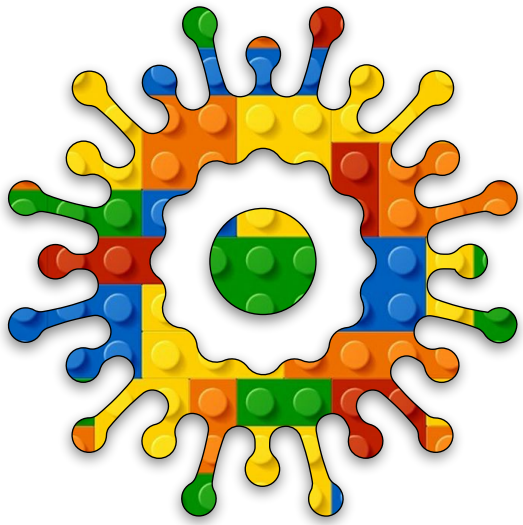
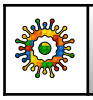
Creative Coding Module C Unit #3 lights and materials part 1



Content

Module C Unit #3 lights and materials part 1



Introduction to lights and materials part 1

Because we are working in 3D, we can position lights as a separate entity rather than just fill a shape with a colour (which you can do as well).

We can create many pleasing results with just a little practice. There are a number of light options as well as materials. Using the right combination, for not all work as you might expect, you can create some pleasing effects.

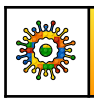
There are a number of materials we can use with WebGL:

- normalMaterial()
- ambientMaterial()
- specularMaterial()
- emissiveMaterial()

For the lights, we have a few to choose from:

- ambientLight()
- pointLight()
- directionalLight()
- lights()

You have a lot to choose from and play with.

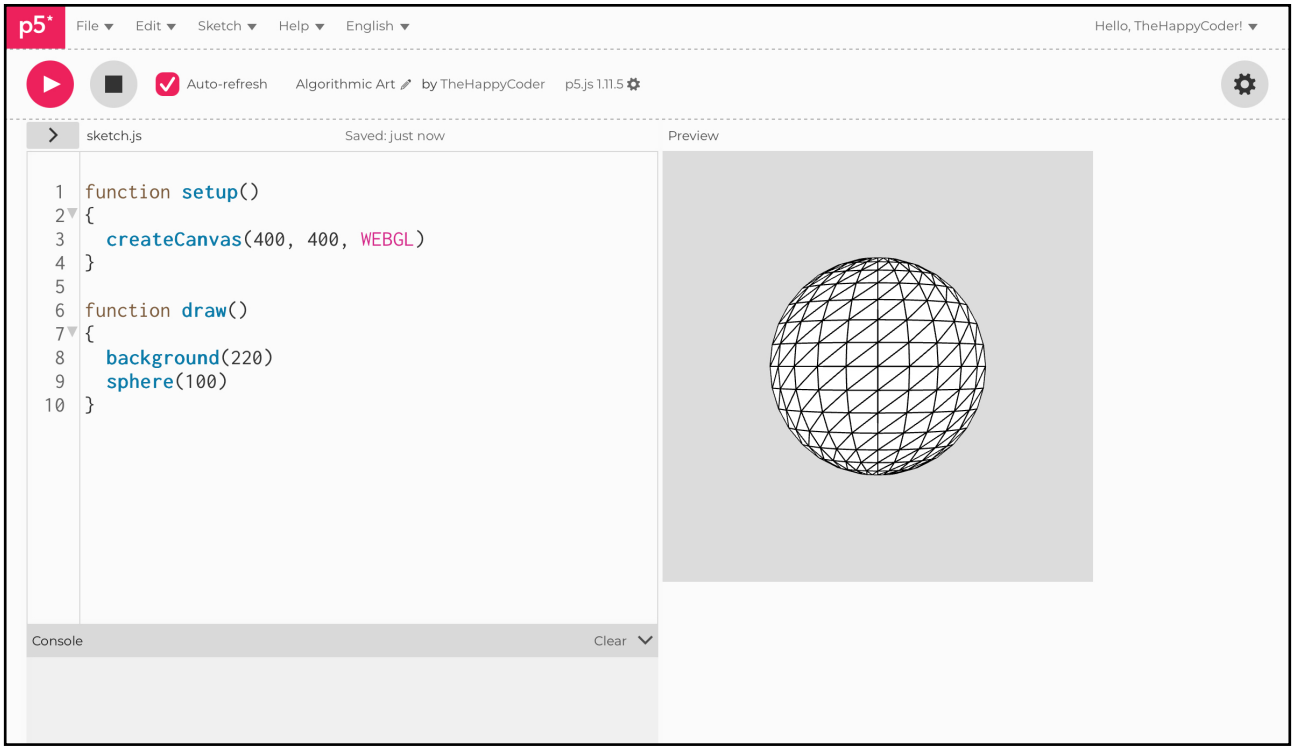


Sketch C3.1 draw a sphere

Our starting sketch, the humble sphere.

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

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

Figure C3.1


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)
sphere(100)
}
Console
Clear

Sketch C3.2 a normal material

Using `normalMaterial()` is just a quick way to give objects a solid colour to make sure everything is working OK.

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

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

Notes

It gives you a strange, almost psychedelic tint to the colours. There isn't much you can do with it other than use it, handy though.

