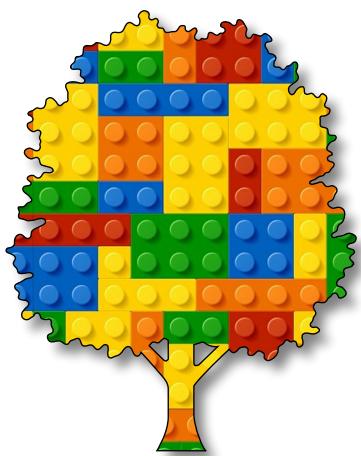


The Joy
of Coding
Module D
Unit #8

3D arrays
part 2





Module D Unit #8 3D arrays part 2

- Sketch D8.1 an array of cubes
- Sketch D8.2 rotating the array
- Sketch D8.3 spaced out
- Sketch D8.4 an array in the y direction
- Sketch D8.5 a cube of cubes
- Sketch D8.6 adding a splash of colour
- Sketch D8.7 a bubble factory
- Sketch D8.8 not a cube
- Sketch D8.9 now a circle
- Sketch D8.10 creating a spiral
- Sketch D8.11 more spirally



Introduction to 3D arrays part 2

We are going to develop the concept of a 3D array and create some interesting shapes and patterns. First off is a cube of cubes and then we will have a spiral of bubbles!



Sketch D8.1 an array of cubes

Starting a new sketch with many of the previous elements. Making an array of **cubes** where we have **0** to **100** in steps of **20** for a cube size of **20** giving us **5** cubes. We do this with a simple **for()** loop.

```
let cubes = []
let cube

function setup()
{
    createCanvas(400, 400, WEBGL)
    for (let i = 0; i < 100; i += 20)
    {
        cube = new Cube(i, 0, 0)
        cubes.push(cube)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < cubes.length; i++)
    {
        cubes[i].show()
    }
}

class Cube
{
    constructor(x, y, z)
    {
        this.x = x
    }
}
```

```
    this.y = y
    this.z = z
}

show()
{
    push()
    translate(this.x, this.y, this.z)
    box(20)
    pop()
}
}
```

Notes

As you can see we have a problem. We have drawn the cubes from **(0, 0)** which is the centre of the canvas, we need to start at **-100** rather than **0**. Also we can only see from the front, it would be nice to rotate the array of cubes to see them properly.

Figure D8.1

The screenshot shows the p5.js code editor interface. At the top, there are buttons for File, Edit, Sketch, Help, and Language (English). To the right, it says "Hello, TheHappyCoder!" with a dropdown arrow. The main area has tabs for "sketch.js" and "Preview". The code in "sketch.js" is as follows:

```
20 }
21 }
22
23 class Cube
24{
25   constructor(x, y, z)
26  {
27    this.x = x
28    this.y = y
29    this.z = z
30  }
31
32  show()
33{
34  push()
35  translate(this.x, this.y, this.z)
36  box(20)
37  pop()
38}
39}
```

The "Preview" window shows a 3D perspective view of a cube centered at the origin. The cube is composed of six faces, each rendered in a light gray color.



Sketch D8.2 rotating the array

Now we rotate the array about the centre of the canvas. Also we have now got 10 cubes

```
let cubes = []
let cube
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 20)
    {
        cube = new Cube(i, 0, 0)
        cubes.push(cube)
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < cubes.length; i++)
    {
        cubes[i].show()
    }
    angle += 0.5
}
```

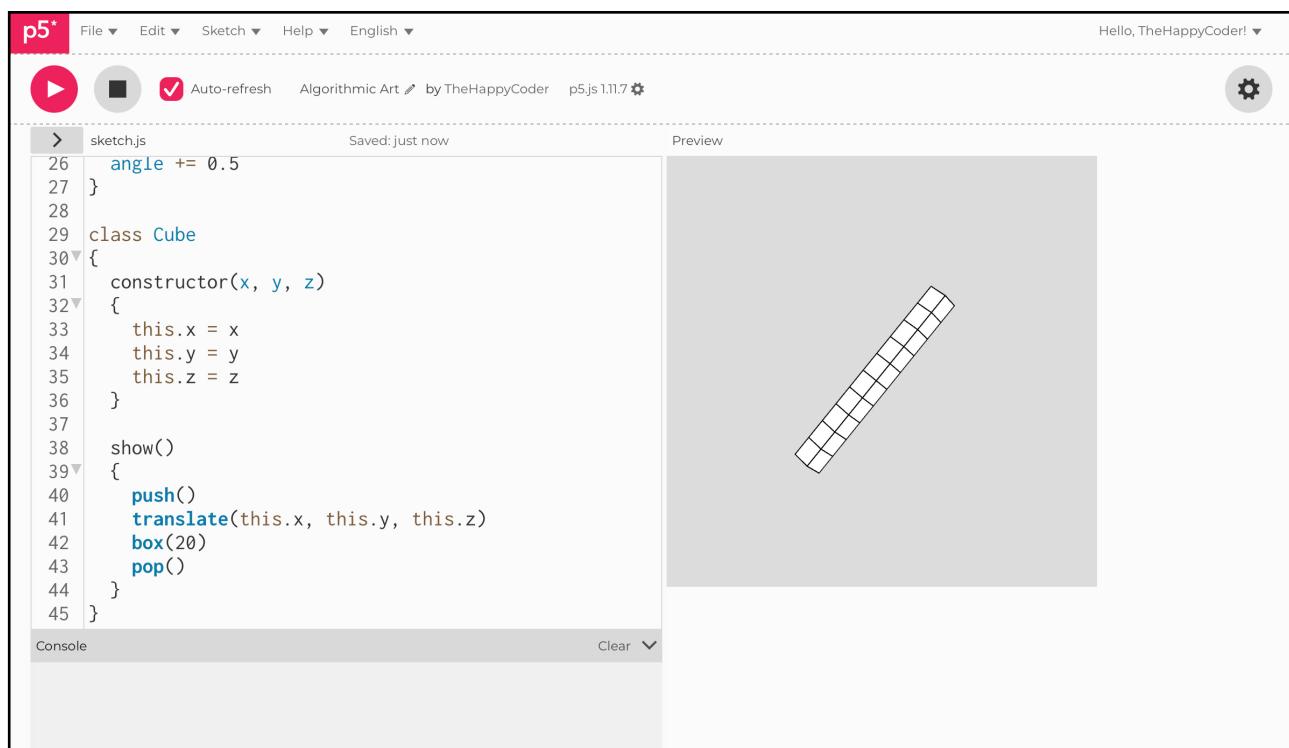
```
class Cube
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        box(20)
        pop()
    }
}
```

