USC IML 288 __ 2018.11.12
Flocking

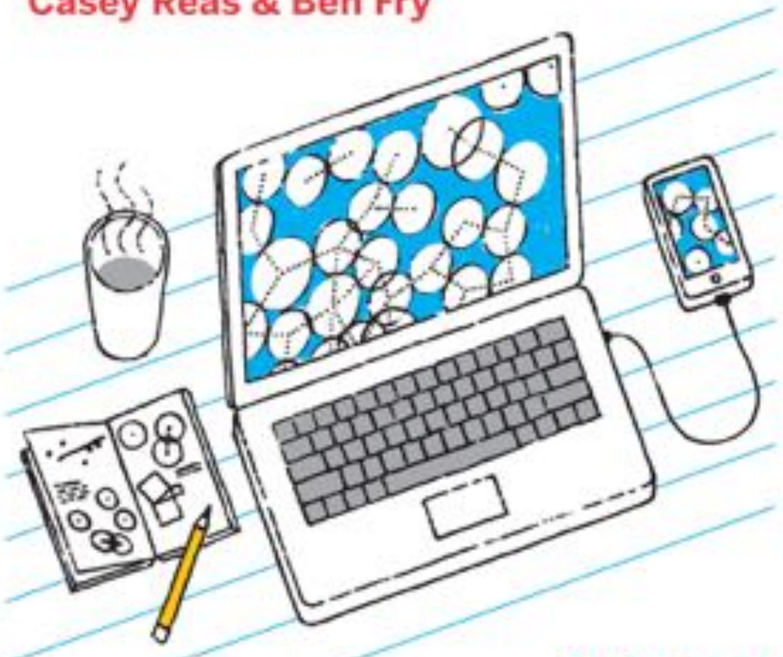
all code illustrations taken from the processing.org website +

