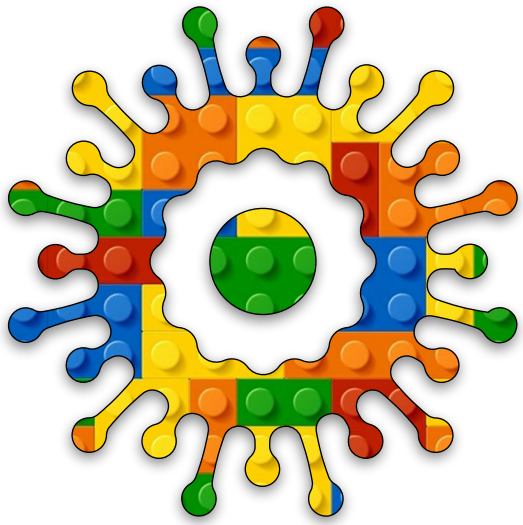
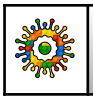
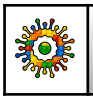
Creative Coding Module C Unit #4 lights and materials part 2





Content

Module C Unit #4 lights and materials part 2



Introduction to lights and materials part 2

Quick recap:

Since we're working in 3D, we can position lights as separate entities rather than simply filling a shape with a colour (which is also possible). With just a little practice, we can create many pleasing results. There are many light options and materials to choose from, and by combining the right ones, you can create some pleasing effects that don't always work as you might expect.

- normalMaterial()
- ambientMaterial()
- specularMaterial()
- emissiveMaterial()
- ambientLight()
- pointLight()
- directionalLight()
- lights()

Sketch C4.1 new sketch new lights

❗ start a new sketch

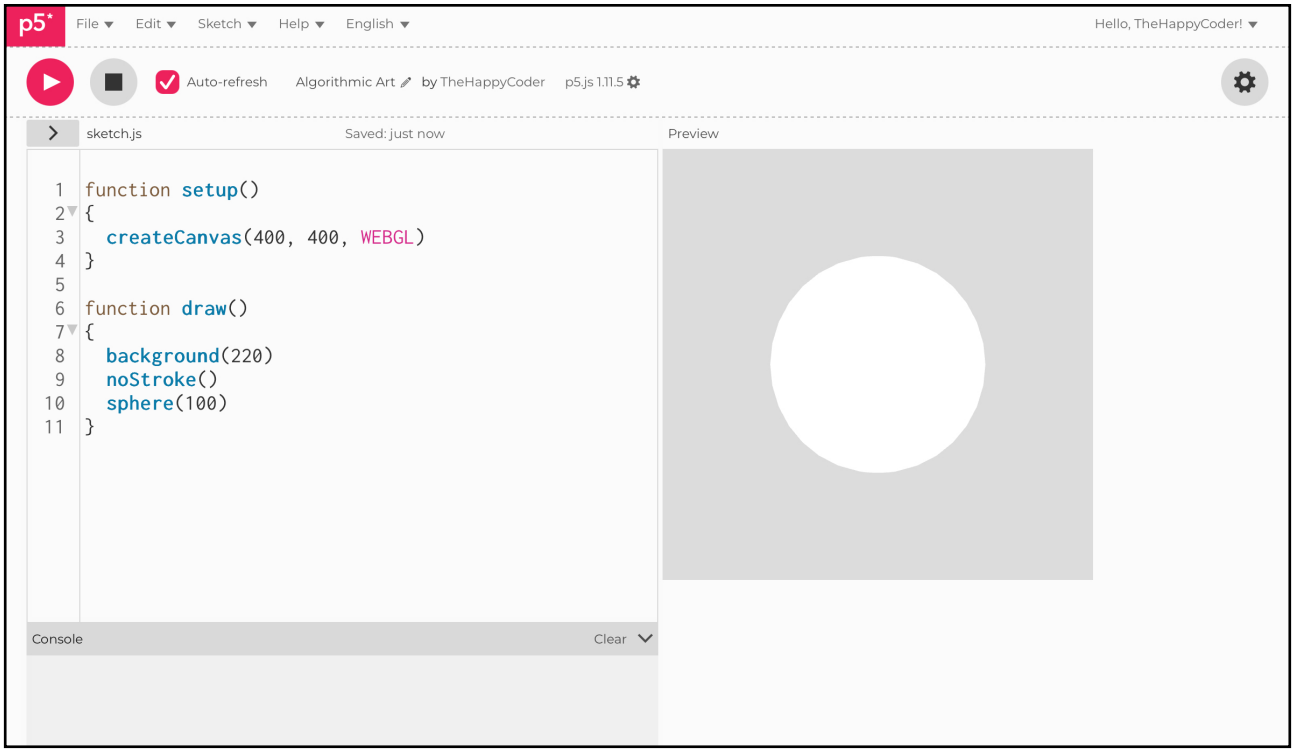
We are going to introduce directional light. It is similar to the point light because it has six arguments: the first three are the RGB colours and the next three are the vectors which tell you the direction but not the position. They have one of three values: 1, 0, and −1. Let's start with a basic sketch again and demonstrate it. We get a white sphere with no detail.

