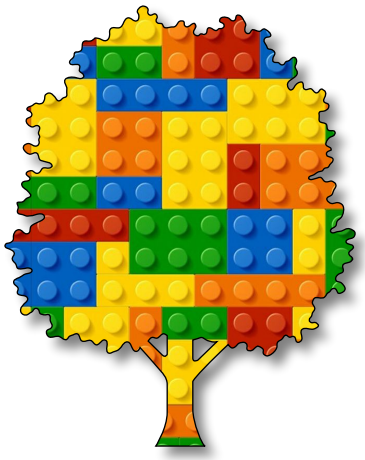
Algorithmic Art
Module F
Unit #4
image files



Content

Module F Unit #4 image files

Introduction to creating image files

We can save our work as sketches and share the full-screen version online, but it would be better if we could save our creations as an image or a video (and GIF). Then we could share our hard-earned work so others can share in our brilliance.

This unit gives you the mechanisms to save your work in various formats.

Creating a png (or jpg) image

We can create a simple PNG or JPEG image of the canvas using some very simple code. Yet we can also create PNG images that are transparent, which could be very useful if you want to add interactive characters in a game or some dynamic artwork.

Sketch F4.1 bubbles

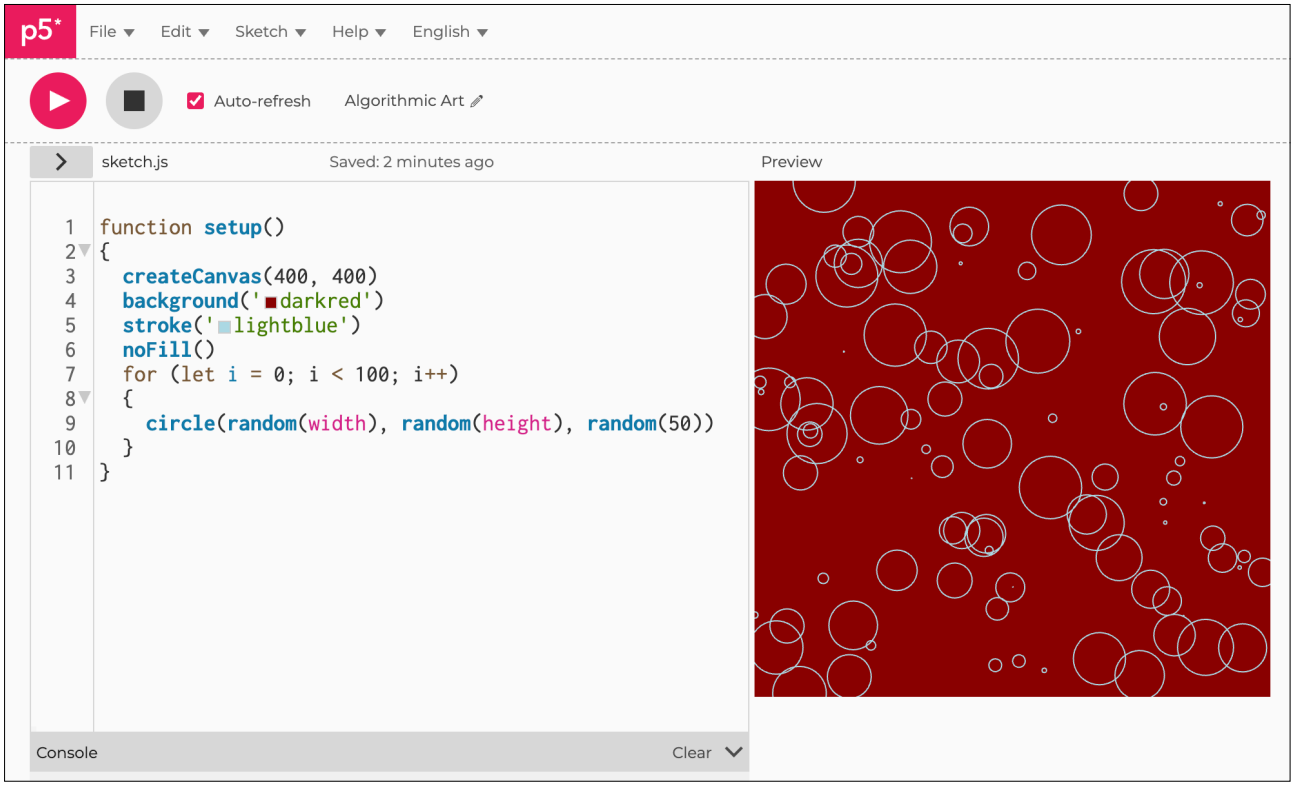
First, we will create a small canvas of bubbles.

```
function setup()
{
  createCanvas(400, 400)
  background('darkred')
  stroke('lightblue')
  noFill()
  for (let i = 0; i < 100; i++)
  {
    circle(random(width), random(height), random(50))
  }
}
```

Notes

100 randomly sized and positioned circles.

Figure F4.1



Sketch F4.2 save canvas as a .png image

When you click on the canvas, you will save an image of the canvas as a .png image.

```
function setup()
{
  createCanvas(400, 400)
  background('darkred')
  stroke('lightblue')
  noFill()
  for (let i = 0; i < 100; i++)
  {
    circle(random(width), random(height), random(50))
  }
}
```

```
function mouseClicked()
{
  save('bubbles.png')
}
```

🗒️ Notes

You can also save this as a `.jpg` image by replacing `png` with `jpg`.