Notes

A long stick of cubes or a long cuboid

Figure D8.2



The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play, stop, auto-refresh (which is checked), and help. The title bar says "sketch.js" and "Saved: just now". On the right, it shows "Hello, TheHappyCoder!" and a gear icon. The main area has two panes: "Preview" on the right displaying a 3D perspective drawing of a cube, and "Code" on the left containing the following JavaScript code:

```
angle += 0.5
}
class Cube
{
  constructor(x, y, z)
  {
    this.x = x
    this.y = y
    this.z = z
  }
  show()
  {
    push()
    translate(this.x, this.y, this.z)
    box(20)
    pop()
  }
}
```

Below the code pane is a "Console" section which is currently empty.



Sketch D8.3 spaced out

If we space them out a little bit more by increasing the steps from **20** to **25**, this will also mean that we have only **8** cubes.

```
let cubes = []
let cube
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 25)
    {
        cube = new Cube(i, 0, 0)
        cubes.push(cube)
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < cubes.length; i++)
    {
        cubes[i].show()
    }
    angle += 0.5
}
```

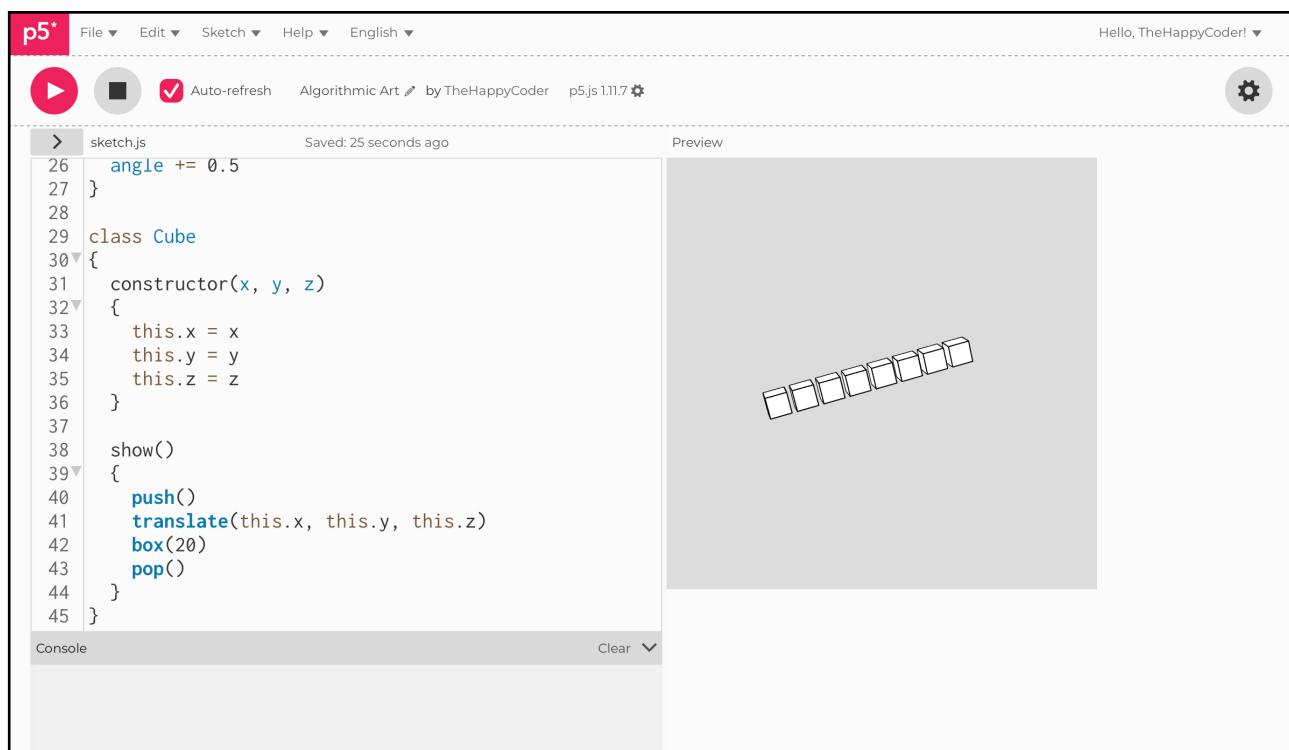
```
class Cube
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        box(20)
        pop()
    }
}
```