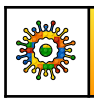
```
function setup()
{
  createCanvas(400, 400, WEBGL)
}

function draw()
{
  background(220)
  noStroke()
  sphere(100)
}
```

🗒️ Notes

Just a white, indistinct sphere.

Figure C4.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
}
function draw()
{
background(220)
noStroke()
sphere(100)
}
Console
Clear



Sketch C4.2 a more directional light

Now, add the directional light (a nice yellow). The `directionalLight()` has six arguments.

The first three are the RGB (depending on the `colourMode()`), the next three require a little attention but are quite logical. They are:
The x direction with values between −1 and +1
The y direction with values between −1 and +1
The z direction with values between −1 and +1

As an example for the x direction, a value of +1 shines from the left-hand side, whereas −1 the light shines from the right-hand side (see fig. C4.2). Then, the y direction +1 is from above, whereas −1 is from below; for the z value +1 is from behind, −1 is from in front. A value of zero seems to switch them off altogether.

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

function draw()
{
  background(220)
  directionalLight(255, 255, 0, -1, 0, 0)
  noStroke()
  sphere(100)
}
```

Notes

The light in this instance is coming from the right (x is −1). It is best to play with these values to get a feel for them.

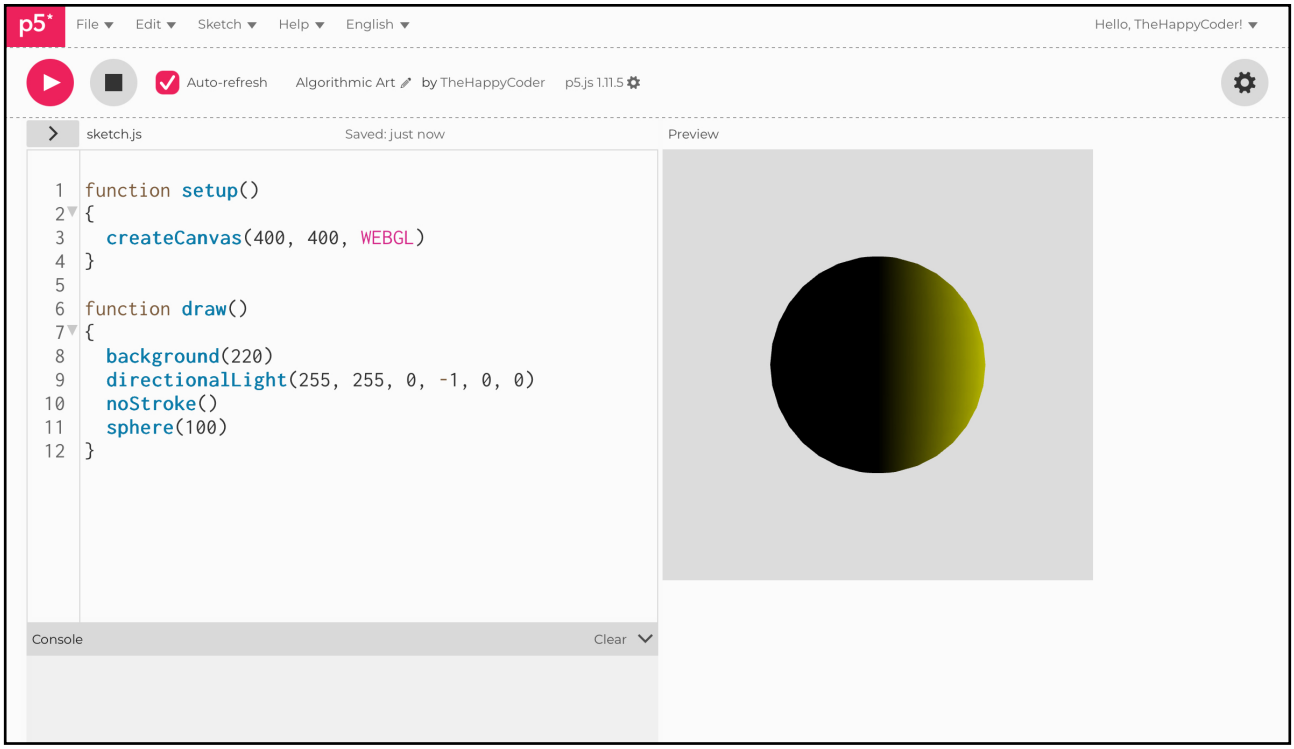
🌻 Challenges

1. Change the values from 1 to 0 to −1.
2. Change the strength of the colour.
3. Add more directional lights from different directions (maximum number is five).
4. Add more directional lights from the same direction.
5. Use different colours on the same position.

🛠️ Code Explanation

`directionalLight(255, 255, 0, -1, 0, 0)`	A yellow directional light pointing from the right

Figure C4.2


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
}
function draw()
{
background(220)
directionalLight(255, 255, 0, -1, 0, 0)
noStroke()
sphere(100)
}
Console
Clear

Sketch C4.3 just lights!

❗ removing the sphere replacing with a cube and adding some rotation and shiny material.
One final option is a function called `lights()` which is a mixture of ambient and directional light. It's a quick throw-in so that you don't have to think about direction or position.

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

function draw()
{
  background(220)
  lights()
  specularMaterial(50)
  rotateX(frameCount/50)
  rotateY(frameCount/50)
  rotateZ(frameCount/50)
  noStroke()
  box(100)
}
```

