

Section G
Unit #2
Matter.js
Constraints



Section G Unit #2 constraints

Sketch G2.1	blocks to bubbles
Sketch G2.2	two bubbles
Sketch G2.3	constraint
Sketch G2.4	drawing a line
Sketch G2.5	making a chain
Sketch G2.6	fixing one end
Sketch G2.7	rest length
Sketch G2.8	adding the lines



Introduction to unit #2 constraints

We are going to constrain a line of circles (bubbles). We need to add the matter.js line of code and add a file called bubble.js as we did previously with the blocks.

One of the reasons we aren't using the word circle is that it is a taken word, the plural is OK so I have decided to use the words bubble and bubbles instead of circle and circles (as I did with block and blocks)



Sketch G2.1 index.html

The addition of the bubble.js file (and matter.js)

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/p5.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/addons/p5.sound.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/matter-js/0.20.0/matter.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8" />

  </head>
  <body>
    <main>
    </main>
    <script src="sketch.js"></script>
    <script src="bubble.js"></script>
  </body>
</html>
```

p5* File Edit Sketch Help English Hello, TheHappyCoder!

Auto-refresh matter.js new by TheHappyCoder

Sketch Files index.html*

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/p5.js"></script>
5   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.10.0/addons/p5.sound.min.js"></script>
6   <script src="https://cdnjs.cloudflare.com/ajax/libs/matter-js/0.20.0/matter.min.js"></script>
7   <link rel="stylesheet" type="text/css" href="style.css">
8   <meta charset="utf-8" />
9
10 </head>
11 <body>
12 <main>
13 </main>
14 <script src="sketch.js"></script>
15 <script src="bubble.js"></script>
16 </body>
17 </html>
18
```



Sketch G2.2 bubbles (part 1)

Our starting sketch

sketch.js

```
const {Engine, Body, Bodies, Composite} = Matter
let engine
let bubble
let world
let bubbles = []

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
}

function mousePressed()
{
  bubbles.push(new Bubble(mouseX, mouseY, 20))
}

function draw()
{
  background(220)
  Engine.update(engine)
  for (let i = 0; i < bubbles.length; i++)
  {
    bubbles[i].show()
  }
}
```



Sketch G2.2 bubbles (part 2)

For our Bubble class, again very similar

bubble.js

```
class Bubble
{
  constructor(x, y, r)
  {
    let options = {
      friction: 0.3,
      restitution: 0.6
    }
    this.body = Bodies.circle(x, y, r, options)
    this.r = r
    Composite.add(world, this.body)
  }

  show()
  {
    let pos = this.body.position
    let angle = this.body.angle
    push()
    translate(pos.x, pos.y)
    rotate(angle)
    fill(200, 0, 0)
    circle(0, 0, this.r * 2)
    pop()
  }
}
```

Notes

Every time you click on the canvas you should get a bubble (red circle)



Constraints

Using much of the code we have just created we are going to look at constraints. This is where you might join two objects that have a fixed or elastic constraint. Like two spheres connected by a string or a piece of elastic.

We can remove all reference to the slopes and ground. (If you are unsure about making all these changes then I suggest duplicating what you have already so that you have something to fall back on.

We also remove the `mousePressed()` function.



Sketch G2.3 two bubbles (part 1)

We want just two bubbles for now. They will behave like the blocks and fall through the floor (for now), we will call them `bubble1` and `bubble2` for simplicity.

sketch.js

```
const {Engine, Body, Bodies, Composite} = Matter
let engine
let bubble1
let bubble2
let world
let bubbles = []

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  bubble1 = new Bubble(200, 100, 10)
  bubble2 = new Bubble(250, 150, 10)
  bubbles.push(bubble1)
  bubbles.push(bubble2)
}

// function mousePressed()
// {
//   bubbles.push(new Bubble(mouseX, mouseY, 20))
// }

function draw()
{
  background(220)
```

```
Engine.update(engine)
for (let i = 0; i < bubbles.length; i++)
{
  bubbles[i].show()
}
}
```

Notes

All we now have is two bubbles falling, they are offset slightly from each other. They still fall through the bottom of the canvas.



Sketch G2.4 constraint

Now to make a constraint and give it options, also offset the two bubbles

```
sketch.js

const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let bubble1
let bubble2
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  bubble1 = new Bubble(200, 100, 10)
  bubble2 = new Bubble(250, 150, 10)
  bubbles.push(bubble1)
  bubbles.push(bubble2)
  let options = {
    bodyA: bubble1.body,
    bodyB: bubble2.body,
    length: 40,
    stiffness: 0.4
  }
  constraint = Constraint.create(options)
  Composite.add(world, constraint)
}
```

```
function draw()
{
  background(220)
  Engine.update(engine)
  for (let i = 0; i < bubbles.length; i++)
  {
    bubbles[i].show()
  }
}
```



Sketch G2.5 drawing a line

Drawing a line between the two bubbles, be warned it is a ridiculously long line of code!