🌻 Challenge

Save as a `.jpg`

Sketch F4.3 png object image

We can do something interesting with a png image. We can remove the background and have what we draw on the canvas as an image. Then we can use that in another sketch. But first, we need to make some adjustments. I have changed the canvas size and the thickness of the lines.

```
let img

function setup()
{
  img = createCanvas(200, 200)
  stroke('lightblue')
  strokeWeight(2)
  noFill()
  for (let i = 0; i < 100; i++)
  {
    circle(random(width), random(height), random(50))
  }
}

function mouseClicked()
{
  save(img, 'bubbles', 'png')
}
```

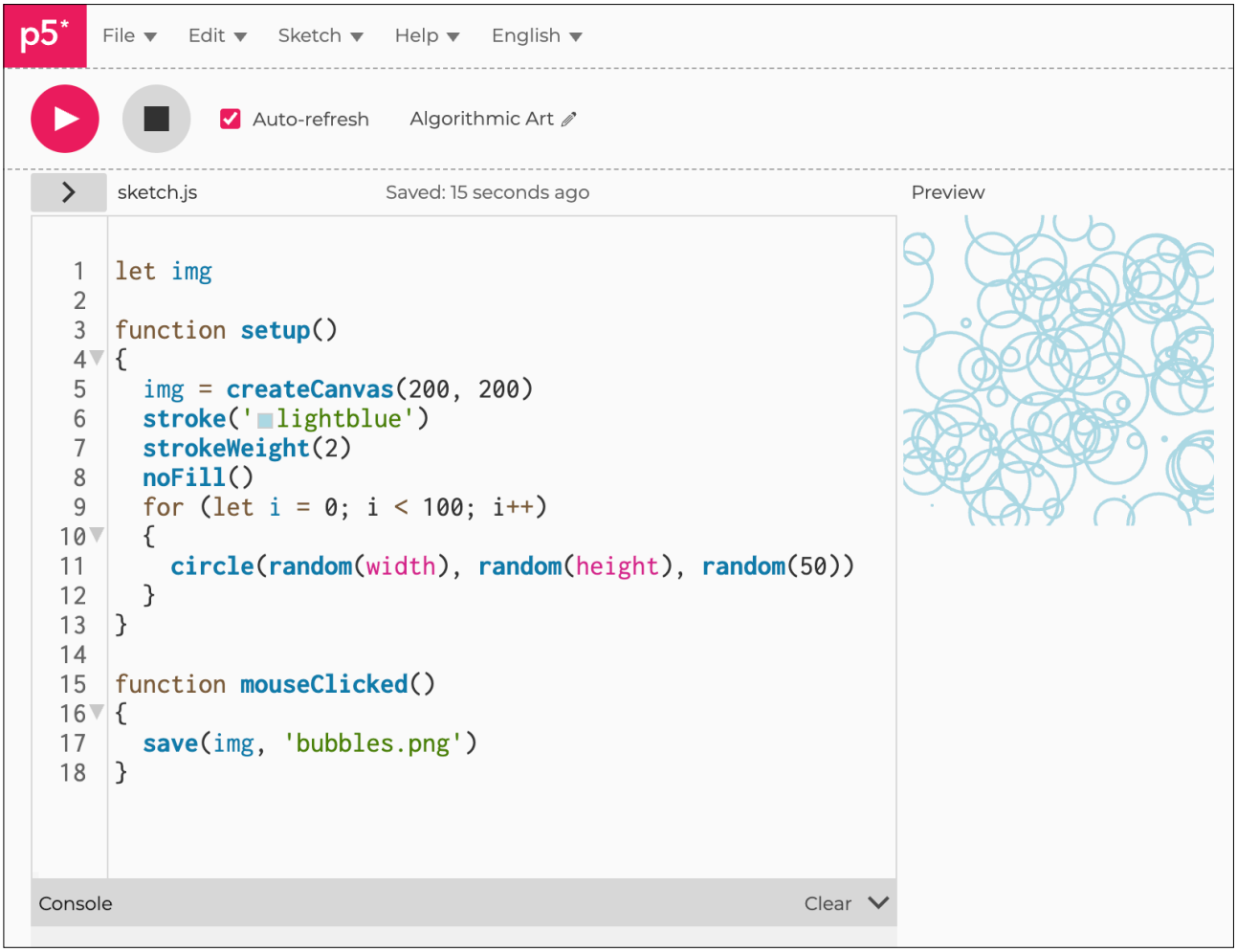
🗒️ Notes

When you click on the canvas now, you will get another file called bubbles, probably with a suffix (2). Note where it is. Most likely in the downloads folder.

🌻 Challenge

Create a button to click on to save the png file.