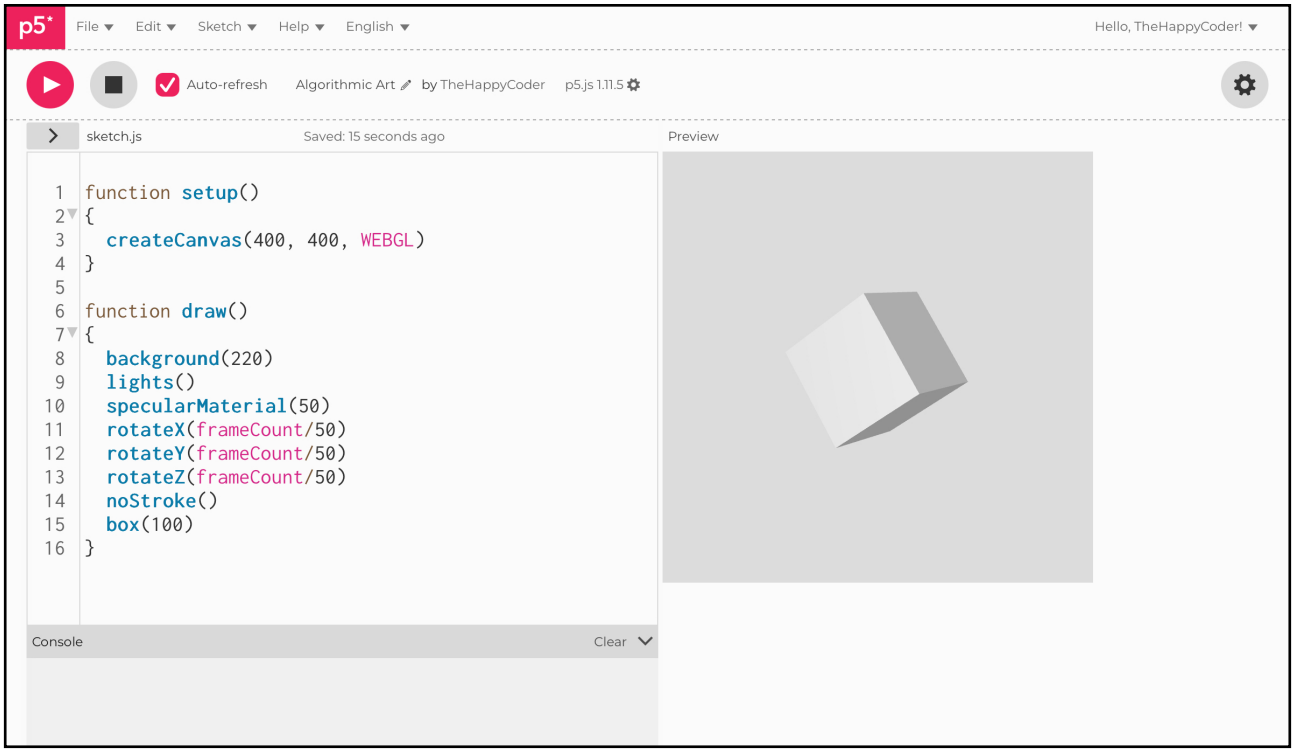
🗒️ Notes

This shows how you can quickly show your 3D graphics at work.

🛠️ Code Explanation

`lights()`	Has ambient and directional light built in

Figure C4.3


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 15 seconds ago
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
}
function draw()
{
background(220)
lights()
specularMaterial(50)
rotateX(frameCount/50)
rotateY(frameCount/50)
rotateZ(frameCount/50)
noStroke()
box(100)
}
Console
Clear

Sketch C4.4 basic torus

Here we have a torus with the default level of detail (24, 16).

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

function draw()
{
  background(220)
  lights()
  specularMaterial(50)
  rotateX(frameCount/50)
  rotateY(frameCount/50)
  rotateZ(frameCount/50)
  noStroke()
