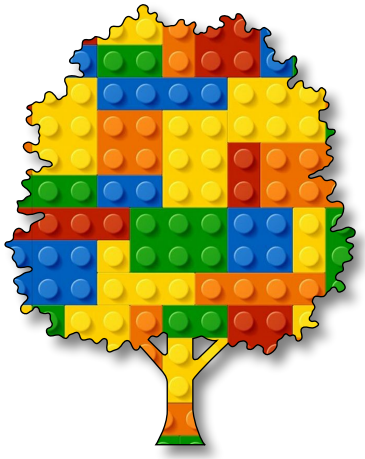
The Joy of Coding
Module D
Unit #7
3D arrays
part 1



Content

Module D Unit #7 3D arrays part 1

Sketch D7.1	new sketch
Sketch D7.2	mouse pressed cubes
Sketch D7.3	three arguments
Sketch D7.4	drawing the cube
Sketch D7.5	translation
Sketch D7.6	adding an angle
Sketch D7.7	angle to the object
Sketch D7.8	rotating in all directions
Sketch D7.9	incrementing the angle
Sketch D7.10	another newish sketch
Sketch D7.11	rotating the array
Sketch D7.12	spaced out
Sketch D7.13	an array in the y direction
Sketch D7.14	an array in the z direction
Sketch D7.15	adding a splash of colour

Introduction to 3D arrays part 1

In this unit we are going to create a cube class and introduce 3D arrays. We will have a load of spinning cubes created every time you click on the canvas.

Sketch D7.1 new sketch

❗ new sketch for this unit

We will call them cubes (rather than boxes) and because they are objects we will create a class called Cube with a `constructor()`, `move()` and `show()` function.

```
function setup()
{
  createCanvas(400, 400, WEBGL)
}

function draw()
{
  background(220)
}

class Cube
{
  constructor()
  {

  }

  move()
  {

  }

  show()
  {

  }
```

```
}
```

🗒️ Notes

This is our basic class sketch

Sketch D7.2 mouse pressed cubes

We want to click on the canvas and put a cube there. So we need a `mousePressed()` function, an array called cubes and a variable called cube (which is one box).

```
let cubes = []
let cube

function setup()
{
  createCanvas(400, 400, WEBGL)
}

function mousePressed()
{

}

function draw()
{
  background(220)
}

class Cube
{
  constructor()
  {

  }

  move()
  {
```

```
  }

  show()
  {

  }
}
```

🗒️ Notes

As you may be aware we won't see anything yet

🌻 Challenge

Can you guess how we might create a cube where we click on the canvas

Sketch D7.3 three arguments

We are going to create a new cube at the place where we point the mouse and click. The cube will have three arguments. The `x` position (`mouseX`), the `y` position (`mouseY`) and the `z` position (`0`). So in the constructor we will have `x`, `y`, `z` and when we click on the canvas we pass on those values into the array as a new object. When we create a new cube we push it into the array (`cubes[]`)

```
let cubes = []
let cube

function setup()
{
  createCanvas(400, 400, WEBGL)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 0)
  cubes.push(cube)
}

function draw()
{
  background(220)
}

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