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let bubble1
let bubble2
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  bubble1 = new Bubble(200, 100, 10)
  bubble2 = new Bubble(250, 150, 10)
  bubbles.push(bubble1)
  bubbles.push(bubble2)
  let options = {
    bodyA: bubble1.body,
    bodyB: bubble2.body,
    length: 40,
    stiffness: 0.4
  }
  constraint = Constraint.create(options)
  Composite.add(world, constraint)
}
```

```
function draw()
{
  background(220)
  Engine.update(engine)
  for (let i = 0; i < bubbles.length; i++)
  {
    bubbles[i].show()
  }
  line(bubbles[0].body.position.x, bubbles[0].body.position.y,
  bubbles[1].body.position.x, bubbles[1].body.position.y)
}
```

Notes

What you should see is a line drawn between the two bubbles as they fall.



Sketch G2.6 making a chain

Let's make a chain of them, commenting out the line for now and removing reference to `bubble1` and `bubble2`, have two new variables called `prevBubble` and `otherBubble`. You should see a row of bubbles fall to the ground stretched in a row starting at 100 pixels from the top. There a lot to take in really, mostly it is renaming but also reorganising. Take your time to work through the code.

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let prevBubble
let otherBubble
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  prevBubble = null
  for (let x = 40; x < 360; x += 40)
  {
    otherBubble = new Bubble(x, 100, 10)
    bubbles.push(otherBubble)
    if (prevBubble)
    {
      let options = {
        bodyA: otherBubble.body,
        bodyB: prevBubble.body,
```

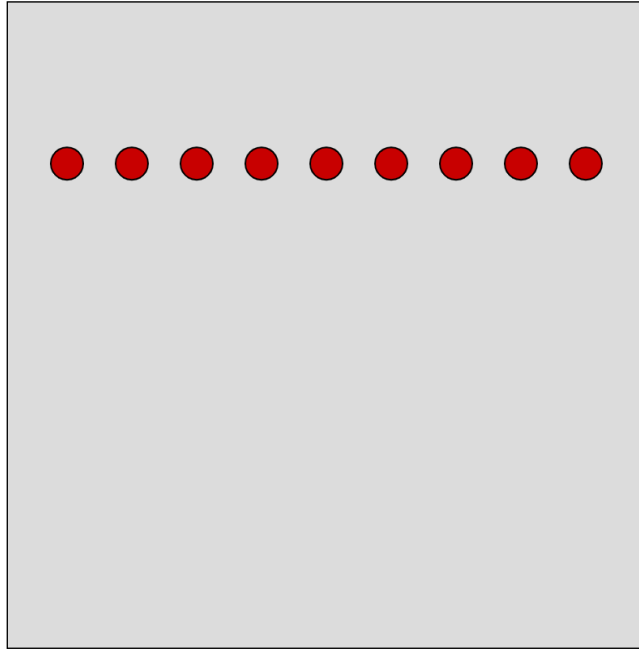
```

        length: 40,
        stiffness: 0.4
    }
    constraint = Constraint.create(options)
    Composite.add(world, constraint)
}
prevBubble = otherBubble
}
}

function draw()
{
    background(220)
    Engine.update(engine)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    // line(bubbles[0].body.position.x,
    bubbles[0].body.position.y, bubbles[1].body.position.x,
    bubbles[1].body.position.y)
}

```


A row falling





Sketch G2.7 fixing one end (part 1)

Now to fix one of the bubbles in **bubble.js** by adding another argument to the Bubble constructor and including that in the options, then move back to the **sketch.js**

bubble.js

```
class Bubble
{
  constructor(x, y, r, fixed)
  {
    let options = {
      friction: 0.3,
      restitution: 0.6,
      isStatic: fixed
    }
    this.body = Bodies.circle(x, y, r, options)
    this.r = r
    Composite.add(world, this.body)
  }

  show()
  {
    let pos = this.body.position
    let angle = this.body.angle
    push()
    translate(pos.x, pos.y)
    rotate(angle)
    fill(200, 0, 0)
    circle(0, 0, this.r * 2)
    pop()
  }
}
```



Sketch G2.7 fixing one end (part 2)