  torus(100, 25)
}
```

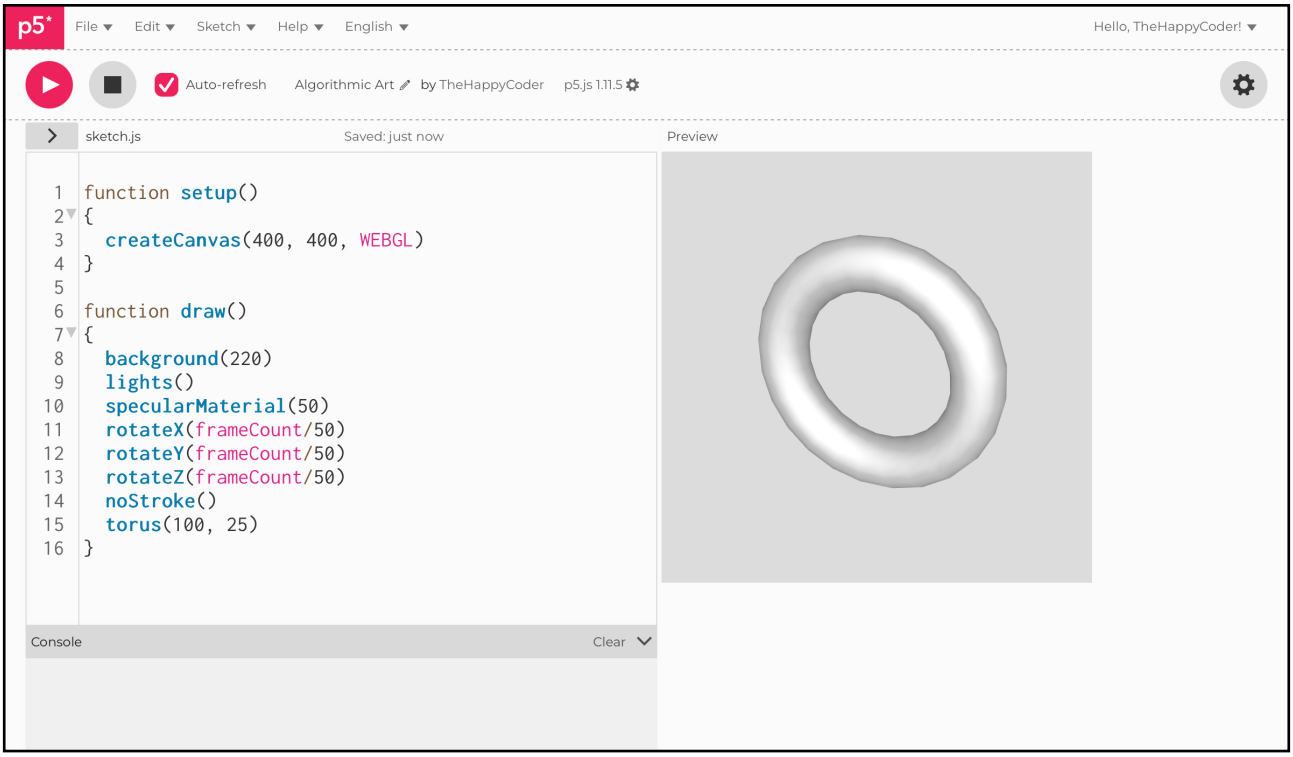
🗒️ Notes

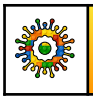
We have our standard torus.

🌻 Challenge

Make the background darker to highlight the effect.

Figure C4.4


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
}
function draw()
{
background(220)
lights()
specularMaterial(50)
rotateX(frameCount/50)
rotateY(frameCount/50)
rotateZ(frameCount/50)
noStroke()
torus(100, 25)
}
Console
Clear



Sketch C4.5 more detail added

Increasing the amount of detail to see the difference compared to the default.

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

function draw()
{
  background(220)
  lights()
  specularMaterial(50)
  rotateX(frameCount/50)
  rotateY(frameCount/50)
  rotateZ(frameCount/50)
  noStroke()
  torus(100, 25, 48, 32)
}
```

