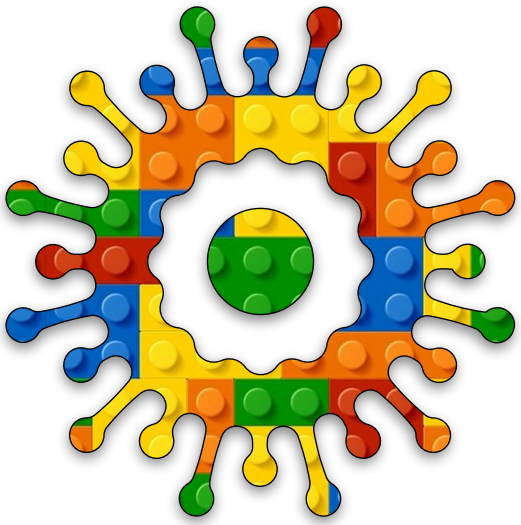
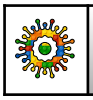
Making Games
Module A
Unit #3
flappy bird





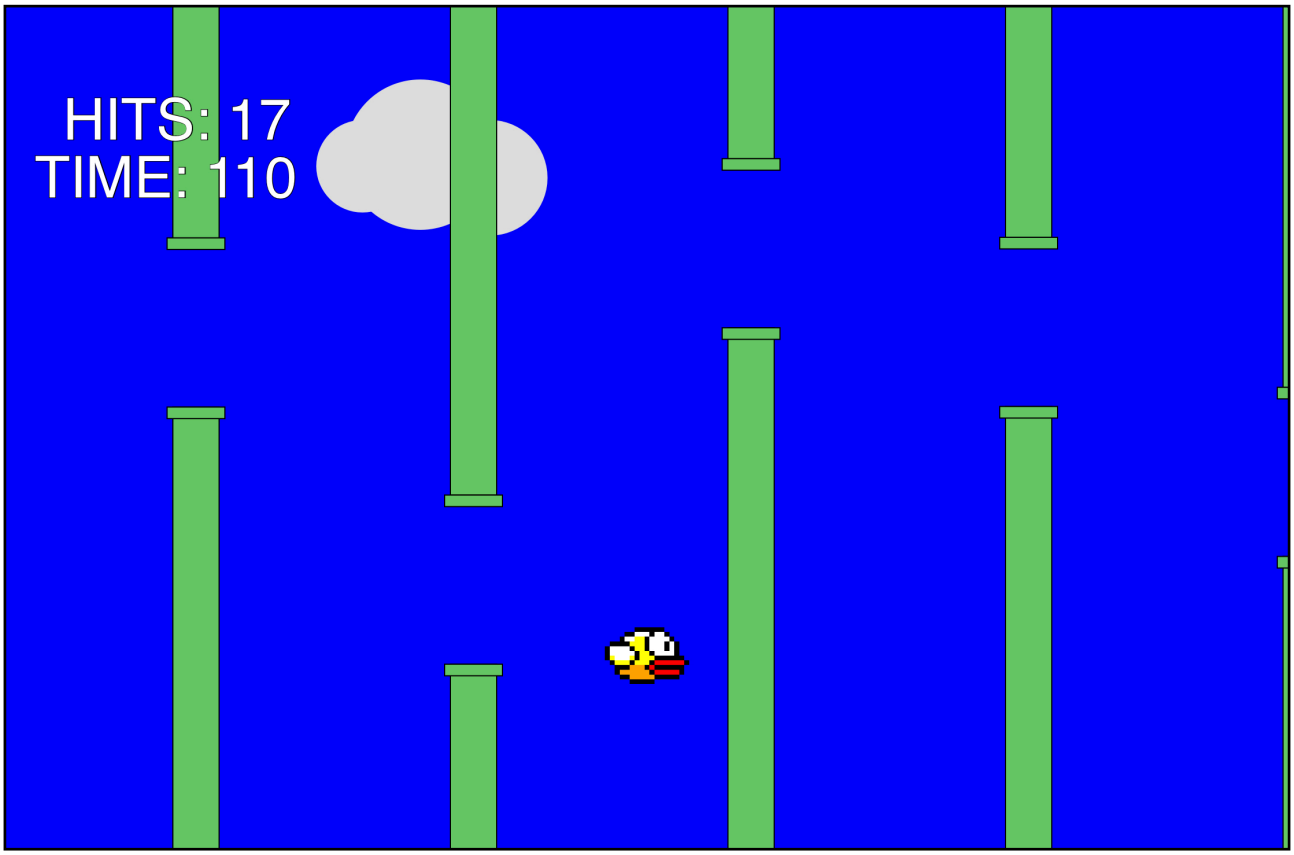
Content

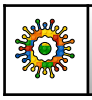
Module A Unit #3 Flappy Bird

Introduction

If you have never played Flappy Bird, don't worry, it is the simplest game ever and it is probably one of the most popular. It is simple, it is fun, and it is addictive. It is a great place to start. If you Google it, you will find some YouTube clips of people playing it.

We are going to make a version of Flappy Bird in JavaScript using the p5.js library. This is free software for use to create art or games and even teachable machines (AI). It is web-based, so you code directly into the browser with no need to download any software.





Let's get going...

I have assumed that you will be familiar with the basics of using p5.js. If you aren't, then I recommend that you cover at least the first couple of modules; otherwise, you might struggle a bit at first. However, everything is explained as you go along.

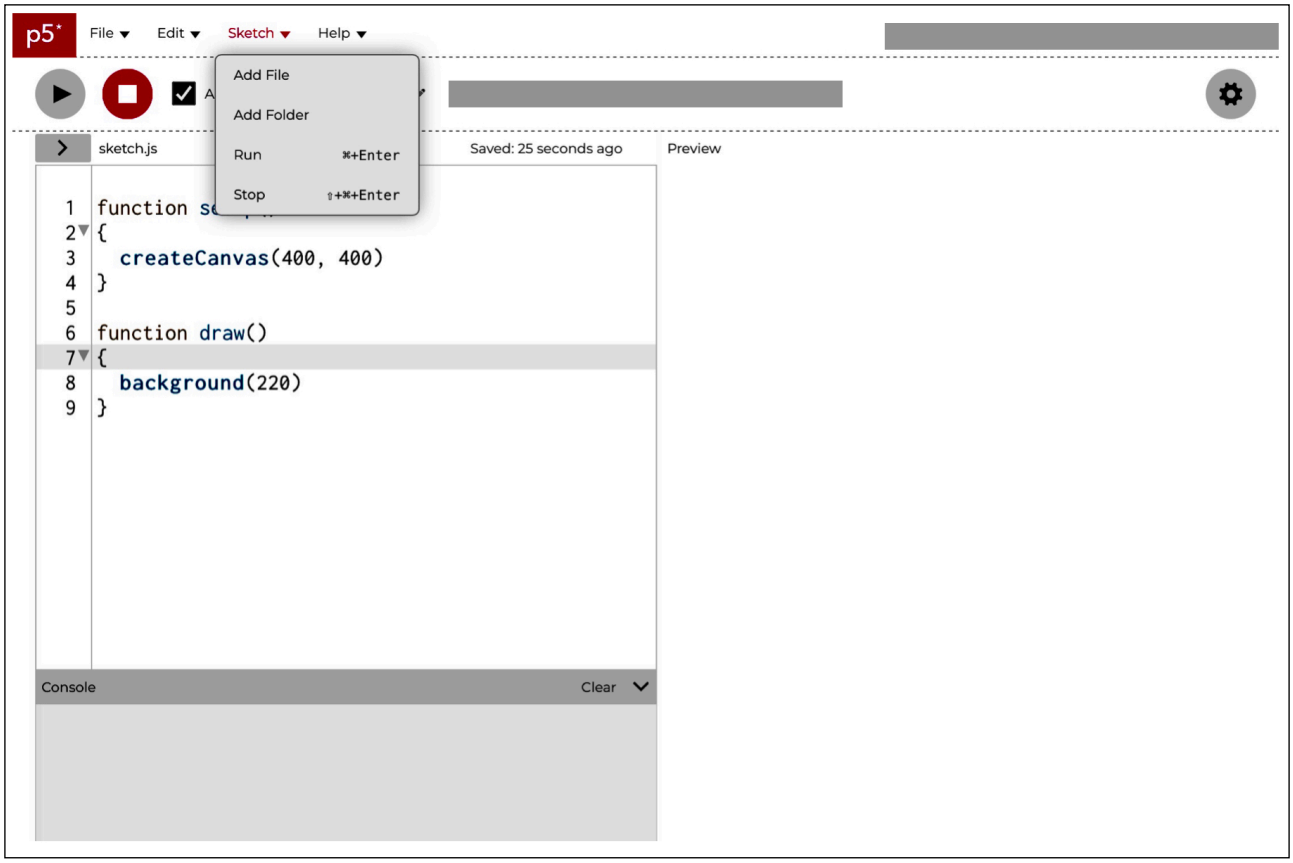
A. Create two new files called bird.js and pipe.js. The next few pages will take you carefully through the process.

B. Open up the html.index file and add them to the list below the sketch.js. You should have three JavaScript files: sketch.js, bird.js, and pipe.js.

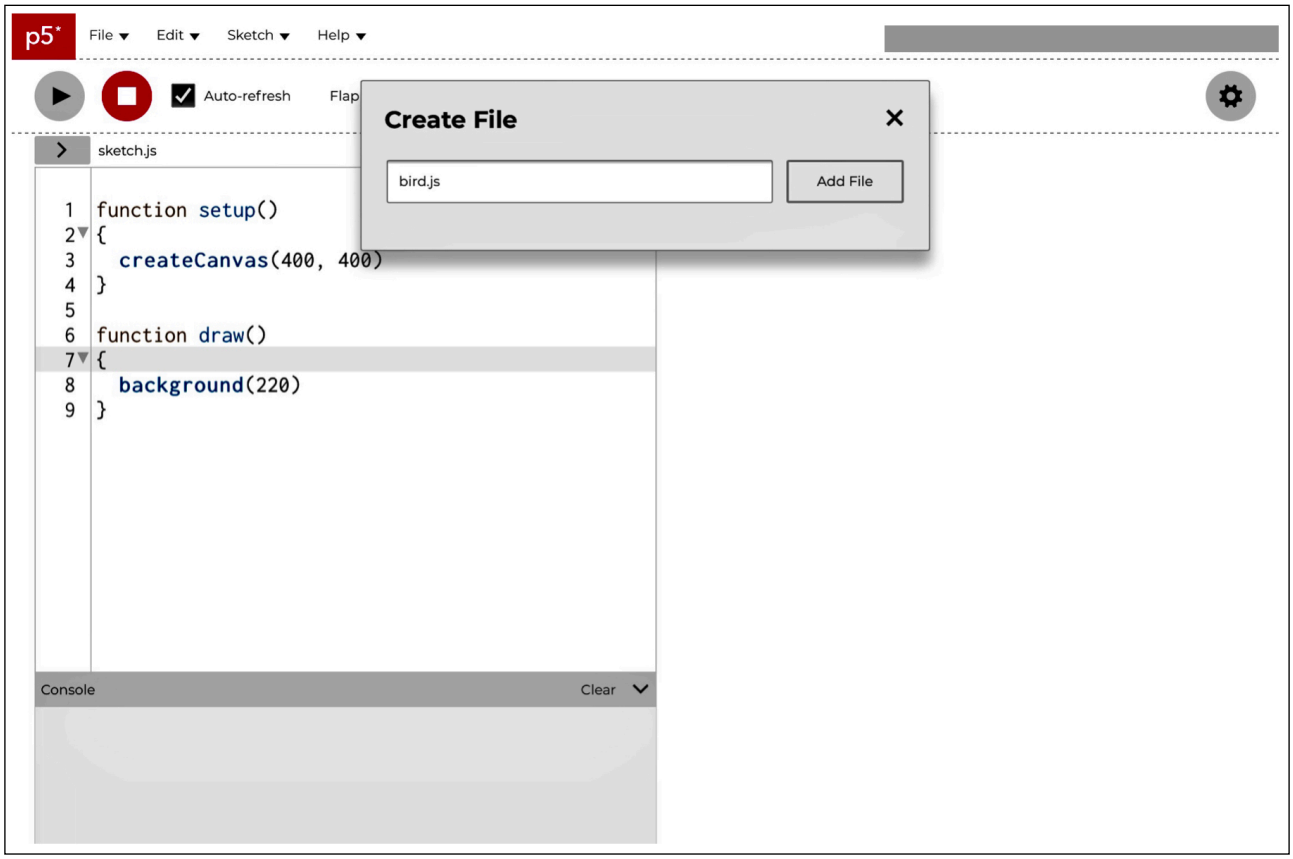
C. Start typing in the code as I show you, press the run (play) button to see the impact of the code (if any!). You may have to update more than one file, e.g. a line of code in sketch.js and bird.js.

D. If you get to the very end, I suggest that you customise it for yourself in any way that your own creativity suggests.