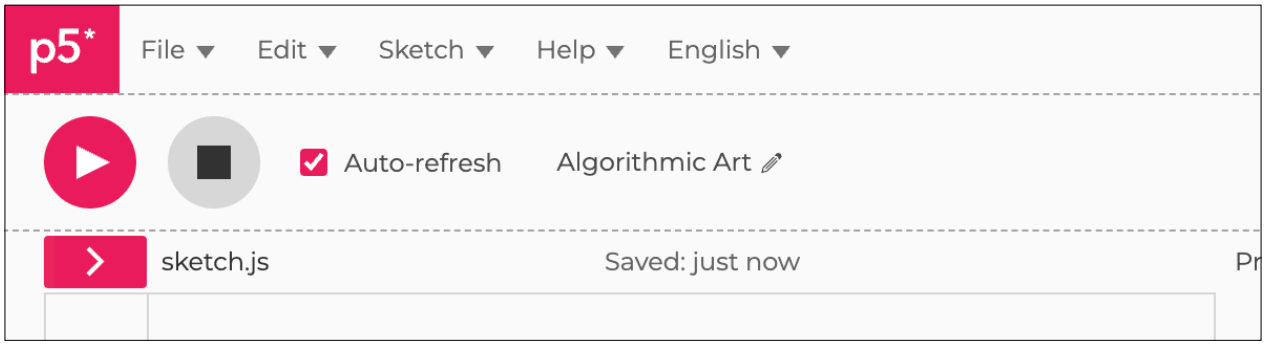
Figure F4.3


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let img
function setup()
{
img = createCanvas(200, 200)
stroke('lightblue')
strokeWeight(2)
noFill()
for (let i = 0; i < 100; i++)
{
circle(random(width), random(height), random(50))
}
}
function mouseClicked()
{
save(img, 'bubbles.png')
}
Console
Clear

Uploading the image

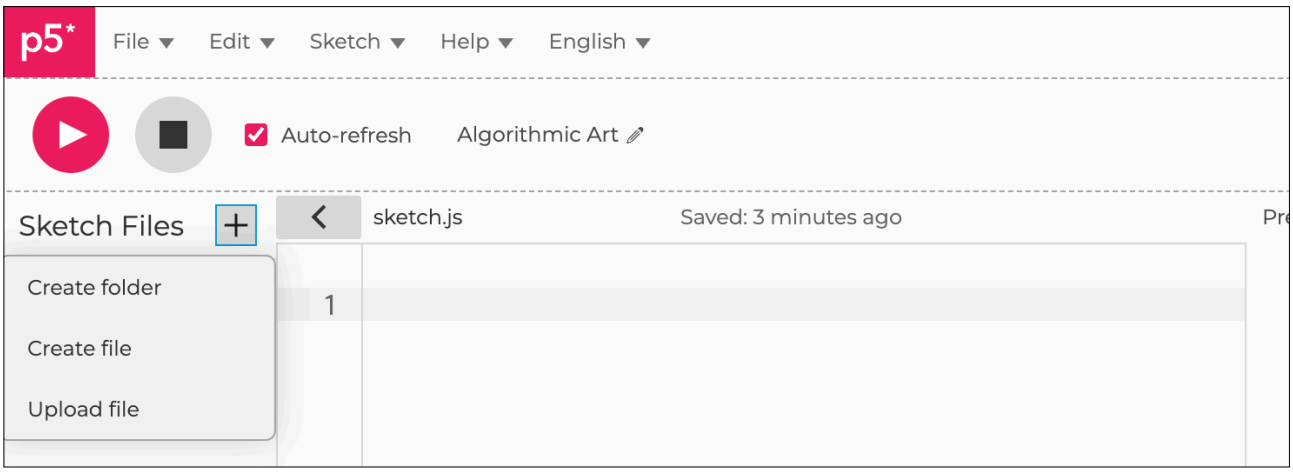
To add the image, we need to upload it as a file. To do that, we click on the side arrow next to where it says sketch.js, shown in Fig.1 below.

Figure 1: rectangular arrow button



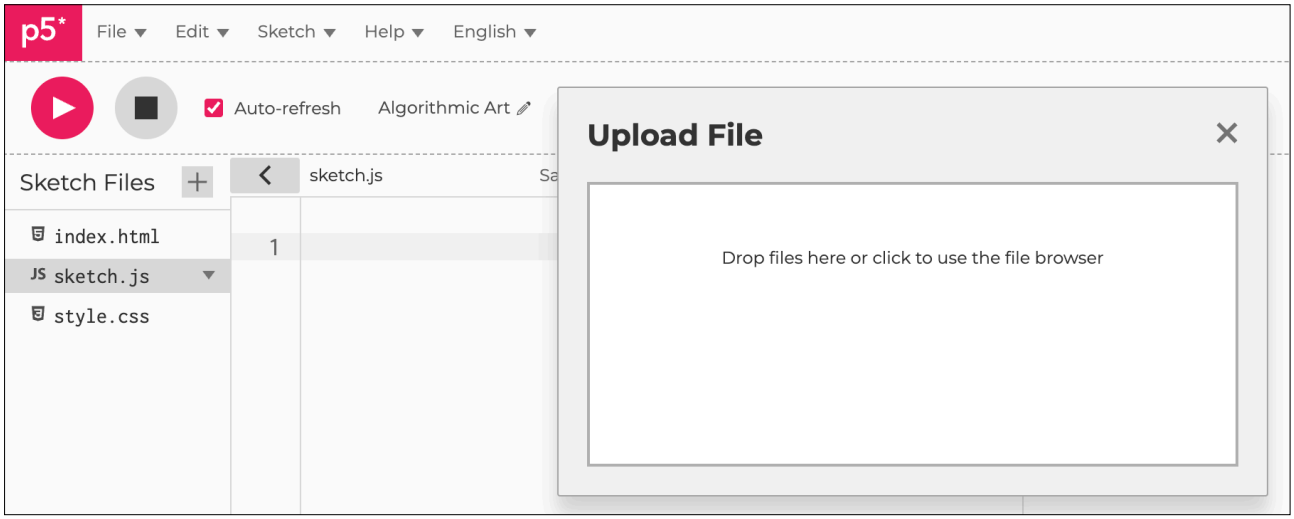
When you click on that arrow, it opens up a list of files. To add anything, you need to click on the + sign at the top, as shown in Figure 2 below.

