5

Lorem Ipsum III

Caius Incorrigibilus

Version 1.0
2014-04-15

What is Processing?

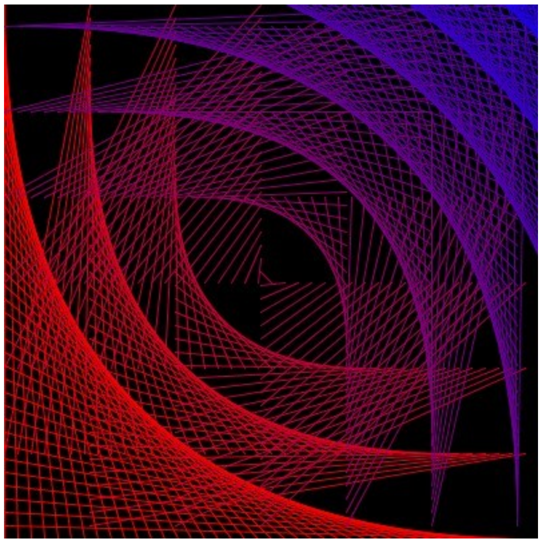


Figure 1. Artwork

Processing is a language intended for visual artists or, more generally, anyone who needs to create images, whether they be graphs, representations of data, works of art, or something else entirely. Processing can be used to produce still images like the one on the left, moving images like those on the right, and even images with which a single user or a live audience can interact. It can be used to give still or dynamic views of social networks, or to simulate physical systems, as in the *Bouncing Ball* video. With the addition of libraries like minim, Processing programs can synthesize sound and music. Because the output of a program is immediate and visual, processing is also an excellent language for learning to program.

An open source language created by Casey Reas and Ben Fry, Processing has a large and active community of people who contribute to it in many ways. Go to the official web site, processing.org, to download the program (Linux, Mac, Windows), to see examples of what people have done with Processing', and to consult the tutorials and reference manual.

The lessons that follow give a rapid introduction to Processing. While they are by no means a complete introduction to the language, they should give an idea of what can be done with it. For more information see processing.org or some of the excellent books, e.g., Daniel Shiffman's Learning Processing and its segue, Nature of Code.

<hr />

Examples from later in the course

Magic marker was created by drawing with a graphics tablet. You can also use a mouse, though you then have less control. Processing tracks the stylus position and uses it to make square marks on the screen.

#

The xvideo above was constructed using *random walk*. A starting square is chosen with a starting color value. At each "tick of the clock", one moves randomly up or down, left and right to a new square. The color is changed by a small random amount. This process — repeatedly making small random changes to a quantity — is a random walk. Successive steps in the random walk are taken rapidly 24 to 60 times a second.

The video above simulates the motion of a ball dropped onto a small surface with a small initial velocity in the horizontal direction. A discrete solution to Newton’s laws of motion is used to "compute" the motion of the bouncing ball. :imagesdir: images

Running a processing program

To get started with Processing, follow the directions below.

1. Install Processing on your computer using the download tab at Processing.org.
2. Open Processing. You will see a window like that in Figure 1.
3. Paste the code for gray_on_black.pde into the Processing window. The code is below: copy and paste! The Processing window is displayed on the right.
4. In the Processing window, press the button with the triangle icon (upper left). This is the **run** button. The program should produce output as in the third figure.

```
// gray_on_black.pde
// Draw gray square on black square

void setup() {

  size(400, 400);
  background(0);

  fill(125);
  rect(100,100,200,200);
}
```

Exercises

1. Run the program above, if you haven't already.
2. Vary some of the parameters. For example, change size(400,400) to size(300,600). What is the result? Try changing other parameters, run the program, and try to understand what happened.
3. Run some of the other programs in this course using the copy and paste method.