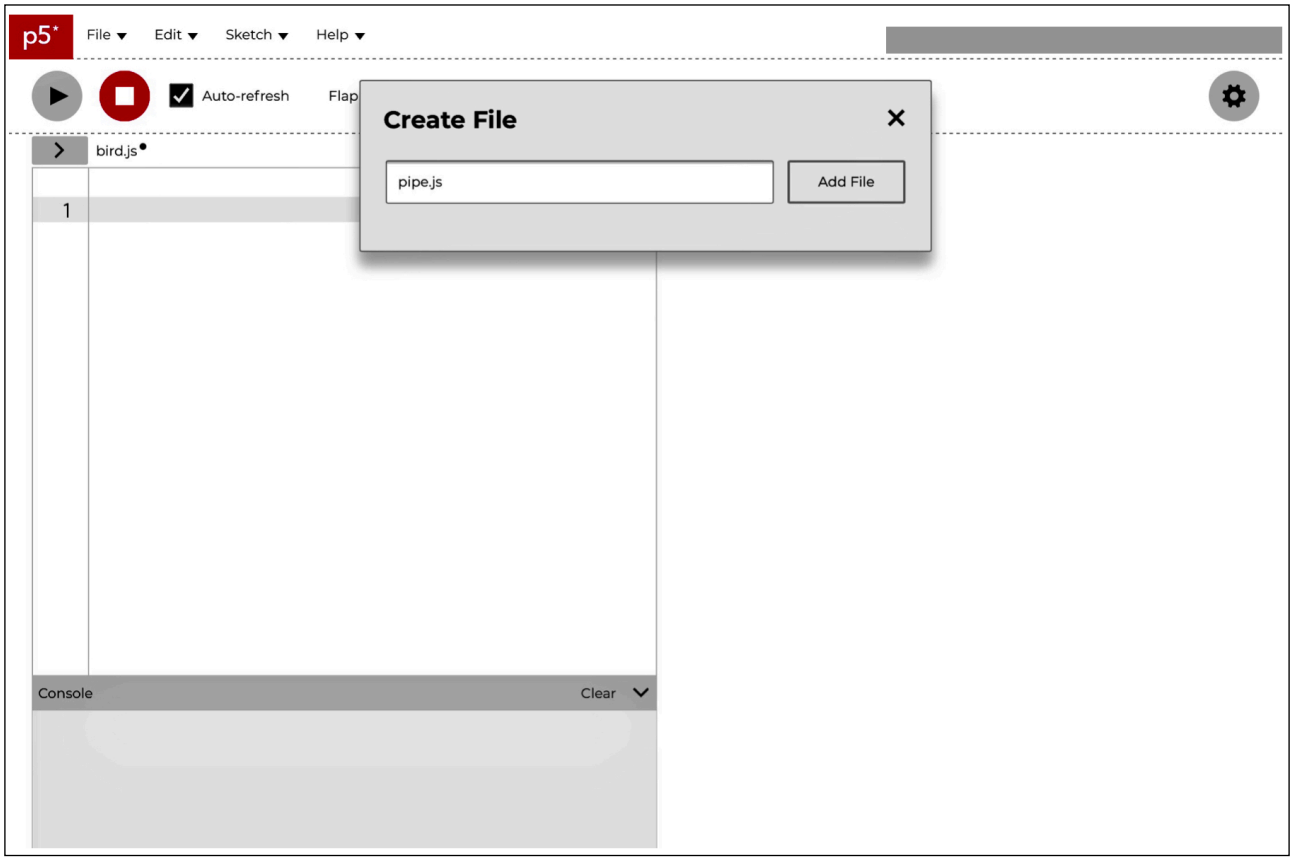
Click Sketch to add file



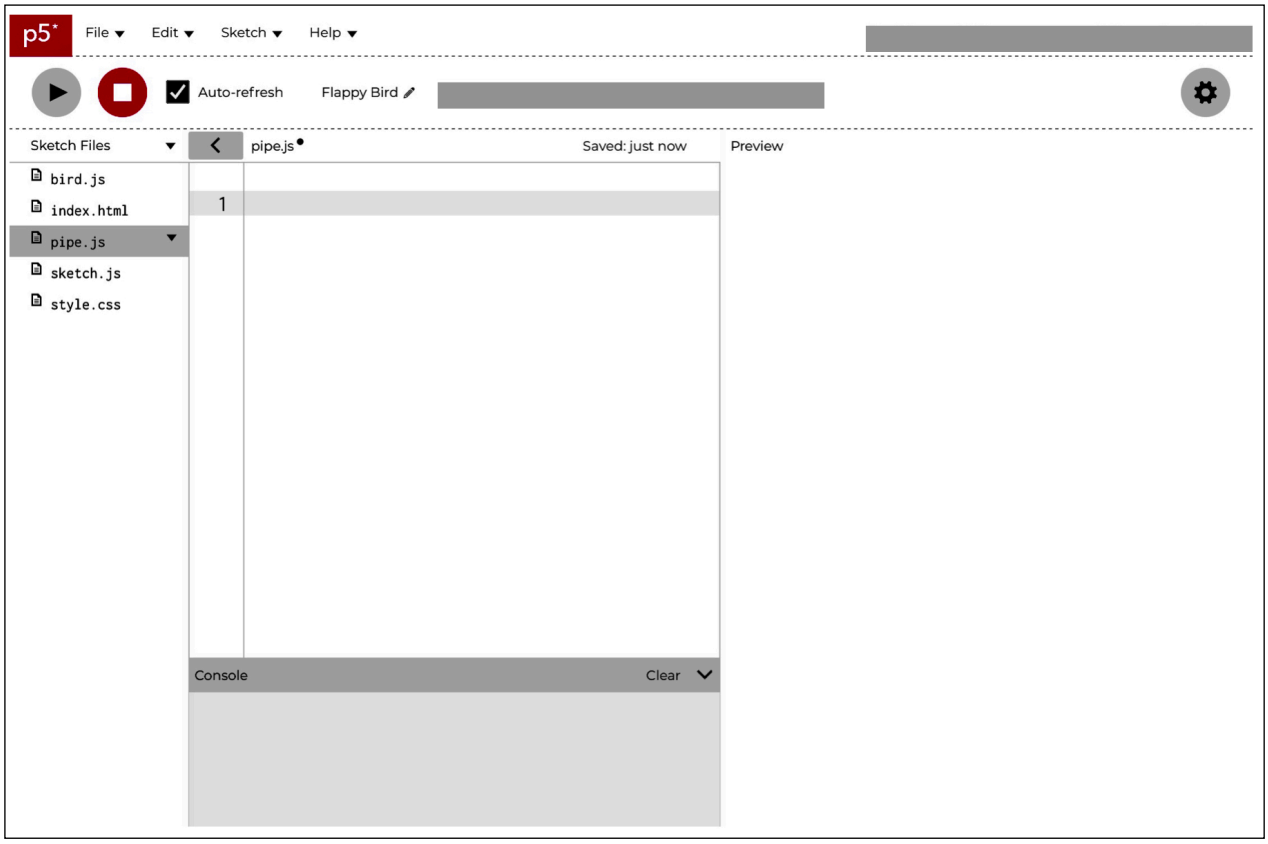
Give it the name `bird.js`



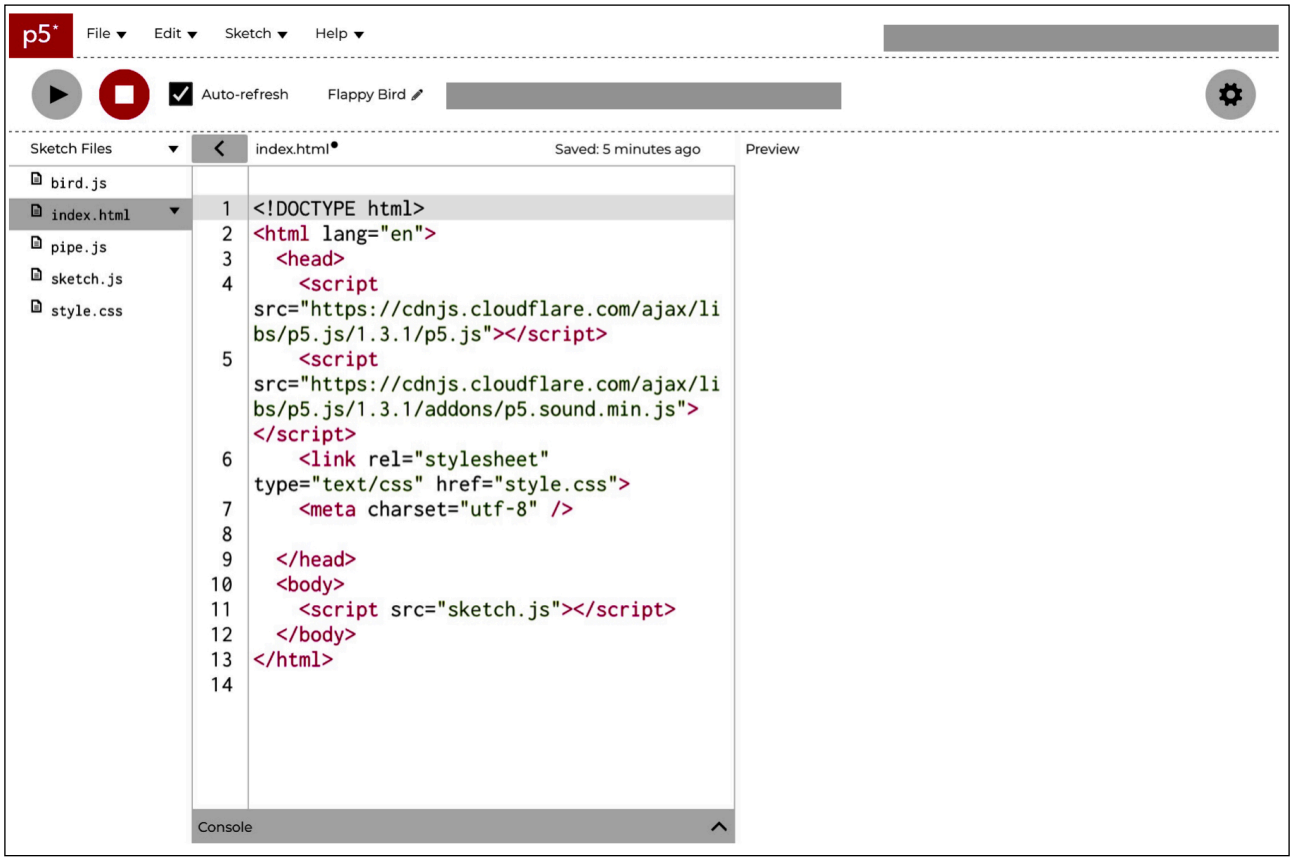
Same for `pipe.js`



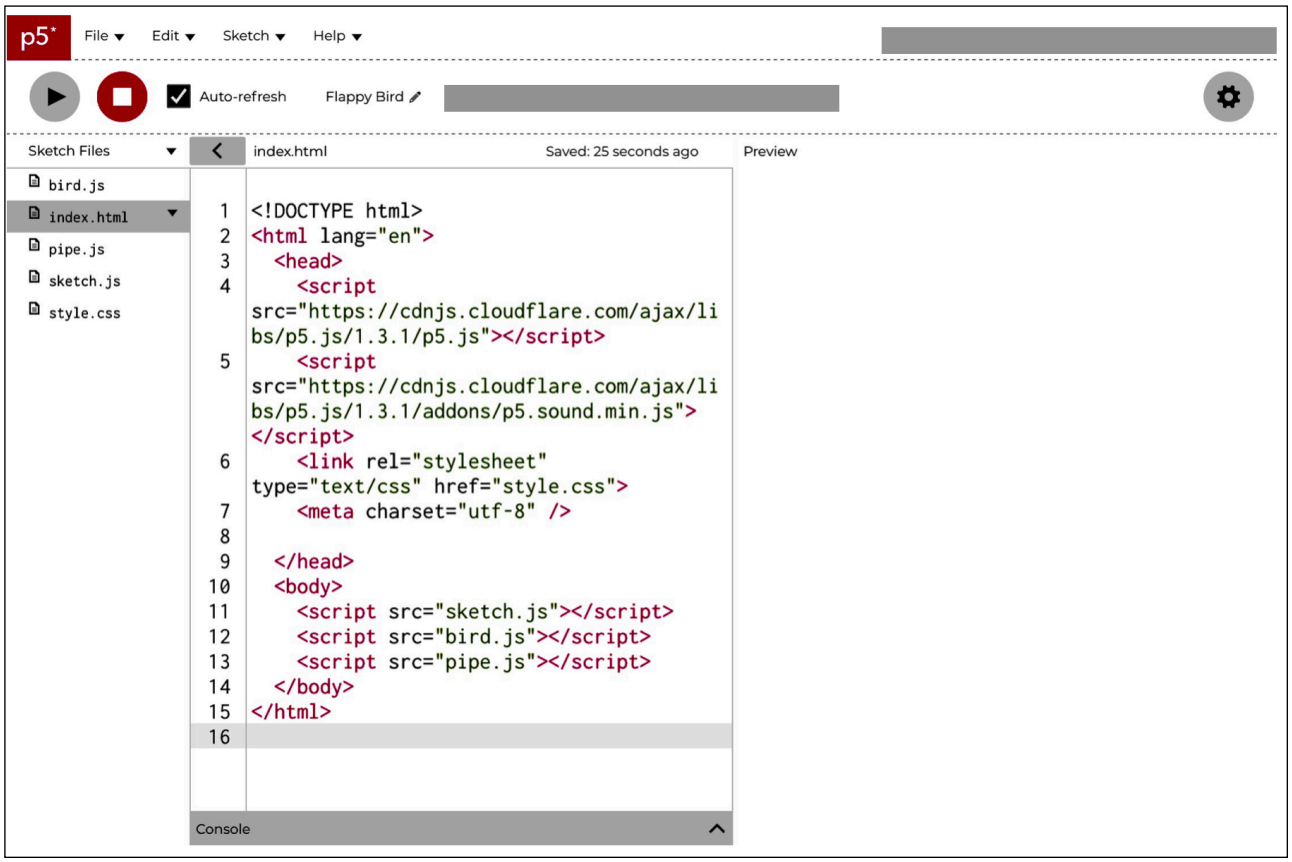
Open the side panel to see the `files` are there

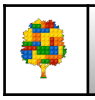


Open the index.html file



Add the two lines of code after `sketch.js`





These are the files that are needed...

`index.html`
The following files need to be referenced here, always very easy to forget:

`sketch.js`
This already exists by default and doesn't need adding.

`bird.js`
This one needs to be added as a file and referenced in the index.html file.

`pipe.js`
This one also needs to be added as a file and referenced in the index.html file.

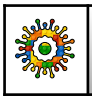
We will start by adding these files. Click on `Sketch` and click on `Add File`. It will ask you to name the file, type in `bird.js` and it will create an empty file called `bird.js`. Save it, now add `pipe.js` and save what you have done.

For this next bit, we need to add these files to the `index.html` file.

If your computer has given it a capital B for bird, that is OK, but when you add it to the `HTML` file, you will need to do the same; it is case-sensitive. I suggest that you always use lowercase as a default.

We are now going to take a peek at the files we have. To do this, we click on the little grey arrow button. The one in the illustration below. Then click on the `index.html`, this will bring up the code that is there by default. There is a lot of stuff there that you may or may not be familiar with. If not, please don't worry about what it all means; at this stage, you really don't need to understand it.

It may seem a bit busy in the `HTML.index` file, but if you locate the `sketch.js` file and copy and paste it twice and change the name of one to bird.js and pipe.js as seen in the sketch below.



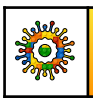
Creating the files

You will create an extra two files, making it four altogether. These will be colour-coded for easy identification. In essence, you are making one sketch with several components. Each of the JavaScript files will be its own class. This is good practice for using classes.

index.html	HTML file that will need additional lines of code
sketch.js	main body of the sketch
bird.js	bird class will sit here
pipe.js	pipe class will sit here

Any changes to the previous sketch will be highlighted in blue as shown below, to help you speed up the process. It may be that some lines of code have been removed (replaced), so you will need to check through the lines of code to see what has been removed. I will try to alert you to anything removed, but check anyway.