Figure 2: the + means to add a file



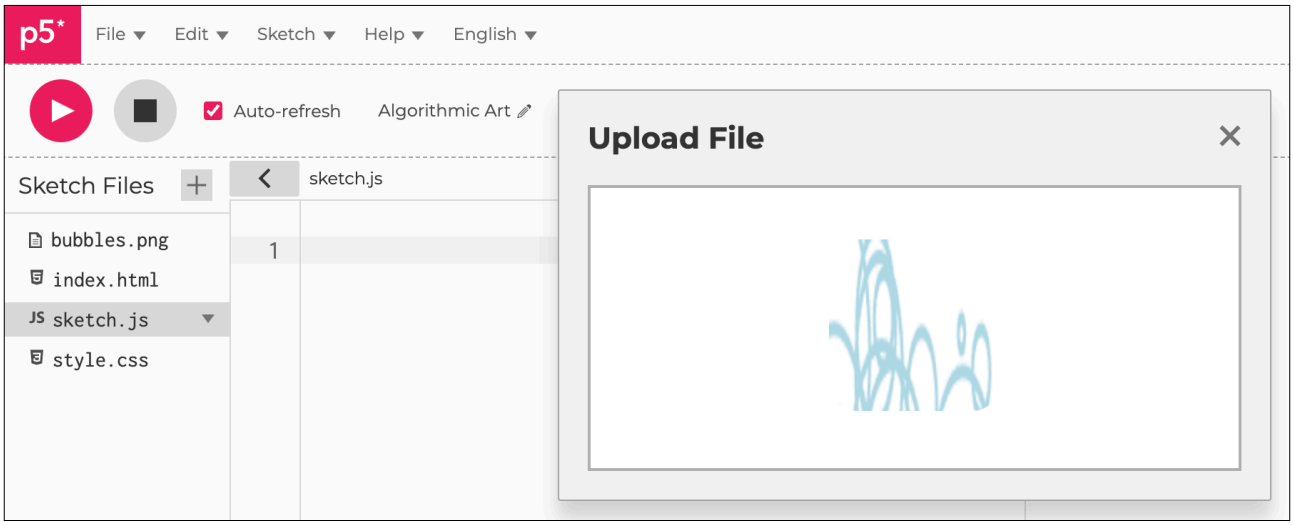
Click on the upload file, and a box appears. You can either click inside that box to browse to where your bubble file is or drag and drop it if it is easily accessible. See Fig. 3 below.

Figure 3: drop or browse window box



Once you have selected the file, then you should see it appear in the upload file box with a big tick inside it. See Fig.4 below. Notice also that the name of the file plus the extension is now listed in the list of files. If you have other versions of the png you can rename it using the arrow next to it with an additional drop-down list.

Figure 4: file (image) uploaded



Sketch F4.4 uploading the bubbles png

This will upload the image file `bubbles.png`. It places the image (at a size of your choosing) onto the canvas and follows the movement of the mouse.

```
let img

async function setup()
{
  createCanvas(400, 400)
  imageMode(CENTER)
  img = await loadImage('bubbles.png')
}

function draw()
{
  background('darkred')
  image(img, mouseX, mouseY, 100, 100)
}
```