```
    this.x = z
  }

  move()
  {

  }

  show()
  {

  }
}
```

Notes

We create a new cube and add it to the array (nothing to see though)

Sketch D7.4 drawing the cube

Nothing will happen until we draw the `cube` so in the `show()` function we will first have to translate to where we click on the canvas and then draw the box but we need to loop through the array in `draw()`. You will notice something odd when you click on the canvas, the `cube` isn't where it should be. We will rectify this.

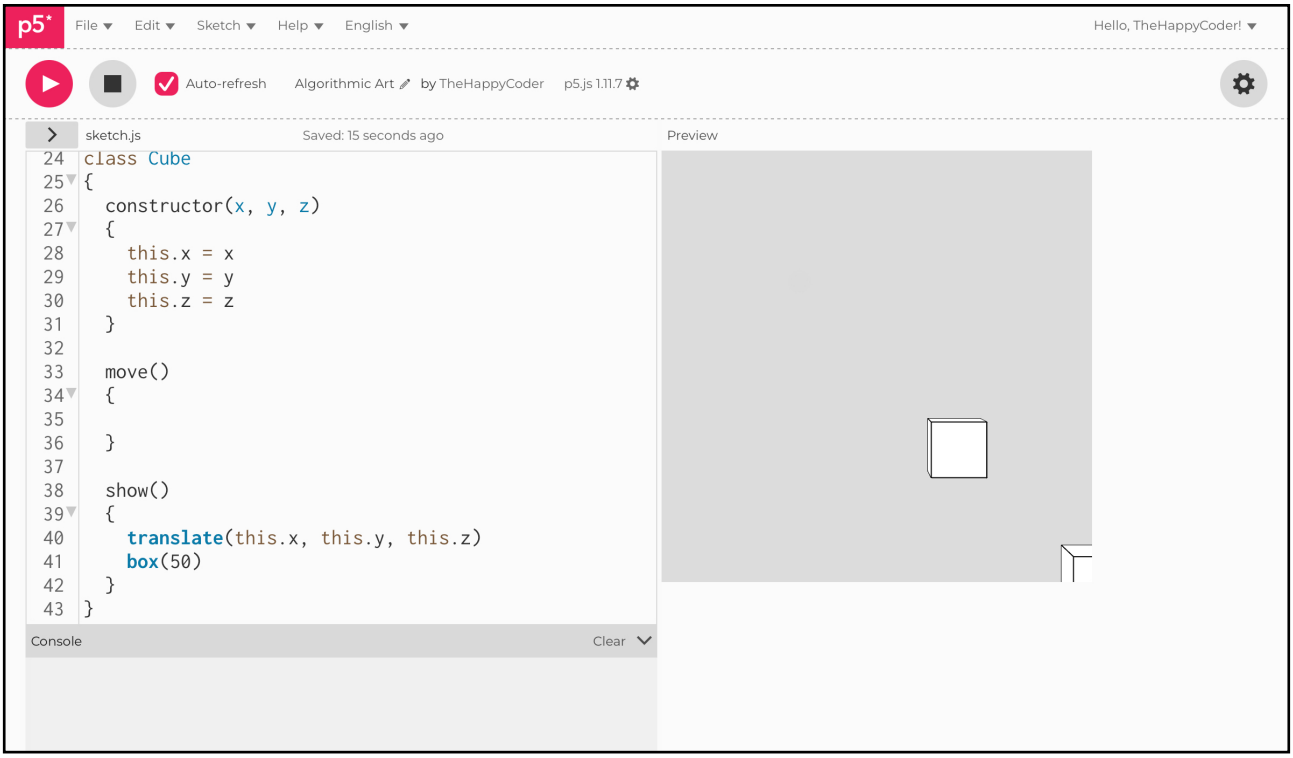
```
let cubes = []
let cube

function setup()
{
  createCanvas(400, 400, WEBGL)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 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
  }