File Name
Code to change highlighted

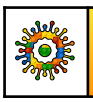


Sketch A3.1 index.html default

The `index.html` file, this is what the default should look like...

index.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <script src="https://cdn.jsdelivr.net/npm/p5@1.11.10/lib/
p5.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/p5@1.11.10/lib/
addons/p5.sound.min.js"></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>
```

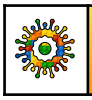


Sketch A3.2 index.html setup

We are now going to add in the bird.js and the pipe.js. It is quite easy to cut and paste and then edit the names. The additional text is highlighted.

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://cdn.jsdelivr.net/npm/p5@1.11.10/lib/
p5.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/p5@1.11.10/lib/
addons/p5.sound.min.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8" />

  </head>
  <body>
    <main>
    </main>
    <script src="sketch.js"></script>
    <script src="bird.js"></script>
    <script src="pipe.js"></script>
  </body>
</html>
```



Sketch A3.3 creating a new bird

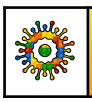
❗ Creating a new bird in sketch.js.
We will be shifting between all three files. You could just have everything in one file (`sketch.js`), but it will get very long very quickly. The top tab in the code below indicates which file we are working in; just be aware.

sketch.js

```
let bird

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
}

function draw()
{
  background(220)
  bird.show()
}
```

Notes

You will get an error message if you run it because we haven't created a Bird class yet. Next, you want to go to the side tab (click on the arrow next to `sketch.js`) and you will see a side window with the names of the sketches; select `bird.js` by clicking on it.



Sketch A3.4 a bird class

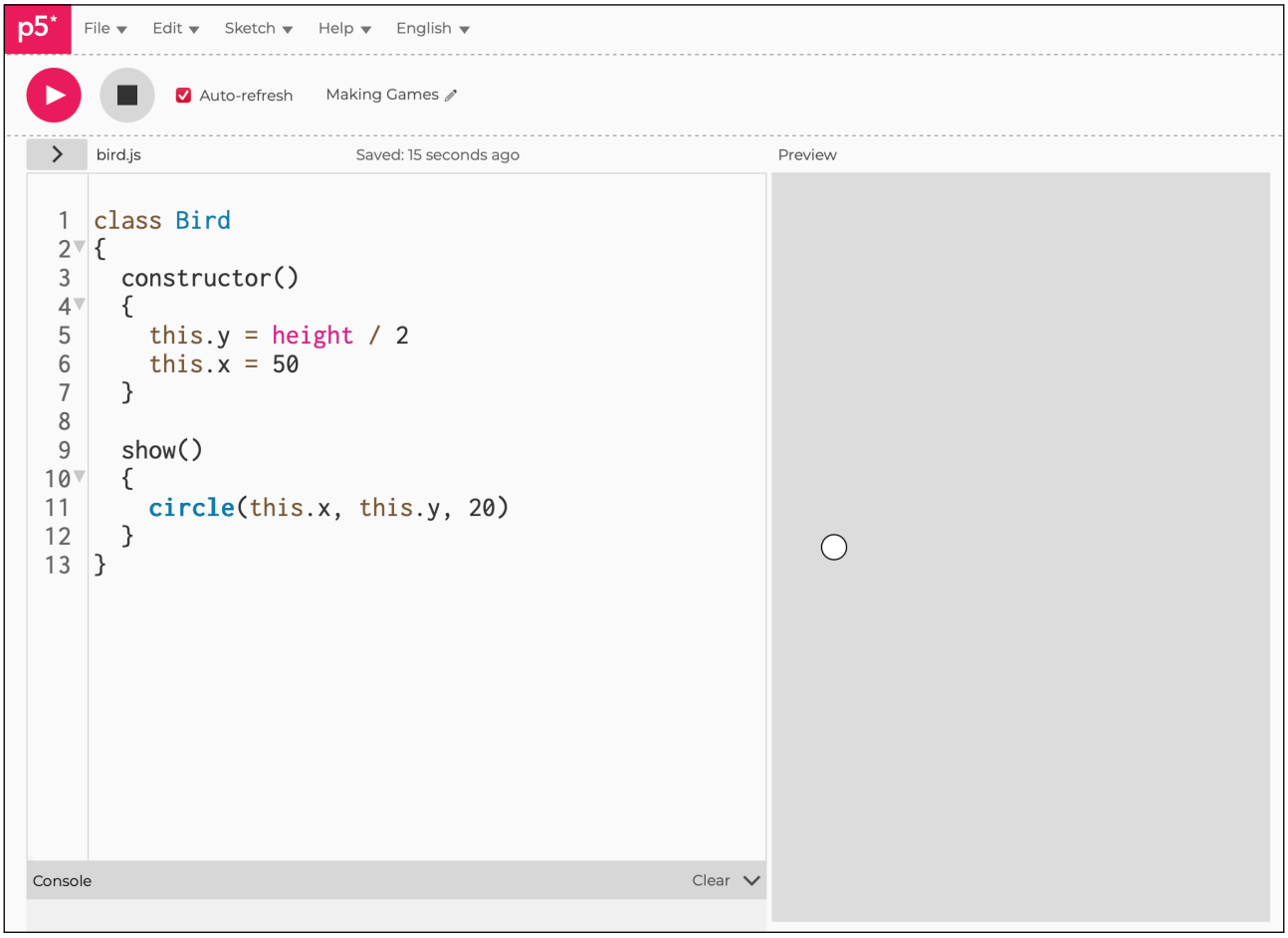
❗ bird.js

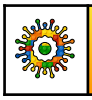
The bird at this stage is just a circle; you will be creating a more realistic bird (well, not a real bird!) later as a PNG. A `constructor()` function and `show()` function are created for the bird (a simple circle).

bird.js

```
class Bird
{
  constructor()
  {
    this.y = height / 2
    this.x = 50
  }

  show()
  {
    circle(this.x, this.y, 20)
  }
}
```

Figure A3.3


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
bird.js
Saved: 15 seconds ago
Preview
class Bird
{
constructor()
{
this.y = height / 2
this.x = 50
}
show()
{
circle(this.x, this.y, 20)
}
}
Console
Clear



Sketch A3.5 adding some gravity

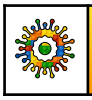
The bird will be given some gravity; this is a value that you can experiment with. A value of `0.5` is a reasonable value, but try others to see the difference. The velocity starts at zero, but it increases by the gravity on each iteration. Hence, it falls to the ground accelerating. This is to give the bird some gravity and a velocity so the bird is pushed down.

bird.js

```
class Bird
{
  constructor()
  {
    this.y = height/2
    this.x = 50
    this.gravity = 0.5
    this.velocity = 0
  }

  show()
  {
    circle(this.x, this.y, 20)
  }

  move()
  {
    this.velocity += this.gravity
    this.y += this.velocity
  }
}
```

🗒️ Notes

Nothing will happen just yet, we need to include the `move()` function in `sketch.js`.



Sketch A3.6 making gravity work

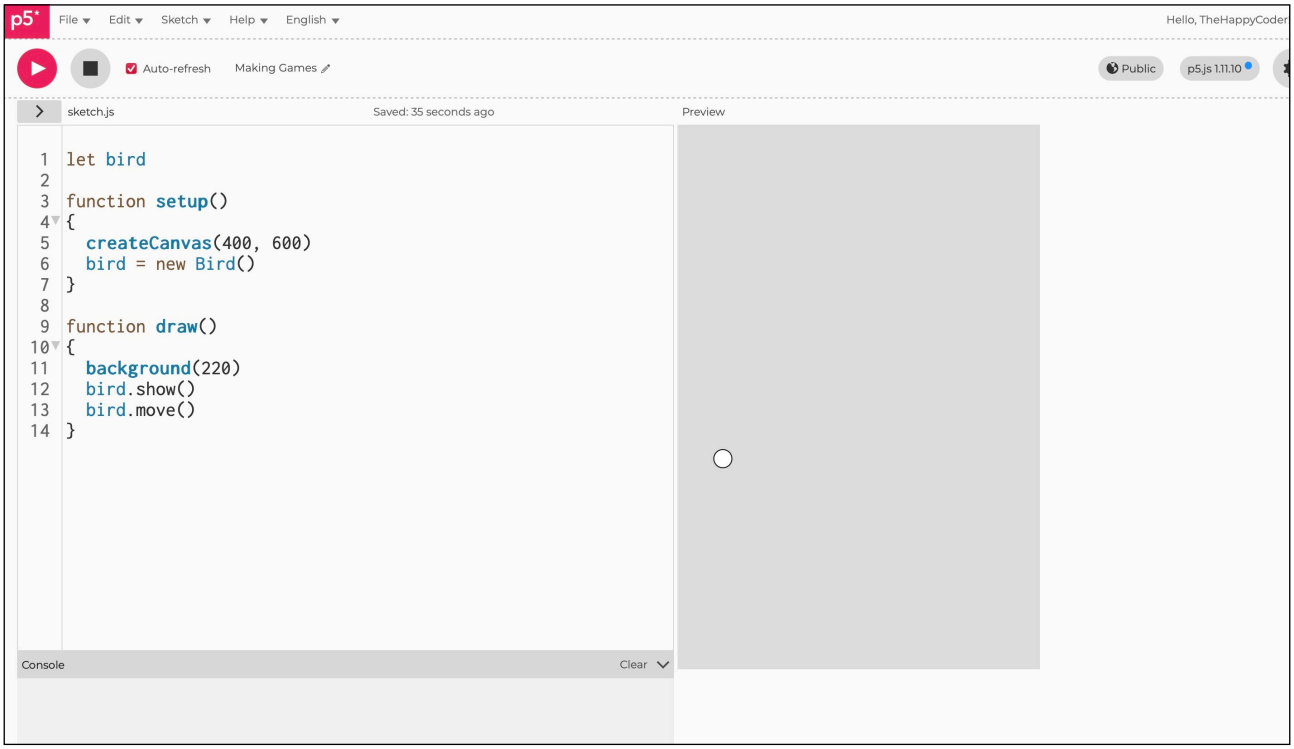
❗ back to the main sketch

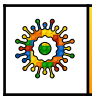
Now to add the `move()` function into `sketch.js`, however, the bird just falls through the ground, not what we want.

sketch.js

```
let bird

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
}
```

Figure A3.6


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
sketch.js
Saved: 35 seconds ago
Preview
let bird
function setup()
{
createCanvas(400, 600)
bird = new Bird()
}
function draw()
{
background(220)
bird.show()
bird.move()
}
Console
Clear



Sketch A3.7 upper and lower limits

❗ returning to bird.js file

Giving the bird upper and lower limits of movement in `bird.js`.

bird.js

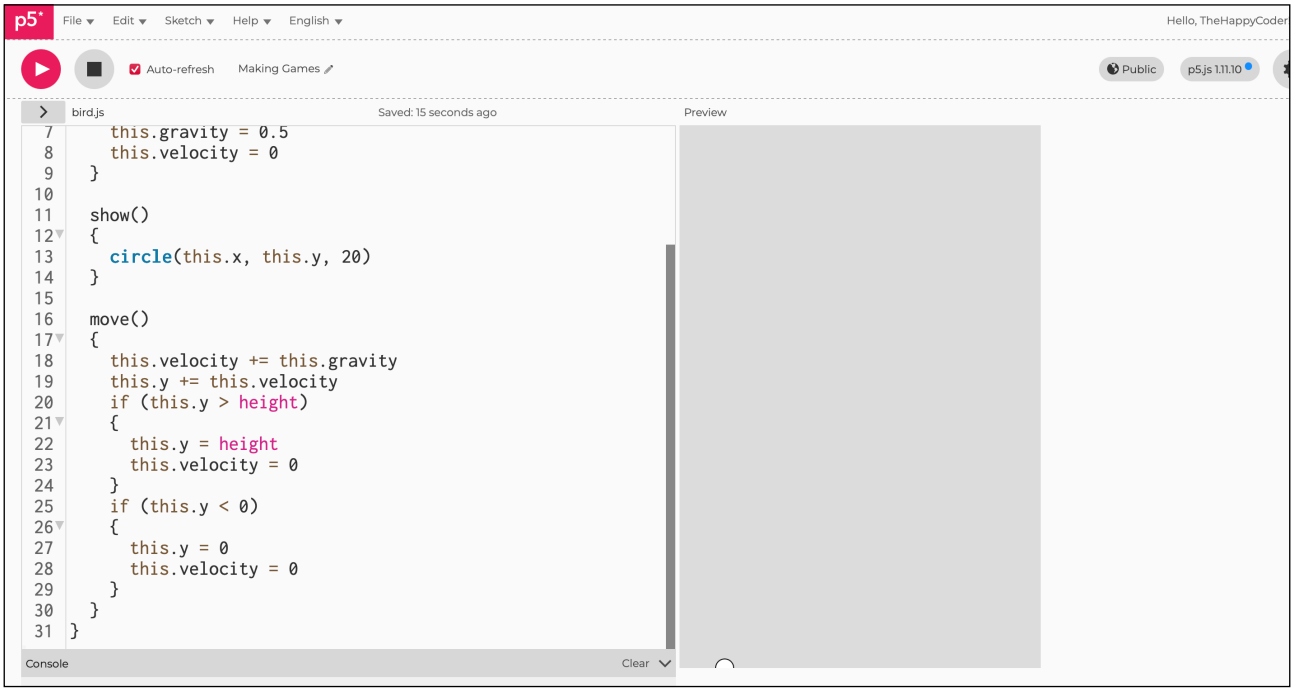
```
class Bird
{
  constructor()
  {
    this.y = height/2
    this.x = 50
    this.gravity = 0.5
    this.velocity = 0
  }

  show()
  {
    circle(this.x, this.y, 20)
  }

  move()
  {
    this.velocity += this.gravity
    this.y += this.velocity
    if (this.y > height)
    {
      this.y = height
      this.velocity = 0
    }
    if (this.y < 0)
    {
```

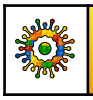
```
      this.y = 0
      this.velocity = 0
    }
  }
}
```

🗒️ Notes

This has the effect of keeping it on the screen

Figure A3.7


p5
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
bird.js
Saved: 15 seconds ago
Preview
this.gravity = 0.5
this.velocity = 0
}
show()
{
circle(this.x, this.y, 20)
}
move()
{
this.velocity += this.gravity
this.y += this.velocity
if (this.y > height)
{
this.y = height
this.velocity = 0
}
if (this.y < 0)
{
this.y = 0
this.velocity = 0
}
}
Console
Clear



Sketch A3.8 tapping the spacebar

❗ and back to sketch.js

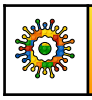
We want to move the bird upwards by tapping the spacebar. We add another function called `keyPressed()`.

