

LINEAR ALGEBRA IN GAME DEVELOPMENT

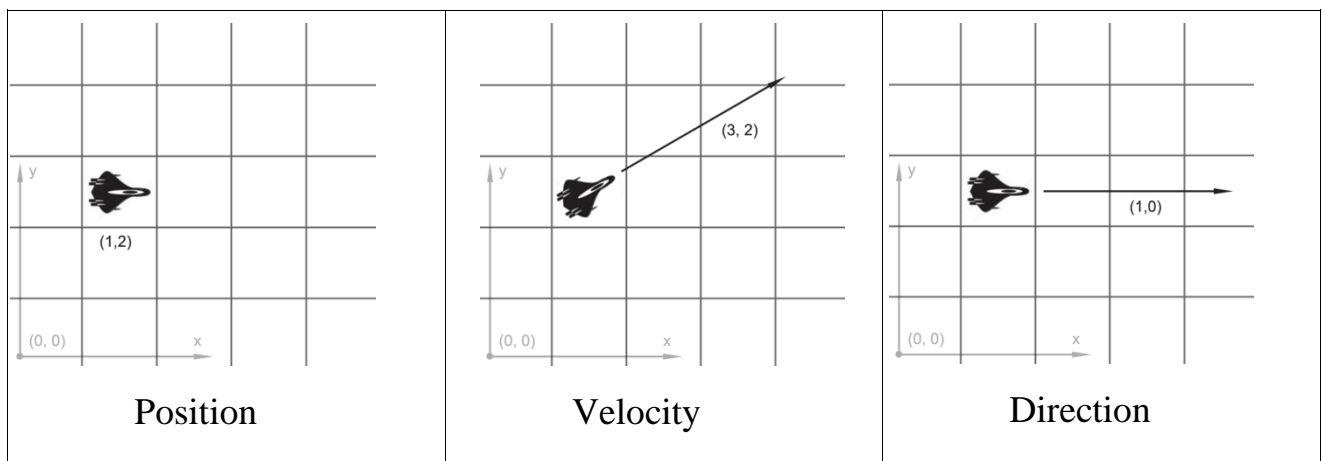
Кореняк Богдан

Київський національний університет технологій та дизайну

Науковий керівник – к. пед.н. Гальченко О. Ю.

Linear algebra is the study of vectors. If your game involves position, direction and velocity of some object, you will have to use vectors. The better one understand linear algebra, the more control you will have over the behavior of these vectors.

In games, vectors are used to store positions, directions, and velocities. Here are some 2-Dimensional examples:

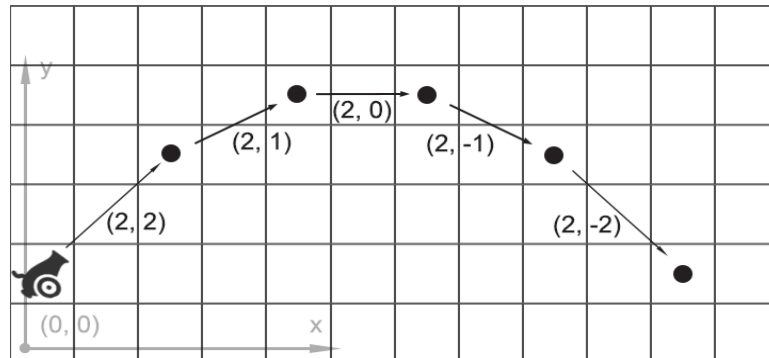


A vector by itself is just a set of numbers - it is only given meaning by its context. *Position vector* shows us that the spaceship is in the cell with coordinates (1, 2) from the initial point. *Velocity vector* indicates that in one time unit the spaceship moves two cells up and three cells to the right. Finally, the *direction vector* tells us that the ship is pointing to the right.