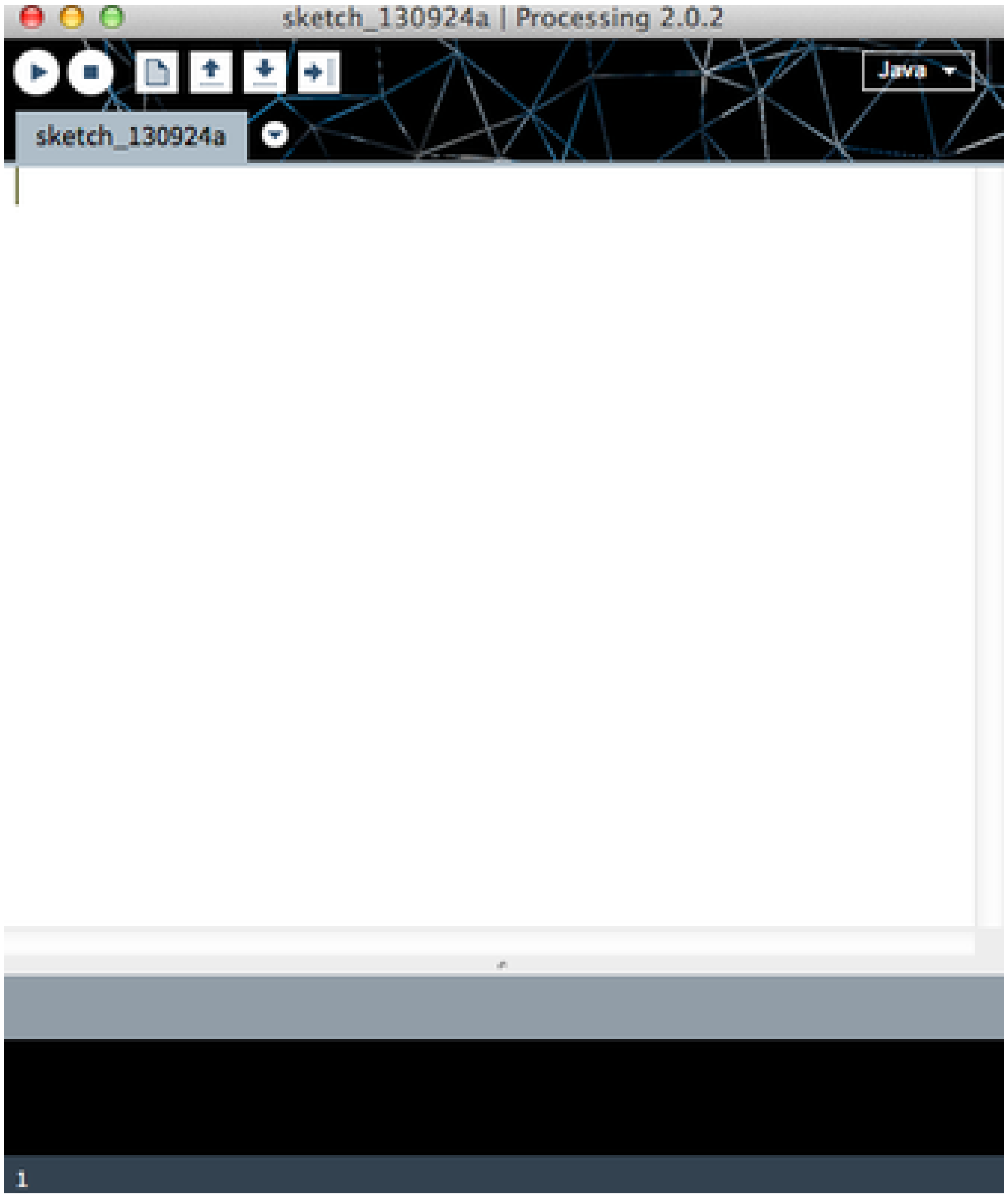
