

Particle systems with forces

So far in this section, we've been focusing on structuring our code in an object-oriented way to manage a collection of particles. Maybe you noticed, or maybe you didn't, but during this process we unwittingly took a couple steps backward from where we were in previous sections. Let's examine the constructor of our simple Particle object:

```
var Particle = function(position) {  
    this.acceleration = new PVector(0, 0.05);  
    this.velocity = new PVector(random(-1, 1), random(-1, 0));  
    this.position = new PVector(position.x, position.y);  
    this.timeToLive = 255.0;  
};
```

And now let's look at the `update()` method:

```
Particle.prototype.update = function() {  
    this.velocity.add(this.acceleration);  
    this.position.add(this.velocity);  
    this.timeToLive -= 2;  
};
```

Notice that our acceleration is constant, it's never set beyond the constructor. A much better framework would be to follow Newton's second law ($\vec{F} = M\vec{A}$ $\vec{F} = M\vec{A}$ \vec{F} , with, vector, on top, equals, M, A, with, vector, on top) and incorporate the force accumulation algorithm we worked so hard on in the Forces section.

The first step is to add in an `applyForce()` method. (Remember, we need to make a copy of the `PVector` before we divide it by mass.)

```
Particle.prototype.applyForce = function(force) {
```

```
var f = force.get();  
f.div(this.mass);  
this.acceleration.add(f);  
};
```

Once we have this, we can add in one more line of code to clear the acceleration at the end of `update()`.

```
Particle.prototype.update = function() {  
    this.velocity.add(this.acceleration);  
    this.position.add(this.velocity);  
    this.acceleration.mult(0);  
    this.timeToLive -= 2.0;  
};
```

And thus, we have a `Particle` object that can have force applied to it. Now, where should we call the `applyForce()` function? Where in the code is it appropriate to apply a force to a particle? The truth of the matter is that there's no right or wrong answer; it really depends on the exact functionality and goals of a particular program. Still, we can create a generic situation that would likely apply to most cases and craft a model for applying forces to individual particles in a system.