🗒️ Notes

It makes it much smoother

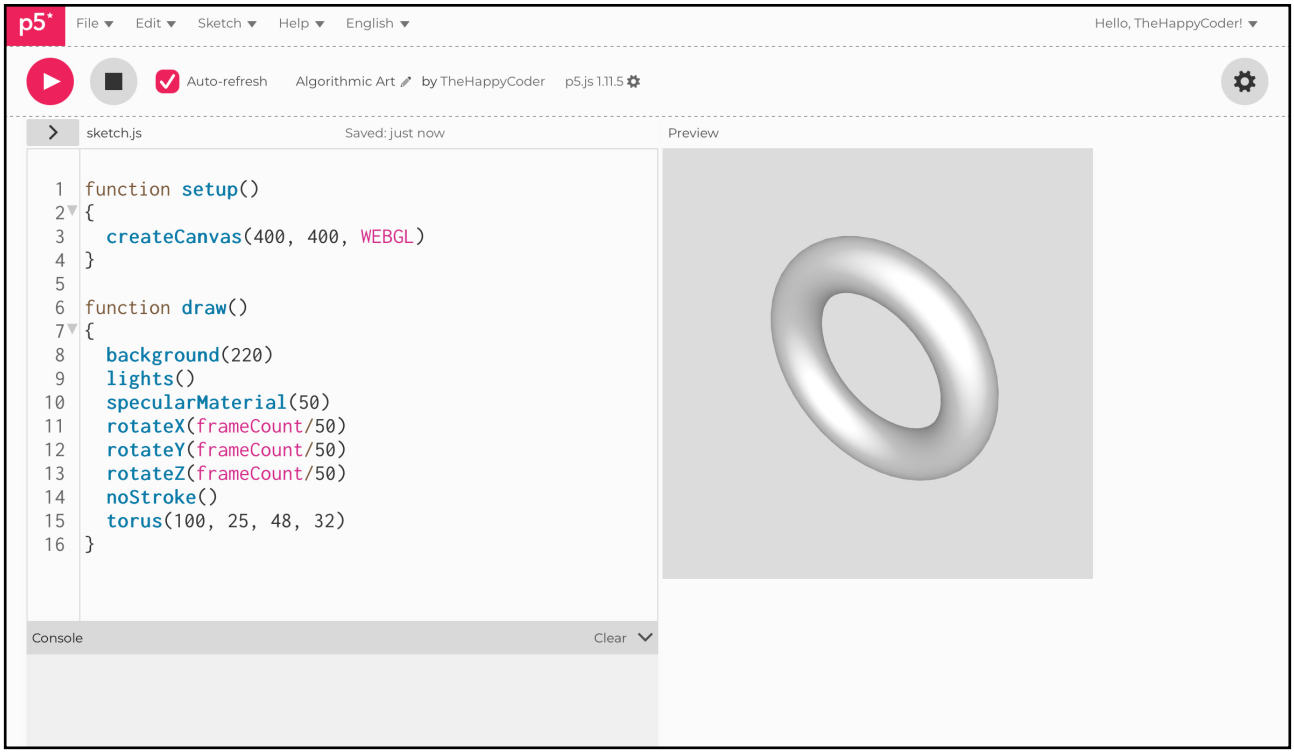
🌻 Challenges

1. Try higher values
2. Try it with other shapes

🛠️ Code Explanation

`torus(100, 25, 48, 32)`	A torus shape with double the detail

Figure C4.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
}
function draw()
{
background(220)
lights()
specularMaterial(50)
rotateX(frameCount/50)
rotateY(frameCount/50)
rotateZ(frameCount/50)
noStroke()
torus(100, 25, 48, 32)
}
Console
Clear

