Oscillation with angular velocity

An understanding of the concepts of oscillation, amplitude, and frequency/period is often required in the course of simulating real-world behaviours. However, there is a slightly easier way to rewrite the above example with the same result. Let’s take one more look at our oscillation formula:

```
var x = amplitude * sin(TWO_PI * frameCount / period);
```

And let’s rewrite it a slightly different way:

```
var x = amplitude * sin(some value that increments slowly);
```

If we care about precisely defining the period of oscillation in terms of frames of animation, we might need the formula the way we first wrote it, but we can just as easily rewrite our example using the concept of angular velocity (and acceleration) from the [Angular Motion lesson](). Assuming:

```
var angle = 0;

var aVelocity = 0.03;
```

...in `draw()`, we can simply say:

```
angle += aVelocity;

var x = amplitude * sin(angle);
```

...where `angle` is our “some value that increments slowly.”

Here's our modified program:

```
angleMode = "radians";


var angle = 0;
var aVelocity = 0.03;


var draw = function() {
   background(255);


   var amplitude = 100;
   var x = amplitude * sin(angle);
   angle += aVelocity;


   resetMatrix();
   translate(width/2, height/2);


   stroke(0, 0, 0);
   fill(175, 175, 175);
   line(0, 0, x, 0);
   ellipse(x, 0, 20, 20);
};
```

Just because we’re not referencing it directly doesn’t mean that we’ve eliminated the concept of period. After all, the greater the angular velocity, the faster the circle will oscillate (therefore lowering the period). In fact, the number of times it takes to add up the angular velocity to get to TWO_PI is the period or:

period = TWO_PI / angular velocity

Let’s expand this example a bit more and create an `Oscillator` object. And let’s assume we want the oscillation to happen along both the x-axis (as above) and the y-axis. To do this, we’ll need two angles, two angular velocities, and two amplitudes (one for each axis). Another perfect opportunity for `PVector`!

Read through the code in the program below:

```
angleMode = "radians";

var Oscillator = function() {
  this.angle = new PVector();
  this.velocity = new PVector(random(-0.05, 0.05), random(-0.05, 0.05));
  this.amplitude = new PVector(random(20, width/2), random(20, height/2));
};

Oscillator.prototype.oscillate = function() {
  this.angle.add(this.velocity);
};

Oscillator.prototype.display = function() {
  var x = sin(this.angle.x) * this.amplitude.x;
  var y = sin(this.angle.y) * this.amplitude.y;

  pushMatrix();
  translate(width/2, height/2);
  stroke(0, 0, 0);
  strokeWeight(2);
  fill(127, 127, 127, 127);
  line(0, 0, x, y);
  ellipse(x, y, 32, 32);
  popMatrix();
};

var oscillators = [];
for (var i = 0; i < 10; i++) {
    oscillators.push(new Oscillator());
}

var draw = function() {
  background(255, 255, 255);
  for (var i = 0; i < oscillators.length; i++) {
    oscillators[i].oscillate();
    oscillators[i].display();
  }
};
```