sketch.js

```
let bird

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

Notes

If you try it you will get an error message, we have to write the function `up()` in `bird.js` to take effect, so lets hop on over there to do just that.



Sketch A3.9 giving it some lift

❗ hopping over to bird.js

The `up()` function in bird.js creates an upwards force called lift. Now when you tap the spacebar, it moves upwards.

❗ Remember to click on the canvas first!

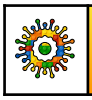
bird.js

```
class Bird
{
  constructor()
  {
    this.y = height/2
    this.x = 50
    this.gravity = 0.5
    this.lift = -10
    this.velocity = 0
  }

  show()
  {
    circle(this.x, this.y, 20)
  }

  up()
  {
    this.velocity += this.lift
  }

  move()
  {
    this.velocity += this.gravity
```

```
    this.y += this.velocity
    if (this.y > height)
    {
      this.y = height
      this.velocity = 0
    }
    if (this.y < 0)
    {
      this.y = 0
      this.velocity = 0
    }
  }
}
```

🗒️ Notes

After you have typed the code, click on the canvas and tap the spacebar; the circle should go up and fall back to the ground. Keep pressing the space bar to get a feel for how it moves.



Sketch A3.10 air resistance

This can be improved by adding some air resistance.

bird.js

```
class Bird
{
  constructor()
  {
  this.y = height/2
  this.x = 50
  this.gravity = 0.5
  this.lift = -10
  this.velocity = 0
  }

  show()
  {
    circle(this.x, this.y, 20)
  }

  up()
  {
    this.velocity += this.lift
  }

  move()
  {
    this.velocity += this.gravity
    this.velocity *= 0.9
    this.y += this.velocity
    if (this.y > height)
```

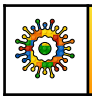
```
    {
      this.y = height
      this.velocity = 0
    }
    if (this.y < 0)
    {
      this.y = 0
      this.velocity = 0
    }
  }
}
```

🗒️ Notes

As you tap the spacebar, it moves more sluggishly, as if there is air resistance.

🌻 Challenge

You can play around with this figure to see what difference it makes.



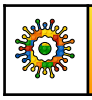
Sketch A3.11 the pipes

❗ go to the pipe.js file

Now we need the pipes. We will draw a rectangle from the top to some random position half the height, and another rectangle from some random height (up to half the height) to the bottom. If you sketch it out, it does make sense, honestly. We are going to make it move from the right to the left.

pipe.js

```
class Pipe
{
  constructor()
  {
    this.top = random(0, height/2)
    this.bottom = random(height/2)
    this.x = width
    this.w = 20
    this.speed = 1
  }

  show()
  {
    fill(255)
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
  }

  move()
  {
    this.x -= this.speed
  }
}
```



Sketch A3.12 an array of pipes

❗ back to the main sketch

Next, let's create an array of pipes so that they move constantly across the screen as if the bird is flying through them. This produces one set of pipes.

sketch.js

```
let bird
let pipes = []

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
  pipes.push(new Pipe())
}

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

function keyPressed()
{
  if (key === ' ')
  {
```

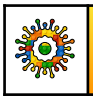
```
    bird.up()
  }
}
```

🗒️ Notes

You should get two pipes moving across the canvas from right to left and then out of picture

Figure A3.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Making Games
Public
p5.js 1.11.10
sketch.js
Saved: 25 seconds ago
Preview
{
createCanvas(400, 600)
bird = new Bird()
pipes.push(new Pipe())
}
function draw()
{
background(220)
bird.show()
bird.move()
for (let i = 0; i < pipes.length; i++)
{
pipes[i].show()
pipes[i].move()
}
}
function keyPressed()
{
if (key === ' ')
{
bird.up()
}
}
Console
Clear



Sketch A3.13 the speed of the pipes

Now, having a new set every 100 frames (between 1 and 2 seconds), we can now delete: `pipes.push(new Pipe())` in `setup()`.

sketch.js

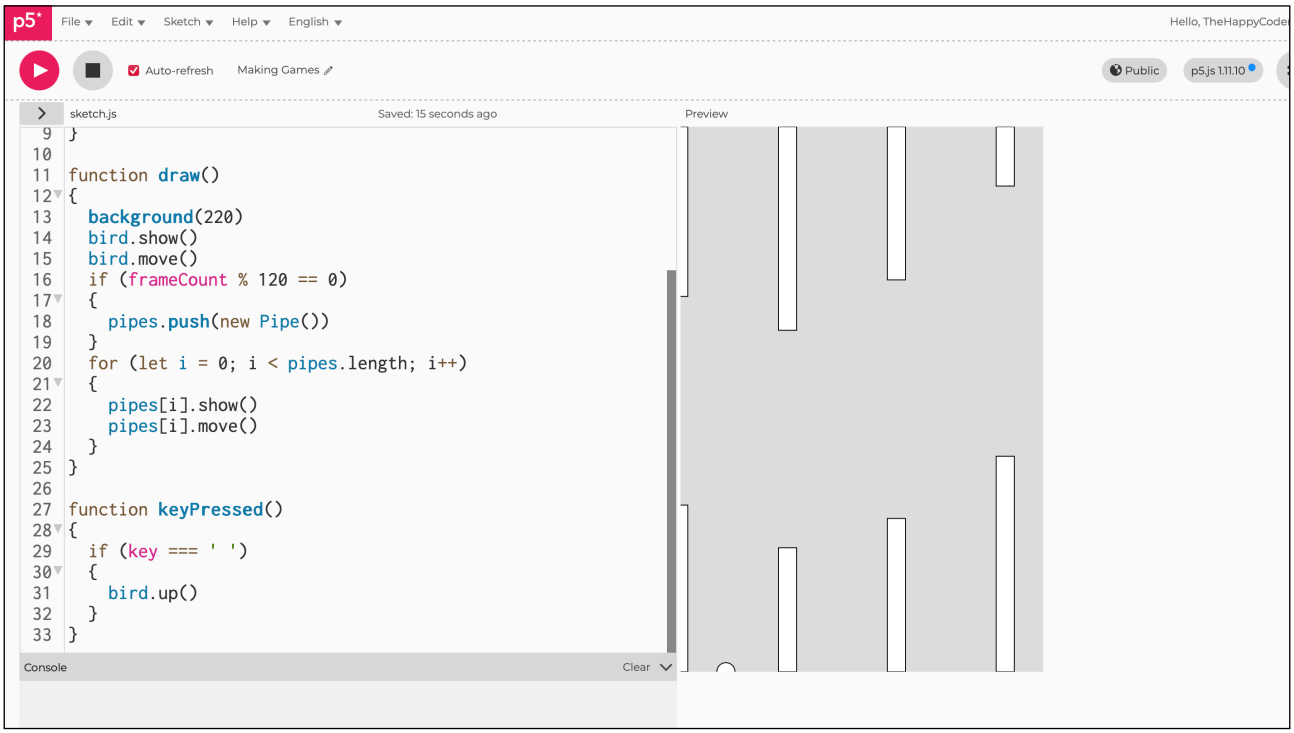
```
let bird
let pipes = []

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
  // pipes.push(new Pipe())
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
  for (let i = 0; i < pipes.length; i++)
  {
    pipes[i].show()
    pipes[i].move()
  }
}
```

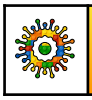
```
function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

🗒️ Notes

You should now have lots of pipes flowing across the canvas, see if you can get the circle through each of the gaps in the pipes!

Figure A3.13


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
sketch.js
Saved: 15 seconds ago
Preview
}
function draw()
{
background(220)
bird.show()
bird.move()
if (frameCount % 120 == 0)
{
pipes.push(new Pipe())
}
for (let i = 0; i < pipes.length; i++)
{
pipes[i].show()
pipes[i].move()
}
}
function keyPressed()
{
if (key === ' ')
{
bird.up()
}
}
Console
Clear



Sketch A3.14 deleting the pipes

Deleting the pipes once they wander off the screen. To achieve this, we create a function called `offscreen()`. It checks through the array backwards; hence, see replacement: for `(let i = pipes.length - 1; i >= 0; i--)`.

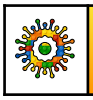
sketch.js

```
let bird
let pipes = []

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
  for (let i = pipes.length - 1; i >= 0; i--)
  {
    pipes[i].show()
    pipes[i].move()
    if (pipes[i].offscreen())
    {
      pipes.splice(i, 1)
```

```
    }
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

Notes

You should get an error message when you try to run this sketch because the `offscreen()` function hasn't been created yet.



Sketch A3.15 deleting the pipes

! off to pipe.js.
Then the offscreen function detection in `pipe.js`.

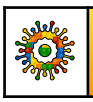
pipe.js

```
class Pipe
{
  constructor()
  {
    this.top = random(0, height/2)
    this.bottom = random(height/2)
    this.x = width
    this.w = 20
    this.speed = 1
  }

  show()
  {
    fill(255)
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
  }

  move()
  {
    this.x -= this.speed
  }

  offscreen()
  {
    if (this.x < -this.w)
```

```
    {
      return true
    }
    else
    {
      return false
    }
  }
}
```

🗒️ Notes

You can use `console.log()` to check how long the pipes array is, e.g. `console.log(pipes.length)` at the end of the function `draw()` in `sketch.js`, otherwise nothing new to see here.



Sketch A3.16 collision

❗ back to sketch.js.

Now the collision! Checking to see if the bird hit the pipes. We need to write a function called `hit()`. The `console.log('hit')` is just to check that it works.

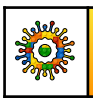
sketch.js

```
let bird
let pipes = []

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
  pipes.push(new Pipe())
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
  for (let i = pipes.length - 1; i >= 0; i--)
  {
    pipes[i].show()
    pipes[i].move()
    if (pipes[i].hits(bird))
    {
```

```
      console.log('HIT')
    }
    if (pipes[i].offscreen())
    {
      pipes.splice(i, 1)
    }
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

Notes

You should get an error message at this point because we haven't added `hits()` function to `pipe.js`.



Sketch A3.17 the x factor

❗ In pipes.

We create a function with an argument bird object. This tests whether it is in line with the x-value of the pipe and between the two ends of the pipe (the y-value).

pipe.js

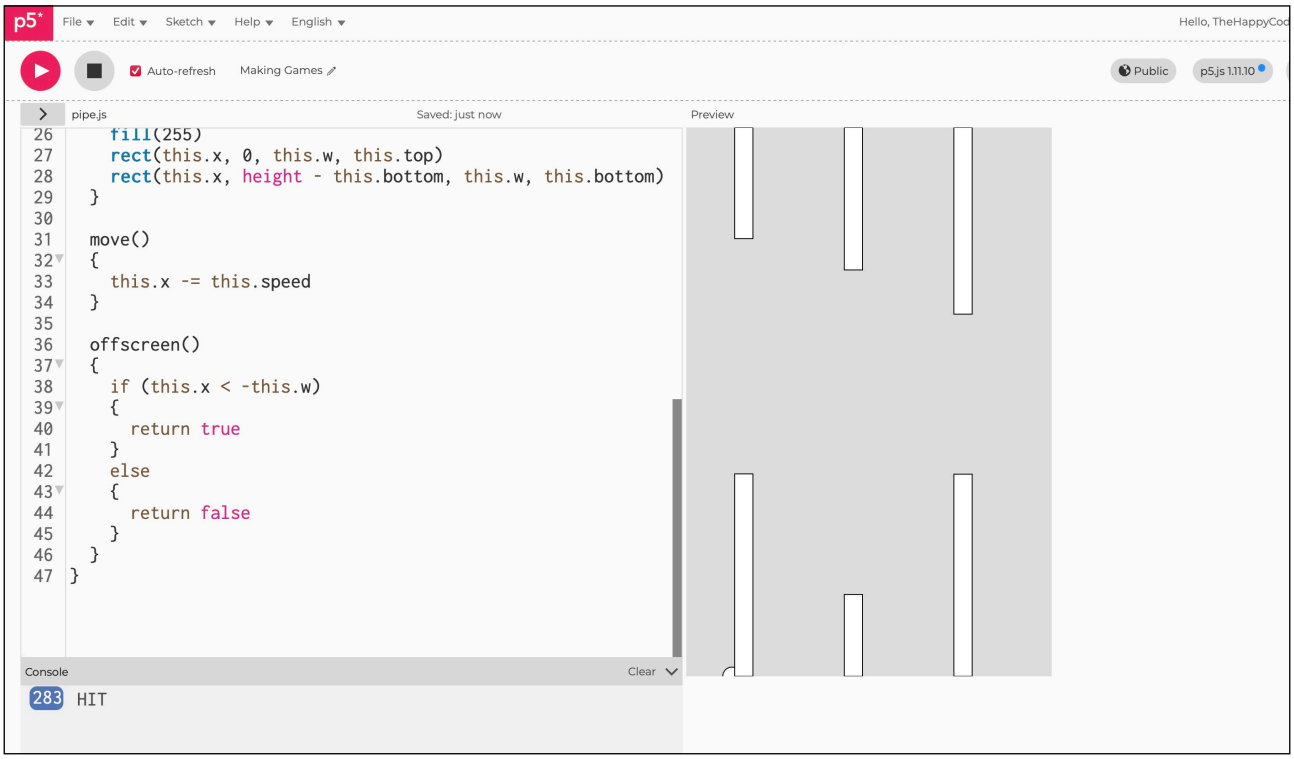
```
class Pipe
{
  constructor()
  {
    this.top = random(0, height/2)
    this.bottom = random(height/2)
    this.x = width
    this.w = 20
    this.speed = 1
  }

  hits(bird)
  {
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x > this.x && bird.x < this.x + this.w)
      {
        return true
      }
    }
    return false
  }

  show()
  {
```

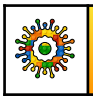
```
    fill(255)
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
  }

  move()
  {
    this.x -= this.speed
  }

  offscreen()
  {
    if (this.x < -this.w)
    {
      return true
    }
    else
    {
      return false
    }
  }
}
```

📝 Notes

You should be getting a message in the console every time the circle hits one of the pipes. See below.

