The Nature of Code

SUBTITLE

X Edition

Dan Shiffman

° ° ° ° °

Table of Contents

Preface

is there a preface? bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Dan Shiffman
Oslo/Antibes
July 2012

Chapter 1

Vectors

Roger, Roger. What's our vector, Victor?

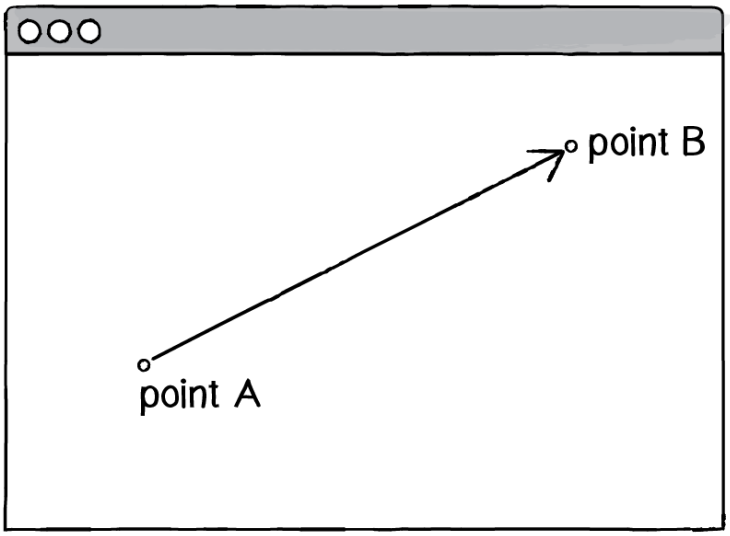
This book is all about looking at the world around us and coming up with clever ways to simulate that world with code. Divided into three parts, the book will start by looking at basic physicsâ€"how an apple falls from a tree, a pendulum swings in the air, the earth revolves around the sun, etc. Absolutely everything contained within the first five chapters of this book requires the use of the most basic building block for programming motionâ€"the *vector*. And so this is where we begin our story.

Now, the word *vector* can mean a lot of different things. Vector is the name of a new wave rock band formed in Sacramento, CA in the early 1980s. Itâ€™s the name of a breakfast cereal manufactured by Kelloggâ€™s Canada. In the field of epidemiology, a vector is used to describe an organism that transmits infection from one host to another. In the C++ programming language, a Vector (std::vector) is an implementation of a dynamically resizable array data structure. While all these definitions are interesting, theyâ€™re not what we are looking for. What we want is called a *Euclidean vector* (named for the Greek mathematician Euclid and also known as a geometric vector). When you see the term â€œvectorâ€? in this book, you can assume it refers to a Euclidean vector defined as:

A vector is an entity that has both magnitude and direction

A vector is typically drawn as a arrow; the direction is indicated by where the arrow is pointing, and the magnitude by the length of the arrow itself.

Figure 1: A caption



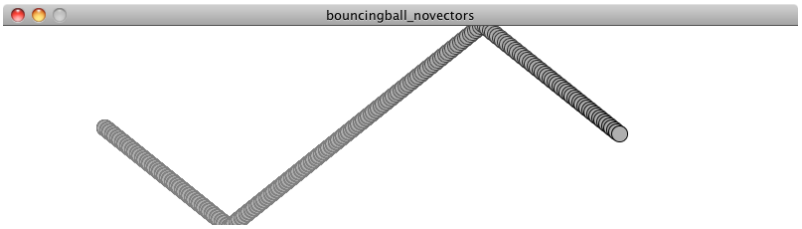
A vector (drawn as an arrow)
has magnitude (length of arrow)
and direction (which way it is pointing).

In the diagram to the left [notetoself]*[reference figure # instead?]*, the vector is drawn as an arrow from point A to point B and serves as an instruction for how to travel from A to B.

1.1 VECTORS, YOU COMPLETE ME

Before we dive into more of the details about vectors, letâ€™s look at a basic Processing example that demonstrates why we should care about vectors in the first place. If youâ€™ve read any of the introductory Processing textbooks or taken a class on programming with Processing (and hopefully youâ€™ve done one of these things to help prepare you for this book), you probably, at one point or another, learned to how to write a simple bouncing ball sketch.

Figure 2: A caption


bouncingball_novectors

A vector is an entity that has both magnitude and direction

EXAMPLE 1.1: BOUNCING BALL WITH NO VECTORS

```
// Variables for location and speed of ball. float x = 100; float y = 100; float xspeed = 1; float yspeed = 3.3; // Remember how Processing works? setup() is executed once when the sketch starts and draw() loops forever and ever (until you quit). void setup() { size(200,200); smooth(); background(255); } void draw() { background(255); // Move the ball according to its speed. x = x + xspeed; y = y + yspeed; }
```