Vectors in Two Dimensions

Leon Tabak

14 June 2011

You are standing in front of the Empire State Building at Fifth Avenue and West 34th Street. Carnegie Hall is at Seventh Avenue and West 57th Street. How do you get to Carnegie Hall?

You know the answer! You will walk two blocks west (to get from Fifth Avenue to Seventh Avenue) and 23 blocks north (to get from 34th Street to 57th Street).

Even if you have never before heard the word "vector," you already know something about how to work with vectors.

There is an old joke whose punchline gives us a different route to the concert hall. "How do I get to Carnegie Hall?," a vistor asks a stranger on the street. "Practice, practice, practice!," the native New Yorker replies. We will use that answer too. Practice!

Vectors have applications in many fields. For example, physicists use vectors to describe forces. In computer graphics, we use vectors to describe the positions and orientations of objects in space. We will show you an application of vectors in computer graphics in the programming part of this exercise.

Vectors, like the arrows often used to represent them, have lengths and directions.

As we have just seen, we can do arithmetic with vectors.

Constructing, accessing, mutating, and printing.

In two dimensions, a vector can be completely specified with two numbers. The first number represents a displacement along the x axis. The second number represents a displacement along the y axis.

We can give the elements of a vector names. Here we have two vectors. The first vector's elements are u_x and u_y. The second vector's elements are v_x and v_y.

$$\vec{u} = (u_x, u_y)$$
$$\vec{v} = (v_x, v_y)$$

We can assign particular numeric values to the elements.

$$u_x = 3$$
$$u_y = 4$$
$$\vec{u} = (3, 4)$$
$$v_x = 5$$
$$v_y = 12$$
$$\vec{v} = (5, 12)$$

At the start, we said that a vector has a length and a direction. However, the vector whose elements are both zero ($\vec{v} = (0, 0)$) has a length of zero but no direction. We will find it convenient to include $(0, 0)$ in the set of vectors. For example, we will define the addition of vectors so that, for all vectors $\vec{v}$, $\vec{v} + (0, 0) = \vec{v}$. Similarly, we will define the subtraction of vectors so that $\vec{v} - \vec{v} = (0, 0)$.

A class that models vectors will define constructors, accessors, and mutators. A constructor creates a vector. An accessor is a method that retrieves the value of one of the vector's elements. A mutator is a method that changes the value of one of the vector's elements. We will also find it convenient to give ourselves a method that produces a printable representation of a vector.

Addition and subtraction.

We can add and subtract vectors. The sum or difference of two vectors is another vector.

The elements of the sum of two vectors are the sums of the corresponding elements of the two vectors. Similarly, the elements of the differences of two vectors are the differences of the corresponding elements of the two vectors.

$$\vec{u} = (3, 4)$$
$$\vec{v} = (5, 12)$$
$$\vec{u} + \vec{v} = (3 + 5, 4 + 12)$$
$$= (8, 16)$$
$$\vec{v} - \vec{u} = (5 - 3, 12 - 4)$$
$$= (2, 8)$$

We can produce the sum of two vectors geometrically. Draw two arrows to represent the vectors. Place the tail end of the second vector on the head of the first vector. Then draw a third arrow that extends from the tail of the first vector to the head of the second vector. That third arrow represents the sum.

Figure **??** shows an example.