Figure A3.17


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
p5.js 1.11.10
pipe.js
Saved: just now
Preview
fill(255)
rect(this.x, 0, this.w, this.top)
rect(this.x, height - this.bottom, this.w, this.bottom)
}
move()
{
this.x -= this.speed
}
offscreen()
{
if (this.x < -this.w)
{
return true
}
else
{
return false
}
}
}
Console
Clear
283 HIT



Sketch A3.18 highlighting collision

Highlights when a bird hits the pipe, and make it turn red.

pipe.js

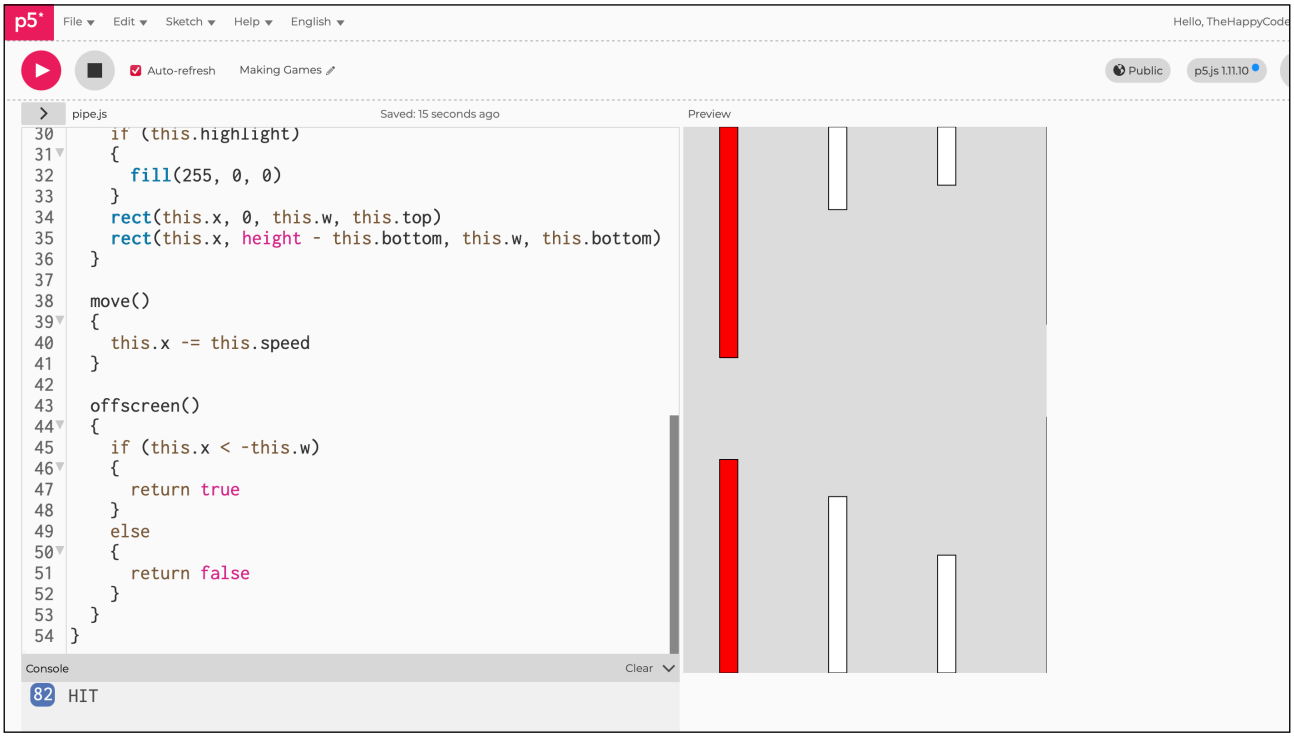
```
class Pipe
{
  constructor()
  {
    this.top = random(0, height/2)
    this.bottom = random(height/2)
    this.x = width
    this.w = 20
    this.speed = 1
    this.highlight = false
  }

  hits(bird)
  {
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x > this.x && bird.x < this.x + this.w)
      {
        this.highlight = true
        return true
      }
    }
    this.highlight = false
    return false
  }

  show()
```

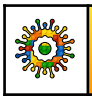
```
  {
    fill(255)
    if (this.highlight)
    {
      fill(255, 0, 0)
    }
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
  }

  move()
  {
    this.x -= this.speed
  }

  offscreen()
  {
    if (this.x < -this.w)
    {
      return true
    }
    else
    {
      return false
    }
  }
}
```

🗒️ Notes

When the circle (bird) hits the pipe, it turns red.

Figure A3.18


p5
File
Edit
Sketch
Help
English
Hello, TheHappyCode
Auto-refresh
Making Games
Public
p5.js 1.11.10
pipe.js
Saved: 15 seconds ago
Preview
30 if (this.highlight)
31 {
32 fill(255, 0, 0)
33 }
34 rect(this.x, 0, this.w, this.top)
35 rect(this.x, height - this.bottom, this.w, this.bottom)
36 }
37
38 move()
39 {
40 this.x -= this.speed
41 }
42
43 offscreen()
44 {
45 if (this.x < -this.w)
46 {
47 return true
48 }
49 else
50 {
51 return false
52 }
53 }
54 }
Console
Clear
82 HIT



Sketch A3.19 mind the gap

Now to make the gaps the same size, whatever the length of the pipe. We create a fixed spacing. All the randomness will be contained in the top pipe.

pipe.js

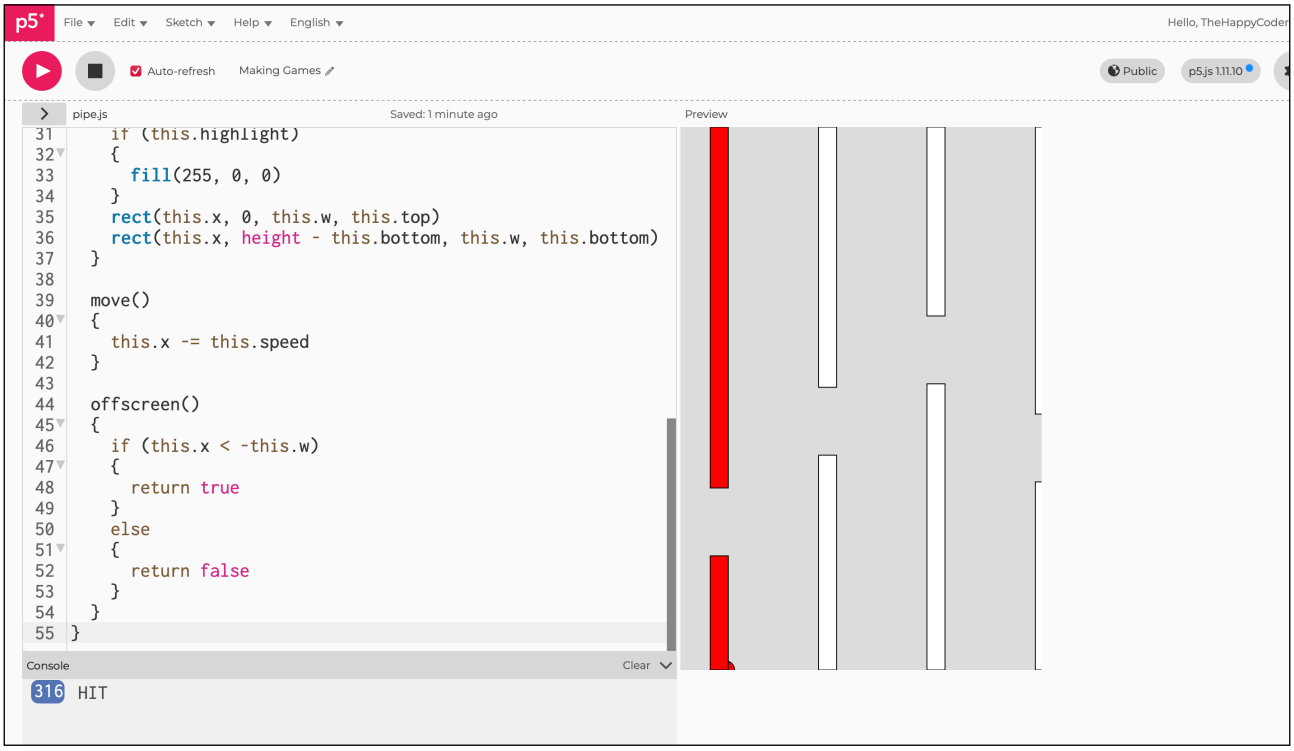
```
class Pipe
{
  constructor()
  {
    this.spacing =  75
    this.top = random(height/6, 3*height/4)
    this.bottom = height - (this.top + this.spacing)
    this.x = width
    this.w = 20
    this.speed = 1
    this.highlight = false
  }

  hits(bird)
  {
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x > this.x && bird.x < this.x + this.w)
      {
        this.highlight = true
        return true
      }
    }
    this.highlight = false
    return false
  }
```

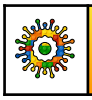
```
  show()
  {
    fill(255)
    if (this.highlight)
    {
      fill(255, 0, 0)
    }
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
  }

  move()
  {
    this.x -= this.speed
  }

  offscreen()
  {
    if (this.x < -this.w)
    {
      return true
    }
    else
    {
      return false
    }
  }
}
```

🗒️ Notes

Now all the gaps are the same, see if you can get through without hitting the pipes. You could try changing the gap size.

Figure A3.19


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
pipe.js
Saved: 1 minute ago
Preview
31 if (this.highlight)
32 {
33 fill(255, 0, 0)
34 }
35 rect(this.x, 0, this.w, this.top)
36 rect(this.x, height - this.bottom, this.w, this.bottom)
37 }
38
39 move()
40 {
41 this.x -= this.speed
42 }
43
44 offscreen()
45 {
46 if (this.x < -this.w)
47 {
48 return true
49 }
50 else
51 {
52 return false
53 }
54 }
55 }
Console
Clear
316 HIT



Sketch A3.20 remove the log

❗ back in sketch.js

And finally, in the main sketch, remove the `console.log()`.

sketch.js

```
let bird
let pipes = []

function setup()
{
  createCanvas(400, 600)
  bird = new Bird()
  pipes.push(new Pipe())
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
  for (let i = pipes.length - 1; i >= 0; i--)
  {
    pipes[i].show()
    pipes[i].move()
    pipes[i].hits(bird)
    if (pipes[i].offscreen())
    {
```

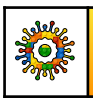
```
      pipes.splice(i, 1)
    }
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

🗒️ Notes

We are just putting the pieces of the game together, at the moment it is virtually unplayable.

Refinements

Here we will add all the stuff that moves it from a theoretical game to one that looks like the real game. We create the Flappy Bird itself, enhance the look of the pipes, change the colour of the background, etc. You can definitely improve upon it.



Sketch A3.21 a png of the bird

❗ start a brand new sketch

Save what you have done already, open a new sketch and call it `bird.png` because we are going to create a PNG image of the bird used in the original game. You will be saving onto your desktop so that you can later drag and drop it into the Flappy Bird sketch.

So delete all the default code because you only need two functions: `setup()` and `mouseClicked()`. There is no background (and that is important). I found an image and drew a `17x12` grid over it. Then drew and coloured each square/rectangle in p5.js. The resulting code is what you can see below. It may seem a bit tedious to copy out all the lines of code but you can use copy and paste as you go along.

However tedious, think of it as creating your own bit of pixel artwork. You can create any PNG image. The reason we are creating a PNG without a background is because it is then transparent where there are no pixels.

When you are finished, just click on the image in the canvas and it will download it somewhere on your computer. Wherever it has downloaded, find it so that you can drag and drop it in the next part.

🗒️ Notes

you do not need to write the comments bit but they will help you keep track!

Making a png of the bird sketch.js

```
let c


function setup()
{
  c = createCanvas(170, 120)


  noStroke()


  // black
  fill(0)
  // line 0
  rect(60, 0, 60, 10)
  // line 1
  rect(40, 10, 20, 10)
  square(90, 10, 10)
  square(120, 10, 10)
  // line 2
  square(30, 20, 10)
  rect(80, 20, 10, 30)
  square(130, 20, 10)
  // line 3
  rect(10, 30, 40, 10)
  rect(120, 30, 10, 20)
  rect(140, 30, 10, 30)
```

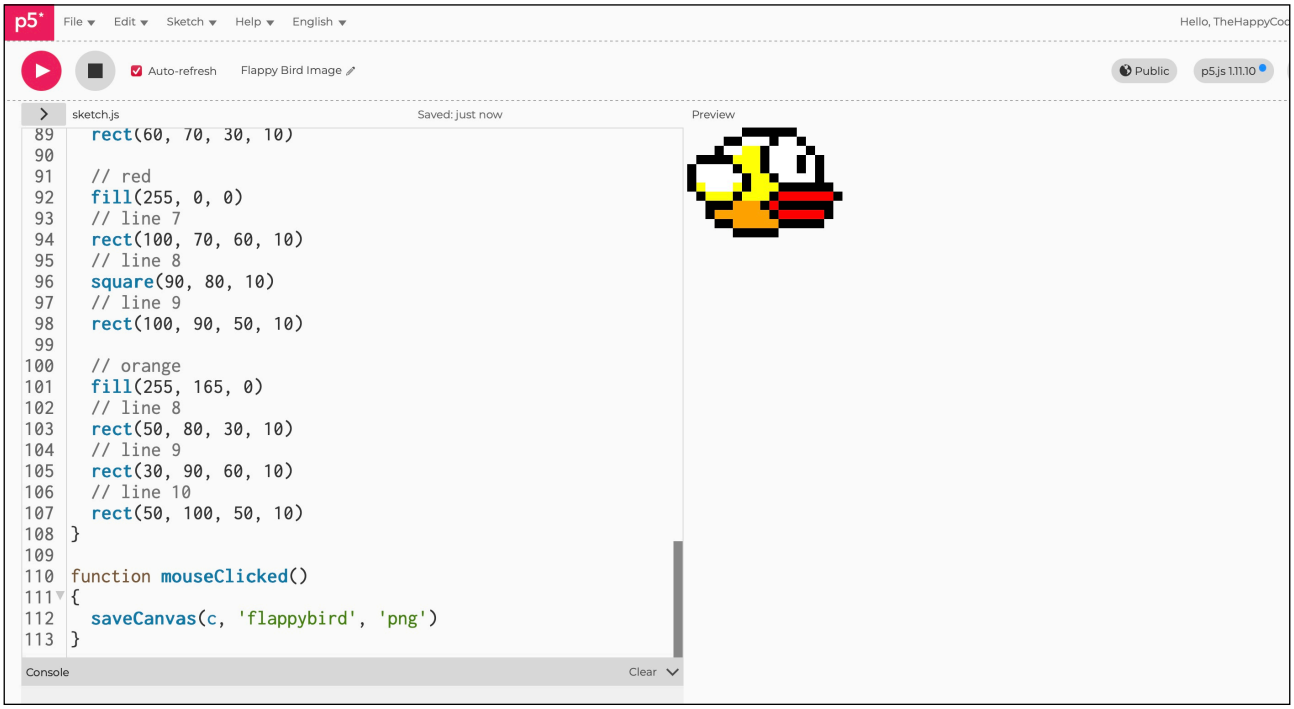
```
// line 4
rect(0, 40, 10, 30)
square(50, 40, 10)
// line 5
rect(60, 50, 10, 20)
square(90, 50, 10)
// line 6
rect(100, 60, 60, 10)
// line 7
square(10, 70, 10)
square(50, 70, 10)
square(90, 70, 10)
square(160, 70, 10)
// line 8
rect(20, 80, 30, 10)
square(80, 80, 10)
rect(100, 80, 60, 10)
// line 9
square(20, 90, 10)
square(90, 90, 10)
square(150, 90, 10)
// line 10
rect(30, 100, 20, 10)
rect(100, 100, 50, 10)
// line 12
rect(50, 110, 50, 10)

