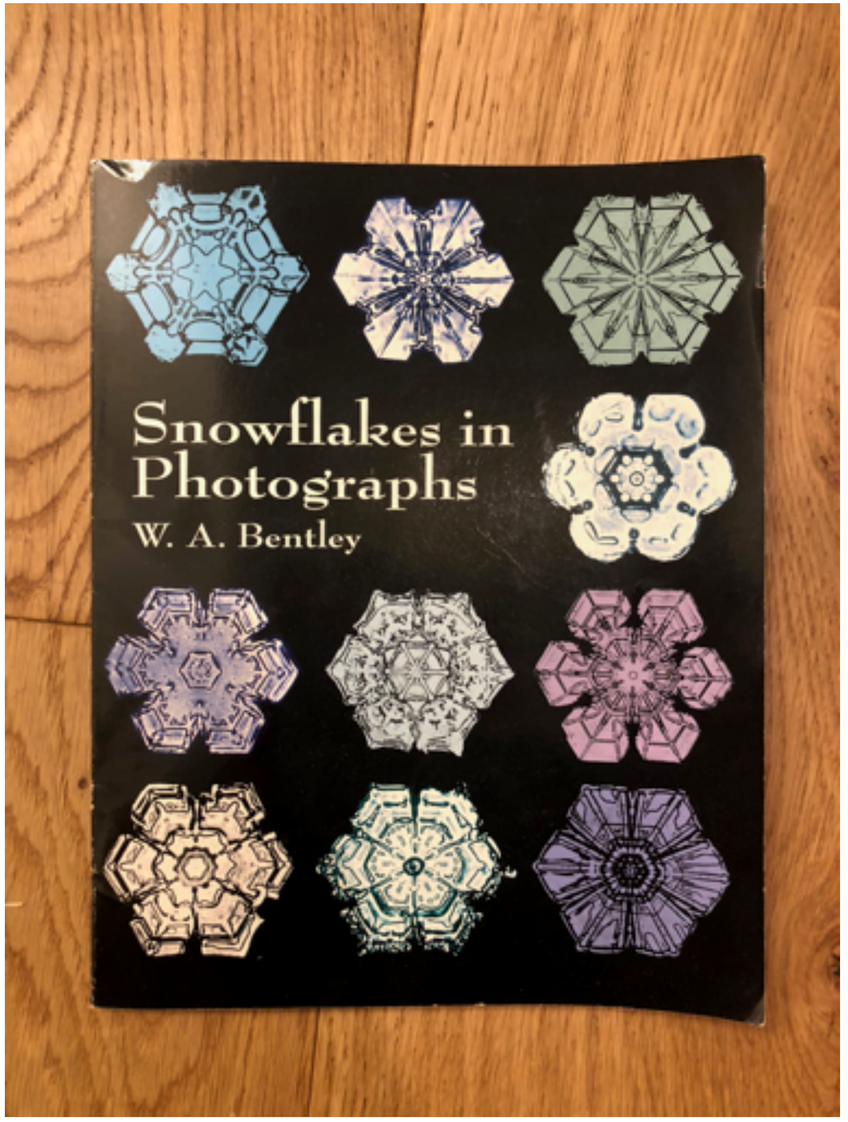
Shapes

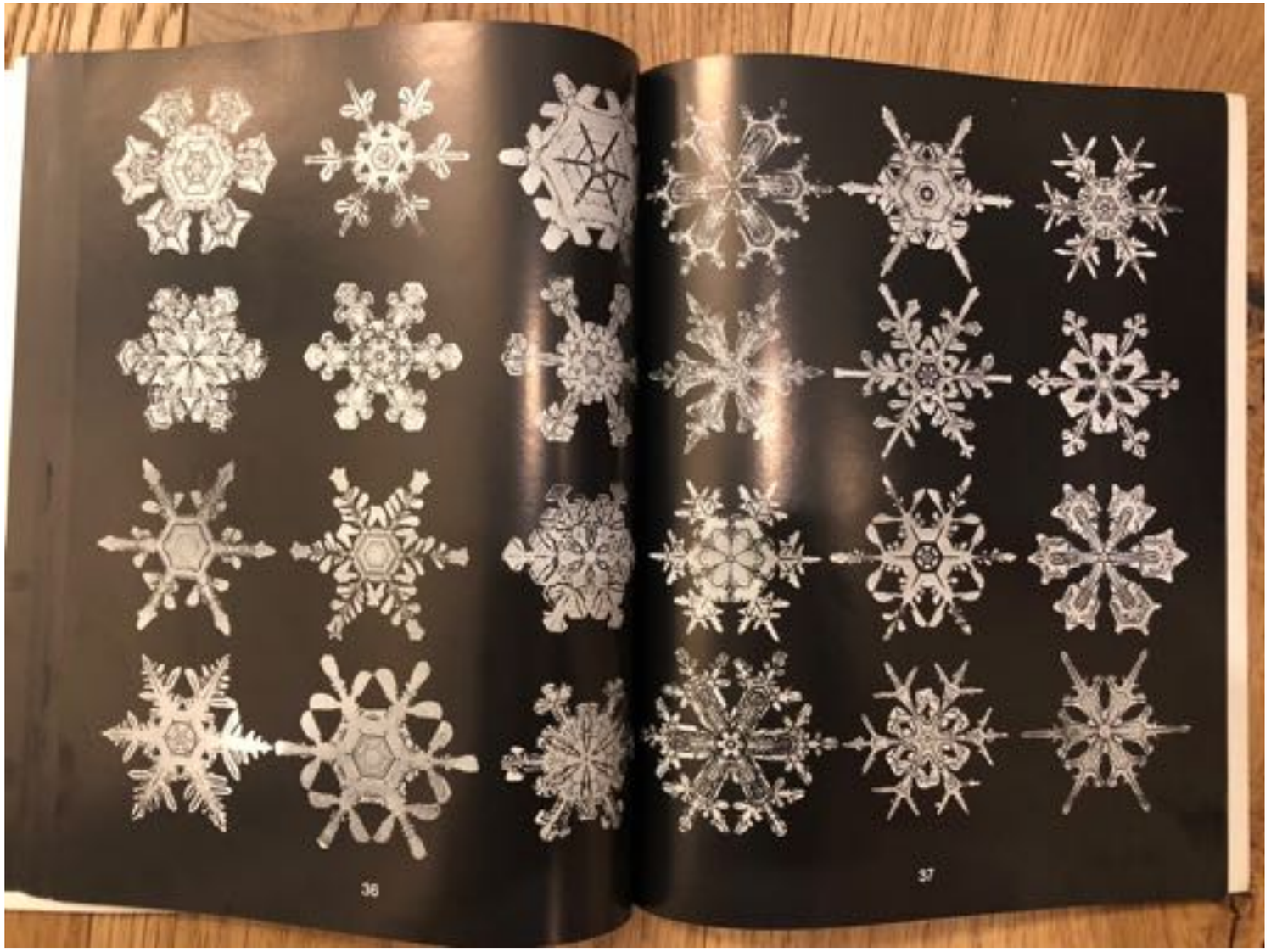
If you make vector shapes in a program like Inkscape or Illustrator, you can load them into Processing directly. This is helpful for shapes you'd rather not build with Processing's drawing functions. As with images, you need to add them to your sketch before they can be loaded.

There are three steps to load and draw an SVG file:

1. Add an SVG file to the sketch's *data* folder.
2. Create a *PShape* variable to store the vector file.
3. Load the vector file into the variable with *loadShape()*.

Generate a snowflake using a *for loop* and an svg vector file





***Snowflakes in Photographs* by W A Bentley (1931)**
https://books.google.com/books?id=4O25SZtrMfkC

***Boquet of Eyes* by Hannah Hoch (1930)**
https://theartstack.com/artist/hannah-hoch/bouquet-eyes-1930

Pearblossom Highway #1 by David Hockney **(11th-18th April 1986)**
http://www.davidhockney.co/works/photos/photographic-collages

***We are all Nature II & III* by Christopher Relander (2012-13)**
https://www.christofferrelander.com/projects/we-are-nature-3/

Homework 05 // Due 2018.10.08

Part I. Use a for loop to generate a kaleidoscopic image
Using either a *PImage, PFont,* or *PShape*

Size: 720x720 pixels

Part II. Generate your own photomontage, collage, and or composite image
Think about an interesting way to use code with imagery.

Size: 1280x720 pixels or 720x720 pixels