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

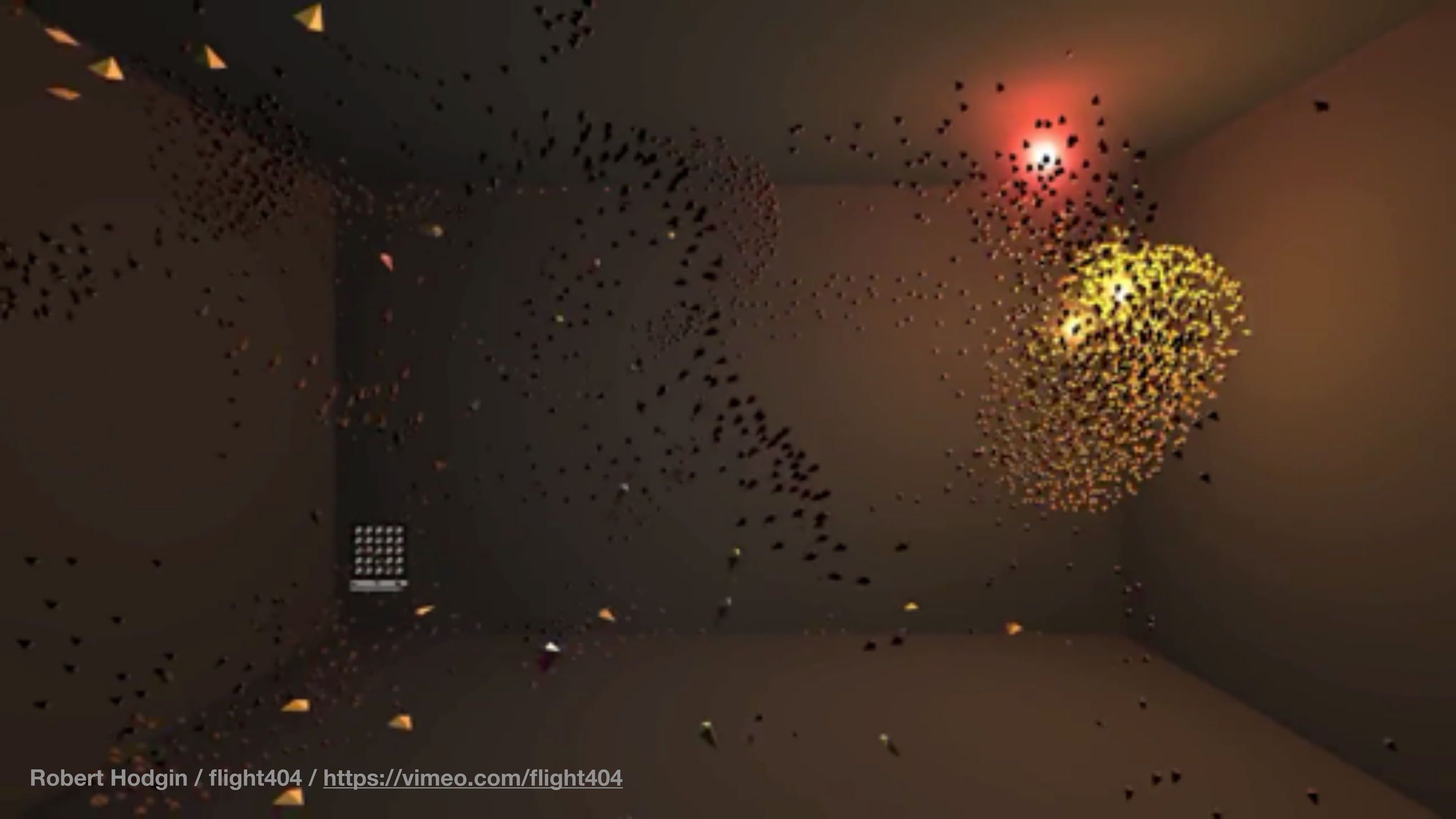
O'REILLY

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

Robert Hodgin / flight404 : murmuration+fox movies / https://vimeo.com/flight404