// white
```

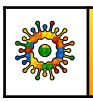
```
fill(255)
// line 1
rect(60, 10, 30, 10)
rect(100, 10, 20, 10)
// line 2
rect(40, 20, 20, 10)
rect(90, 20, 40, 10)
// line 3
rect(90, 30, 30, 10)
square(130, 30, 10)
// line 4
rect(10, 40, 40, 10)
rect(90, 40, 30, 10)
square(130, 40, 10)
// line 5
rect(10, 50, 50, 10)
rect(100, 50, 40, 10)
// line 6
rect(20, 60, 30, 10)

// yellow
fill(255, 255, 0)
// line 2
rect(60, 20, 20, 10)
// line 3
rect(50, 30, 30, 10)
// line 4
rect(60, 40, 20, 10)
```

```
  // line 5
  rect(70, 50, 20, 10)
  // line 6
  square(10, 60, 10)
  square(50, 60, 10)
  rect(70, 60, 30, 10)
  // line 7
  rect(20, 70, 30, 10)
  rect(60, 70, 30, 10)

  // red
  fill(255, 0, 0)
  // line 7
  rect(100, 70, 60, 10)
  // line 8
  square(90, 80, 10)
  // line 9
  rect(100, 90, 50, 10)

  // orange
  fill(255, 165, 0)
  // line 8
  rect(50, 80, 30, 10)
  // line 9
  rect(30, 90, 60, 10)
  // line 10
  rect(50, 100, 50, 10)
}
```

```
function mouseClicked()
{
  saveCanvas(c, 'flappybird', 'png')
}
```

Figure 1: flappy bird image





Sketch A3.22 adding the png to our game

❗ back to sketch.js (flappy bird)

Now you will need to go back to your code for Flappy Bird. Under the tab 'Sketches', where you created your files for bird.js and pipe.js, this time click on 'Add Folder' and give it the name images.

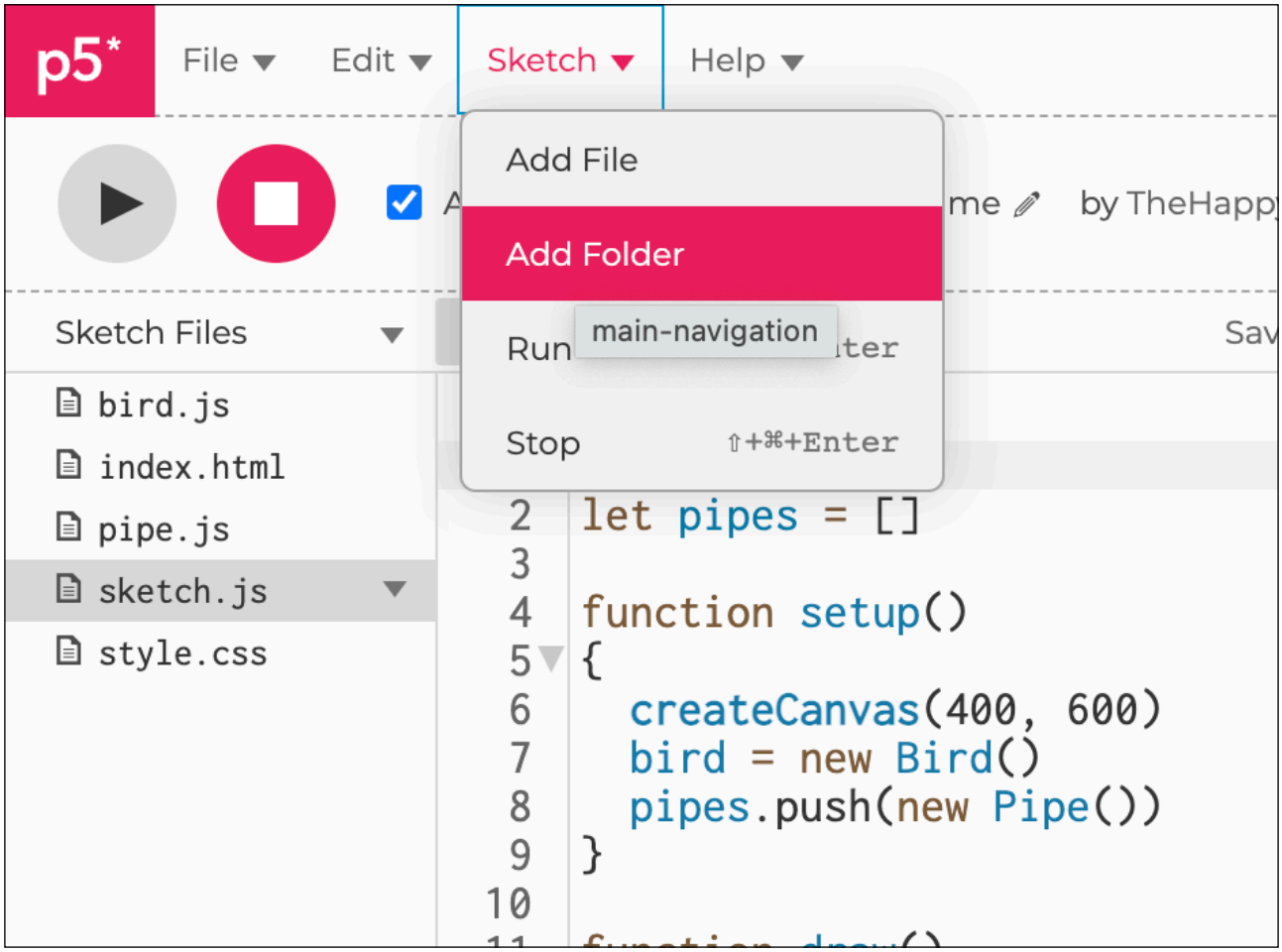
After you have done that, head over to the list of sketches on the left-hand side. You will see a folder called images, click on it, there will be a small triangle (on the right of the name), click on that and you will get a drop-down menu, select `Upload file`, now drag and drop the image from your desktop.

The name of the image will now appear in that folder with the extension PNG. If the name is not the one you want, then rename it either in its own drop-down box or rename it on the desktop and redo the drag and drop.

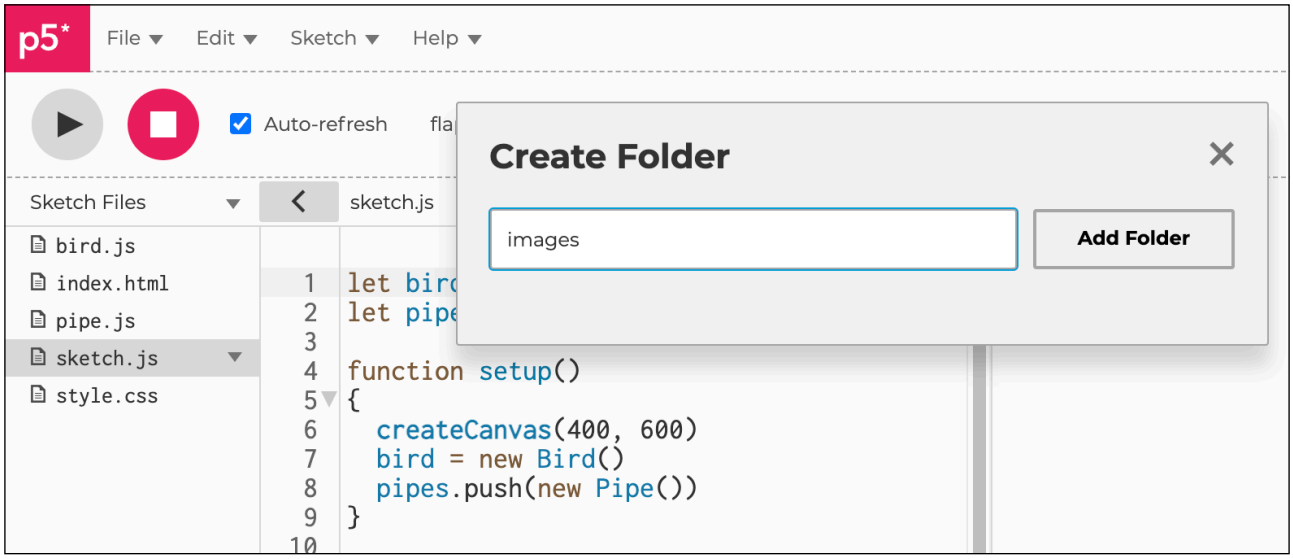
I have called mine `flappybird.png`, yours should be the same, although if you have made several copies it might have an index number as well. The name that appears here is important for what you put in the code in a moment.

The `preload()` function is to make sure the bird image is loaded properly before starting the game. The commented-out bit can be uncommented to make sure that the image is there before we replace the circle with the bird image.

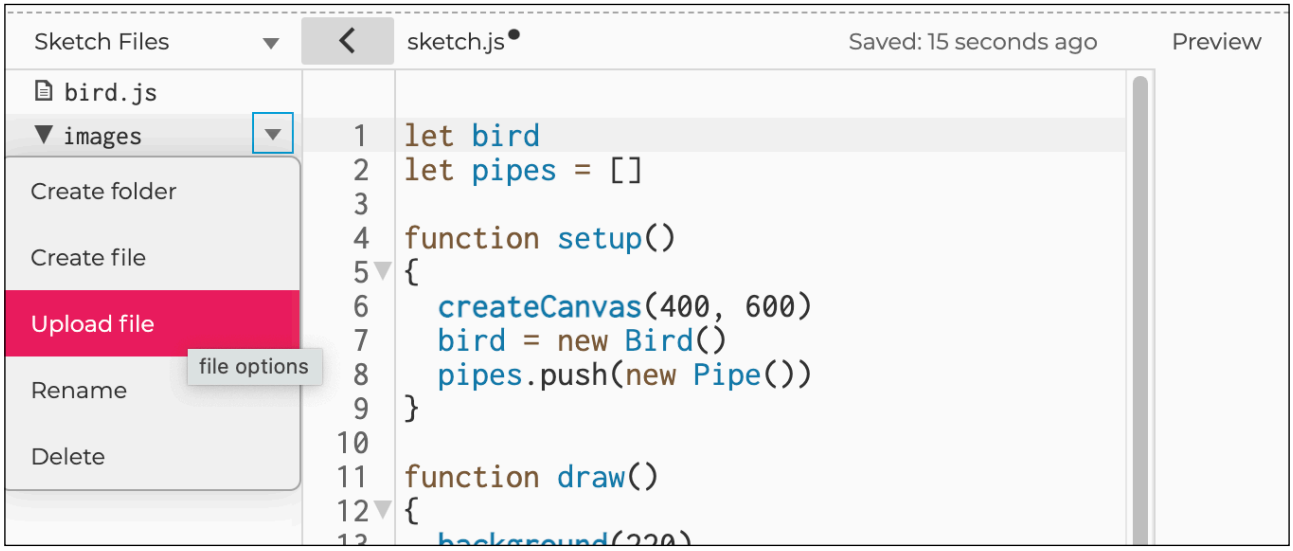
Add a folder


p5*
File
Edit
Sketch
Help
Add File
Add Folder
Run
main-navigation
Stop
⇧+⌘+Enter
Sketch Files
bird.js
index.html
pipe.js
sketch.js
style.css
let pipes = []
function setup()
{
createCanvas(400, 600)
bird = new Bird()
pipes.push(new Pipe())
}

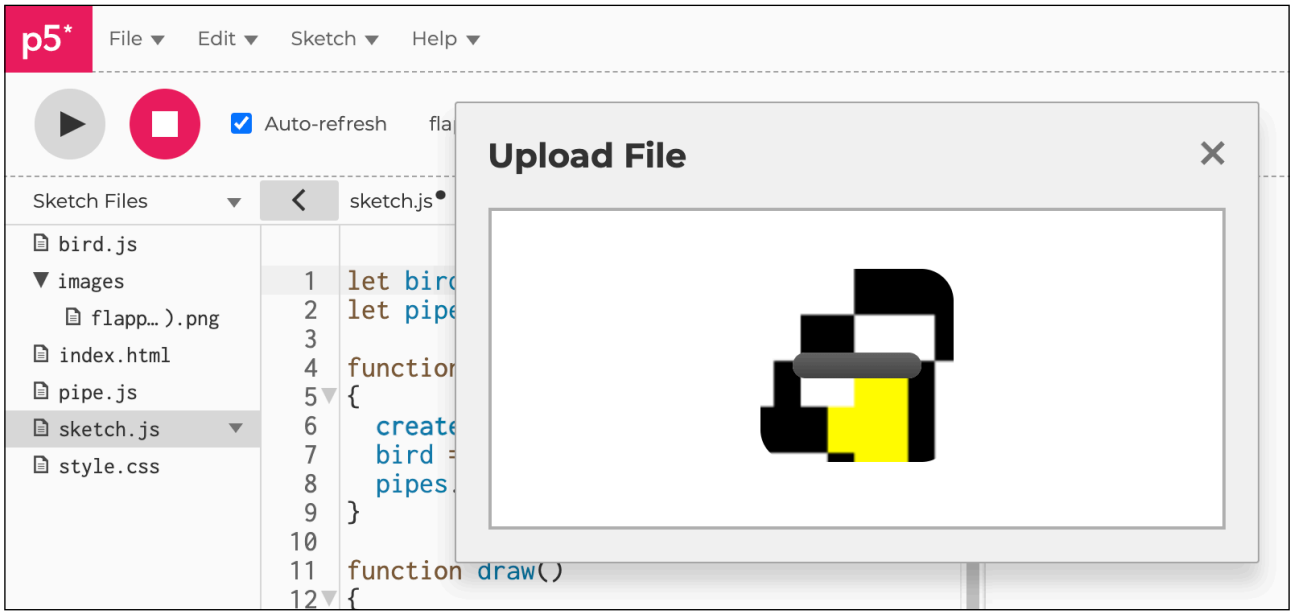
Give it the name images



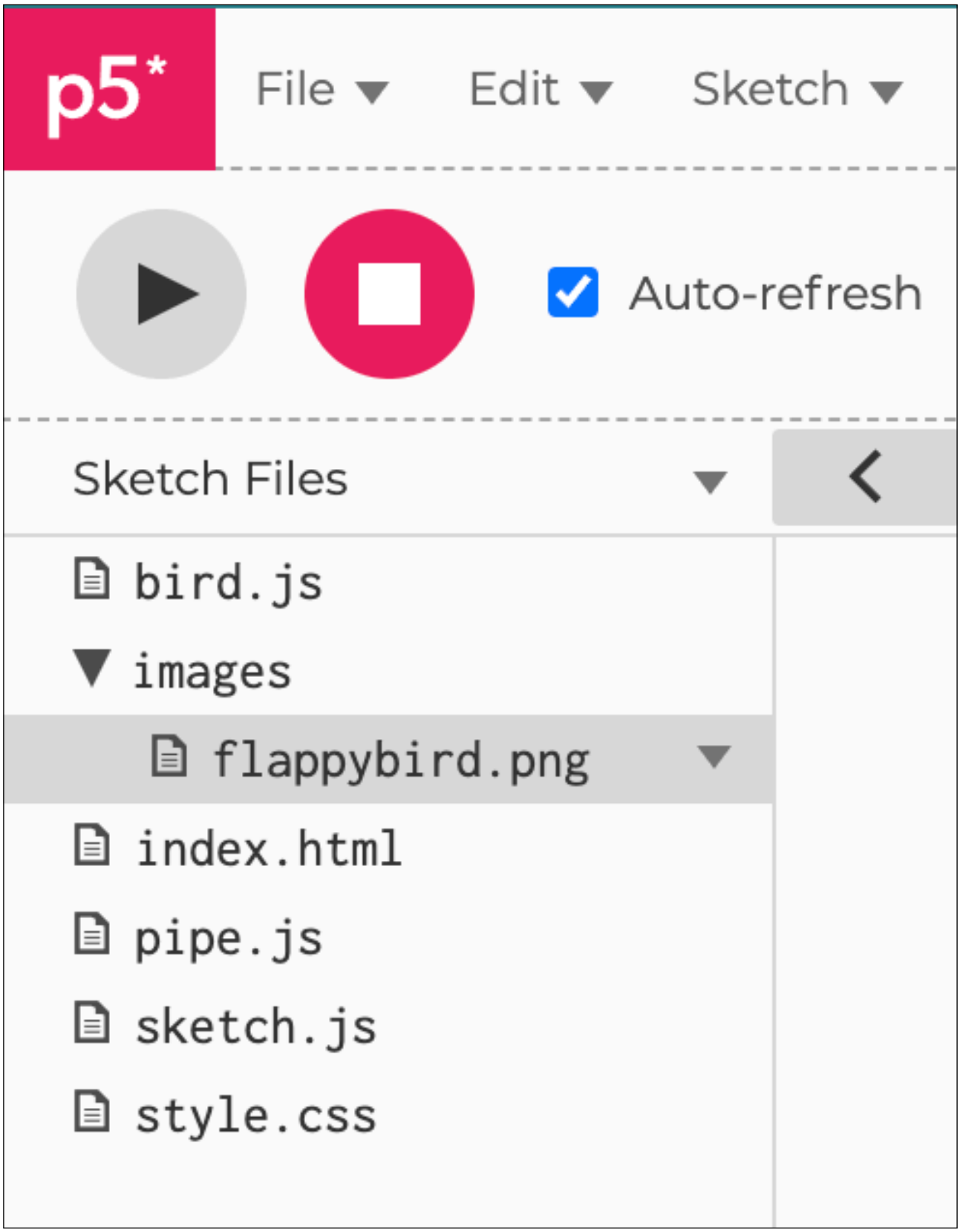
Click on upload file

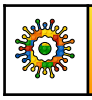


Drag and drop the flappy bird file



The flappy bird file is in the images folder





Sketch A3.23 adding the bird image

The image is preloaded so that the code doesn't start running without the image being downloaded. We will also change the dimensions to make it a bit more playable.

sketch.js

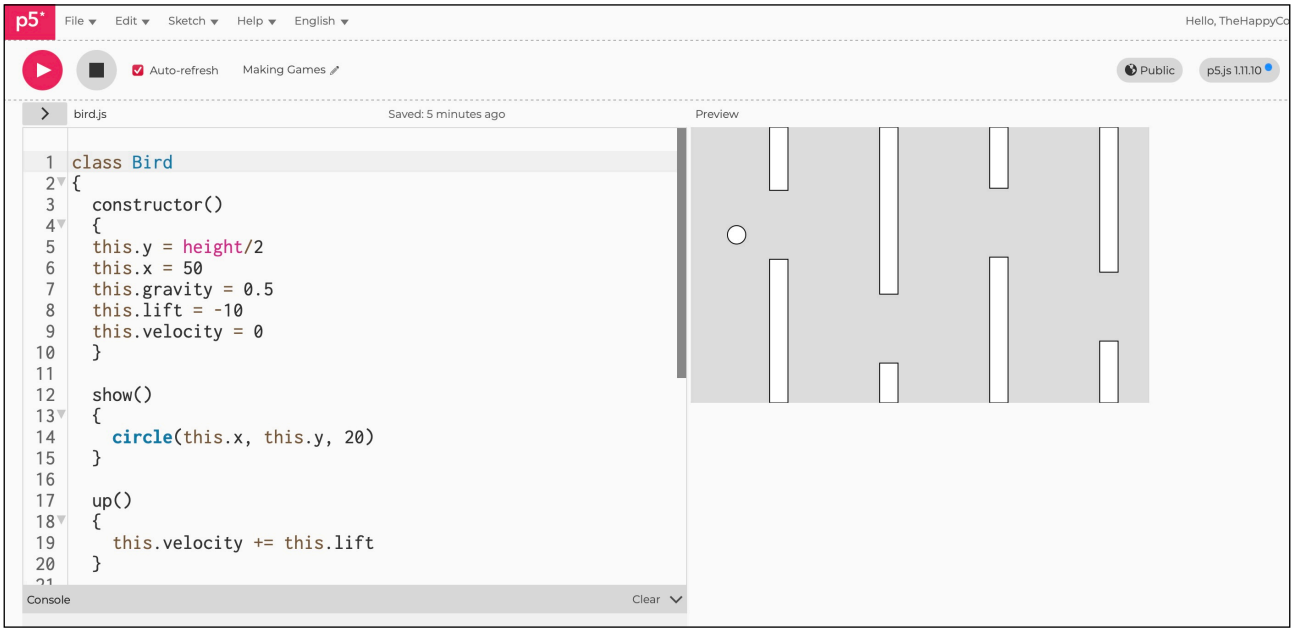
```
let bird
let pipes = []
let img