We have represented the two vectors we want to add with arrows that span two sides of a triangle. We have represented the sum of the vectors with an arrow that spans the third side of the triangle.
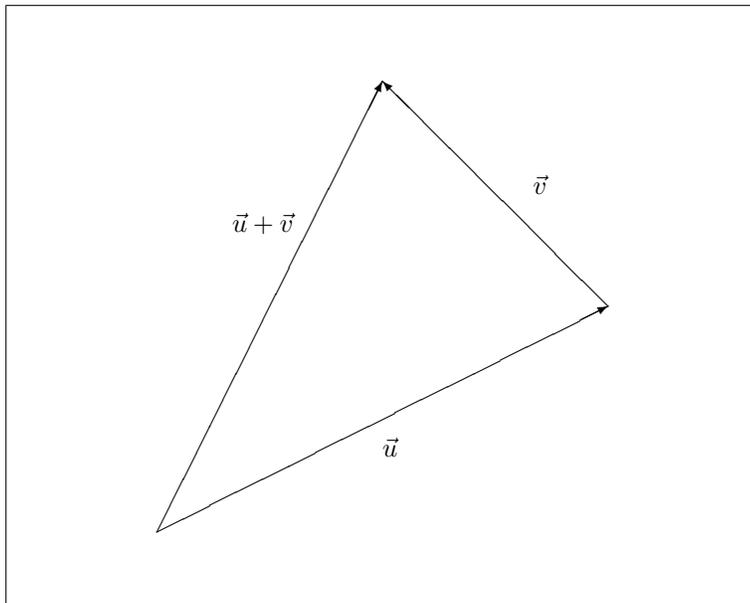
Figure 1: Sum of two vectors.

Multiplication.

The dot product of two vectors is a number (not another vector). The dot product is the sum of the products of the corresponding elements of the two vectors.

$$\vec{u} = (3, 4)$$
$$\vec{v} = (5, 12)$$
$$\vec{u} \cdot \vec{v} = 3 \cdot 5 + 4 \cdot 12$$
$$= 15 + 48$$
$$= 63$$

The dot product of two vectors $\vec{u}$ and $\vec{v}$ equals the product of the lengths (magnitudes) of the vectors ($|\vec{u}|$ and $|\vec{v}|$) and the cosine of the angle θ between the vectors.

If we could make the lengths of two vectors each equal to one, then we could find the cosine of the angle between the two vectors by computing their cross product. This turns out to be a very useful operation. We can use it to shade a surface (by brightly illuminating a surface when rays of light strike it perpendicularly and dimly shading the surface when rays strike it obliquely) and to compute reflected rays.

$$\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos\theta$$

Magnitude, scaling, and normalization.

The magnitude of a vector is its length. We compute the magnitude of a vector in the same way that we compute the length of the hypotenuse of a right triangle.

This follows because we can represent the vector as an arrow that corresponds to the hypotenuse of a right triangle. The value of a vector's x component is the length of the base of the right triangle. The value of a vector's y component is the height of the right triangle. The vector in this picture extends from $(0,0)$ to (x,y).

The magnitude of a vector $\vec{v}$ (denoted $|\vec{v}|$ is the square root of the dot product of the vector with itself.

$$\vec{u} = (3,4)$$
$$|\vec{u}| = \sqrt{3^2 + 4^2}$$
$$= \sqrt{\vec{u} \cdot \vec{u}}$$

Given a vector $\vec{a}$, it is possible to produce another vector $\vec{b}$ that has the same direction as $\vec{a}$ but a different length.

$$s = 3$$
$$a_x = 1$$
$$a_y = 2$$
$$\vec{a} = (a_x, a_y)$$
$$= (1,2)$$
$$\vec{b} = s\vec{a}$$
$$= (sa_x, sa_y)$$
$$= (3 \cdot 1, 3 \cdot 2)$$
$$= (3,6)$$

If the scaling factor s is negative, then the product is a vector that points in the opposite direction.

$$s = -1$$
$$a_x = 1$$
$$a_y = 2$$
$$\vec{a} = (a_x, a_y)$$
$$= (1, 2)$$
$$\vec{b} = s\vec{a}$$
$$= (sa_x, sa_y)$$
$$= (-1 \cdot 1, -1 \cdot 2)$$
$$= (-1, -2)$$

The normalization of a vector $\vec{v}$ produces a new vector that has the same direction as $\vec{v}$ but a length of one. Normalization is a special case of scaling.