Robert Hodgin / flight404 : murmuration / https://vimeo.com/132161478

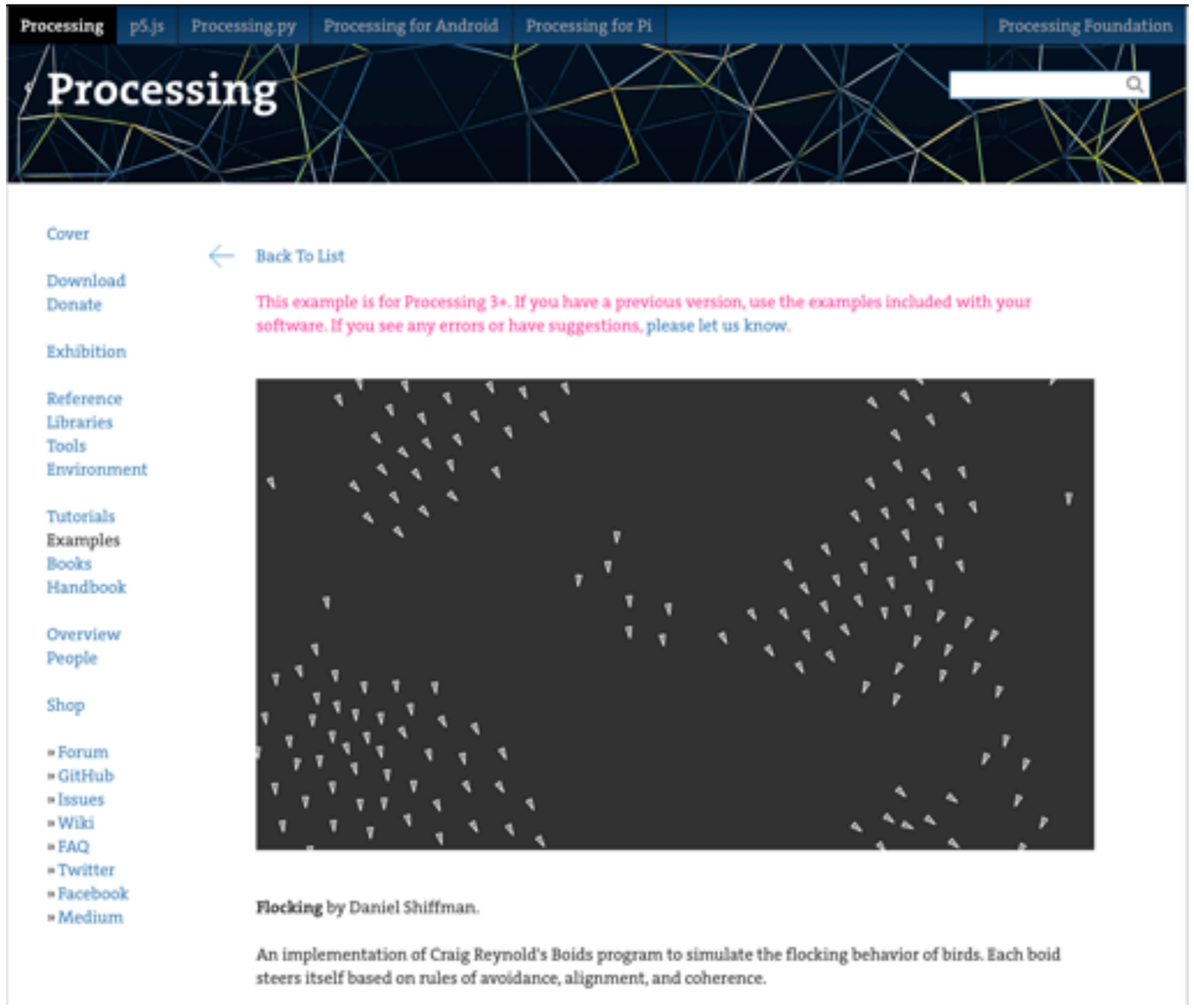
Robert Hodgin / flight404 : murmuration with obstacles / https://vimeo.com/141463798