Sketch C4.6 orbiting object

❗ new sketch

We have a large sphere in the centre and a smaller sphere orbiting it. The `translate()` function scribes a circular motion, think circle equation with `sin()` and `cos()`. The radius of the orbit is `120`.

```
function setup()
{
  createCanvas(400, 400, WEBGL)
  strokeWeight(0.5)
}


function draw()
{
  background(220)
  ambientLight(255)
  push()
  translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount
/ 30))
  sphere(10)
  pop()
  sphere(50)
}
```

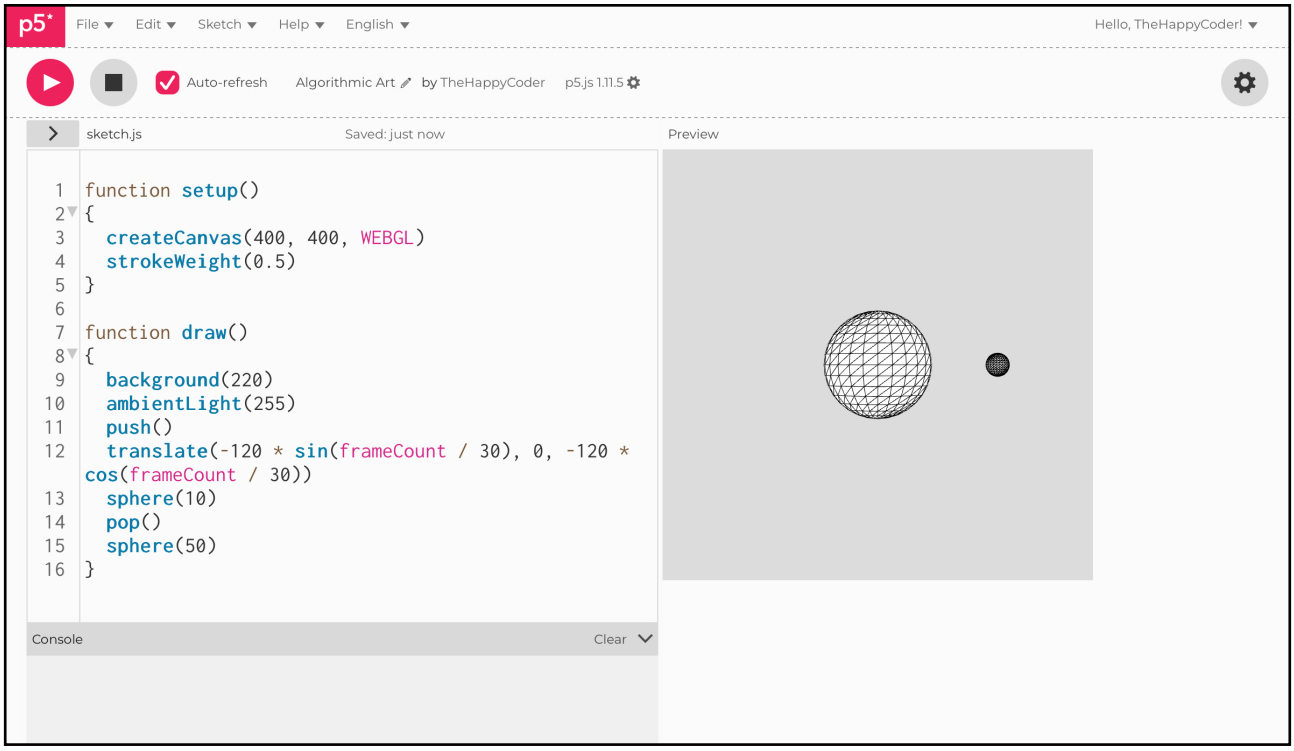
🗒️ Notes

The rotation is on the `x` axis and the `z` axis, if it were on the `x` and `y` axis it would make more sense. We `push()` and `pop()` it so that the large sphere doesn't start orbiting as well.