$$\vec{u} = (3, 4)$$

$$|\vec{u}| = \sqrt{\vec{u} \cdot \vec{u}}$$
$$= \sqrt{3 \cdot 3 + 4 \cdot 4}$$
$$= \sqrt{9 + 16}$$
$$= \sqrt{25}$$
$$= 5$$

$$\hat{u} = \frac{1}{|\vec{u}|}\vec{u}$$
$$= \frac{1}{5}\vec{u}$$
$$= (\frac{3}{5}, \frac{4}{5})$$

$$|\hat{u}| = \sqrt{\frac{3^2}{5^2} + \frac{4^2}{5^2}}$$
$$= \sqrt{\frac{9}{25} + \frac{16}{25}}$$
$$= \sqrt{\frac{25}{25}} = 1$$

Rotation.

Given a vector $\vec{v}$, we can produce a new vector $\vec{u}$ that has the same length as $\vec{v}$ but a different direction.

$$u_x = v_x \cos\theta - v_y \sin\theta$$
$$u_y = v_x \sin\theta + v_y \cos\theta$$

We can also express this as the product of a 2×2 matrix with the vector. The product of the matrix and the vector is another vector.

The elements of the matrix are the cosines and sines of the angle of rotation. The matrix describes the rotation. Matrix multiplication is the operation that produces the rotation.

$$\vec{v} = (v_x, v_y)$$
$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\vec{u} = \mathbf{R}\vec{v}^T$$
$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$
$$= ((\cos\theta)v_x - (\sin\theta)v_y, (\sin\theta)v_x + (\cos\theta)v_y)$$

Here, the elements of the vector appear in a column. We may think of each row of the matrix as a vector. The two elements of our result are each the dot product of a row of the matrix with the column vector.

The dot product is a very handy operation!

$$\vec{v} = (1, 1)$$
$$\mathbf{R} = \begin{bmatrix} \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix}$$

$$\vec{u} = \mathbf{R}\,\vec{v}^T$$
$$= \begin{bmatrix} \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
$$= ((\cos\frac{\pi}{2}) \cdot 1 - (\sin\frac{\pi}{2}) \cdot 1, (\sin\frac{\pi}{2}) \cdot 1 + (\cos\frac{\pi}{2}) \cdot 1)$$
$$= (-1, 1)$$

How might you test code that rotates a vector? Which vectors would you use in your tests? Which angles of rotation would you use in your tests? (Hints: How about vectors that are aligned with the coordinate axes? How about rotations of 0, $\pi/2$, or π radians?)

Weighted averages.

In computer graphics, we often want to split a line segment into two smaller line segments. Given points **a** and **b**, we will find a third point **c** that lies on the line segment **ab** between the endpoints. Then we will create new line segments **ac** and **cb**.

We want a way to find a point on the segment that lies one fourth, one third, one half, or some other fraction of the distance from the first endpoint to the second. Weighted averages will let us do this.

Let t be the "weight" in a weighted average. The value of the weight lies between zero and one. The value of the weighted average lies between the two given values a and b whose weighted average we are computing.

$$f(a, b, t) = (1 - t)a + tb$$

For example, a student's grade in a course might be a weighted average of the score earned on a mid-term examination and the score earned on a final examination. Let's say that the mid-term examination determines 40% of the grade, the final examination determines 60% of the grade, and the student earned 80 points on the midterm and 90 points on the final.

$$t = 0.6$$
$$midterm = 80$$
$$final = 90$$
$$grade = (1 - 0.6) \cdot 80 + 0.6 \cdot 90$$
$$= 32 + 54$$
$$= 86$$

We can compute the weighted averages of vectors, too. The elements of the weighted average of two vectors are just the weighted average of the x components and the weighted average of the y components.

$$\vec{f}(\vec{u}, \vec{v}, t) = ((1 - t)u_x + tv_x, (1 - t)u_y + tv_y)$$

We can again write this as product of a matrix and a vector.

$$\vec{f}(\vec{u}, \vec{v}, t) = \begin{bmatrix} u_x & v_x \\ u_y & v_y \end{bmatrix} \begin{bmatrix} (1 - t) \\ t \end{bmatrix}$$

If $\vec{u}$ and $\vec{v}$ describe the positions of points in the plane, then that weighted average of $\vec{u}$ and $\vec{v}$ describes the position of a point on the line defined by the given points. If $0 \leq t \leq 1$, then the computed point lies between the two given points. If $t = 0.5$, it is the midpoint of the line segment. If $t = 1/3$, it is a point one third of the distance from the first to the second point, and so on.

