Simulating Natural Phenomena

Lec3

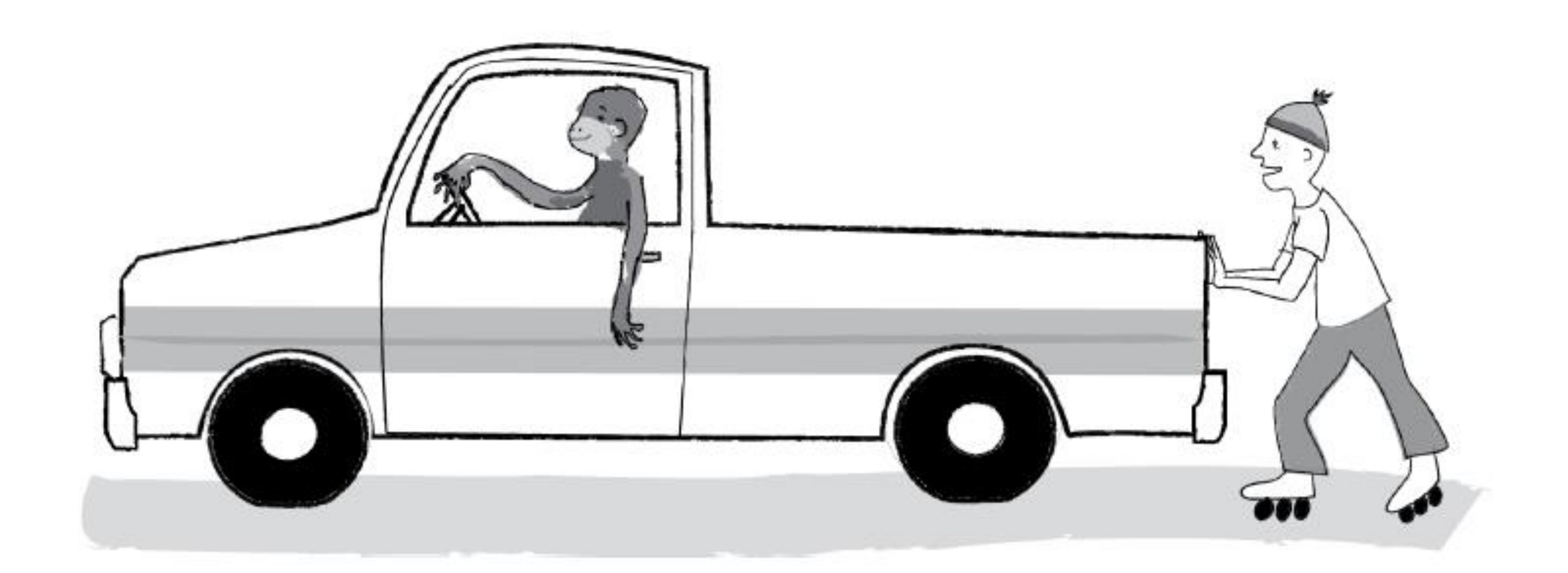
Force and Acceleration

2023-2024

High Diploma 2nd Course

Nyan Dawood

Force and Acceleration



Newton First law:
“An object at rest stays at rest and an object in motion stays in motion at a constant speed and direction unless acted upon by an unbalanced force”.

Newton second law:
“Force equals mass times acceleration”.

Or:

$$\vec{F} = M \times \vec{A}$$

$$\vec{A} = \vec{F} \,/\, M$$

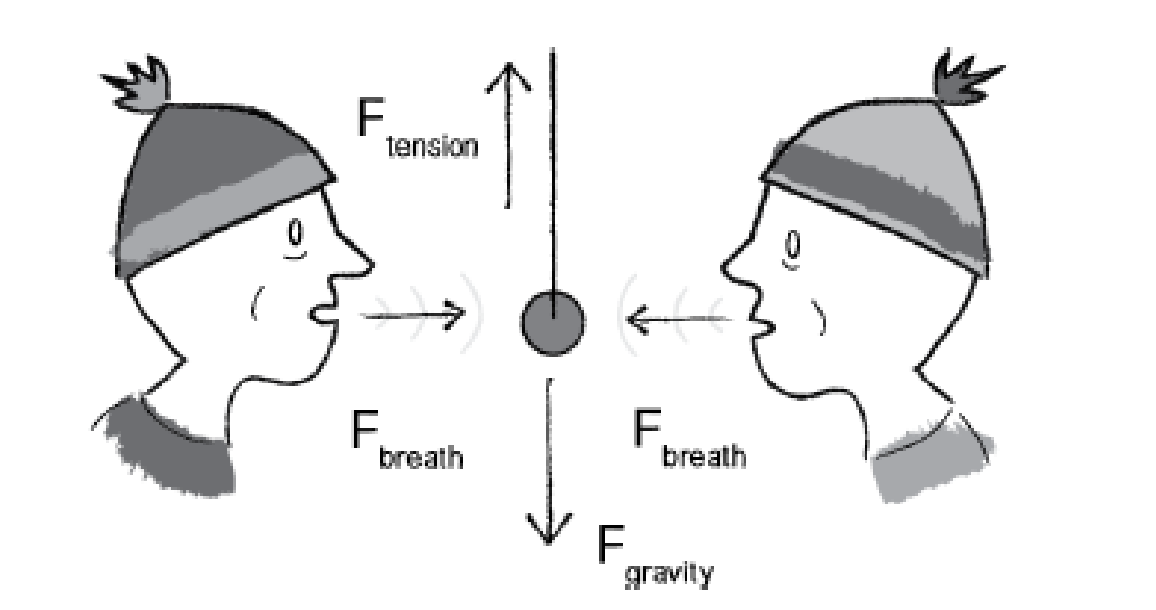
Acceleration is directly proportional to force and inversely proportional to mass. This means that if you get pushed, the harder you are pushed, the faster you’ll move (accelerate). The bigger you are, the slower you’ll move.