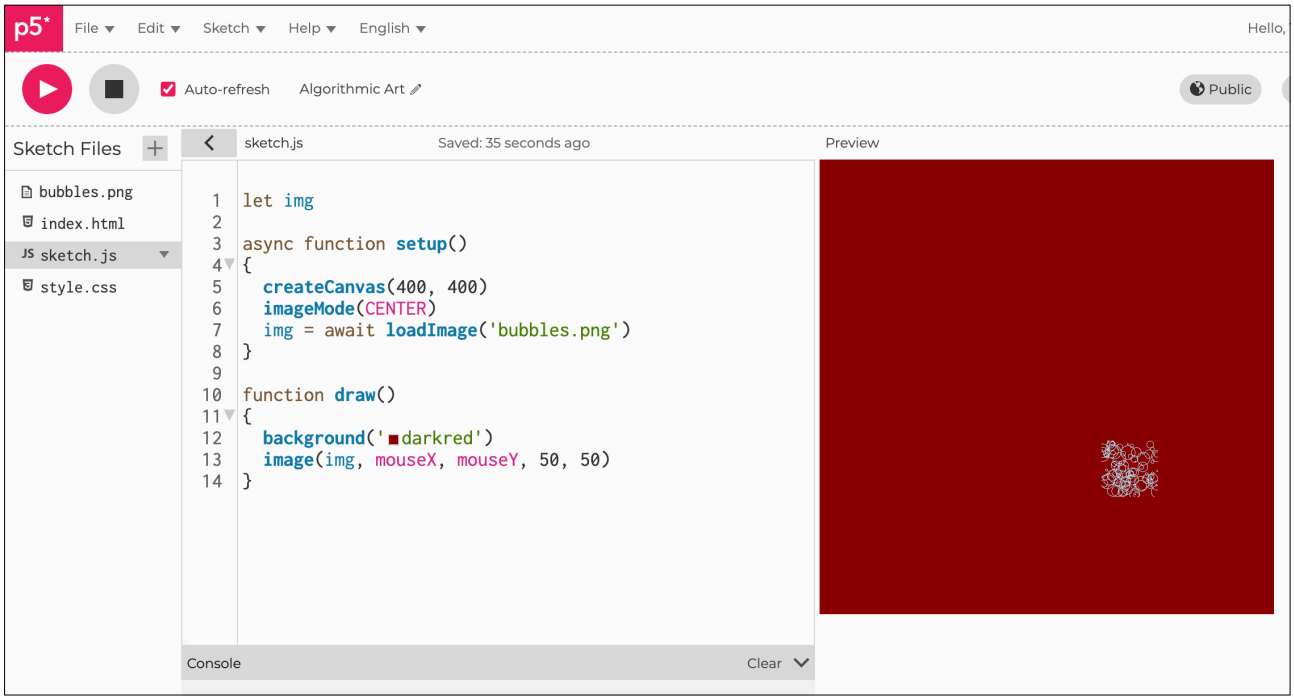
🗒️ Notes

The main takeaway is that there is no background.

🌻 Challenge

Create another shape that you can manipulate by rotating or bouncing around the canvas.

Figure F4.4


p5*
File
Edit
Sketch
Help
English
Hello,
Auto-refresh
Algorithmic Art
Public
Sketch Files
bubbles.png
index.html
sketch.js
style.css
sketch.js
Saved: 35 seconds ago
Preview
let img
async function setup()
{
createCanvas(400, 400)
imageMode(CENTER)
img = await loadImage('bubbles.png')
}
function draw()
{
background('darkred')
image(img, mouseX, mouseY, 50, 50)
}
Console
Clear

Creating a gif

You may well be familiar with GIF images, which are those mini videos that repeat continuously. We can create one quite easily using the `.gif` extension as we save the file as an image. If you want to be very clever, you could count the number of frames so that it repeats seamlessly.

Sketch F4.5 rotating torus

This should be pretty familiar to you.

```
function setup()
{
  createCanvas(400, 400, WEBGL)
  rectMode(CENTER)
  frameRate(30)
}

function draw()
{
  background('darkred')
  fill('yellow')
  lights()
  rotateX(frameCount * 0.02)
  rotateY(frameCount * 0.03)
  torus(width * 0.2, width * 0.07, 32, 32)
}
```

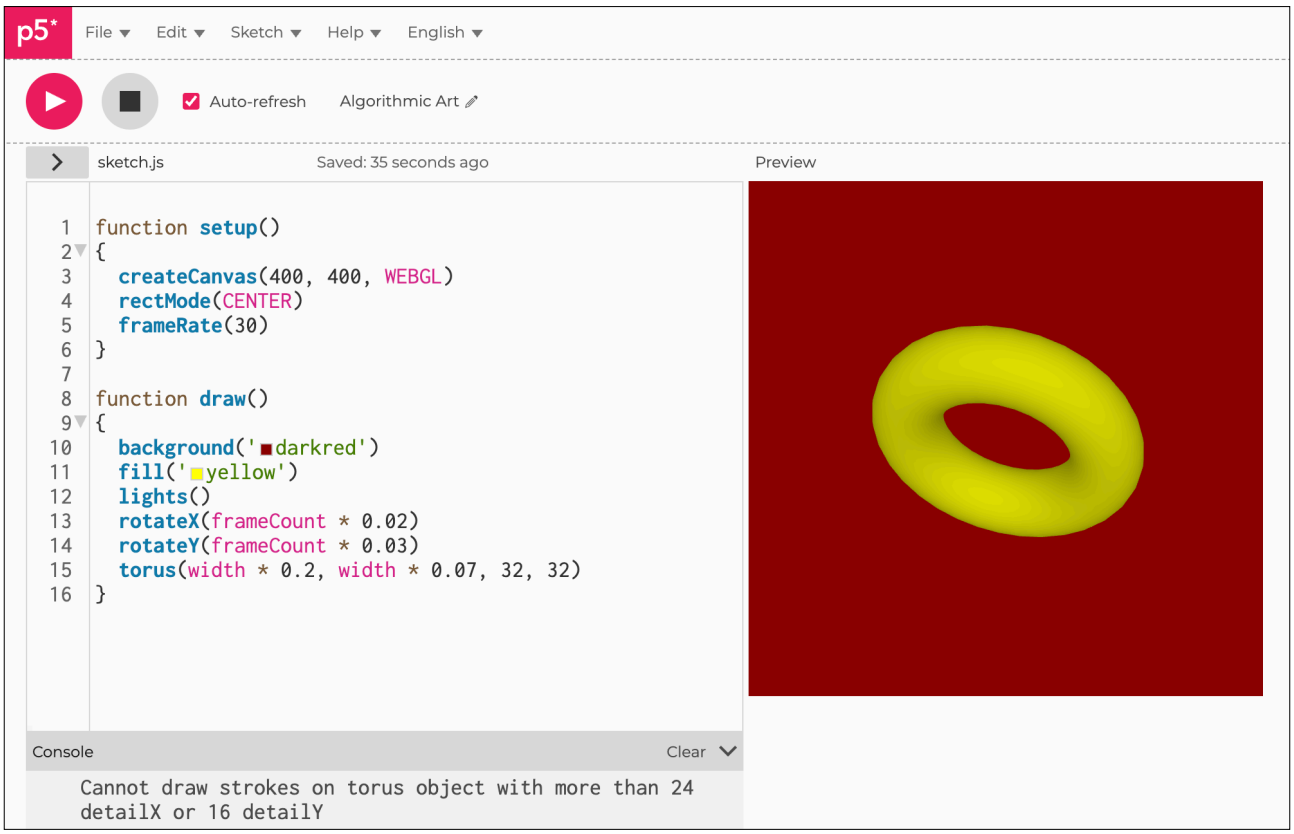
🗒️ Notes

The only new addition is specifying the frame rate. This is useful if you want your GIF to end where it starts, so that it seems like a continuous motion with no breaks.