Classes, objects, and methods.

A class is a blueprint for the construction of objects. An object is a bundle of related data and methods for working with that data. A method is a sequence of instructions for the computer that accomplish a single well-defined goal.

We can write code for testing our methods before writing the methods. We can write explanations of what each part of our program does before writing code. Indeed, the tests and documentation can be our guide as we write the code.

Success in our field requires an ability to express ideas with diagrams, mathematical notation, programming languages, and English (or another natural language). We are showing you in this exercise how several forms of expression can complement one another. Very often we will find it easier to understand a design if we look at it in more than one way.

Review of vector arithmetic.

Try your hand at this.

$$\vec{u} = (7, 24) \tag{1}$$
$$\vec{v} = (8, 15) \tag{2}$$
$$\tag{3}$$
$$|\vec{u}| = ?? \tag{4}$$
$$|\vec{v}| = ?? \tag{5}$$
$$\vec{u} + \vec{v} = ?? \tag{6}$$
$$\vec{u} - \vec{v} = ?? \tag{7}$$
$$\vec{u} \cdot \vec{v} = ?? \tag{8}$$
$$2\vec{u} = ?? \tag{9}$$

Given a vector $\vec{v} = (1, 1)$, what do get if you normalize the vector? What do you get if you rotate the vector through an angle of $\pi/2$ radians?

Preview of what comes next.

At the very beginning of your mathematical education, you learned about the counting numbers. Later you learned about fractions. You learned one rule for adding whole numbers, then another rule for adding fractions.

A fraction is a number with two parts, a numerator and denominator. Other kinds of numbers also have two parts. For example, you know how to add weights. The sum of 1lb., 8oz. and 2lb., 10oz. is not 3lb., 18oz.!

In fact, there are many kinds of mathematical objects that have two (or more) parts...

- angles (degrees and minutes of arc)

- geographic coordinates (latitude and longitude)

- lengths (feet and inches)

- times (hours and minutes)

- currency (dollars and cents, or pounds and shillings)

- complex numbers (real and imaginary parts),

- points or vectors (x and y components)

. . . and so on.

There are different rules for the addition of each of these kinds of measures and numbers.

In our study of computer science, we will learn how to design classes that model many kinds of mathematical objects and their arithmetic. We will reuse the pattern that we are learning in this exercise.

For other classes, we will again want. . .

- a means of creating objects (for example, a means of creating a particular vector or a particular fraction)

- a means of peeking at the data stored within an object (for example, reading the value of the vector's x component or the fraction's denominator)

- a means of changing the values stored within an object

- a means of producing a representation of an object for human eyes (for example, we show a fraction by printing the value of the numerator, then a slash, and then the denominator)

- a means of combining two objects to get a third object (for example, add two weights to get a third weight)

You will not be able to repeat all that we have done with vectors in every case. We will have to adapt our pattern. Here are two differences that you might recognize right way.

- We will not find an operation that is analogous to the dot product for weights.

- Although we have no means of comparing one vector to another, we might want a way of deciding that one weight is less than, greater than, or equal to a second weight.

Through practice, you will gain experience. With experience will come knowledge, skill, and confidence. Your understanding of the pattern we are showing you here will deepen when you write other, similar classes.

In later lessons, we will learn how to reuse a pattern without rewriting all of the code. We will define relationship among the classes that we write. The computer will then copy relevant parts of our previous work for us automatically into our new work. When we write a new class, we will only have to define the parts that distinguish it from existing classes.

A big part of computer science is the search for ways to minimize the amount of work that we must do!