Robert Hodgin / flight404 / https://vimeo.com/flight404

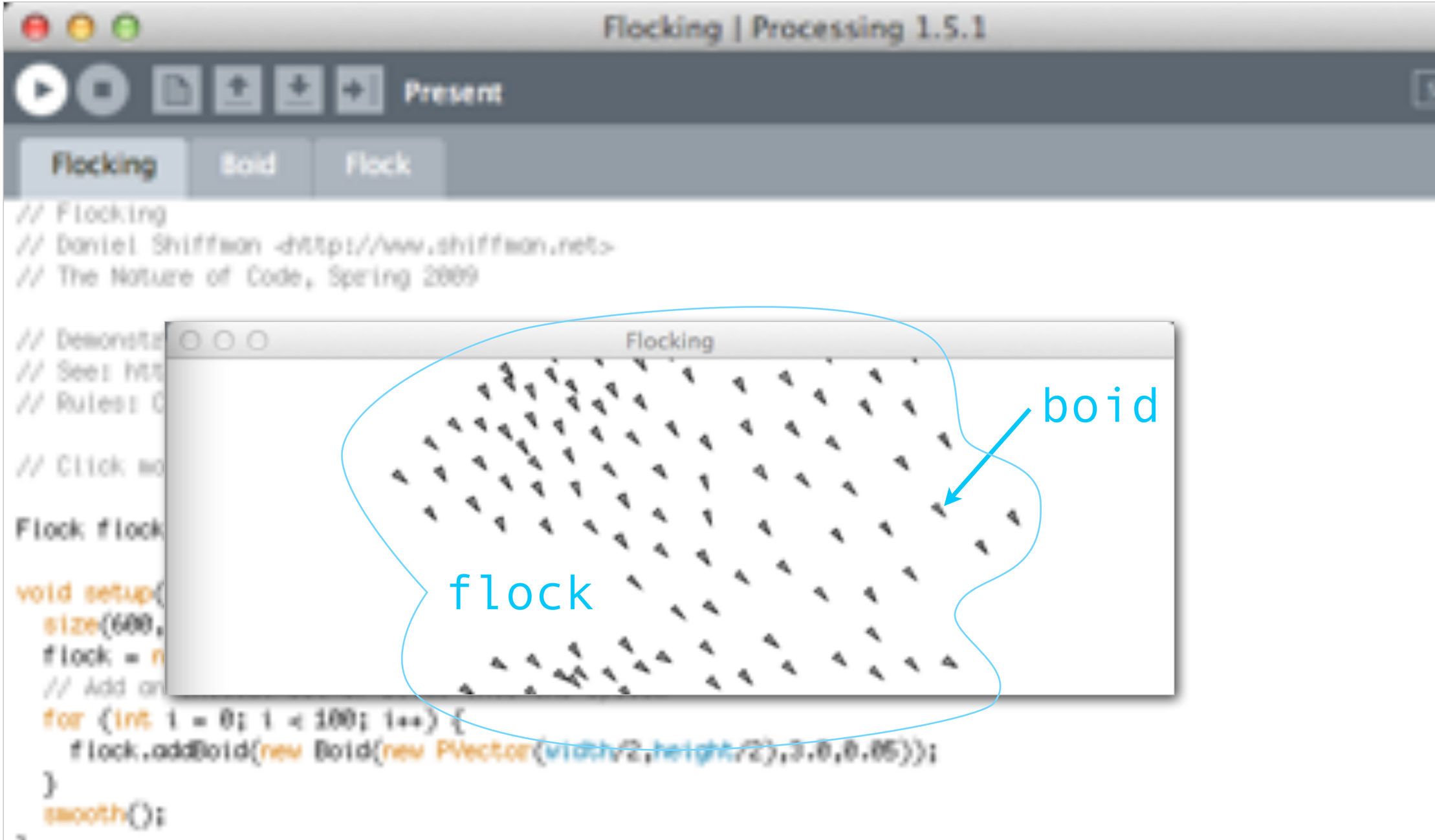
Robert Hodgin / flight404 : Fish Tornado, revisited / https://vimeo.com/132161782

Robert Hodgin / flight404 : Dense bait ball, 5000 fish / https://vimeo.com/64937983