🌻 Challenge

Create your own.

Figure F4.5


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 35 seconds ago
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
rectMode(CENTER)
frameRate(30)
}
function draw()
{
background('darkred')
fill('yellow')
lights()
rotateX(frameCount * 0.02)
rotateY(frameCount * 0.03)
torus(width * 0.2, width * 0.07, 32, 32)
}
Console
Clear
Cannot draw strokes on torus object with more than 24 detailX or 16 detailY

Sketch F4.6 the space bar

A simple `keyPressed()` function that waits for the space bar to be pressed (" ").

```
function setup()
{
  createCanvas(400, 400, WEBGL)
  rectMode(CENTER)
  frameRate(30)
}

function draw()
{
  background('darkred')
  fill('yellow')
  lights()
  rotateX(frameCount * 0.02)
  rotateY(frameCount * 0.03)
  torus(width * 0.2, width * 0.07, 32, 32)
}
```

```
function keyPressed()
{
  if (key == " ")
  {
    // something happens
  }
}
```

Notes

Nothing will happen yet.

Sketch F4.7 options and frames

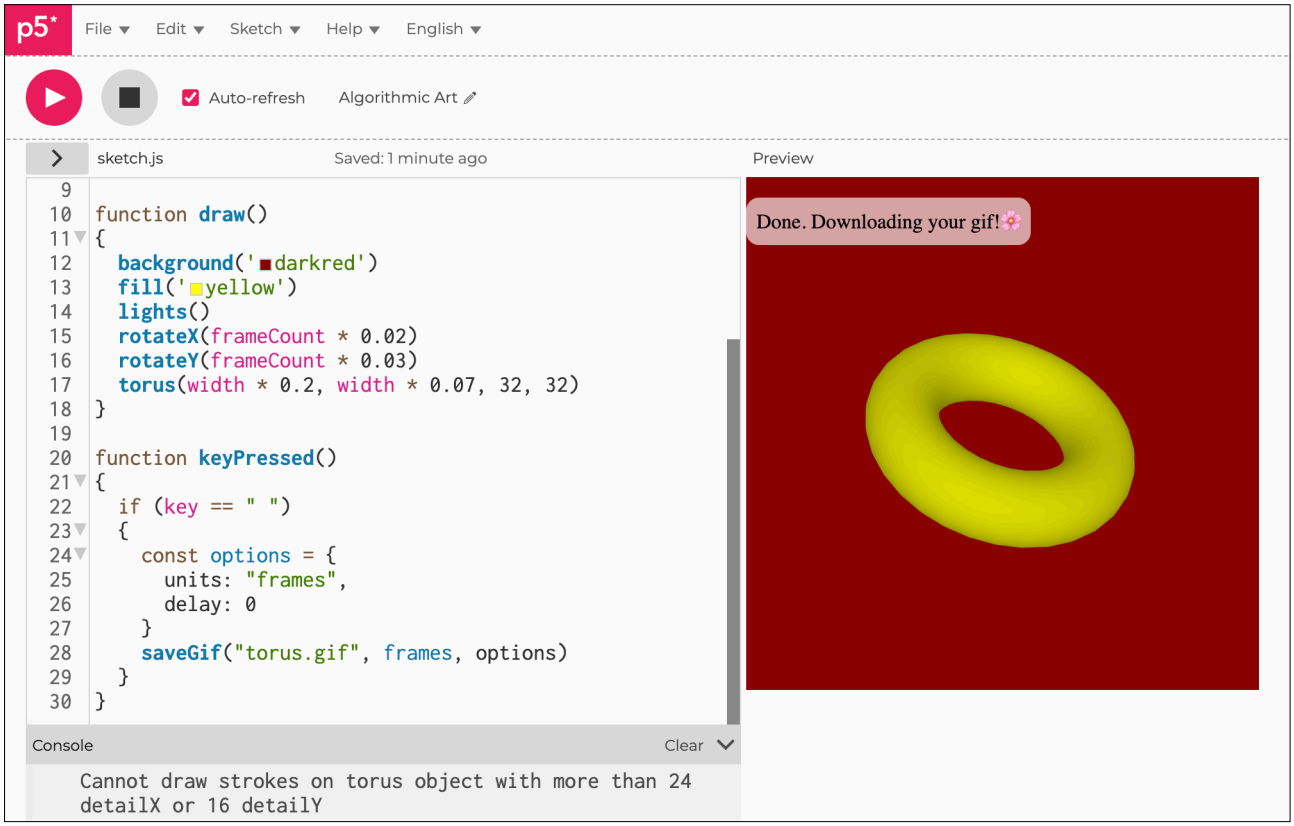
We are going to record and create a GIF. We can give the GIF a number of options: the number of frames it will last for, and if we want any delay before it starts. We have already set the `frameRate()` to `30`. We will record for `120` frames and have zero delay.

```
let frames = 120

function setup()
{
  createCanvas(400, 400, WEBGL)
  rectMode(CENTER)
  frameRate(30)
}

function draw()
{
  background('darkred')
  fill('yellow')
  lights()
  rotateX(frameCount * 0.02)
  rotateY(frameCount * 0.03)
  torus(width * 0.2, width * 0.07, 32, 32)
}

function keyPressed()
{
  if (key == " ")
  {
    const options = {
      units: "frames",
      delay: 0
    }
    saveGif("torus.gif", frames, options)
  }
}
```

Notes

The GIF will be downloaded somewhere; you might have to search for the file, but it is there somewhere.