Notes

A bit more spaced out

Figure D8.3



The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play/pause, stop, auto-refresh (which is checked), and other options like 'Algorithmic Art' and 'Help'. The title bar says 'sketch.js' and 'Saved: 25 seconds ago'. On the right, there's a preview window showing a 3D perspective view of a series of white cubes arranged in a diagonal line.

```
> sketch.js          Saved: 25 seconds ago      Preview
26   angle += 0.5
27 }
28
29 class Cube
30{
31   constructor(x, y, z)
32{
33   this.x = x
34   this.y = y
35   this.z = z
36 }
37
38 show()
39{
40   push()
41   translate(this.x, this.y, this.z)
42   box(20)
43   pop()
44 }
45 }
```

Below the code editor is a 'Console' section which is currently empty. At the bottom right of the code editor area, there's a 'Clear' button with a dropdown arrow.



Sketch D8.4 an array in the y direction

Yet we can do even better because we can create an array in the **y** direction also. For that we need a nested loop.

```
let cubes = []
let cube
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 25)
    {
        for (let j = -100; j < 100; j += 25)
        {
            cube = new Cube(i, j, 0)
            cubes.push(cube)
        }
    }
}

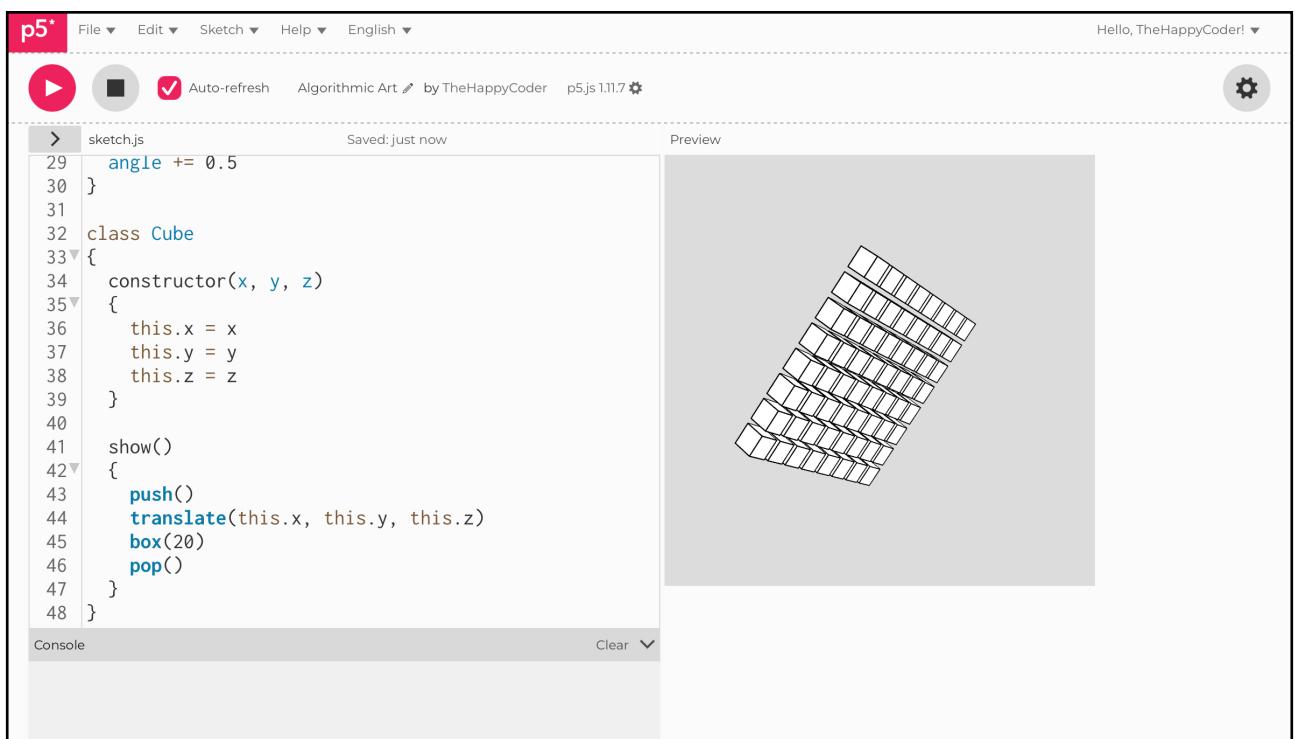
function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < cubes.length; i++)
    {
        cubes[i].show()
    }
}
```

```
    angle += 0.5
}

class Cube
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        box(20)
        pop()
    }
}
```

Figure D8.4



The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play, stop, auto-refresh (which is checked), and settings. The title bar says "Hello, TheHappyCoder! ▾". The main area has tabs for "sketch.js" and "Preview". The code editor contains the following JavaScript code:

```
sketch.js
Saved: just now
29 |   angle += 0.5
30 |
31
32 class Cube
33 {
34   constructor(x, y, z)
35   {
36     this.x = x
37     this.y = y
38     this.z = z
39   }
40
41   show()
42 {
43   push()
44   translate(this.x, this.y, this.z)
45   box(20)
46   pop()
47 }
48 }
```

The "Preview" window shows a 3D perspective view of a cube composed of multiple smaller cubes stacked along its edges, creating a segmented or wireframe-like appearance.