🛠️ Code Explanation

`translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount / 30))`	This is the motion of the small sphere, it describes a circle motion

Figure C4.6


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
strokeWeight(0.5)
}
function draw()
{
background(220)
ambientLight(255)
push()
translate(-120 * sin(frameCount / 30), 0, -120 *
cos(frameCount / 30))
sphere(10)
pop()
sphere(50)
}
Console
Clear

Sketch C4.7 rotating directional light

We now add in the directional light, which mirrors the movement of the small sphere. We reduce the ambient light and give it a black background.

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


function draw()
{
  background(0)
  ambientLight(100)
  directionalLight(200, 200, 0, sin(frameCount / 30), 0,
cos(frameCount / 30))
  push()
  translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount
/ 30))
  sphere(10)
  pop()
  sphere(50)
}
```

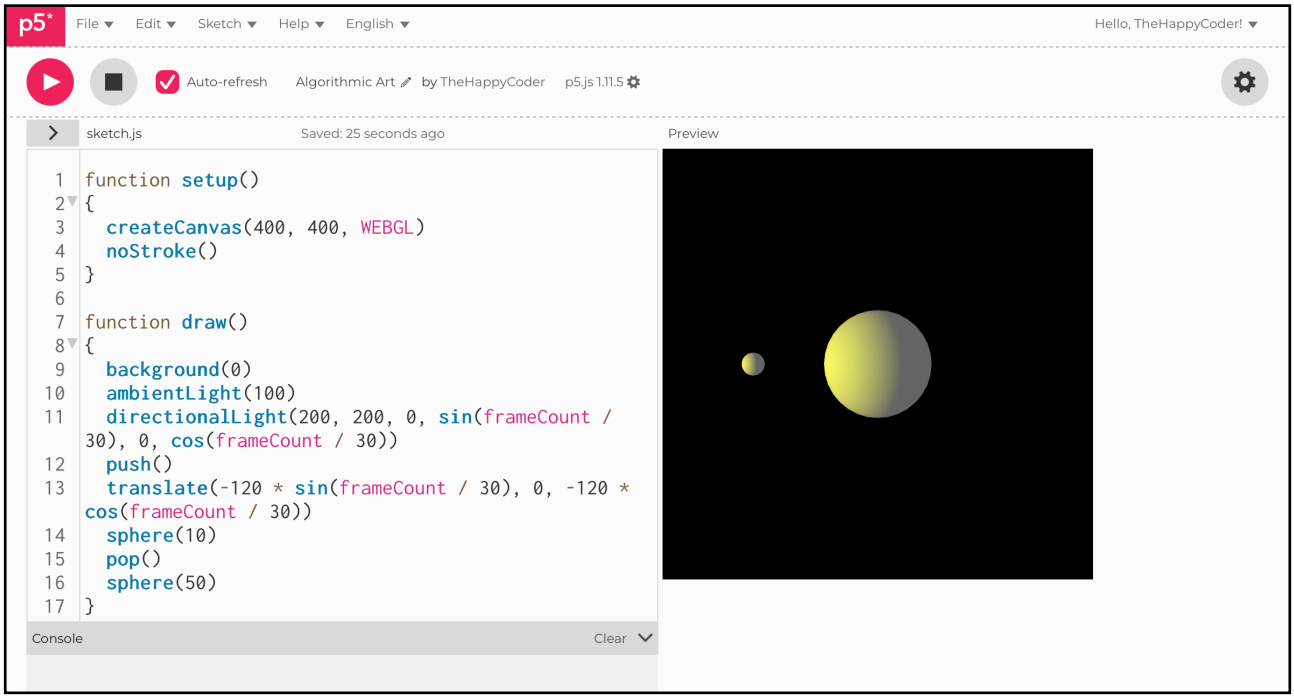
Notes

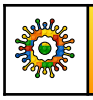
The vectors (the last three arguments) change direction according to the position of the sphere; they still return values between −1 and +1.

Code Explanation

`directionalLight(200, 200, 0, sin(frameCount / 30), 0, cos(frameCount / 30))`	The directional light changes direction as it navigates the larger sphere

Figure C4.7


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 25 seconds ago
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
noStroke()
}
function draw()
{
background(0)
ambientLight(100)
directionalLight(200, 200, 0, sin(frameCount / 30), 0, cos(frameCount / 30))
push()
translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount / 30))
sphere(10)
pop()
sphere(50)
}
Console
Clear



Sketch C4.8 final trick up our sleeve

Adding in a new material called `emissiveMaterial()`, this gives the appearance of glowing but doesn't actually emit light. It takes three arguments.

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


function draw()
{
  background(0)
  ambientLight(100)
  directionalLight(200, 200, 0, sin(frameCount / 30), 0, cos(frameCount / 30))
  push()
  translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount / 30))
  emissiveMaterial(200, 200, 0)
  sphere(10)
  pop()
  sphere(50)
}
```

🗒️ Notes

We give it the same colour as the directional light; if you don't, you may get the light on the small sphere as well.