Figure F4.7



Using mp4 capture

This is a nice way to generate an mp4 video of just the canvas, which means you don't have to crop it later. There are other formats available, but mp4 is pretty universal, and in this example, it will be the default.

Sketch F4.8 capture index.html

We need to add in the lines of code to make this library work. I have set it to mp4 because that works with my computer. You can choose other formats if you wish.

```
<!DOCTYPE html>
<html lang="en"><head>
    <script src="https://cdn.jsdelivr.net/npm/p5@2.2.0/lib/p5.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/p5.capture@1.6"></script>
    <script>
      P5Capture.setDefaultOptions({
        format: "mp4",
        framerate: 30,
        quality: 1,
        verbose: true,
        disableScaling: true,
      });
    </script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">
  </head>
  <body>
    <main>
    </main>
    <script src="sketch.js"></script>
</body></html>
```

🗒️ Notes

As you can see, there are other options in the list that you can set. These are the default, and for now, that does the job.

🌻 Challenge

Visit https://github.com/tapioca24/p5.capture to see more detailed information.

Sketch F4.9 capture main sketch

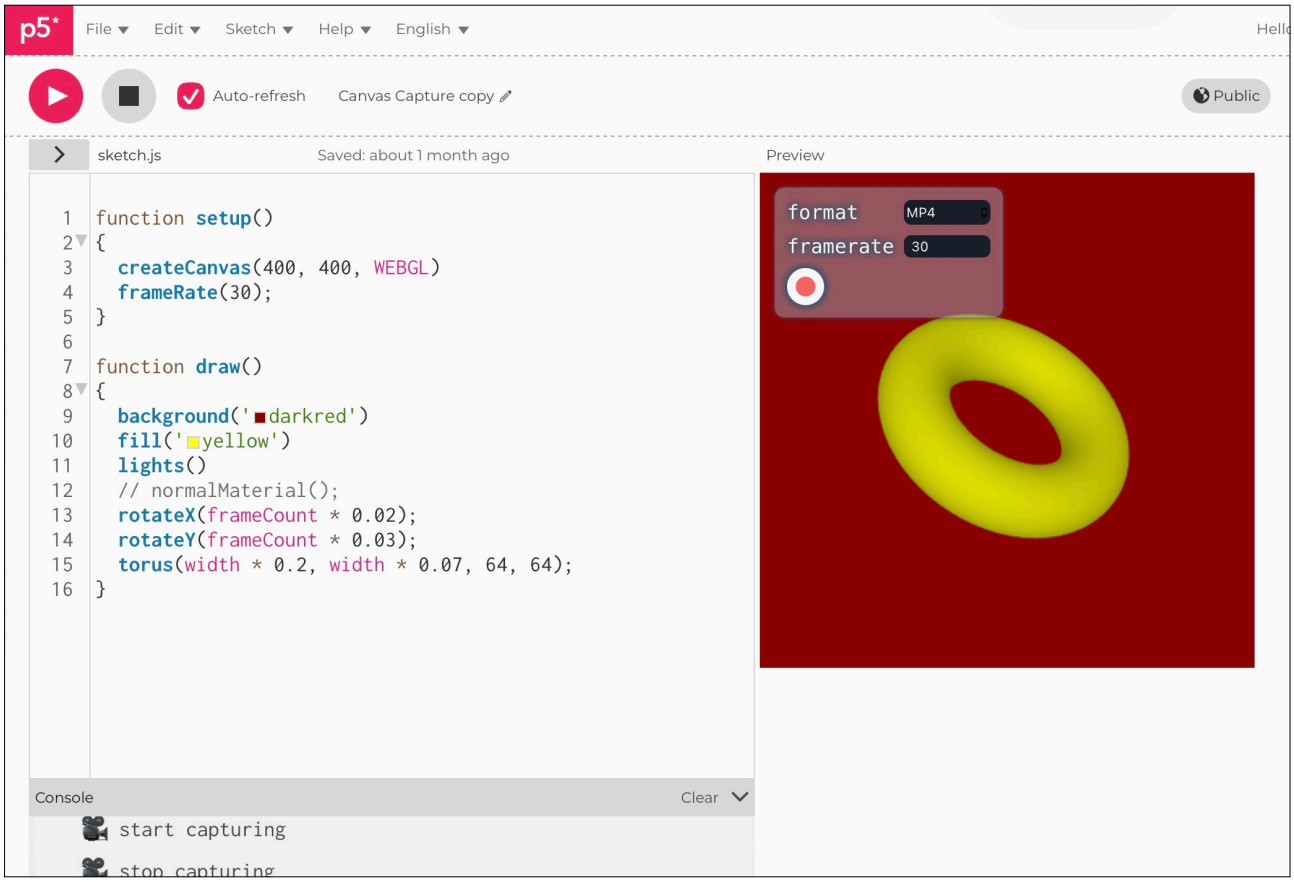
Using the same sketch as before.

```
function setup()
{
  createCanvas(400, 400, WEBGL)
  frameRate(30)
}

function draw()
{
  background('darkred')
  fill('yellow')
  lights()
  rotateX(frameCount * 0.02)
  rotateY(frameCount * 0.03)
  torus(width * 0.2, width * 0.07, 32, 32)
}
```

🗒️ Notes

You will get a small box on the canvas; it won't be in the final video. You can change the format, although I recommend doing that in the options setup in the `index.html` file.