  move()
  {

  }

  show()
  {
    translate(this.x, this.y, this.z)
    box(50)
  }
}
```

🗒️ Notes

We can call the length of the cubes array as it grows

Figure D7.4


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.7
sketch.js
Saved: 15 seconds ago
Preview
class Cube
{
constructor(x, y, z)
{
this.x = x
this.y = y
this.z = z
}
move()
{
}
show()
{
translate(this.x, this.y, this.z)
box(50)
}
}
Console
Clear

Sketch D7.5 translation

Let's rectify this because of the origin is at the centre of the canvas rather than the top left we have to take that into account. Also when we translate we need to only translate for that particular cube otherwise we start to add the translation and things disappear very quickly so we use push() and pop() try without to see the difference

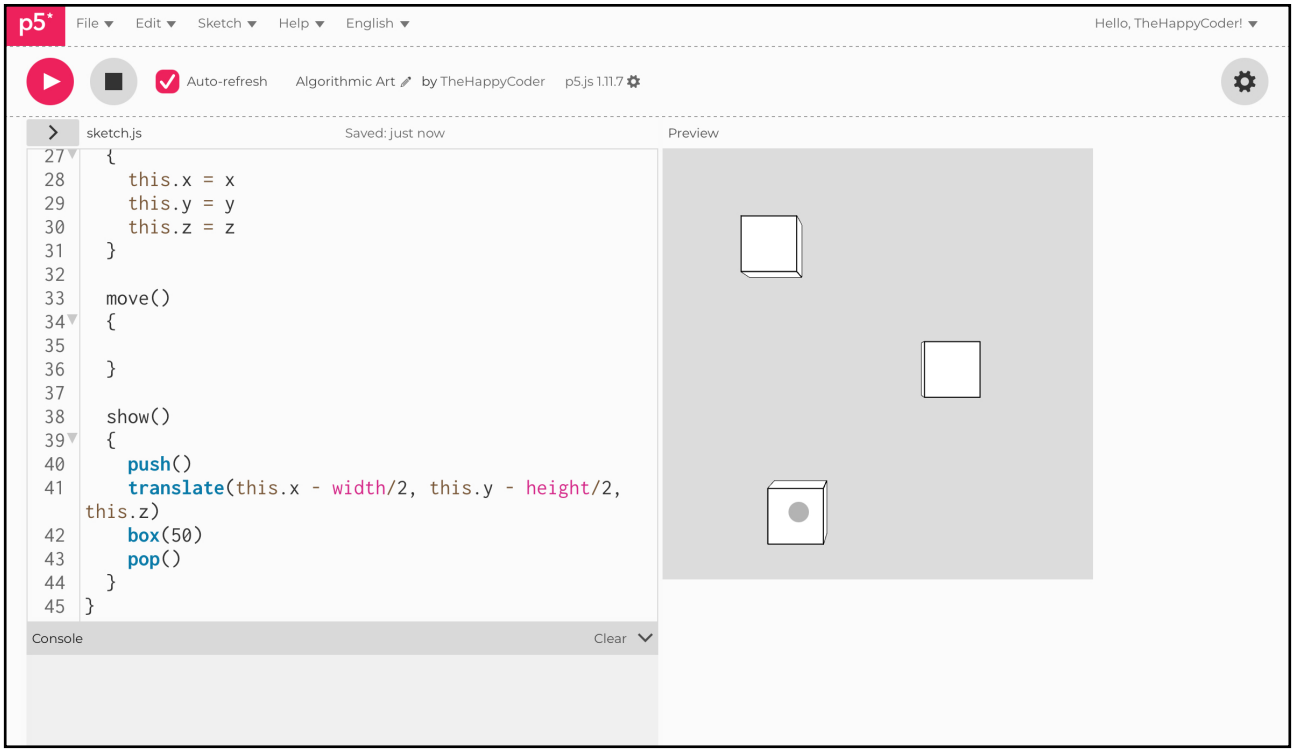
```
let cubes = []
let cube

function setup()
{
  createCanvas(400, 400, WEBGL)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 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
  }

  move()
  {

  }

  show()
  {
    push()
    translate(this.x - width/2, this.y - height/2, this.z)
    box(50)
    pop()
  }
}
```

🗒️ Notes

That's better! Wherever you click on the canvas you should see the cube appear at that position. Next we will look at rotating each one individually.

Figure D7.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.7
sketch.js
Saved: just now
Preview
{
this.x = x
this.y = y
this.z = z
}
move()
{
}
show()
{
push()
translate(this.x - width/2, this.y - height/2, this.z)
box(50)
pop()
}
}
Console
Clear

Sketch D7.6 adding an angle

Using the same sketch we add in a new `angle` variable, initialise it to `0` and change the angle mode from radians to degrees.

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

function setup()
{
  createCanvas(400, 400, WEBGL)
  angleMode(DEGREES)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 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
  }

  move()
  {

  }

  show()
  {
    push()
    translate(this.x - width/2, this.y - height/2, this.z)
    box(50)
    pop()
  }
}
```

🗒️ Notes

No appreciable change yet

Sketch D7.7 angle to the object

Next we need to add angle to the cube object.

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

function setup()
{
  createCanvas(400, 400, WEBGL)
  angleMode(DEGREES)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 0, angle)
  cubes.push(cube)
}

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

class Cube
{
  constructor(x, y, z, angle)
  {
```

```
    this.x = x
    this.y = y
    this.z = z
    this.angle = angle
  }

  move()
  {

  }

  show()
  {
    push()
    translate(this.x - width/2, this.y - height/2, this.z)
    box(50)
    pop()
  }
}
```

🗒️ Notes

Still nowt new to see yet!

Sketch D7.8 rotating in all directions

We want to rotate along the x, y and z axis

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

function setup()
{
  createCanvas(400, 400, WEBGL)
  angleMode(DEGREES)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 0, angle)
  cubes.push(cube)
}

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

class Cube
{
  constructor(x, y, z, angle)
  {
```

```
    this.x = x
    this.y = y
    this.z = z
    this.angle = angle
  }

  move()
  {

  }

  show()
  {
    push()
    translate(this.x - width/2, this.y - height/2, this.z)
    rotateX(this.angle)
    rotateY(this.angle)
    rotateZ(this.angle)
    box(50)
    pop()
  }
}
```

🗒️ Notes

Getting closer

🌻 Challenge

what do you think we need to do to get the cubes rotating?

