

In one tick, we will have moved speed amount along a particular angle, so we represent this as an amount in the x and an amount in the y.

this.speedX = (float)(speed \* Math.cos(Math.toRadians(angleInDegree)));

this.speedY = (float)(-speed \* (float)Math.sin(Math.toRadians(angleInDegree)));

ignoring angles (we are coming off at 90 degrees)

float defaultXVel = (float)Math.Cos(defaultSpeed) \* defaultSpeed;

float defaultYVel = (float)Math.Cos(defaultSpeed) \* defaultSpeed;

In the example below, with our normal axis (origin bottom left), once the ball has hit the rectangle, it bounces off.

As X is increasing, Y is now decreasing as the negative of the rate at which it was increasing prior to the collision.

Y increases ->

X increases ->