To add vectors together, you just add each component together separately. For example: $(2, 6) + (0, -5) = (2, 6-5) = (2, 1)$

Why do we need to add vectors together? The most common application for this is a physics integration. Let`s consider the mechanics of cannon projectile flight. Projectile begins its movement from the (0, 1) point. In the moment of shot its speed is equals to (2, 2). So in the one unit of time the projectile will move two cells up and

two cells to the right. Its velocity is equals to (0, -1) because the gravity is pulling it down. You can see the result of projectile flight below:

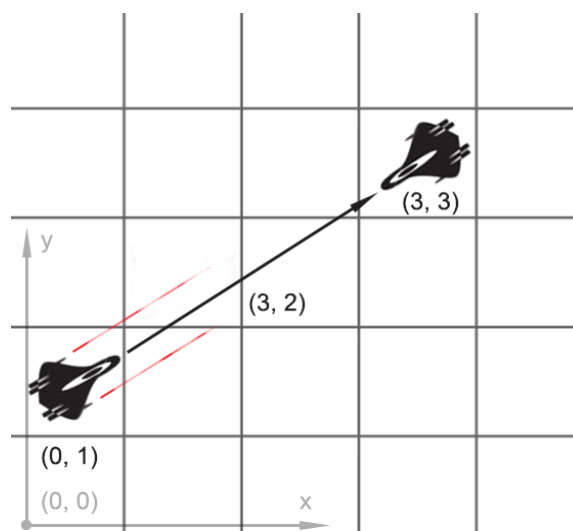


Projectile flight example

When we talk about vectors, we refer to individual numbers as scalars. For example, (0, 1) is a vector, 3 is a scalar. In games, it is often useful to multiply a vector by a scalar. For example, we can simulate basic air resistance by multiplying the player's velocity by 0.7 every frame. To do this, we just multiply each component of the vector by the scalar. If the player's velocity is (20, 30), the new velocity is:

$$0.7 * (20, 30) = (0.7 * 20, 0.7 * 30) = (14, 21).$$

Subtraction works in the same way as addition - subtracting one component at a time. Vector subtraction is useful for getting a vector that points from one position to another. For example, player needs to know in which direction he should shoot the enemy ship. To get the vector of the laser projectile we just subtract the player's position from the enemy position: $(3, 3) - (0, 1) = (3, 2)$

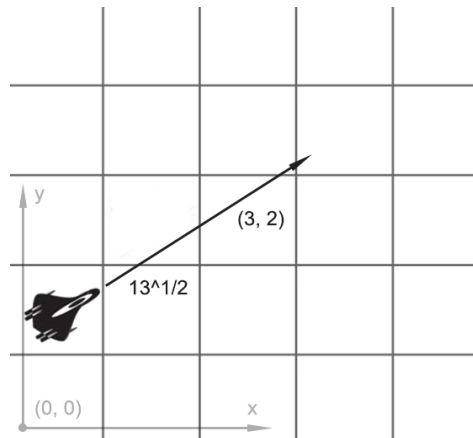


Vector subtraction example

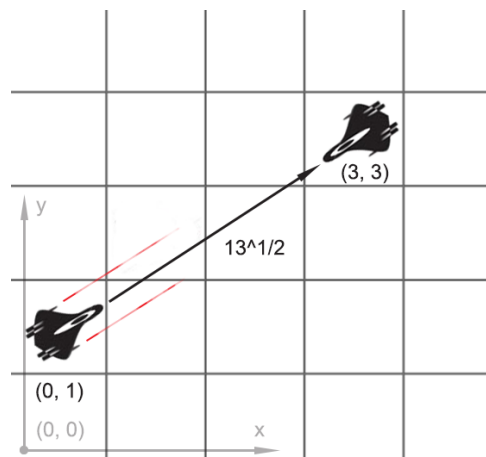
If we have a ship with velocity vector $V(3, 2)$, we might also want to know how fast it is going, in order to calculate how much fuel it should use. To do that, we need to find the length (or magnitude) of vector V .

We can think of V as a right triangle with sides 3 and 2, and use the Pythagorean theorem to find the hypotenuse.