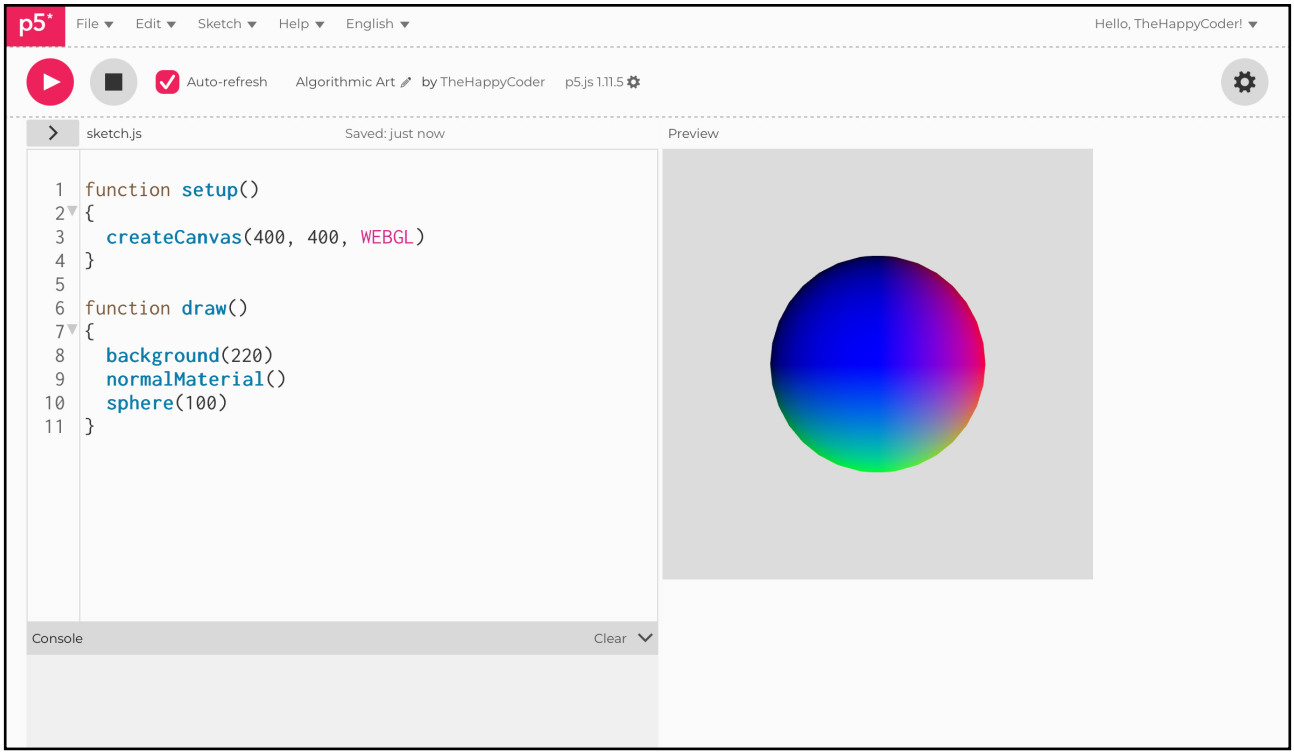
Challenges

1. Try other shapes.
2. Add `orbitControl()` to spin the shapes around.

Code Explanation

`normalMaterial()`	A basic material you can put on the shapes

Figure C3.2


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)
normalMaterial()
sphere(100)
}
Console
Clear

Sketch C3.3 a more ambient material

Another material we could give it is called `ambientMaterial()`. We have to give it a colour, and in this case, we will give it a `yellow` colour.

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

function draw()
{
  background(220)
  ambientMaterial('yellow')
  sphere(100)
}
```

🗒️ Notes

Oh dear, that didn't work.

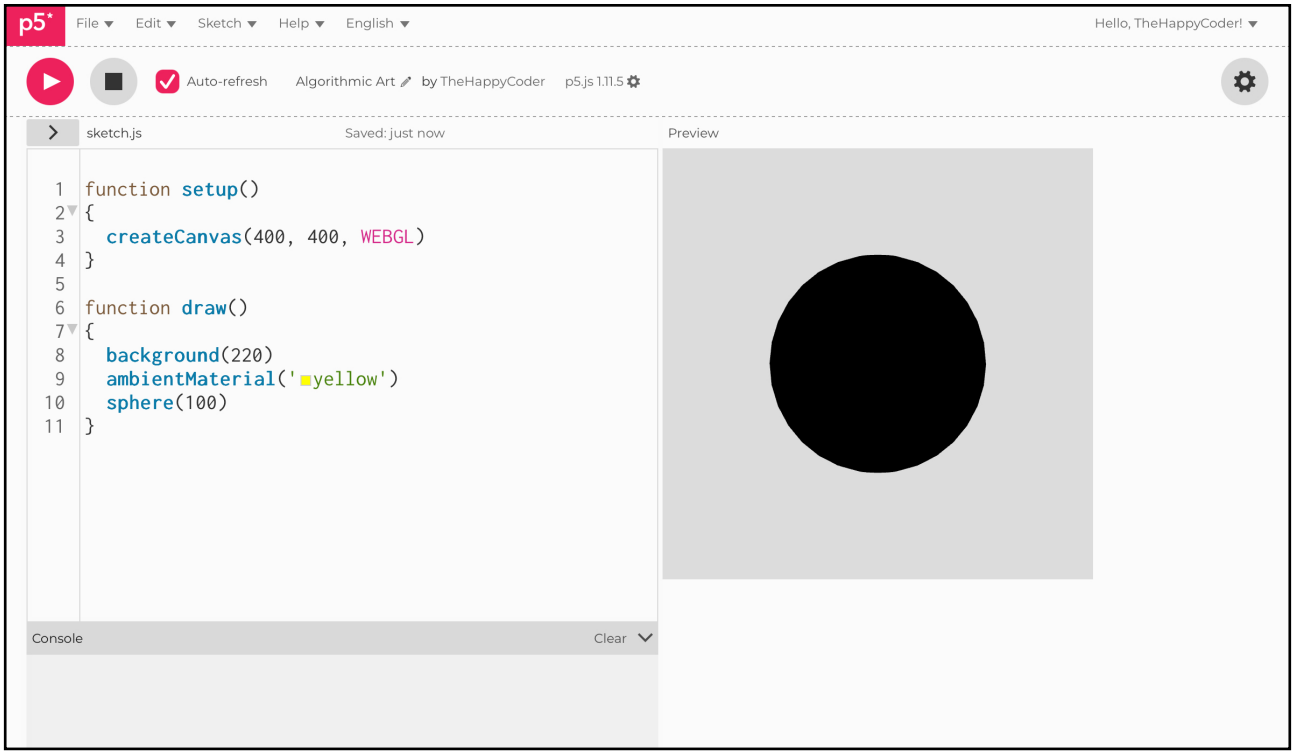
🌻 Challenge

Can you imagine why?

🛠️ Code Explanation

`ambientMaterial('yellow')`	Gives the shape a material colour

Figure C3.3


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)
ambientMaterial('yellow')
sphere(100)
}
Console
Clear

Sketch C3.4 an ambient light

There is a good reason it didn't work; we need to give it some `ambientLight()`. For this to show the true colour of the `ambientMaterial()`, we need to use white light.

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

function draw()
{
  background(220)
  ambientLight('white')
  ambientMaterial('yellow')
  sphere(100)
}
```

🗒️ Notes

We have shone a white light, which means the sphere is now illuminated and its true colour can be seen.