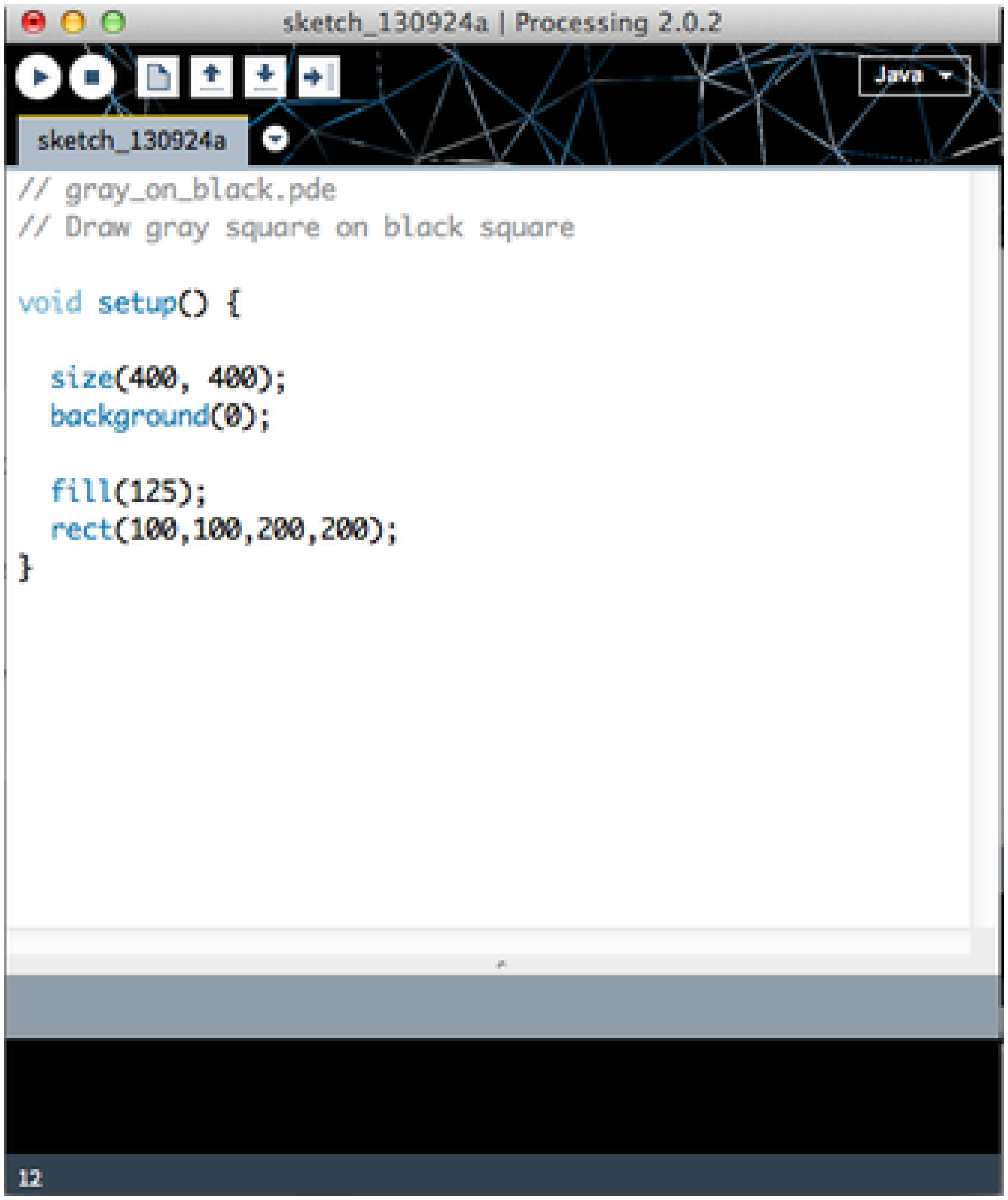