in the world of Processing, what is mass anyway? Aren't we dealing with pixels?

To start in a simpler place, let's say that in our pretend pixel world, all of our objects have a mass equal to 1.

A=F/1 .

Supposing the force applied by wind (horizontal action), and gravity (vertical action)



What about Equilibrium?

Net Force equals mass times acceleration.

The implementation of multi force act on same object will be through a process of “**force accumulation.**”

```
void applyForce(Pvector f) {
        acceleration.add(f);
}
                // adding each force to acceleration one at a time.
```

The acceleration was the key to controlling the movement of our objects on screen.

Location is adjusted by velocity, and velocity by acceleration. Acceleration was where it all began.

Now we learn that force is truly where it all begins.

```
 class Mover {
   PVector location;
   PVector velocity;
   PVector acceleration;
}
```

The goal to be able add force to the object

creating a moving object on the screen that responds to wind and gravity.

```
mover.applyForce(wind);
mover.applyForce(gravity);
mover.update();
mover.display();
```

- if we had forgotten to reset acceleration to zero, the force of wind would still be in effect. Even worse, it would add onto itself from the previous frame.

This problem will be solved by cancel the acceleration each frame by:

```
void update() {
velocity.add(acceleration);
location.add(velocity);
acceleration.mult(0);
}
```

Force analysis

```
void draw() {
    background(255);
       PVector wind = new PVector(0.01,0);
                                                  //horizontal force  " wind "
       PVector gravity = new PVector(0,0.1);
                                                 // vertical force    "gravity"
      m.applyForce(wind);
      m.applyForce(gravity);
      m.update();
      m.display();
      m.checkedges();
}
```

```
class Mover {
    PVector location;
    PVector velocity;
    PVector acceleration;
    float mass;                                   //The object now has mass!
Mover() {
    mass = 1;                                    //for simples the calculation
    location = new PVector(30,30);
    velocity = new PVector(0,0);
    acceleration = new PVector(0,0);
}
```

```
void applyForce(PVector force) {                    //Newton’s second law.
        PVector f = PVector.div(force,mass);
        acceleration.add(f);
}
void update() {
        velocity.add(acceleration);
        location.add(velocity);
        acceleration.mult(0);   }
void display() {
        stroke(0);
        fill(175);
        ellipse(location.x,location.y,mass*16,mass*16);
}                                                   // Scaling the size according to mass.
```

```
void checkEdges() {
    if (location.x > width) {                    // bouncing when reach edges
    location.x = width;
    velocity.x *= -1;
} else if (location.x < 0) {
    velocity.x *= -1;
    location.x = 0;    }
}
if (location.y > height) {
    velocity.y *= -1;
    location.y = height;
}
}   }
```

Q: Sometimes the wind is very strong, sometimes it's weak, and sometimes there's no wind at all, how to Simulate these cases?

Multi mover

```
Mover[] movers = new Mover[100];
void setup() {
//Initializing many Mover objects, all with random mass (and all starting at 0,0)
    for (int i = 0; i < movers.length; i++) {
    movers[i] = new Mover(random(0.1,5),0,0);
                        // Mover(mass, start_X,start_Y)
}
}
```

```
void draw() {
    background(255);
    PVector wind = new PVector(0.01,0);
    PVector gravity = new PVector(0,0.1);              //Make up two forces.
          //Loop through all objects and apply both forces to each object.
for (int i = 0; i < movers.length; i++) {
    movers[i].applyForce(wind);
    movers[i].applyForce(gravity);
    movers[i].update();
    movers[i].display();
    movers[i].checkEdges();   }}
```

Note :

Note how in the multi mover balls , the smaller circles reach the right of the window faster than the larger ones. This is because of our formula: acceleration = force divided by mass. The larger mass has smaller acceleration.

```
for (int i = 0; i < movers.length; i++) {
    PVector wind = new PVector(0.001,0);
    float m = movers[i].mass;          //Scaling gravity by mass to be more accurate

   PVector gravity = new PVector(0,0.1*m);
    movers[i].applyForce(wind);
    movers[i].applyForce(gravity);
    movers[i].update();
    movers[i].display();
    movers[i].checkEdges();
}
```

Exercise :

Using forces, simulate a helium-filled balloon floating upward and bouncing off the top of a window. Can you add a wind force that changes over time.