Sketch D8.5 a cube of cubes

But why stop there. Let's add the third dimension the **Z** direction, we just add another nested loop.

```
let cubes = []
let cube
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 25)
    {
        for (let j = -100; j < 100; j += 25)
        {
            for (let k = -100; k < 100; k += 25)
            {
                cube = new Cube(i, j, k)
                cubes.push(cube)
            }
        }
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < cubes.length; i++)
```

```

{
    cubes[i].show()
}
angle += 0.5
}

class Cube
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        box(20)
        pop()
    }
}

```

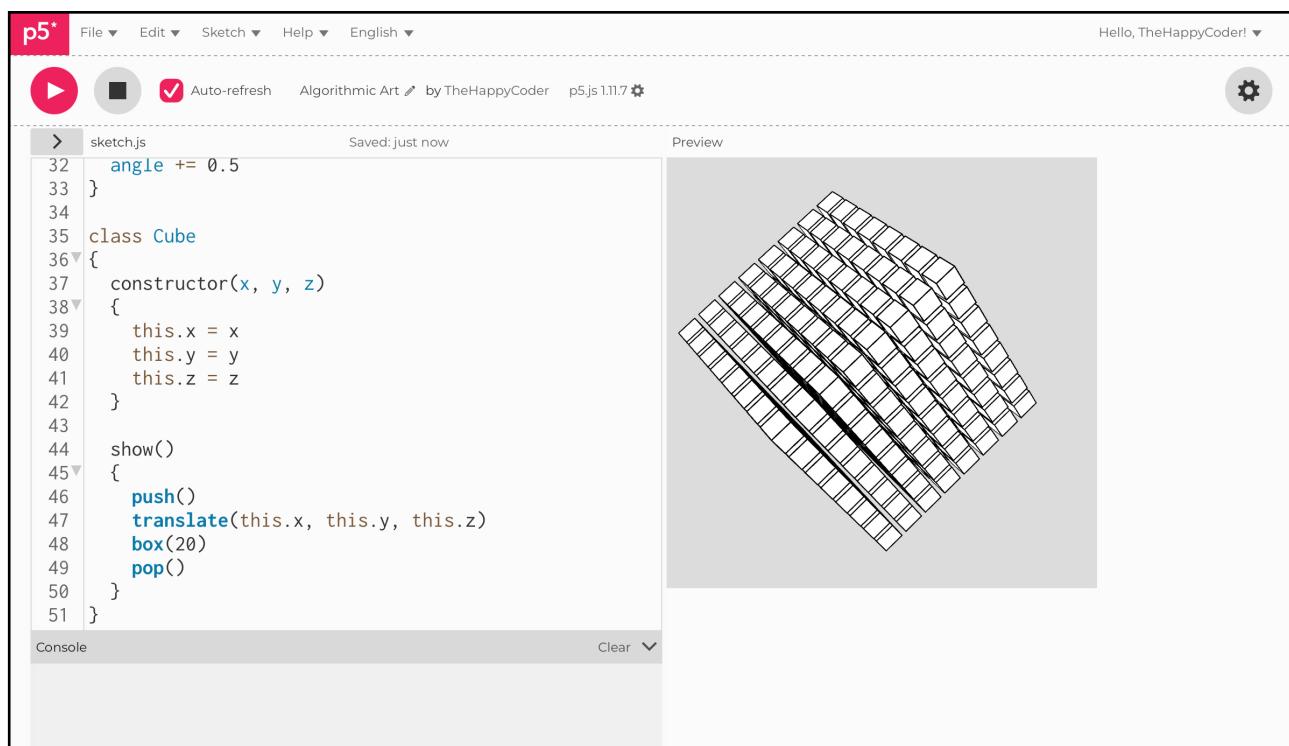
Notes

Inside the `cubes[]` array we have a 3D array

Challenge

Have a look inside the `cubes []` array with `console.log()`

Figure D8.5



The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play/pause, stop, auto-refresh (which is checked), and other options like 'Algorithmic Art' and 'English'. The title bar says 'sketch.js' and 'Saved: just now'. On the right, there's a preview window showing a 3D perspective grid of small cubes arranged in a diamond pattern. The main code area contains the following JavaScript code:

```
sketch.js
32     angle += 0.5
33 }
34
35 class Cube
36 {
37   constructor(x, y, z)
38   {
39     this.x = x
40     this.y = y
41     this.z = z
42   }
43
44   show()
45   {
46     push()
47     translate(this.x, this.y, this.z)
48     box(20)
49     pop()
50   }
51 }
```

Below the code is a 'Console' section which is currently empty.



Sketch D8.6 adding a splash of colour

We can add individual colours (random) to each cube. We use a function called `color(a, b, c)` with three arguments for the red, green and blue. We add a variable (`c`) to carry this information to the new cube being created.

```
let cubes = []
let cube
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 25)
    {
        for (let j = -100; j < 100; j += 25)
        {
            for (let k = -100; k < 100; k += 25)
            {
                let c = color(random(255), random(255), random(255))
                cube = new Cube(i, j, k, c)
                cubes.push(cube)
            }
        }
    }
}

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