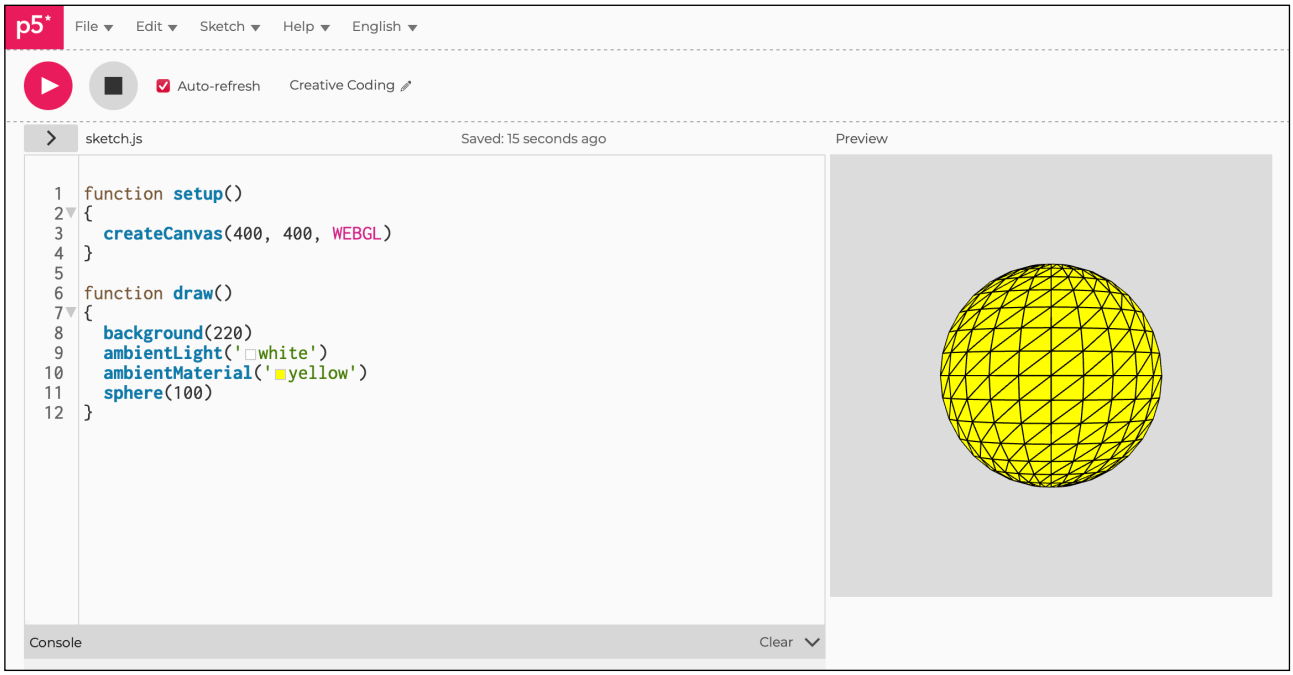
🌻 Challenge

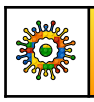
Try other colour combinations.

🛠️ Code Explanation

`ambientLight('white')`	A white ambient light

Figure C3.4


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Creative Coding
sketch.js
Saved: 15 seconds ago
Preview
1 function setup()
2 {
3 createCanvas(400, 400, WEBGL)
4 }
5
6 function draw()
7 {
8 background(220)
9 ambientLight('white')
10 ambientMaterial('yellow')
11 sphere(100)
12 }
Console
Clear



Sketch C3.5 ambient blue

But see what happens if we make the `ambientLight()` blue.