function preload()
{
  img = loadImage('images/flappybird.png')
}

function setup()
{
  createCanvas(500, 300)
  bird = new Bird()
  pipes.push(new Pipe())
}

function draw()
{
  background(220)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
  for (let i = pipes.length - 1; i >= 0; i--)
```

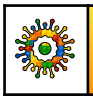
```
  {
    pipes[i].show()
    pipes[i].move()
    pipes[i].hits(bird)
    if (pipes[i].offscreen())
    {
      pipes.splice(i, 1)
    }
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

🗒️ Notes

We won't see the flappy bird image just yet until we add it to the `show()` function in `bird.js`.

Figure A3.23


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCo
Auto-refresh
Making Games
Public
p5.js 1.11.10
bird.js
Saved: 5 minutes ago
Preview
class Bird
{
constructor()
{
this.y = height/2
this.x = 50
this.gravity = 0.5
this.lift = -10
this.velocity = 0
}
show()
{
circle(this.x, this.y, 20)
}
up()
{
this.velocity += this.lift
}
Console
Clear



Sketch A3.24 replacing the circle

❗ to bird.js

In the `bird.js`, we will replace the circle with the bird image and remove the `circle(this.x, this.y, 20)` in the `show()` function.

bird.js

```
class Bird
{
  constructor()
  {
  this.y = height/2
  this.x = 50
  this.gravity = 0.5
  this.lift = -10
  this.velocity = 0
  }

  show()
  {
    imageMode(CENTER)
    image(img, this.x, this.y - 12, 34, 24)
  }

  up()
  {
    this.velocity += this.lift
  }

  move()
  {
    this.velocity += this.gravity
```

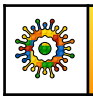
```
    this.velocity *= 0.9
    this.y += this.velocity
    if (this.y > height)
    {
      this.y = height
      this.velocity = 0
    }
    if (this.y < 0)
    {
      this.y = 0
      this.velocity = 0
    }
  }
}
```

🗒️ Notes

Now you have a flappy bird rather than a circle.

Figure A3.24


p5
File
Edit
Sketch
Help
English
Hello, TheHappyCod
Auto-refresh
Making Games
Public
p5.js 1.11.10
bird.js
Saved: just now
Preview
{
this.velocity += this.lift
}
move()
{
this.velocity += this.gravity
this.velocity *= 0.9
this.y += this.velocity
if (this.y > height)
{
this.y = height
this.velocity = 0
}
if (this.y < 0)
{
this.y = 0
this.velocity = 0
}
}
}
Console
Clear



Sketch A3.25 pipe improvements

❗ now pipe.js

Improving the pipes.

pipe.js

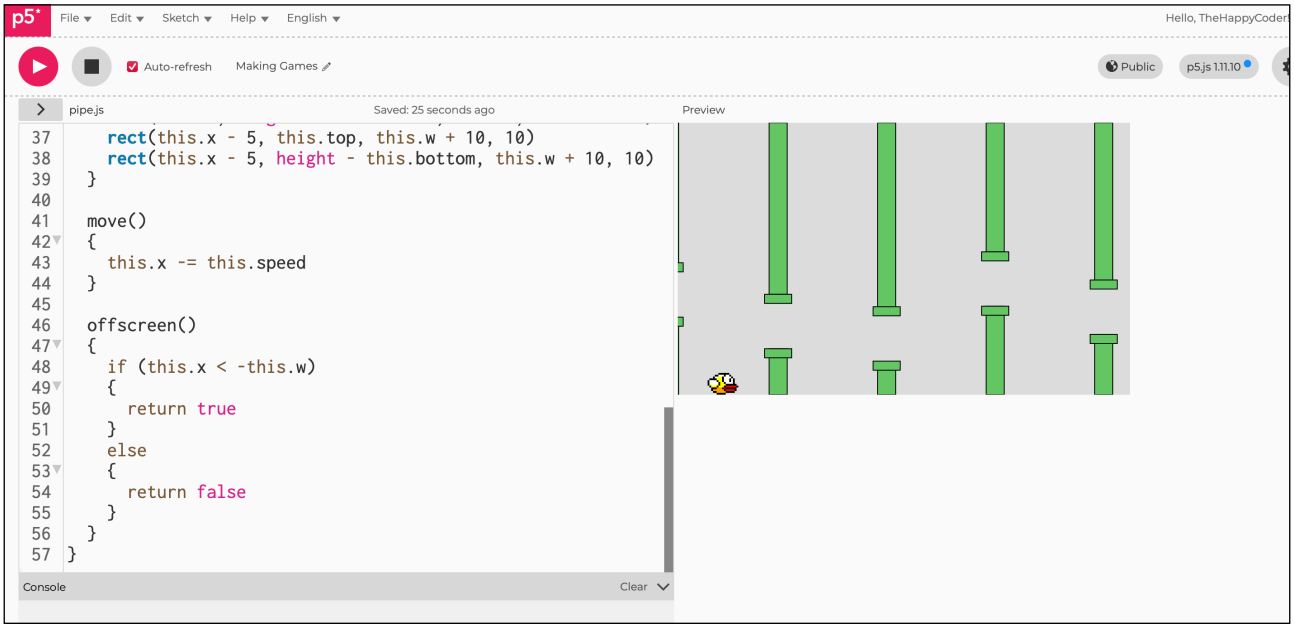
```
class Pipe
{
  constructor()
  {
    this.spacing =  height/5
    this.top = random(height/6, 3*height/4)
    this.bottom = height - (this.top + this.spacing)
    this.x = width
    this.w = 20
    this.speed = 1
    this.highlight = false
  }

  hits(bird)
  {
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x > this.x && bird.x < this.x + this.w)
      {
        this.highlight = true
        return true
      }
    }
    this.highlight = false
    return false
  }
```

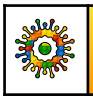
```
  show()
  {
    fill(100, 200, 100)
    if (this.highlight)
    {
      fill(255, 0, 0)
    }
    rect(this.x, 0, this.w, this.top)
    rect(this.x, height - this.bottom, this.w, this.bottom)
    rect(this.x - 5, this.top, this.w + 10, 10)
    rect(this.x - 5, height - this.bottom, this.w + 10, 10)
  }

  move()
  {
    this.x -= this.speed
  }

  offscreen()
  {
    if (this.x < -this.w)
    {
      return true
    }
    else
    {
      return false
    }
  }
}
```

Notes

Much improved pipes

Figure A3.25


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
pipe.js
Saved: 25 seconds ago
Preview
rect(this.x - 5, this.top, this.w + 10, 10)
rect(this.x - 5, height - this.bottom, this.w + 10, 10)
}
move()
{
this.x -= this.speed
}
offscreen()
{
if (this.x < -this.w)
{
return true
}
else
{
return false
}
}
}
Console
Clear



Sketch A3.26 adding a score

❗ and all the way back to sketch.js.
Adding a score and make the background dark blue.

