

Section G

Unit #3

Matter.js

Mouse



## Section G Unit #3 attaching a mouse

- Sketch G3.1 using elt
- Sketch G3.2 attaching to the mouse
- Sketch G3.3 drawing line to the mouse
- Sketch G3.4 the point clicked



## Introduction to unit #3 attaching a mouse

In this section we will learn how to attach a mouse point to an object (body) so we can manipulate it, ie move it around.



## Sketch G3.1 using elt

Having a mouse grab and release, so starting with the previous sketch

1. Start by adding a canvas object in setup()
2. Use the elt property of that canvas object to the mouse constraint
3. Call it canvas mouse to avoid confusion with the options
4. Add it into the world
5. Need change the pixel ratio if you are on a high density display

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint,
MouseConstraint, Mouse} = Matter
```

```
let engine
```

```
let prevBubble
```

```
let otherBubble
```

```
let world
```

```
let mouse
```

```
let bubbles = []
```

```
let constraint
```

```
let mConstraint
```

```
function setup()
```

```
{
```

```
  let canvas = 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
}

```

```

let canvasMouse = Mouse.create(canvas.elt)
canvasMouse.pixelRatio = pixelDensity()
let options = {
  mouse: canvasMouse
}
mConstraint = MouseConstraint.create(engine, options)
Composite.add(world, mConstraint)

```

```

}

```

```

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()  
}  
}
```



## Sketch G3.2 attaching to the mouse

Shows which one has been attached to the mouse by turning it blue, it will also be attached to the mouse pointer as you click on it.

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint,
MouseConstraint, Mouse} = Matter

let engine
let prevBubble
let otherBubble
let world
let mouse
let bubbles = []
let constraint
let mConstraint

function setup()
{
  let canvas = 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
  }
  let canvasMouse = Mouse.create(canvas.elt)
  canvasMouse.pixelRatio = pixelDensity()
  let options = {
    mouse: canvasMouse
  }
  mConstraint = MouseConstraint.create(engine, options)
  Composite.add(world, mConstraint)
}

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()
  }
}

```

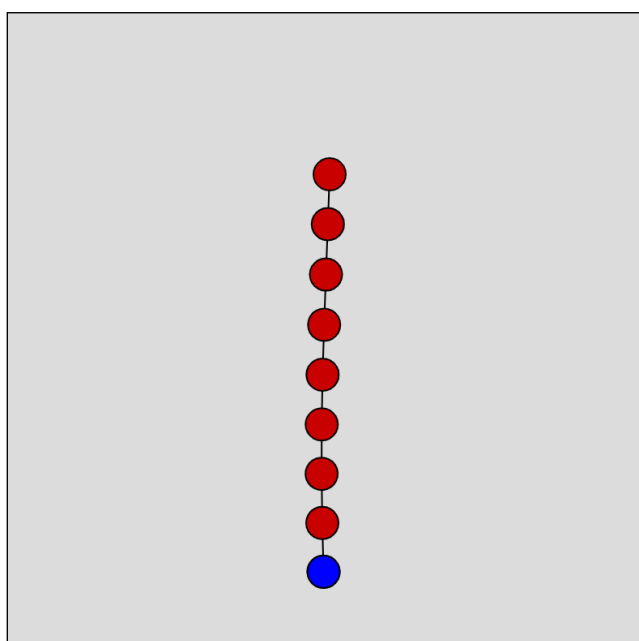


```
}  
if (mConstraint.body)  
{  
  let pos = mConstraint.body.position  
  fill(0, 0, 255)  
  circle(pos.x, pos.y, 20)  
}  
}
```

## Notes

You can do it with any bubble not just the last one

Attached to the mouse and turns blue





## Sketch G3.3 drawing line to mouse

Daw a line from the mouse to the attached particle

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint,
MouseConstraint, Mouse} = Matter

let engine
let prevBubble
let otherBubble
let world
let mouse
let bubbles = []
let constraint
let mConstraint

function setup()
{
  let canvas = 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
}
let canvasMouse = Mouse.create(canvas.elt)
canvasMouse.pixelRatio = pixelDensity()
let options = {
  mouse: canvasMouse
}
mConstraint = MouseConstraint.create(engine, options)
Composite.add(world, mConstraint)
}

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()
  }
}

```

```
}  
  
if (mConstraint.body)  
{  
  let pos = mConstraint.body.position  
  let m = mConstraint.mouse.position  
  line(pos.x, pos.y, m.x, m.y)  
  fill(0, 0, 255)  
  circle(pos.x, pos.y, 20)  
}  
}
```



## Sketch G3.4 the point clicked

Attaching the line to point clicked on rather than the centre of the bubble using **pointB**

sketch.js

```
const {Engine, Body, Bodies, Composite, Constraint,
MouseConstraint, Mouse} = Matter

let engine
let prevBubble
let otherBubble
let world
let mouse
let bubbles = []
let constraint
let mConstraint

function setup()
{
  let canvas = 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
  }
  let canvasMouse = Mouse.create(canvas.elt)
  canvasMouse.pixelRatio = pixelDensity()
  let options = {
    mouse: canvasMouse
  }
  mConstraint = MouseConstraint.create(engine, options)
  Composite.add(world, mConstraint)
}

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()
  }
}

```

```
}  
  
if (mConstraint.body)  
{  
  let pos = mConstraint.body.position  
  let offset = mConstraint.constraint.pointB  
  let m = mConstraint.mouse.position  
  line(pos.x + offset.x, pos.y + offset.y, m.x, m.y)  
  fill(0, 0, 255)  
  circle(pos.x, pos.y, 20)  
}  
}
```