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

function draw()
{
  background(220)
  ambientLight('blue')
  ambientMaterial('yellow')
  sphere(100)
}
```

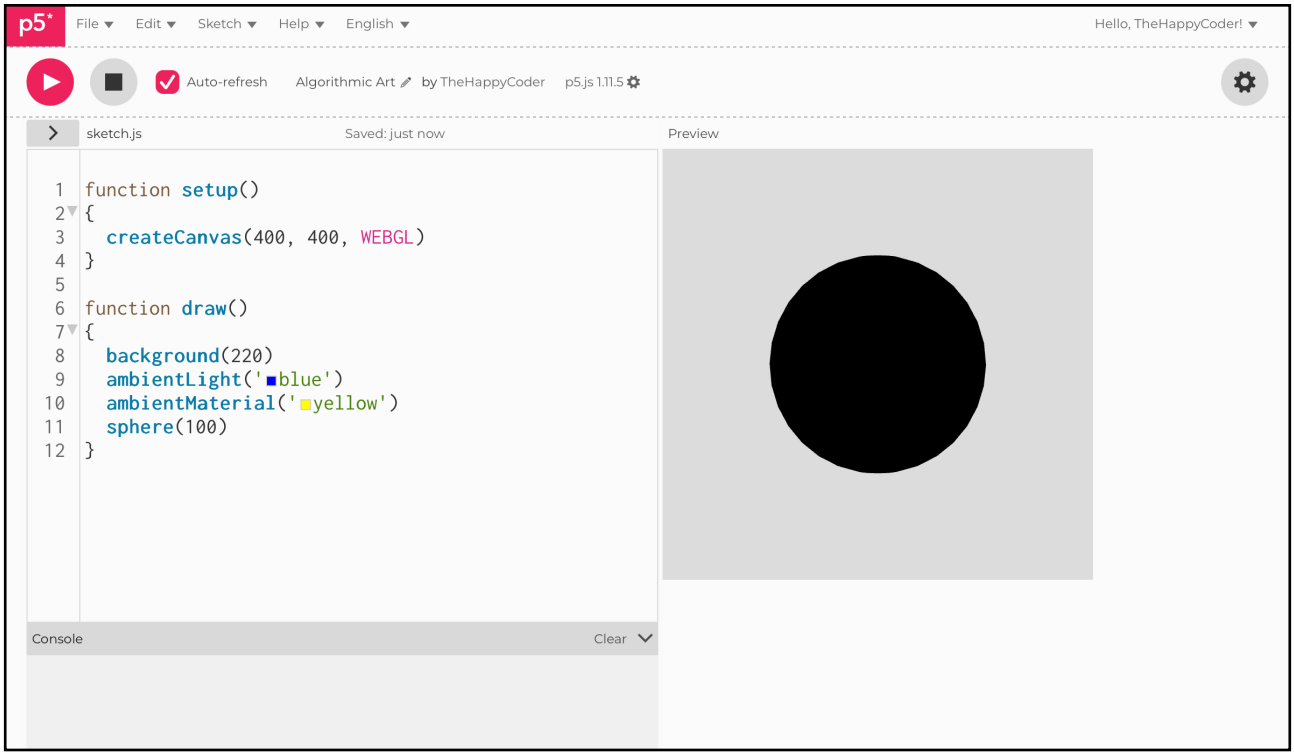
Notes

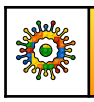
It is a black sphere; you have to be careful which colours you mix.

Challenges

1. Try `ambientLight('green')`
2. Try other colours

Figure C3.5


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)
ambientLight('blue')
ambientMaterial('yellow')
sphere(100)
}
Console
Clear



Sketch C3.6 a green light on a white material

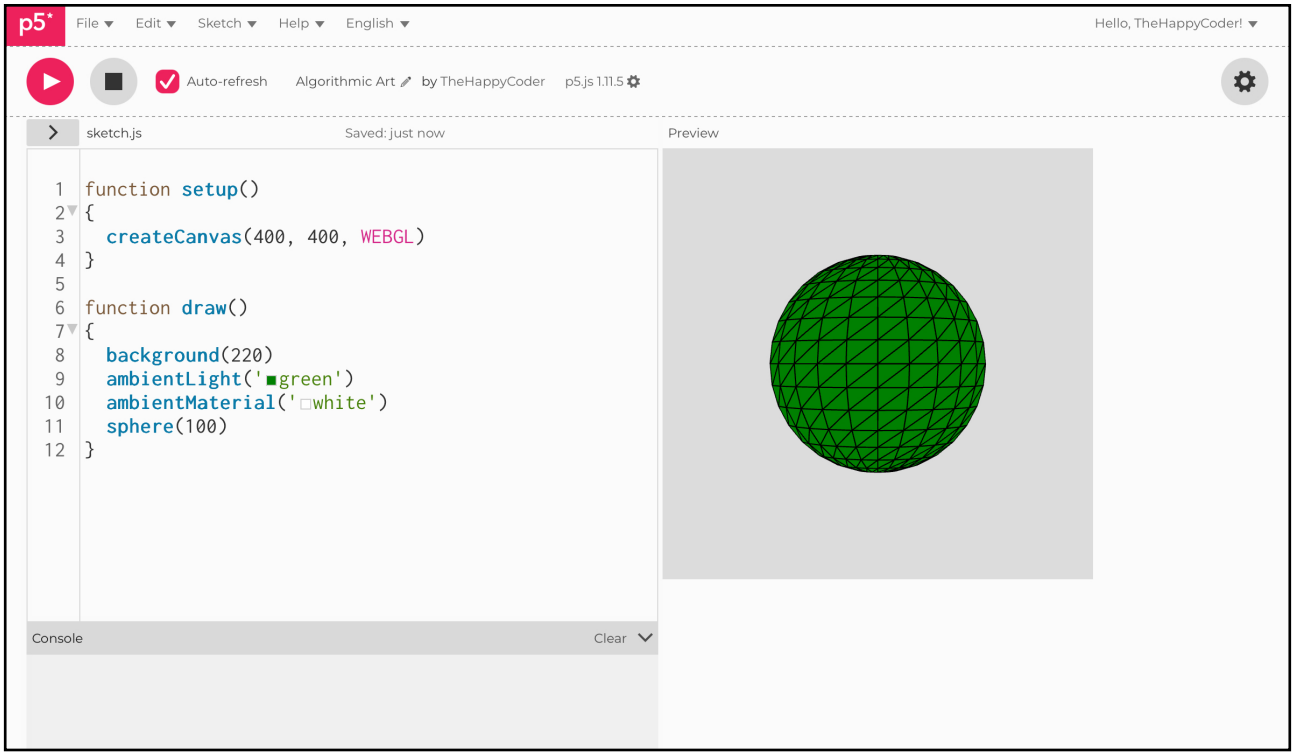
What would happen if we made the material white and shone a green light on it?

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

function draw()
{
  background(220)
  ambientLight('green')
  ambientMaterial('white')
  sphere(100)
}
```

Notes

It all depends on how much of one colour is in the other. If there is none of one in the other (light or material), then it will turn black. As you probably guess, ambient light shines from all directions, not from a source, and it is not coming from a particular direction, so this is why you get a consistent colour from every angle.

Figure C3.6


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)
}
function draw()
{
background(220)
ambientLight('green')
ambientMaterial('white')
sphere(100)
}
Console
Clear

Sketch C3.7 a point of light

❗ Remove the `ambientLight()` and the `ambientMaterial()`.

An alternative is to have a point of light, and for this, we need to still give it a colour but also a position. The function `pointLight()` has six arguments: the first three are the RGB colours, and the second set of three are `x`, `y`, and `z` positions relative to the centre of the canvas space.