Making the fixed particle changes to the **sketch.js**

```
sketch.js

const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let prevBubble
let otherBubble
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  prevBubble = null
  for (let x = 40; x < 360; x += 40)
  {
    let fixed = false
    if(!prevBubble)
    {
      fixed = true
    }
    otherBubble = new Bubble(x, 100, 10, fixed)
    bubbles.push(otherBubble)
    if (prevBubble)
    {
      let options = {
        bodyA: otherBubble.body,
```

```

        bodyB: prevBubble.body,
        length: 40,
        stiffness: 0.4
    }
    constraint = Constraint.create(options)
    Composite.add(world, constraint)
}
prevBubble = otherBubble
}
}

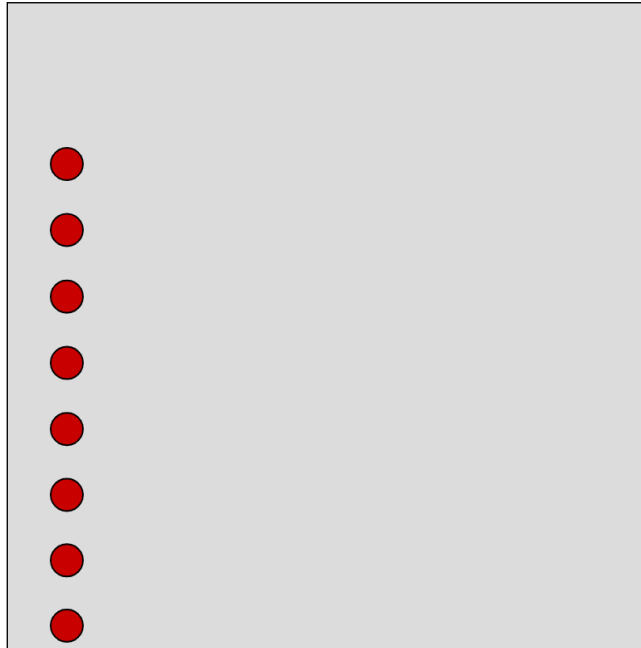
function draw()
{
    background(220)
    Engine.update(engine)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    // line(bubbles[0].body.position.x,
    bubbles[0].body.position.y, bubbles[1].body.position.x,
    bubbles[1].body.position.y)
}

```

Notes

All the bubbles can move except the prevBubble which set to true (fixed) if already false (!prevBubble). The default is false.

They now hang by one





Sketch G2.8 rest length

We will move the bubbles over so that they are in the centre of the canvas and also make the length between them shorter (30)

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let prevBubble
let otherBubble
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  prevBubble = null
  for (let x = 200; x < 560; x += 40)
  {
    let fixed = false
    if(!prevBubble)
    {
      fixed = true
    }
    otherBubble = new Bubble(x, 100, 10, fixed)
    bubbles.push(otherBubble)
    if (prevBubble)
    {
      let options = {
        bodyA: otherBubble.body,
```

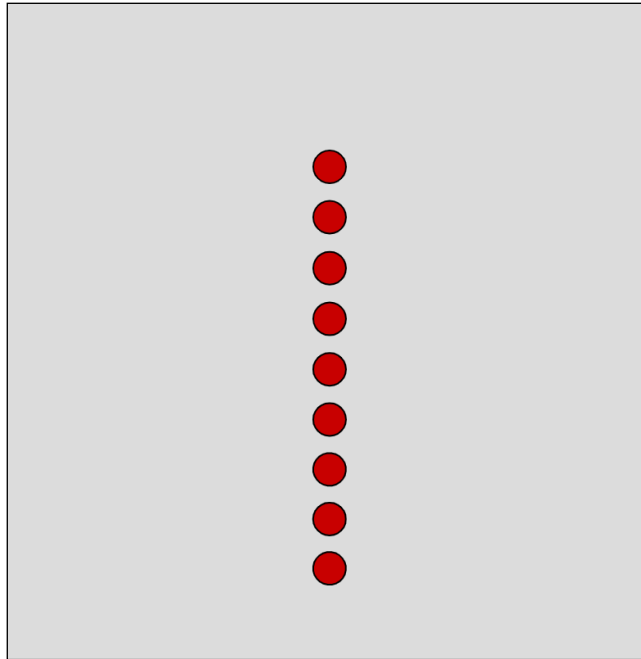
```

        bodyB: prevBubble.body,
        length: 30,
        stiffness: 0.4
    }
    constraint = Constraint.create(options)
    Composite.add(world, constraint)
}
prevBubble = otherBubble
}
}

function draw()
{
    background(220)
    Engine.update(engine)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    // line(bubbles[0].body.position.x,
    bubbles[0].body.position.y, bubbles[1].body.position.x,
    bubbles[1].body.position.y)
}

```

All connected and nicely spaced





Sketch G2.9 adding the lines

We will move the bubbles over so that they are in the centre of the canvas and also make the length between them shorter (30)

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint} = Matter
let engine
let prevBubble
let otherBubble
let world
let bubbles = []
let constraint

function setup()
{
  createCanvas(400, 400)
  engine = Engine.create()
  world = engine.world
  prevBubble = null
  for (let x = 200; x < 560; x += 40)
  {
    let fixed = false
    if(!prevBubble)
    {
      fixed = true
    }
    otherBubble = new Bubble(x, 100, 10, fixed)
    bubbles.push(otherBubble)
    if (prevBubble)
    {
      let options = {
        bodyA: otherBubble.body,
```

```

        bodyB: prevBubble.body,
        length: 30,
        stiffness: 0.4
    }
    constraint = Constraint.create(options)
    Composite.add(world, constraint)
}
prevBubble = otherBubble
}
}

function draw()
{
    background(220)
    Engine.update(engine)
    for (let i = 0; i < bubbles.length-1; i++)
    {
        line(bubbles[i].body.position.x, bubbles[i].body.position.y,
bubbles[i+1].body.position.x, bubbles[i+1].body.position.y)
    }
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
}
}

```

Challenges

1. Change the shape
2. Have the last one static as well

Stringing them together