Figure F4.9


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Canvas Capture copy
Public
sketch.js
Saved: about 1 month ago
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
frameRate(30);
}
function draw()
{
background('darkred')
fill('yellow')
lights()
// normalMaterial();
rotateX(frameCount * 0.02);
rotateY(frameCount * 0.03);
torus(width * 0.2, width * 0.07, 64, 64);
}
format
MP4
framerate
30
Console
Clear
start capturing
stop capturing

Sketch F4.10 playing the video

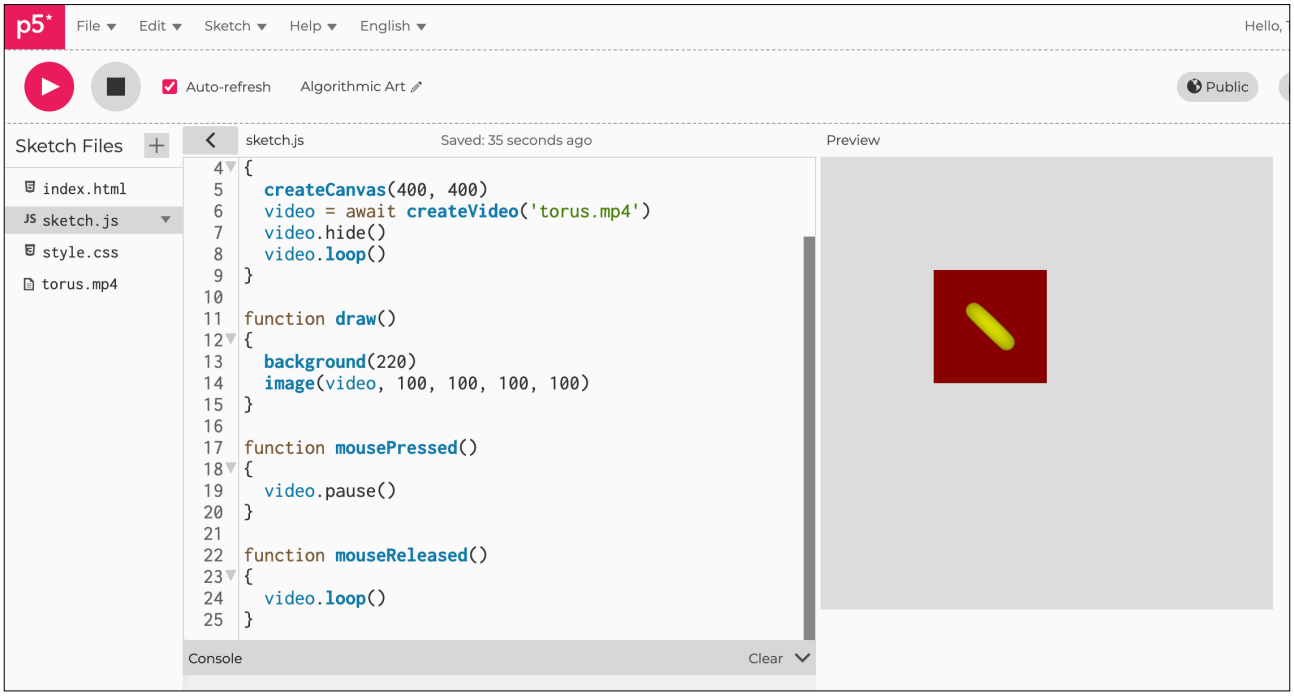
Once you have saved the video, you can include it in a sketch. You upload it just like any other file. You might want to rename it, as I have done. We use `async` and `await` so that we are sure that the video is loaded before we use it. What I have shown below is just a suggestion.

```
let video

async function setup()
{
  createCanvas(400, 400)
  video = await createVideo('torus.mp4')
  video.hide()
  video.loop()
}

function draw()
{
  background(220)
  image(video, 100, 100, 100, 100)
}

function mousePressed()
{
  video.pause()
}

function mouseReleased()
{
  video.loop()
}
```

Notes

You will most likely use the `mp4` on social media or a website so that you can promote your artwork.

Figure F4.10


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
Public
Sketch Files
index.html
sketch.js
style.css
torus.mp4
sketch.js
Saved: 35 seconds ago
Preview
4 {
5 createCanvas(400, 400)
6 video = await createVideo('torus.mp4')
7 video.hide()
8 video.loop()
9 }
10
11 function draw()
12 {
13 background(220)
14 image(video, 100, 100, 100, 100)
15 }
16
17 function mousePressed()
18 {
19 video.pause()
20 }
21
22 function mouseReleased()
23 {
24 video.loop()
25 }
Console
Clear