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

function draw()
{
  background(220)
  pointLight(0, 0, 255, -250, -200, 350)
  sphere(100)
}
```

🗒️ Notes

We have positioned the light source to the left, slightly above and in front of the sphere. Remember that the positions are relative to the origin of the 3D space.

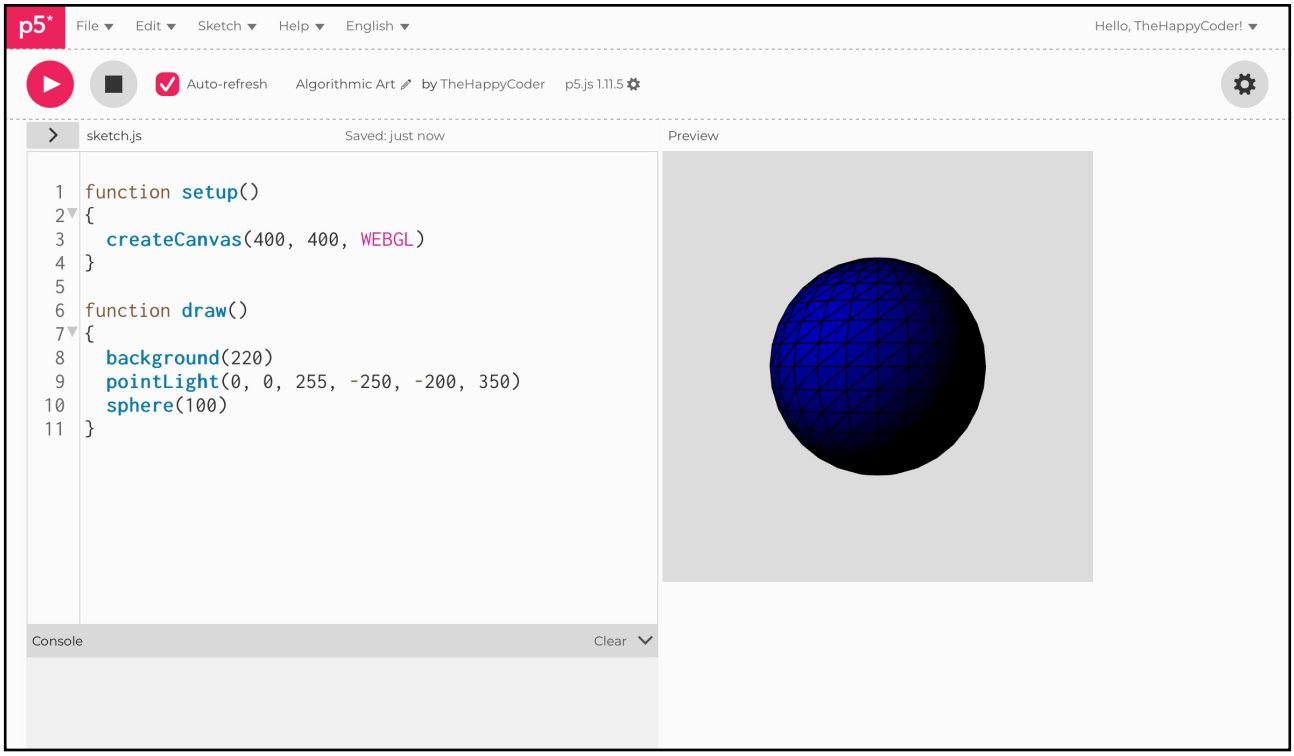
🌻 Challenge

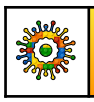
Try other positions.

🛠️ Code Explanation

`pointLight(0, 0, 255, -250, -200, 350)`	This has a colour part (three arguments) and a position part (also three arguments)

Figure C3.7


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)
pointLight(0, 0, 255, -250, -200, 350)
sphere(100)
}
Console
Clear



Sketch C3.8 moving the light

If we give the x and y position the mouseX and mouseY, so you can see the point of light moving as you move the mouse. We subtract 200 because the mouse still thinks it is on a 2D canvas.

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

function draw()
{
  background(220)
  pointLight(0, 0, 255, mouseX - 200, mouseY - 200, 200)
  sphere(100)
}
```