Sketch D7.9 incrementing the angle

Now we need to increment the `angle` by one degree in the `move()` function and add it into `draw()`. You should have nice rotating boxes or cubes.

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

function setup()
{
  createCanvas(400, 400, WEBGL)
  angleMode(DEGREES)
}

function mousePressed()
{
  cube = new Cube(mouseX, mouseY, 0, angle)
  cubes.push(cube)
}

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

class Cube
```

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

  move()
  {
    this.angle++
  }

  show()
  {
    push()
    translate(this.x - width/2, this.y - height/2, this.z)
    rotateX(this.angle)
    rotateY(this.angle)
    rotateZ(this.angle)
    box(50)
    pop()
  }
}
```

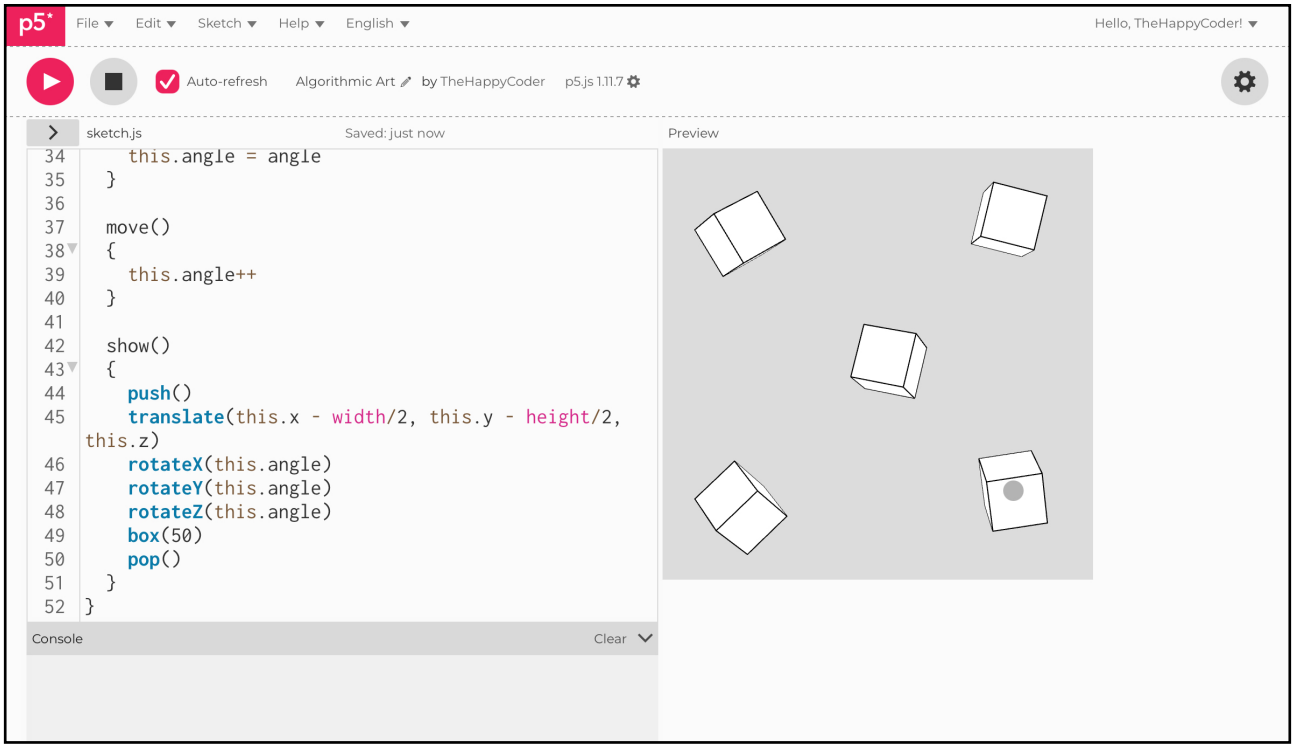
🗒️ Notes

Now we have it, each one rotates individually

🌻 Challenges

1. Add materials, lights and their own colour
2. Create different size cubes

Figure D7.9


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.7
sketch.js
Saved: just now
Preview
34 this.angle = angle
35 }
36
37 move()
38 {
39 this.angle++
40 }
41
42 show()
43 {
44 push()
45 translate(this.x - width/2, this.y - height/2, this.z)
46 rotateX(this.angle)
47 rotateY(this.angle)
48 rotateZ(this.angle)
49 box(50)
50 pop()
51 }
52 }
Console
Clear