```

rotateY(angle)
rotateZ(angle)
for (let i = 0; i < cubes.length; i++)
{
  cubes[i].show()
}
angle += 0.5
}

class Cube
{
  constructor(x, y, z, c)
  {
    this.x = x
    this.y = y
    this.z = z
    this.c = c
  }

  show()
  {
    push()
    fill(this.c)
    translate(this.x, this.y, this.z)
    box(20)
    pop()
  }
}

```

Notes

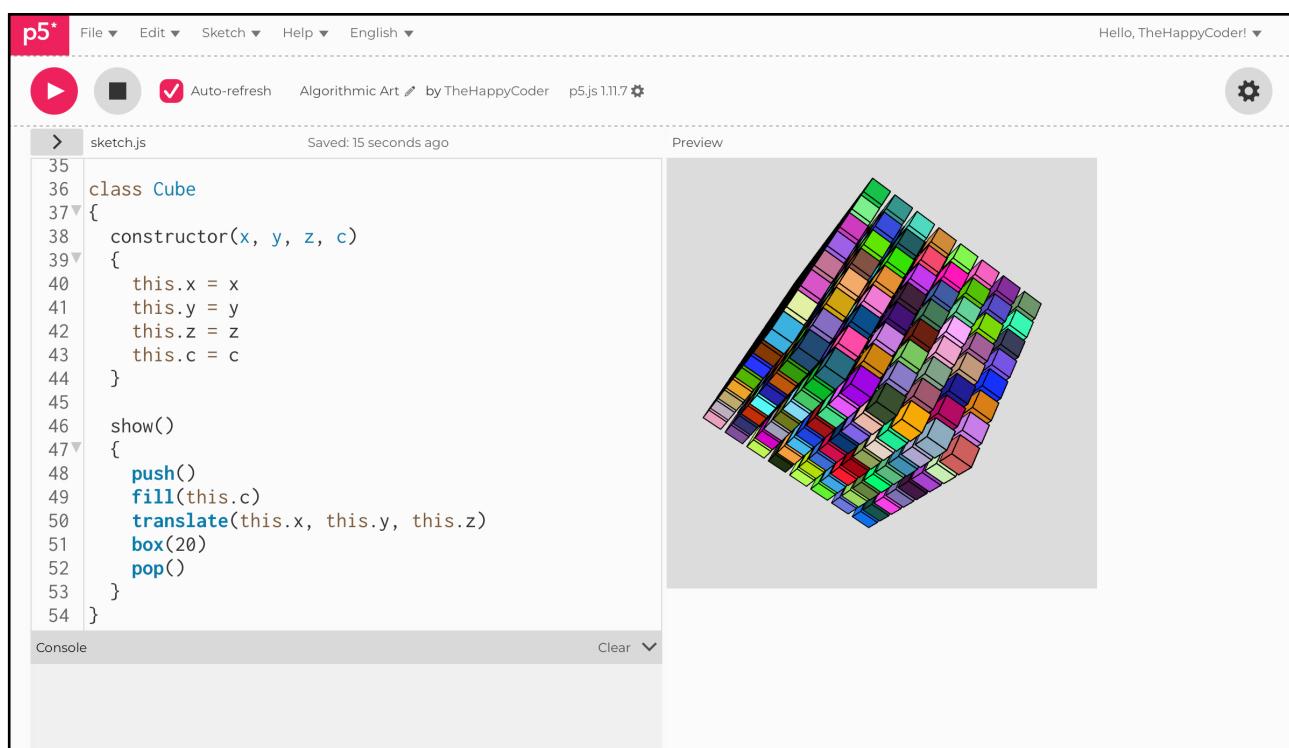
We give each individual cube its own colour



Challenge

1. Add some alpha (fourth argument) to the colour to make it transparent and remove the stroke.
2. Change the shape to a sphere or cylinder

Figure D8.6



The screenshot shows the p5.js code editor interface. At the top, there are menu items: File, Edit, Sketch, Help, and English. On the right side, it says "Hello, TheHappyCoder!" with a gear icon. The main area has tabs for "sketch.js" and "Preview". The code in "sketch.js" is as follows:

```
35
36 class Cube
37 {
38   constructor(x, y, z, c)
39   {
40     this.x = x
41     this.y = y
42     this.z = z
43     this.c = c
44   }
45
46   show()
47   {
48     push()
49     fill(this.c)
50     translate(this.x, this.y, this.z)
51     box(20)
52     pop()
53   }
54 }
```

The "Preview" window shows a 3D perspective view of a cube composed of smaller colored cubes. The colors are a gradient of various hues, creating a multi-colored effect. The cube is oriented diagonally, with its front face visible.



Sketch D8.7 a bubble factory

A bit of refactoring. We are going to use a sphere so we will change the name of the class, variable and array to something more appropriate and familiar: **bubble**. Also added **sphere(10)**, **lights()** and **noStroke()**

```
let bubbles = []
let bubble
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -100; i < 100; i += 25)
    {
        for (let j = -100; j < 100; j += 25)
        {
            for (let k = -100; k < 100; k += 25)
            {
                let c = color(random(255), random(255), random(255))
                bubble = new Bubble(i, j, k, c)
                bubbles.push(bubble)
            }
        }
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
```

```

rotateZ(angle)
for (let i = 0; i < bubbles.length; i++)
{
  bubbles[i].show()
}
angle += 0.5
}

class Bubble
{
  constructor(x, y, z, c)
  {
    this.x = x
    this.y = y
    this.z = z
    this.c = c
  }