Start with the flocking code from:
https://processing.org/examples/flocking.html

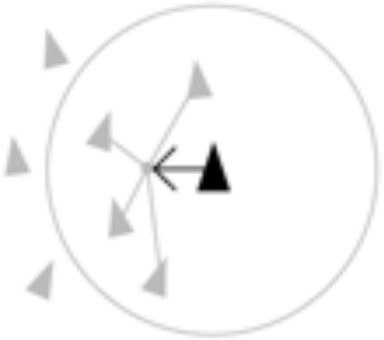
object oriented programming // classes



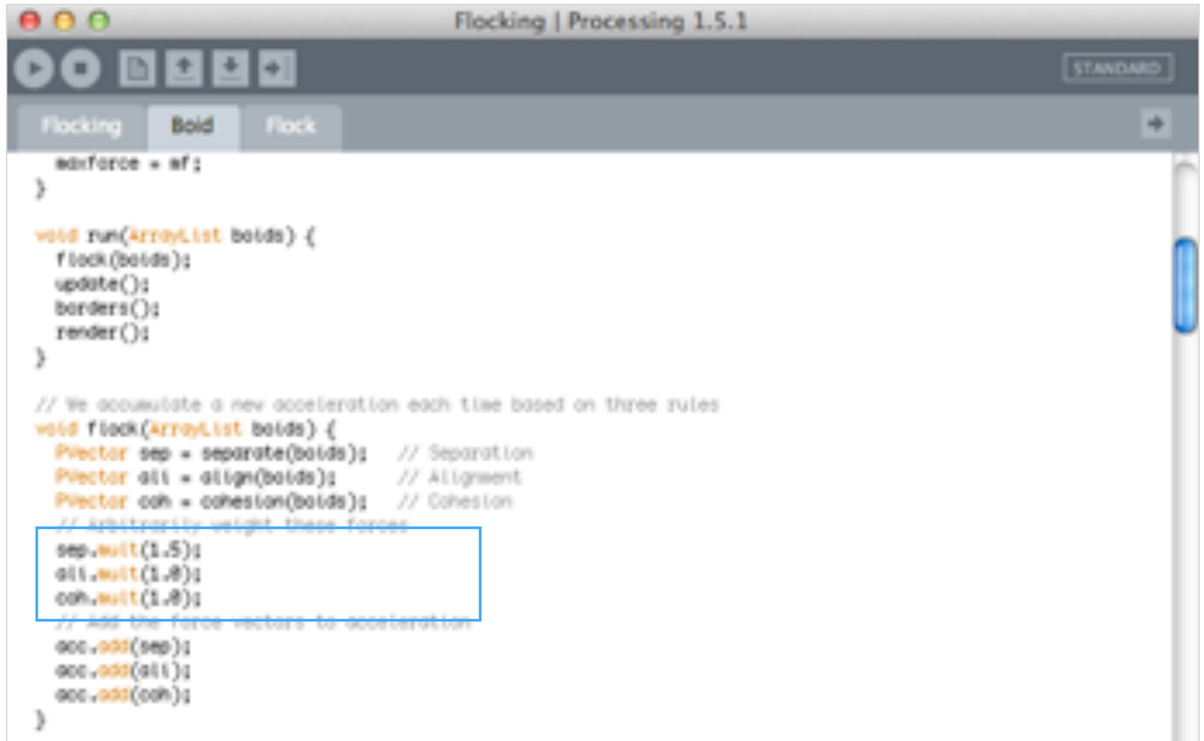
the flocking algorithm...

Separation:
Steer to avoid crowding local flockmates

Alignment:
Steer toward the average heading of local flockmates



Cohesion:
Steer to move toward the average position of local flockmates



Typical Class Structure:

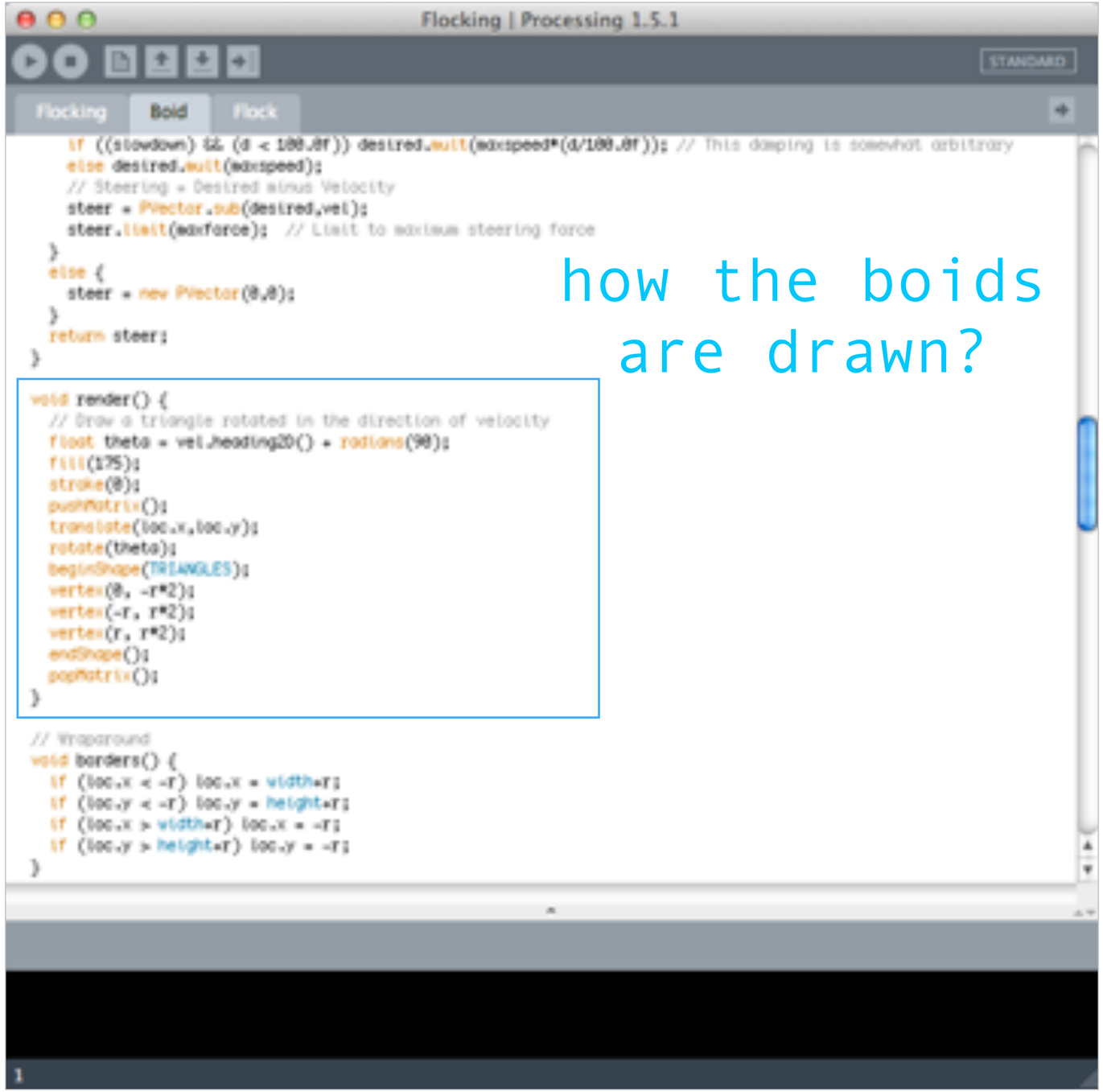
```
class JitterBug {

  float x;
  float y;
  int diameter;
  float speed = 2.5;

  JitterBug(float tempX, float tempY, int tempDiameter) {
    x = tempX;
    y = tempY;
    diameter = tempDiameter;
  }

  void move() {
    x += random(-speed, speed);
    y += random(-speed, speed);
  }

  void display() {
    ellipse(x, y, diameter, diameter);
  }
}
```