Figure 2. Processing window



```
// gray_on_black.pde
// Draw gray square on black square

void setup() {

  size(400, 400);
  background(0);

  fill(125);
  rect(100,100,200,200);
}
```

Figure 3. Processing window with program

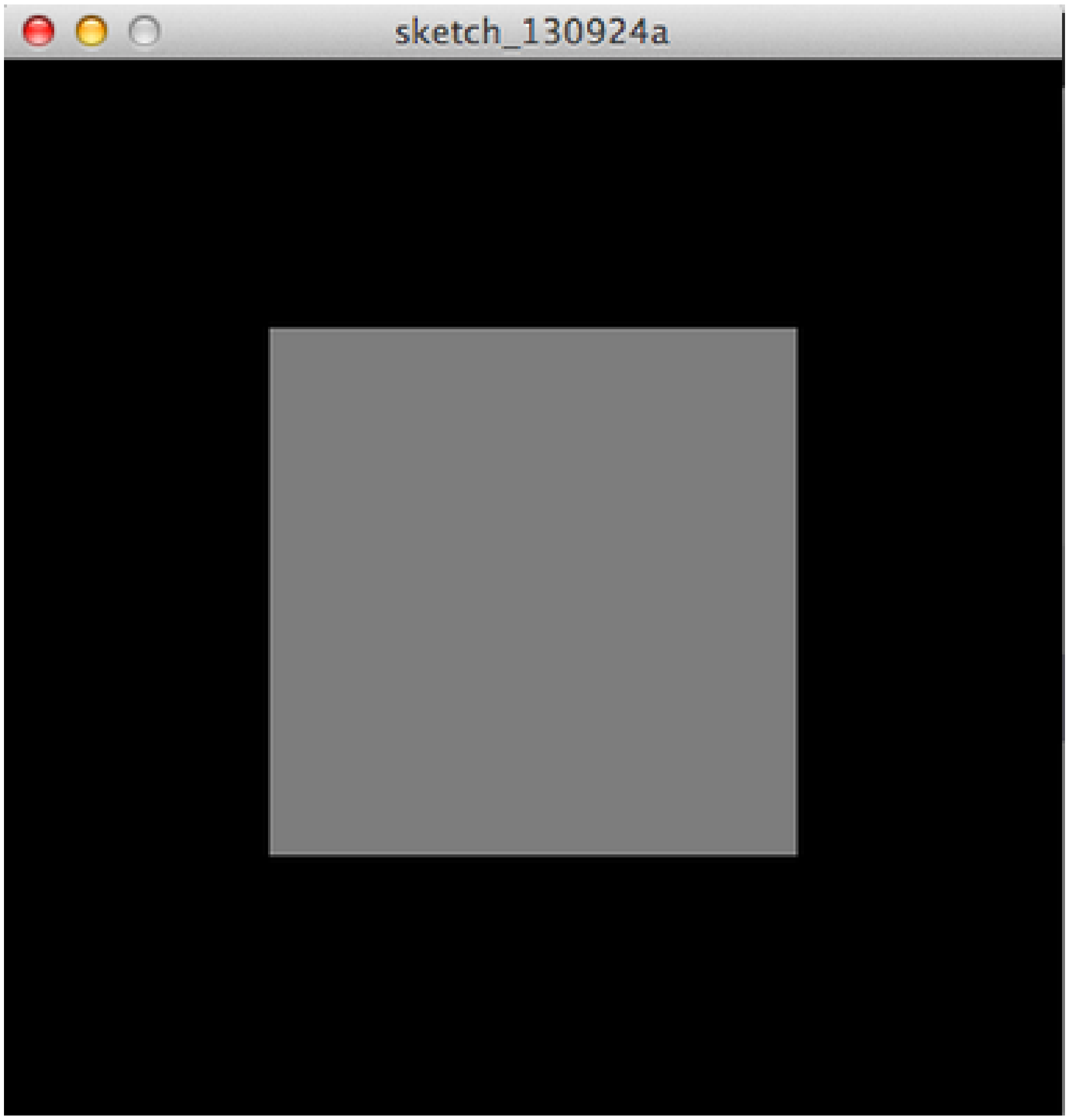


Figure 4. Output of Processing Program