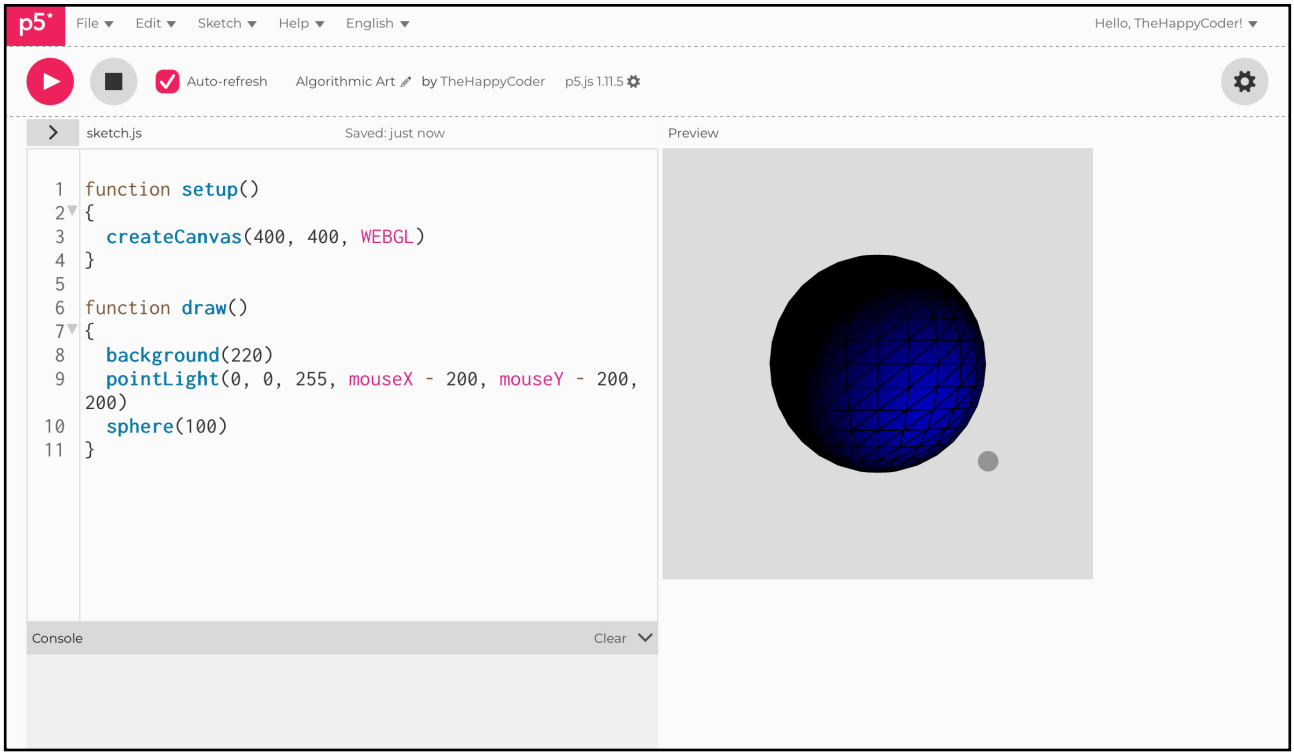
Notes

The light source follows the mouse. The results are a bit dim, though, but you will see that you can use them effectively.

Challenge

Have mouseX or mouseY move the z position.

Figure C3.8


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)
}
function draw()
{
background(220)
pointLight(0, 0, 255, mouseX - 200, mouseY - 200, 200)
sphere(100)
}
Console
Clear

Sketch C3.9 three points of light

Here, we are going to have three points of light and a different colour for each one. Adding a `noStroke()` removes the detail lines.

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

function draw()
{
  background(220)
  noStroke()
  pointLight(0, 0, 255, -200, -200, 200)
  pointLight(255, 0, 0, 200, 200, 200)
  pointLight(0, 255, 0, 200, -200, 200)
  sphere(100)
}
```

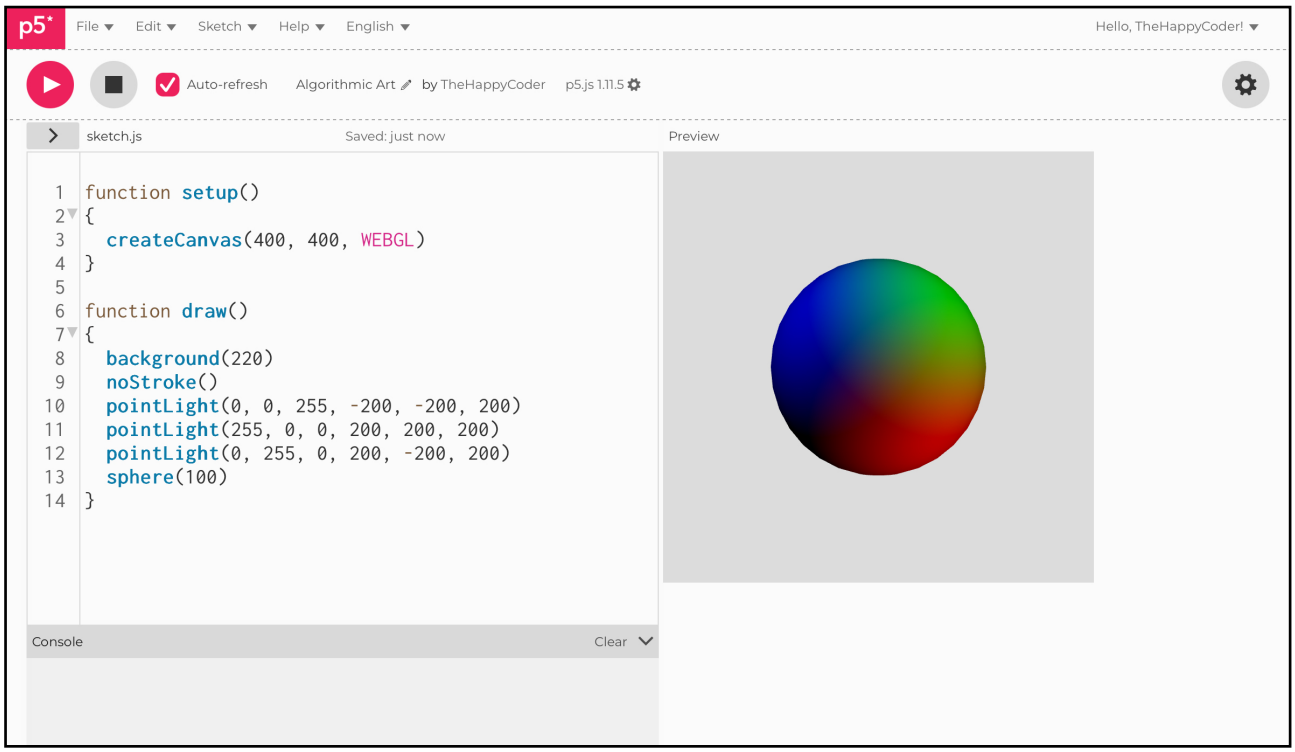
Notes

Creates an interesting effect, which is something you could play with.

Challenge

Add in `ambientLight(100)`.

Figure C3.9


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()
pointLight(0, 0, 255, -200, -200, 200)
pointLight(255, 0, 0, 200, 200, 200)
pointLight(0, 255, 0, 200, -200, 200)
sphere(100)
}
Console
Clear

Sketch C3.10 metallic finish

We have a matte colour by default, but we can make the surface metallic or reflective. We still need to give it a base colour (in this case, white).

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

function draw()
{
  background(220)
  noStroke()
  pointLight(0, 0, 255, -200, -200, 200)
  pointLight(255, 0, 0, 200, 200, 200)
  pointLight(0, 255, 0, 200, -200, 200)
  specularMaterial(255)
  sphere(100)
}
```