So the length of a vector V with components (x, y) is $\sqrt{x^2+y^2}$. In our case the length is $13^{1/2}$.



Distance between our ships can be easily found by combining two operations that we have already considered above: subtraction and length finding. Now we must ensure that our weapons can shoot the enemy. First we should subtract the player's position from the enemy position: $(3, 3) - (0, 1) = (3, 2)$, then we just find the length of the vector $(3, 2) - 13^{1/2}$.



Normalization

When we are dealing with directions, it is important that they have vector length that equal to 1. This makes life a lot easier for us. For example, let's say there is a weapon pointing in the direction of (1, 0) that shoots a projectile at 40 m/s. What is the velocity of the projectile? Since the direction has length 1, we can just multiply the direction and the projectile speed to get its velocity: (40, 0). If the direction vector had any other length, we couldn't do this - the projectile would be too fast or too slow.

A vector with a length of 1 is called "normalized". So how do we set vector's length to 1? We just divide each component by the vector's length. If we want to normalize vector V with components (3, 4), we just divide each component by its length, 5, to get (3/5, 4/5).

What is the dot product (written \bullet)? To get the dot product of two vectors, we multiply the components, and then add them together:

$$(a_1, a_2) \bullet (b_1, b_2) = a_1b_1 + a_2b_2$$

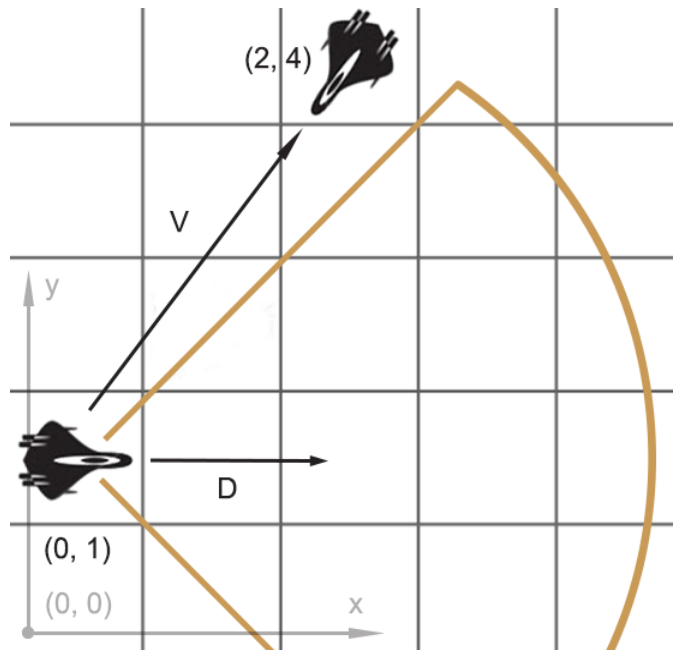
Mathematical expression of the dot product is looks like this:

$$A \bullet B = |A| |B| \cos\Theta$$

Where Θ is «theta» - angle between vectors A and B. This expression allows us to find this angle according to the next formula:

$$\Theta = \arccos([A B] / [|A| |B|]) \text{ or } \Theta = \arccos(A B) \text{ for normalized vectors.}$$

Let's consider following example. The viewing angle from the cockpit is 90° . We want to ensure that our pilot can see the enemy ship from his cockpit. First, we get the normalized vectors for the direction our pilot is facing (D), and the direction from the pilot's ship to the enemy (V). Then, we check the angle between them. If it is greater than 45° (half of the field of view), then the enemy ship is not seen.



To applicate all this stuff in real game one may try to write a simple game engine for action like old Battle City game with the logic describing basic game physics.

REFERENCES

1. S. Roman Advanced Linear Algebra / Steven Roman. – (Springer).
2. G. Strang Linear Algebra and Its Applications / Gilbert Strang. – (4th Edition). J. Hefferon Linear Algebra / Jim Hefferon. – (3d Edition).
3. J. Hefferon Linear Algebra / Jim Hefferon. – (3d Edition).