  show()
  {
    push()
    fill(this.c)
    translate(this.x, this.y, this.z)
    lights()
    noStroke()
    sphere(10)
    pop()
  }
}

```

Notes

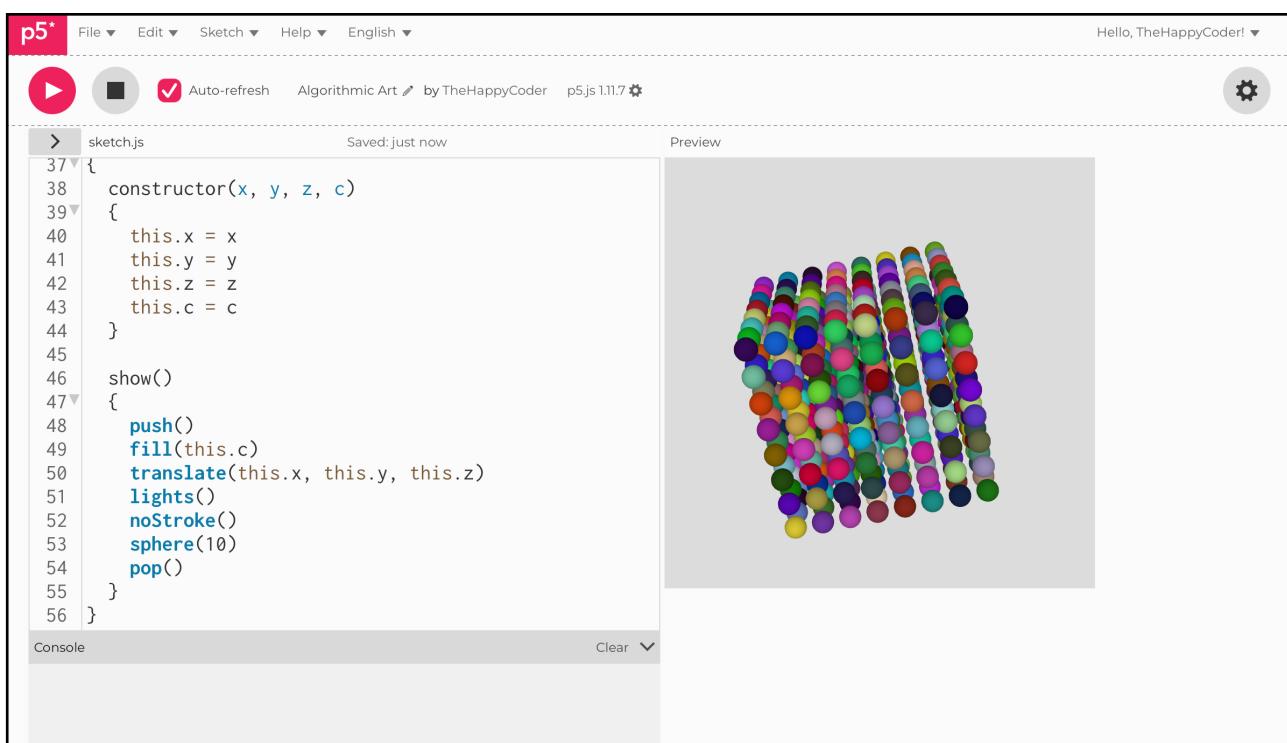
The `noStroke()` is just so that you can see the colours



Challenge

Call the variable, class and array anything you want

Figure D8.7



The screenshot shows the p5.js IDE interface. The top bar includes the p5 logo, file navigation, and a user account section. The main area has tabs for 'sketch.js' and 'Preview'. The code editor on the left contains the following JavaScript code:

```
37 > sketch.js
38 {
39     constructor(x, y, z, c)
40     {
41         this.x = x
42         this.y = y
43         this.z = z
44         this.c = c
45     }
46     show()
47     {
48         push()
49         fill(this.c)
50         translate(this.x, this.y, this.z)
51         lights()
52         noStroke()
53         sphere(10)
54         pop()
55     }
56 }
```

The preview window on the right displays a 3D structure composed of numerous small, semi-transparent spheres in various colors (blue, green, red, yellow, purple, etc.) arranged in a roughly spherical or cluster-like shape.



Sketch D8.8 not a cube

Our next step is putting all those bubbles in a circle rather than a cube, but first remove all the code highlighted and commented, then change the constructor() function

```
let bubbles = []  
let bubble  
let angle = 0  
  
function setup()  
{  
    createCanvas(400, 400, WEBGL)  
    angleMode(DEGREES)  
    // for (let i = -100; i < 100; i += 25)  
    // {  
    //     for (let j = -100; j < 100; j += 25)  
    //     {  
    //         for (let k = -100; k < 100; k += 25)  
    //         {  
    //             let c = color(random(255), random(255), random(255))  
    //             bubble = new Bubble(i, j, k, c)  
    //             bubbles.push(bubble)  
    //         }  
    //     }  
    // }  
}  
  
function draw()  
{  
    background(220)  
    rotateX(angle)  
    rotateY(angle)
```

```

rotateZ(angle)
for (let i = 0; i < bubbles.length; i++)
{
    bubbles[i].show()
}
angle += 0.5
}

class Bubble
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
        // this.c = c
    }

    show()
    {
        push()
        // fill(this.c)
        translate(this.x, this.y, this.z)
        lights()
        noStroke()
        sphere(10)
        pop()
    }
}

```

***** Notes

Don't try running this just yet



Sketch D8.9 now a circle

Creating a circle of bubbles

```
let bubbles = []
let bubble
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = 0; i < 360; i += 30)
    {
        let x = 100 * sin(i)
        let y = 100 * cos(i)
        bubble = new Bubble(x, y, 0)
        bubbles.push(bubble)
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    angle += 0.5
}
```