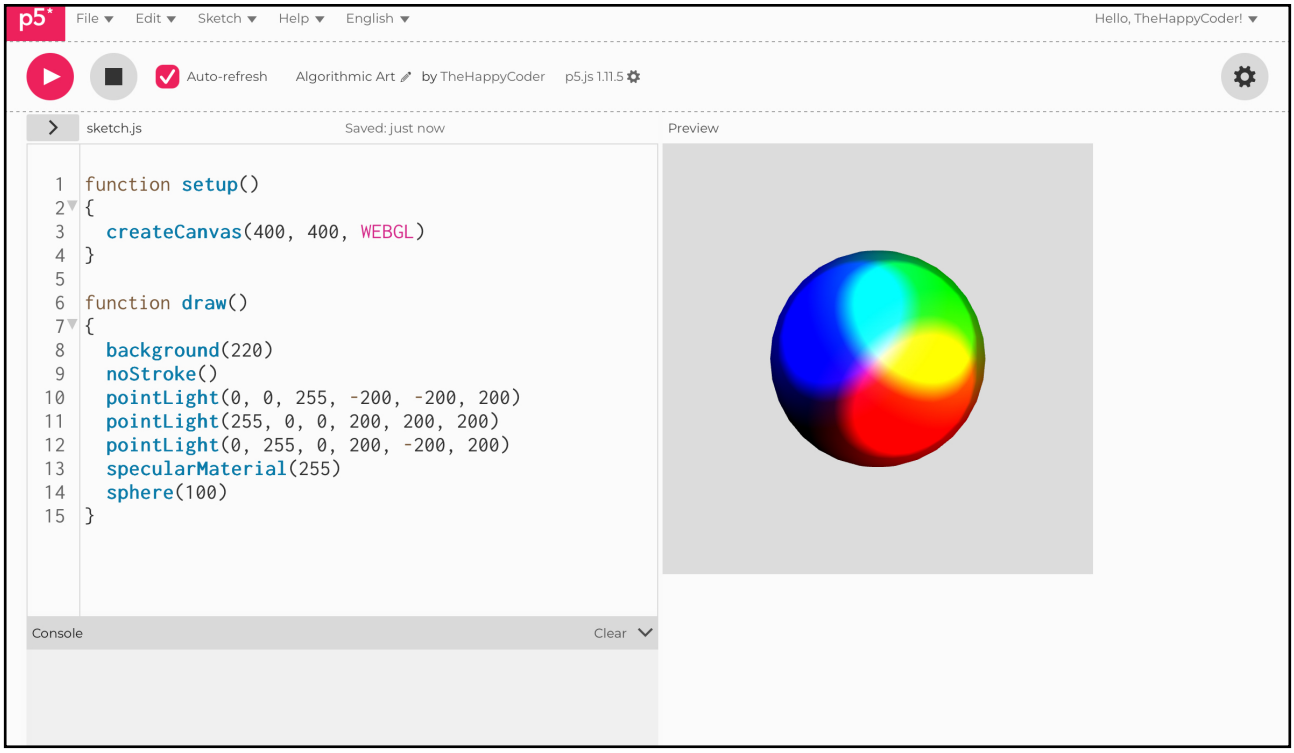
🗒️ Notes

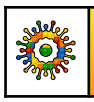
A bit overwhelming, but we can tone it down a bit.

🛠️ Code Explanation

`specularMaterial(255)`	Brings a more reflective surface colour to the shape

Figure C3.10


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()
pointLight(0, 0, 255, -200, -200, 200)
pointLight(255, 0, 0, 200, 200, 200)
pointLight(0, 255, 0, 200, -200, 200)
specularMaterial(255)
sphere(100)
}
Console
Clear



Sketch C3.11 more subtle

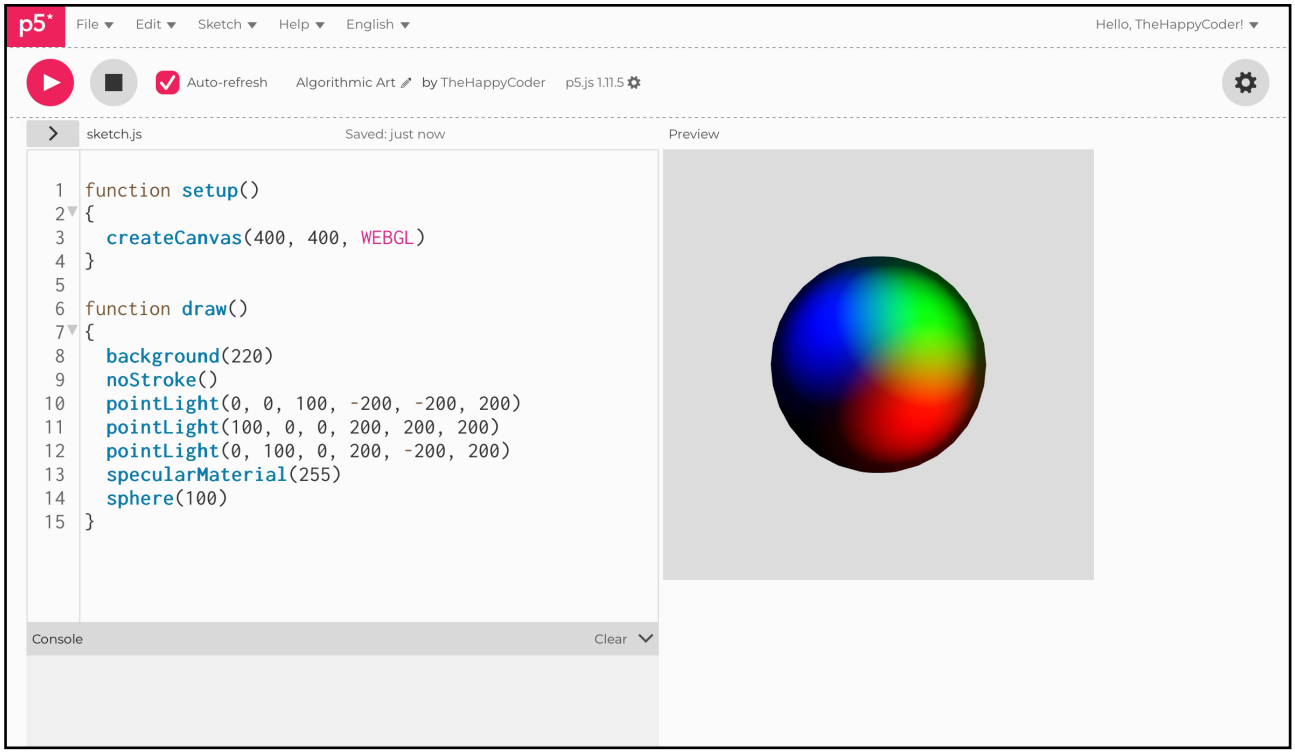
By reducing the amount of red, blue, and green, we can create a more subtle effect.

