

Section C  
Unit #3  
Orbit and  
smooth



## Section C Unit 3 orbit and smooth

Sketch C3.1 default triangles

Sketch C3.2 smoothing the curves

Sketch C3.3 using orbit control

Sketch C3.4 many boxes

Sketch C3.5 frameCount()



## Introduction to Unit #3 orbit and smooth

You will notice that the shapes aren't very smooth as they are drawn with a series of interconnecting triangles. This can be improved by increasing the number of triangles, it will mean that there is a lot more geometry for the code to keep track of so don't go mad, keep them to a minimum where possible to still give a pleasing effect. In this book we won't really need to worry as we are not making big demands on it.



## Sketch C3.1 default triangles

Start a new sketch

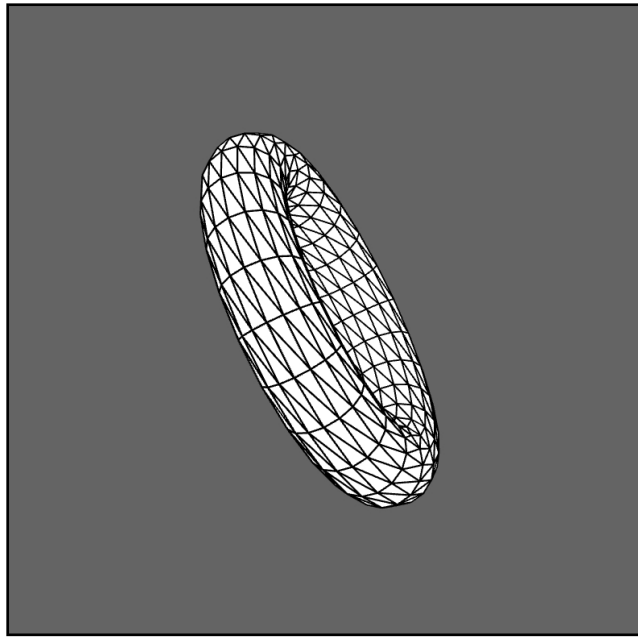
You may notice that the shapes with curves, eg the sphere and the torus are not what you call smooth. The triangles that construct the shape add a certain roughness to them. This is because the default is 24 in the x and 16 in the y. Yet we can increase these values to give us a smoother shape. However you cannot draw all those triangles and vertices so in order to draw the shape we have to remove the lines (use `noStroke()`). We start with our basic torus and we include the third and fourth argument.

```
let angle = 0

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

function draw()
{
  background(100)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  torus(100, 25, 24, 16)
  angle++
}
```

Drawn with the maximum, and default, 24  
and 16





## Sketch C3.2 smoothing the curves

We are going to remove the lines and increase the values by a factor of 2

```
let angle = 0

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

function draw()
{
  background(100)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  noStroke()
  torus(100, 25, 48, 32)
  angle++
}
```

### Notes

This creates a much smoother looking shape. However because we haven't explored materials and light yet, the full benefit is not readily appreciated.

### Challenges

See what difference it makes if you change the detail by half (12 and 8)  
Try even smaller values to see the effect as well as much larger values



## Sketch C3.3 using orbit control

Start a new sketch

You can control the point of view with your mouse. Once you run the code move your mouse over the canvas (click and drag)

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

function draw()
{
  background(220)
  orbitControl()
  box(100)
}
```



## Sketch C3.4 many boxes

To prove that it is the view that changes not the object

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

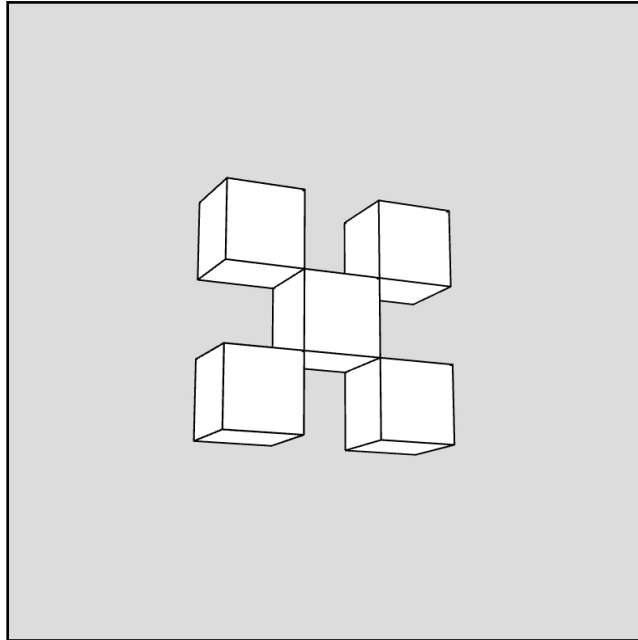
function draw()
{
  background(220)
  orbitControl()
  box(50)
  push()
  translate(50, 50, 0)
  box(50)
  pop()
  push()
  translate(-50, 50, 0)
  box(50)
  pop()
  push()
  translate(50, -50, 0)
  box(50)
  pop()
  push()
  translate(-50, -50, 0)
  box(50)
  pop()
}
```



# Challenge

Make each box rotate separately

Move your mouse over the canvas to rotate





## Sketch C3.5 frameCount()

Start another new sketch

Another quick way to rotate is to use the `frameCount`. Here we can rotate a box just using the `frameCount` variable. It counts how many frames have elapsed since starting the sketch. Note it is a variable not a function. Also use degrees otherwise it will spin out of control.

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

function draw()
{
  background(220)
  rotateX(frameCount)
  rotateY(frameCount)
  rotateZ(frameCount)
  box(100)
}
```