```
class Bubble
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        lights()
        fill('yellow')
        noStroke()
        sphere(10)
        pop()
    }
}
```

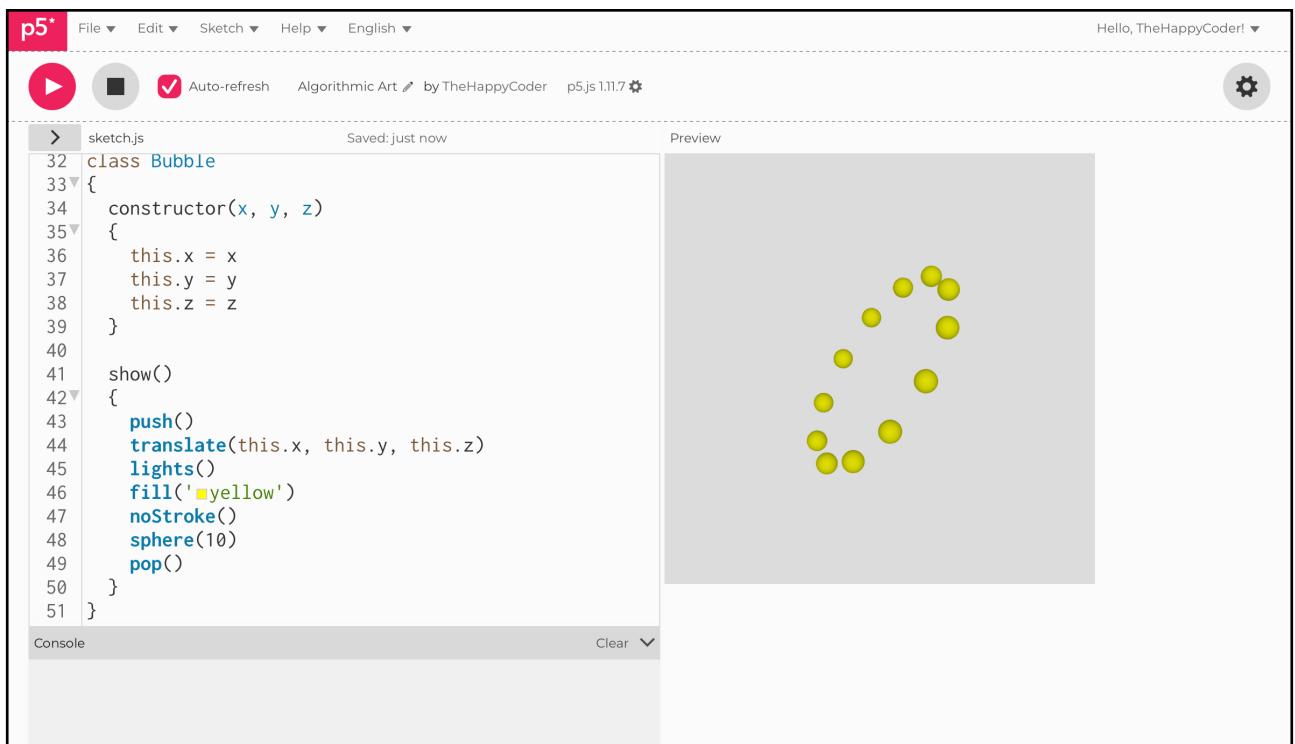
Notes

We have a nice ring of spheres

Challenge

Add other lights or materials

Figure D8.9



The screenshot shows the p5.js IDE interface. The top bar includes the p5 logo, file navigation, and user information "Hello, TheHappyCoder! ▾". The code editor on the left contains the following JavaScript code:

```
sketch.js
32 class Bubble
33 {
34   constructor(x, y, z)
35   {
36     this.x = x
37     this.y = y
38     this.z = z
39   }
40
41   show()
42   {
43     push()
44     translate(this.x, this.y, this.z)
45     lights()
46     fill('yellow')
47     noStroke()
48     sphere(10)
49     pop()
50   }
51 }
```

The preview window on the right displays several yellow spheres scattered in a 3D space.



Sketch D8.10 creating a spiral

Next let's try to make a spiral so that the array of bubbles also has a **Z** component.

```
let bubbles = []
let bubble
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = 0; i < 360; i += 30)
    {
        let x = 100 * sin(i)
        let y = 100 * cos(i)
        bubble = new Bubble(x, y, i)
        bubbles.push(bubble)
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    angle += 0.5
}
```