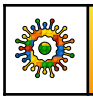
```
function setup()
{
  createCanvas(400, 400, WEBGL)
}

function draw()
{
  background(220)
  noStroke()
  pointLight(0, 0, 100, -200, -200, 200)
  pointLight(100, 0, 0, 200, 200, 200)
  pointLight(0, 100, 0, 200, -200, 200)
  specularMaterial(255)
  sphere(100)
}
```

Notes

That looks a bit more natural.

Figure C3.11


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()
pointLight(0, 0, 100, -200, -200, 200)
pointLight(100, 0, 0, 200, 200, 200)
pointLight(0, 100, 0, 200, -200, 200)
specularMaterial(255)
sphere(100)
}
Console
Clear



Sketch C3.12 box and rotate

❗ making a few changes to the sketch

For a bit of fun, let's change back to a box and rotate.

```
let angle = 0

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

function draw()
{
  background(220)
  noStroke()
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  pointLight(0, 0, 100, -200, -200, 200)
  pointLight(100, 0, 0, 200, 200, 200)
  pointLight(0, 100, 0, 200, -200, 200)
  specularMaterial(255)
  box(100)
  angle++
}
```

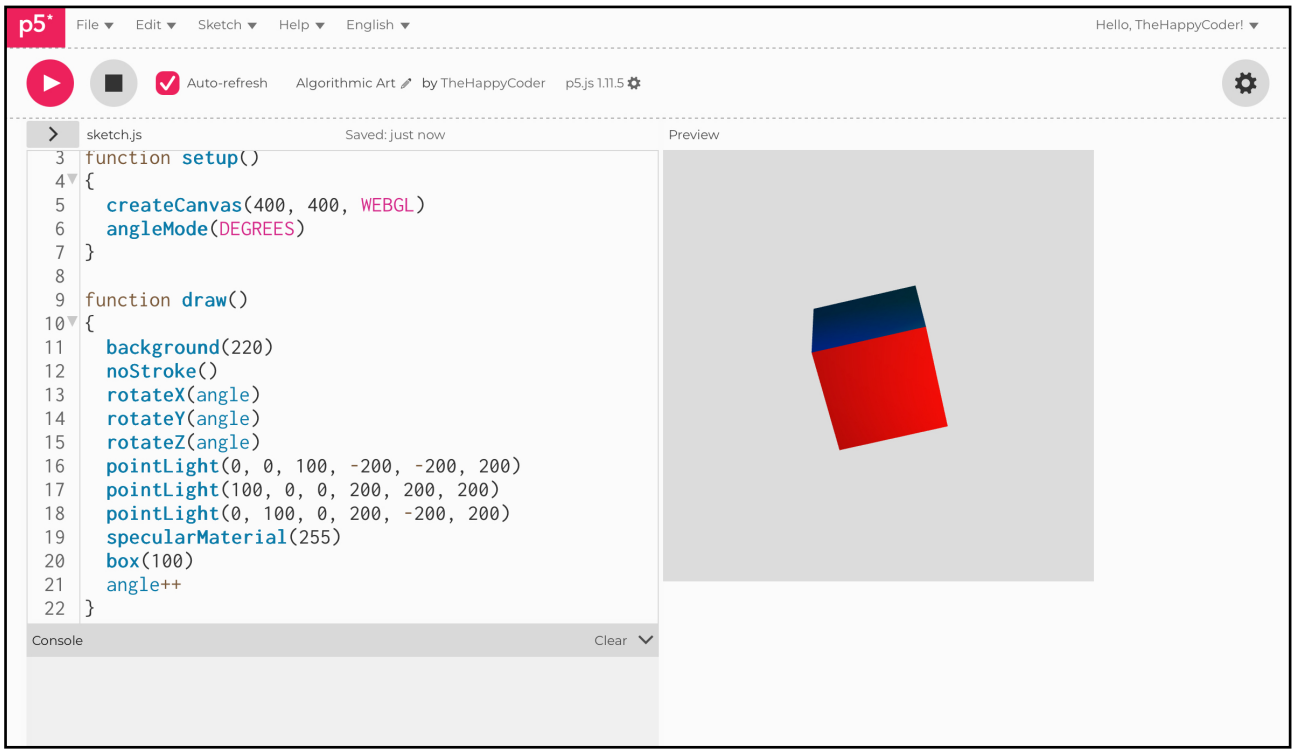
📓 Notes

You can see the more reflective nature of the material with the cube rotating.

🌻 Challenges

1. Try stronger values for the red, green, and blue.
2. Try other shapes and positions of the light source.
3. Give the material different colours.

Figure C3.12


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
3 function setup()
4 {
5 createCanvas(400, 400, WEBGL)
6 angleMode(DEGREES)
7 }
8
9 function draw()
10 {
11 background(220)
12 noStroke()
13 rotateX(angle)
14 rotateY(angle)
15 rotateZ(angle)
16 pointLight(0, 0, 100, -200, -200, 200)
17 pointLight(100, 0, 0, 200, 200, 200)
18 pointLight(0, 100, 0, 200, -200, 200)
19 specularMaterial(255)
20 box(100)
21 angle++
22 }
Console
Clear