sketch.js

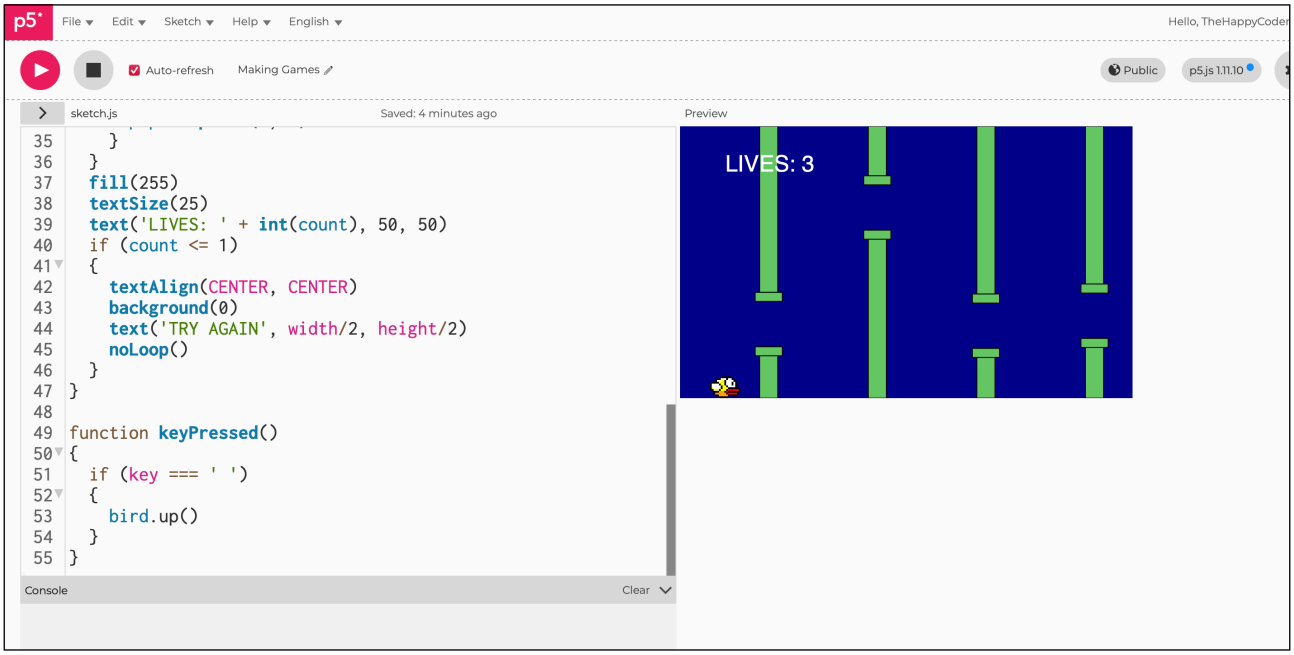
```
let bird
let pipes = []
let img
let count = 3

function preload()
{
  img = loadImage('images/flappybird.png')
}

function setup()
{
  createCanvas(500, 300)
  bird = new Bird()
  pipes.push(new Pipe())
}

function draw()
{
  background('darkblue')
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
```

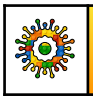
```
  for (let i = pipes.length - 1; i >= 0; i--)
  {
    pipes[i].show()
    pipes[i].move()
    pipes[i].hits(bird)
    if (pipes[i].offscreen())
    {
      pipes.splice(i, 1)
    }
  }
  fill(255)
  textSize(25)
  text('LIVES: ' + int(count), 50, 50)
  if (count <= 1)
  {
    textAlign(CENTER, CENTER)
    background(0)
    text('TRY AGAIN', width/2, height/2)
    noLoop()
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

🗒️ Notes

The score doesn't alter at this stage we need to go into `pipe.js` to count when the pipe is hit.

Figure A3.26


p5
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Making Games
Public
p5.js 1.11.10
sketch.js
Saved: 4 minutes ago
Preview
LIVES: 3
}
}
fill(255)
textSize(25)
text('LIVES: ' + int(count), 50, 50)
if (count <= 1)
{
textAlign(CENTER, CENTER)
background(0)
text('TRY AGAIN', width/2, height/2)
noLoop()
}
}
function keyPressed()
{
if (key === ' ')
{
bird.up()
}
}
Console
Clear



Sketch A3.27 reducing the count

❗ wander over to pipe.js.

Reducing the count by one each time the bird hits a pipe. This doesn't always work perfectly, but it is pretty good.

pipe.js

```
class Pipe
{
  constructor()
  {
    this.spacing =  height/5
    this.top = random(height/6, 3*height/4)
    this.bottom = height - (this.top + this.spacing)
    this.x = width
    this.w = 40
    this.speed = 2
    this.highlight = false
  }

  hits(bird)
  {
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x == this.x)
      {
        count -= 1
      }
    }
    if (bird.y < this.top || bird.y > height - this.bottom)
    {
      if (bird.x > this.x && bird.x < this.x + this.w)
```

```
    {
      this.highlight = true
      return true
    }
  }
  this.highlight = false
  return false
}

show()
{
  stroke(0)
  fill(100, 200, 100)
  if (this.highlight)
  {
    fill(255, 0, 0)
  }
  rect(this.x, 0, this.w, this.top)
  rect(this.x - 5, this.top, this.w + 10, 10)
  rect(this.x, height - this.bottom, this.w, this.bottom)
  rect(this.x - 5, height - this.bottom, this.w + 10, 10)
}

move()
{
  this.x -= this.speed
}

offscreen()
{
  if (this.x < -this.w)
  {
```

```
      return true
    }
    else
    {
      return false
    }
  }
}
```

Figure A3.27a

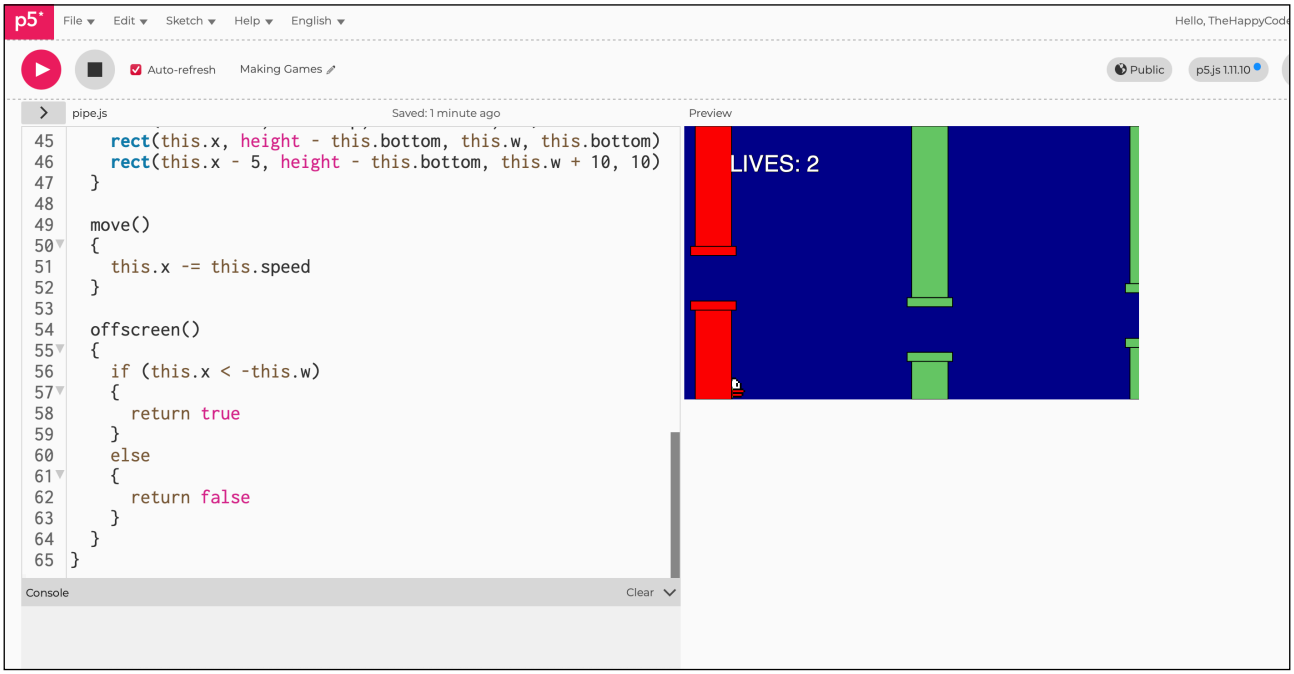
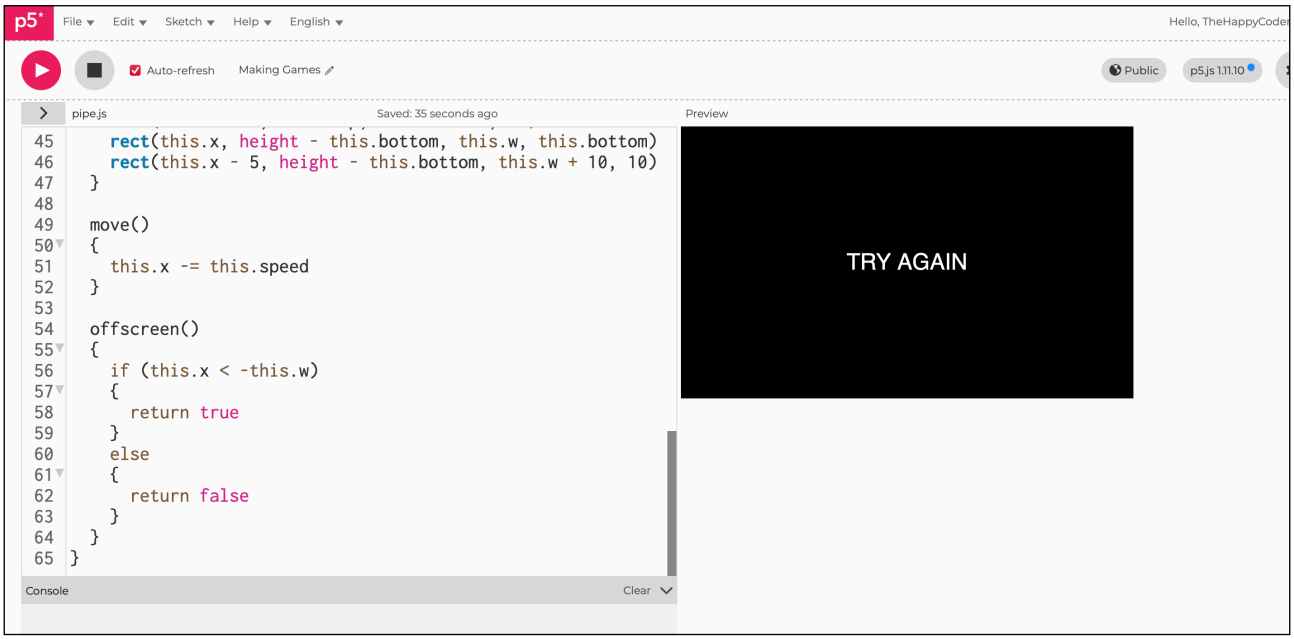
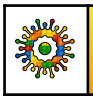


Figure A3.27b





Sketch A3.28 three lives

❗ and for the final time sketch.js
Adding the score is just the time before you hit three pipes.

sketch.js

```
let bird
let pipes = []
let img
let count = 3

function preload()
{
  img = loadImage('images/flappybird.png')
}

function setup()
{
  createCanvas(500, 300)
  bird = new Bird()
  pipes.push(new Pipe())
}

function draw()
{
  background(0, 0, 255)
  bird.show()
  bird.move()
  if (frameCount % 120 == 0)
  {
    pipes.push(new Pipe())
  }
```

```
  for (let i = pipes.length - 1; i >= 0; i--)
  {
    pipes[i].show()
    pipes[i].move()
    pipes[i].hits(bird)
    if (pipes[i].offscreen())
    {
      pipes.splice(i, 1)
    }
  }
  fill(255)
  textSize(25)
  text('LIVES: ' + int(count), 50, 50)
  let score = floor(millis()/1000)
  text('score = ' + score, 50, 75)
  if (count <= 1)
  {
    textAlign(CENTER, CENTER)
    background(0)
    text('TRY AGAIN', width/2, height/2)
    text('YOUR SCORE IS ' + score, width/2, height/3)
    noLoop()
  }
}

function keyPressed()
{
  if (key === ' ')
  {
    bird.up()
  }
}
```

🗒️ Notes

A better way maybe to have a health rectangle which diminishes gradually depending on how hard the bird hits the pipe.

Figure A3.28a

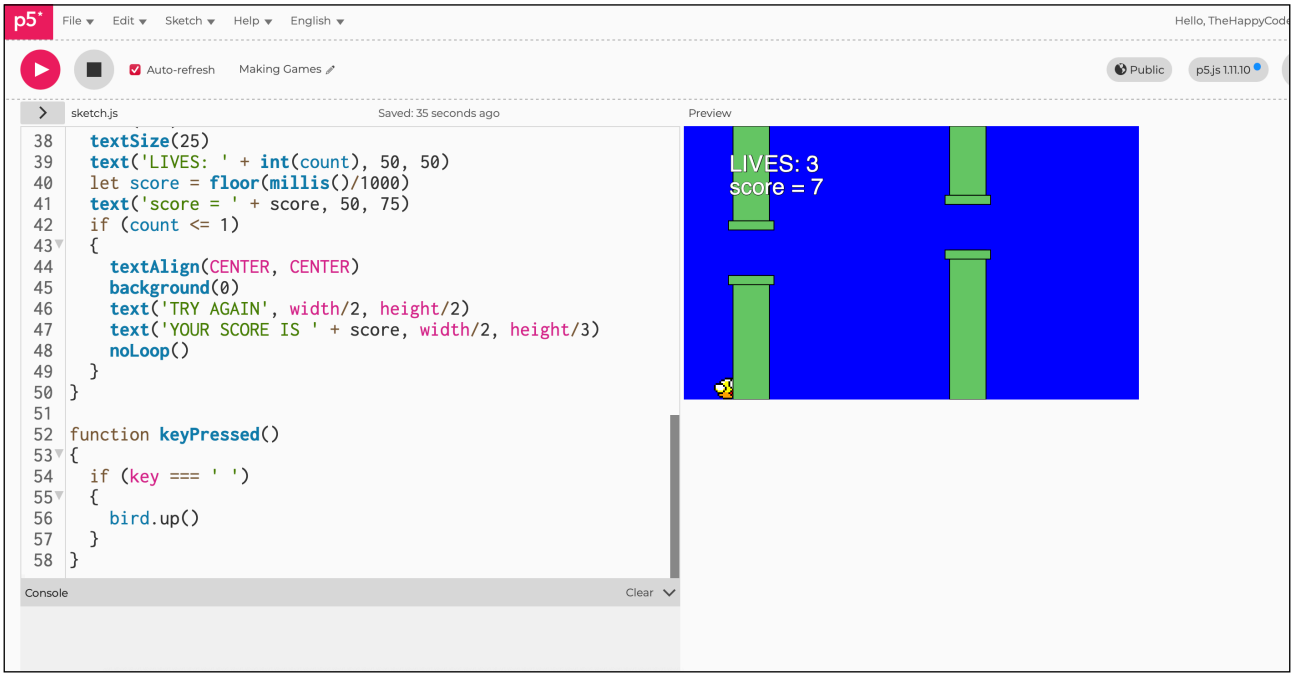


Figure A3.28b