```
}
```

```
class Bubble
```

```
{
```

```
    constructor(x, y, z)
```

```
    {
```

```
        this.x = x
```

```
        this.y = y
```

```
        this.z = z
```

```
    }
```

```
    show()
```

```
{
```

```
    push()
```

```
    translate(this.x, this.y, this.z)
```

```
    lights()
```

```
    fill('yellow')
```

```
    noStroke()
```

```
    sphere(10)
```

```
    pop()
```

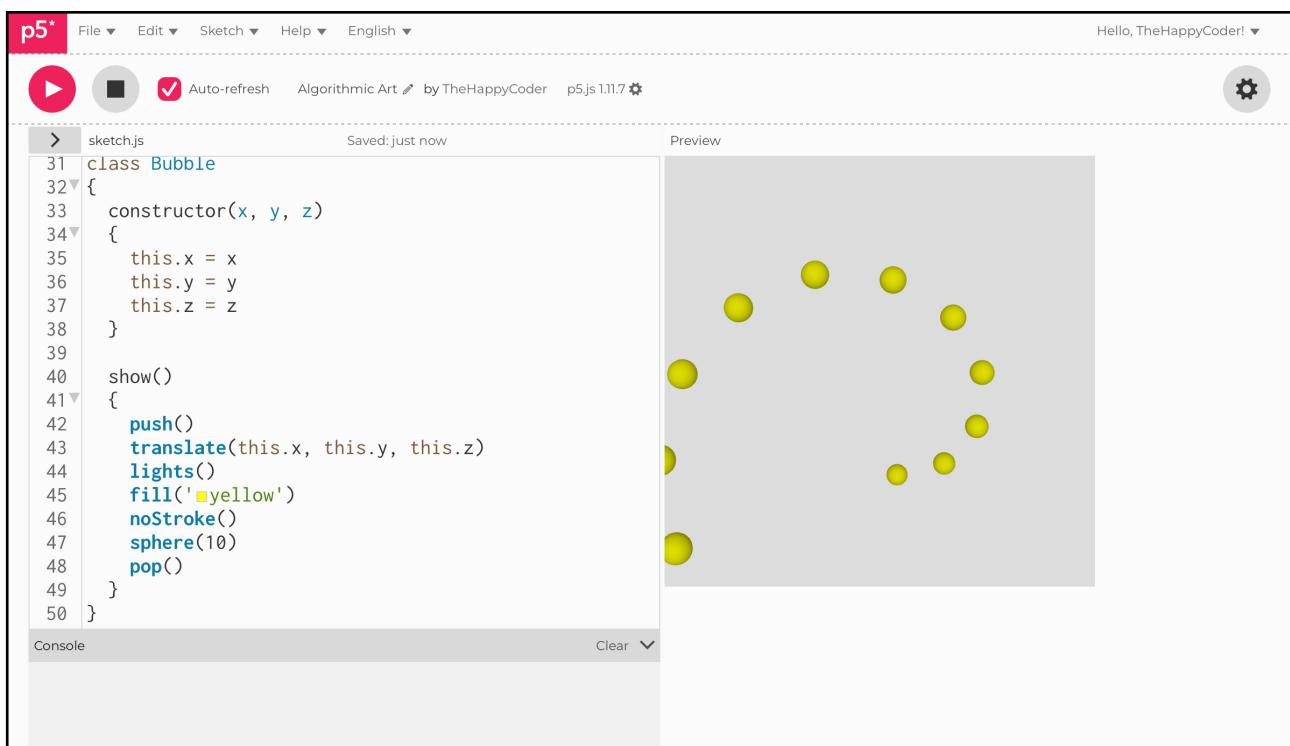
```
}
```

```
}
```

Notes

It's a spiral Jim but not as we know it. Let's see if we can improve

Figure D8.10



The screenshot shows the p5.js code editor interface. At the top, there are buttons for File, Edit, Sketch, Help, and English, along with a status bar that says "Hello, TheHappyCoder! ▾". The main area has tabs for sketch.js and Preview. The code in sketch.js is as follows:

```
sketch.js
Saved: just now
Preview
31 <script>
32 <script>Auto-refresh Algorithmic Art by TheHappyCoder p5.js 1.11.7
33 <script>
34 <script>class Bubble
35 <script>{
36 <script>    constructor(x, y, z)
37 <script>    {
38 <script>        this.x = x
39 <script>        this.y = y
40 <script>        this.z = z
41 <script>    }
42 <script>    show()
43 <script>    {
44 <script>        push()
45 <script>        translate(this.x, this.y, this.z)
46 <script>        lights()
47 <script>        fill('yellow')
48 <script>        noStroke()
49 <script>        sphere(10)
50 <script>    }
</script>
```

The Preview window shows several yellow spheres of varying sizes scattered across a light gray background.



Sketch D8.11 more spirally

We can make some improvements to this to make it more spirally.

```
let bubbles = []
let bubble
let angle = 0

function setup()
{
    createCanvas(400, 400, WEBGL)
    angleMode(DEGREES)
    for (let i = -360; i < 360; i += 10)
    {
        let x = 100 * sin(i)
        let y = 100 * cos(i)
        bubble = new Bubble(x, y, i/4)
        bubbles.push(bubble)
    }
}

function draw()
{
    background(220)
    rotateX(angle)
    rotateY(angle)
    rotateZ(angle)
    for (let i = 0; i < bubbles.length; i++)
    {
        bubbles[i].show()
    }
    angle += 0.5
}
```

```

class Bubble
{
    constructor(x, y, z)
    {
        this.x = x
        this.y = y
        this.z = z
    }

    show()
    {
        push()
        translate(this.x, this.y, this.z)
        lights()
        fill('yellow')
        noStroke()
        sphere(10)
        pop()
    }
}

```

Notes

Much better

Challenges

1. Can you randomise the diameter of the sphere
2. Add random colours to the spheres
3. Change the radius of the spiral so that it starts small and increases
4. How would extend it
5. How would you rotate it about an axis

Figure D8.11

The screenshot shows the p5.js IDE interface. At the top, there's a toolbar with icons for play, stop, and refresh, followed by menu items: File, Edit, Sketch, Help, and English. To the right, it says "Hello, TheHappyCoder!" with a dropdown arrow. Below the toolbar, the code editor has a file named "sketch.js" with the following content:

```
sketch.js
31 class Bubble
32 {
33   constructor(x, y, z)
34   {
35     this.x = x
36     this.y = y
37     this.z = z
38   }
39
40   show()
41   {
42     push()
43     translate(this.x, this.y, this.z)
44     lights()
45     fill('yellow')
46     noStroke()
47     sphere(10)
48     pop()
49   }
50 }
```

To the right of the code editor is a preview window titled "Preview" which displays a 3D rendering of a yellow wireframe heart shape composed of spheres.