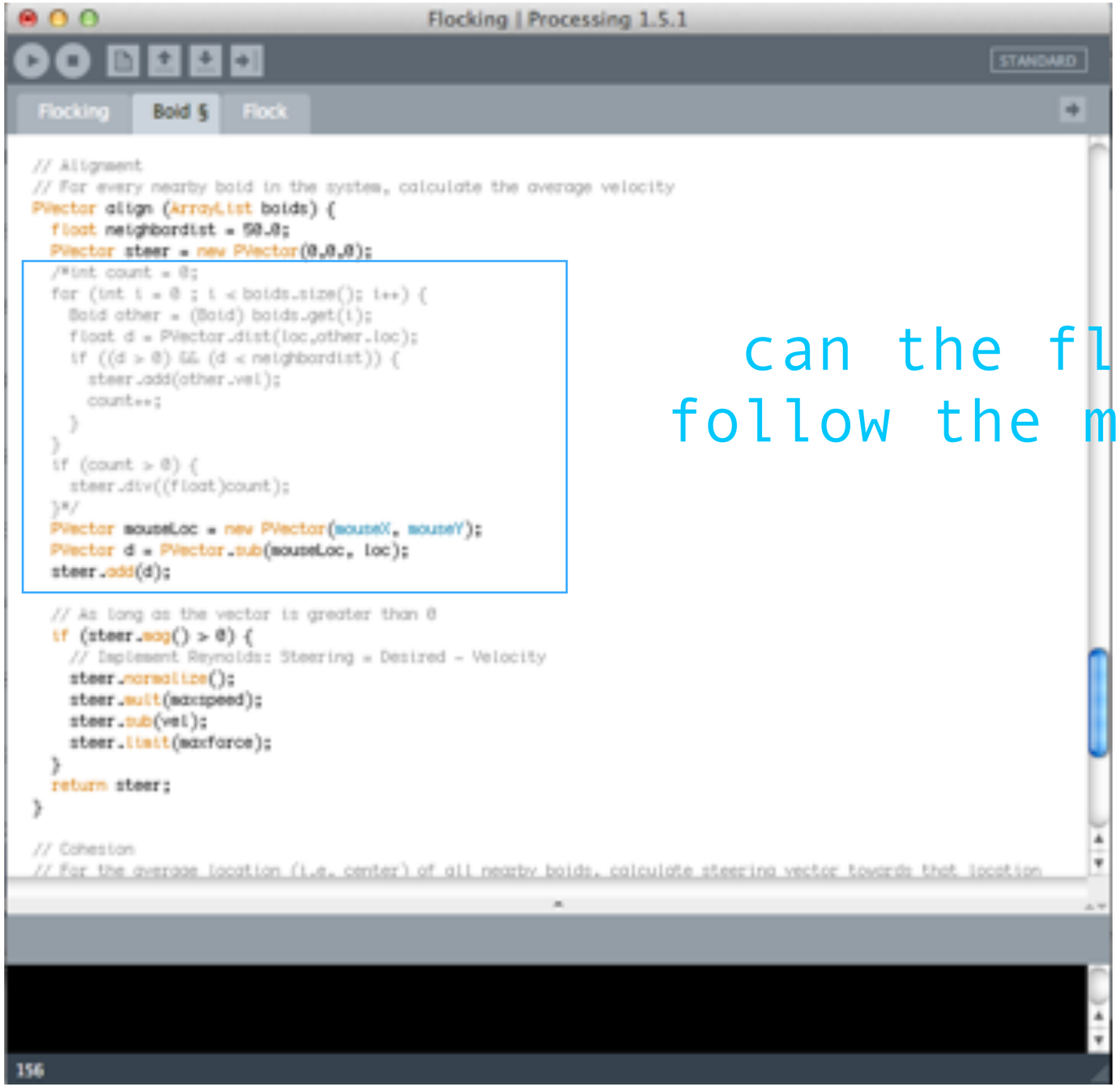
Block

Fields

Constructor

Methods


Flocking | Processing 1.5.1
STANDARD
Flocking
Boid
Flock
if ((slowdown) && (d < 100.0f)) desired.mult(maxspeed*(d/100.0f)); // This damping is somewhat arbitrary
else desired.mult(maxspeed);
// Steering = Desired minus Velocity
steer = PVector.sub(desired,vel);
steer.limit(maxforce); // Limit to maximum steering force
}
else {
steer = new PVector(0,0);
}
return steer;
}
void render() {
// Draw a triangle rotated in the direction of velocity
float theta = vel.heading2D() + radians(90);
fill(175);
stroke(0);
pushMatrix();
translate(loc.x,loc.y);
rotate(theta);
beginShape(TRIANGLES);
vertex(0, -r*2);
vertex(-r, r*2);
vertex(r, r*2);
endShape();
popMatrix();
}
// Wraparound
void borders() {
if (loc.x < -r) loc.x = width+r;
if (loc.y < -r) loc.y = height+r;
if (loc.x > width+r) loc.x = -r;
if (loc.y > height+r) loc.y = -r;
}
how the boids are drawn?
1



can the flock
follow the mouse?

Have the boids follow the cursor.

Use the separation test to draw a connecting line between boids

Create a trail behind the boids.

Homework 11 // Due 2018.11.19

Develop a second draft of your final project in Processing. This could be 1 component of it in more detail, or a rougher overview o the whole project. If applicable (there's something new from last week) submit any notes, diagrams, inspiration references that ar relevent to your final project in a folder inside the Processing sketch folder.

Size: 1280x720 pixels or 720x720 pixels

Continue to work on your own processing sketch based on Shiffman's implementation of the flocking algorithm. Make sure the visuals are sufficiently different from the starting example. Cite the code correctly.

Size: 1280x720 pixels or 720x720 pixels