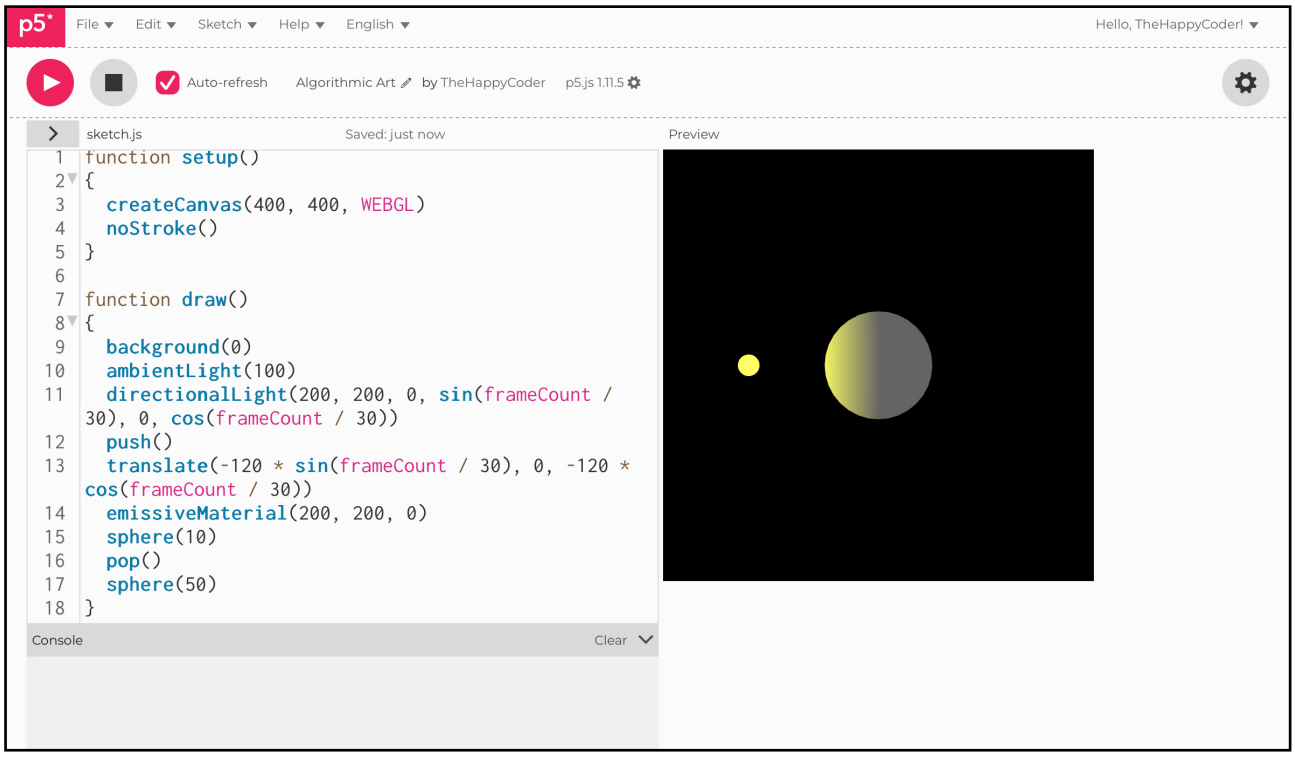
🌻 Challenge

I will leave it to you to explore further.

🛠️ Code Explanation

`emissiveMaterial(200, 200, 0)`	Makes the shape glow, you select the colour you want.

Figure C4.8


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
noStroke()
}
function draw()
{
background(0)
ambientLight(100)
directionalLight(200, 200, 0, sin(frameCount / 30), 0, cos(frameCount / 30))
push()
translate(-120 * sin(frameCount / 30), 0, -120 * cos(frameCount / 30))
emissiveMaterial(200, 200, 0)
sphere(10)
pop()
sphere(50)
}
Console
Clear