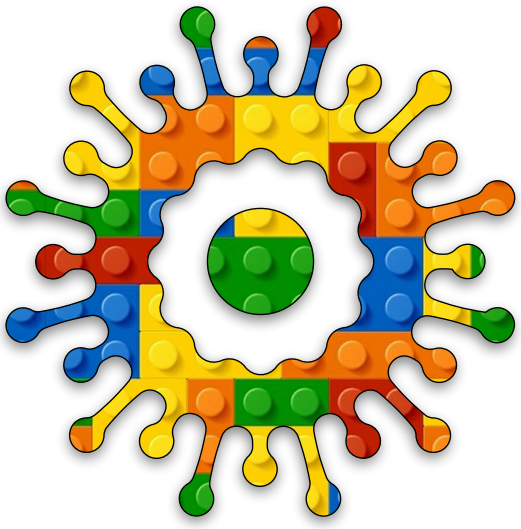
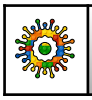
Making Games
Module A
Unit #2
pong



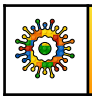


Content

Module A Unit #2 pong game

Introduction to Pong

This is the ultimate classic video game. It looks so simple, and yet there is a lot you can learn about making games and some of the challenges. This can be a two-person game or single-player.

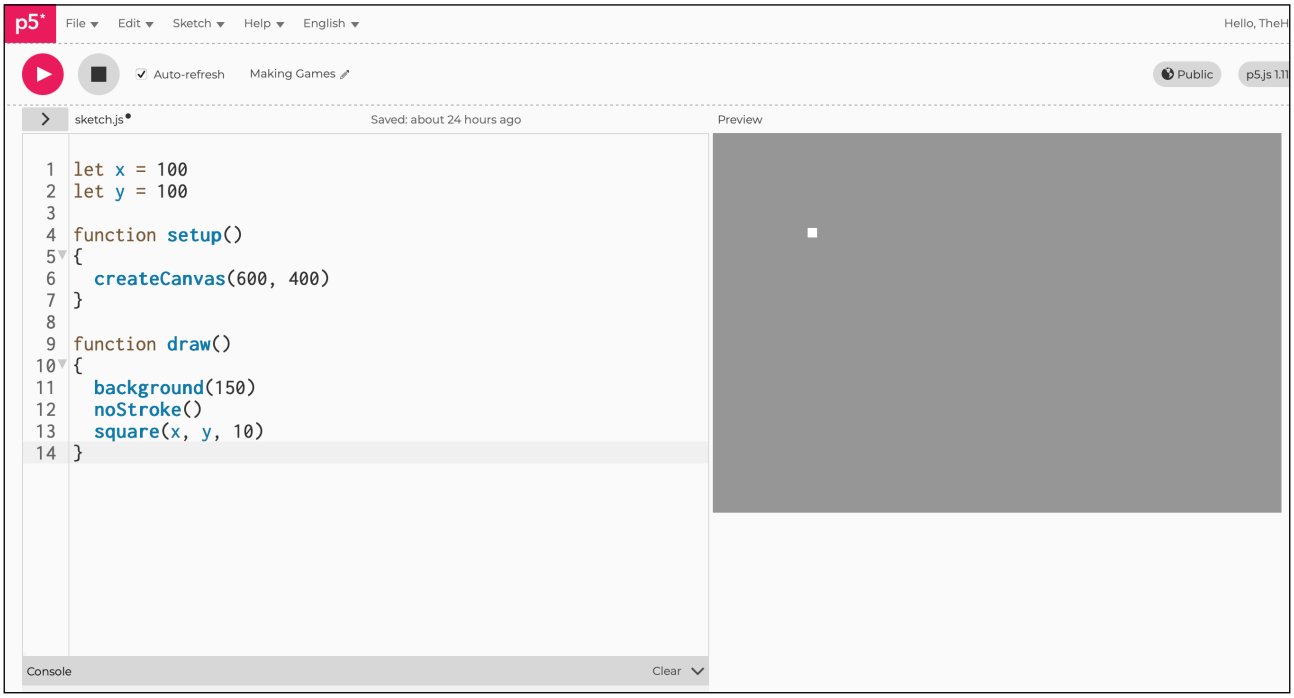


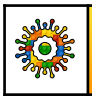
Sketch A2.1 make the pong

This will be the pong square ball.

```
let x = 100
let y = 100

function setup()
{
  createCanvas(600, 400)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
}
```

Figure A2.1


File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: about 24 hours ago
Preview
let x = 100
let y = 100
function setup()
{
createCanvas(600, 400)
}
function draw()
{
background(150)
noStroke()
square(x, y, 10)
}
Console
Clear

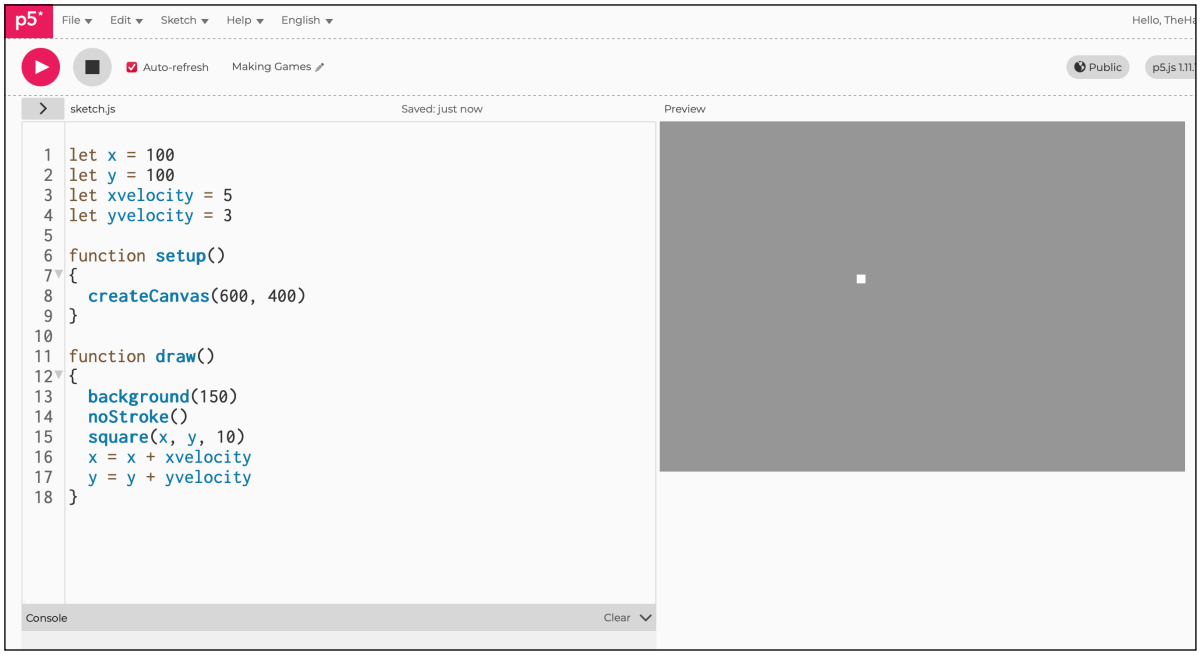


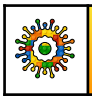
Sketch A2.2 moving the pong

Let's make it move.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3

function setup()
{
  createCanvas(600, 400)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
}
```

Figure A1.2


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: just now
Preview
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
function setup()
{
createCanvas(600, 400)
}
function draw()
{
background(150)
noStroke()
square(x, y, 10)
x = x + xvelocity
y = y + yvelocity
}
Console
Clear



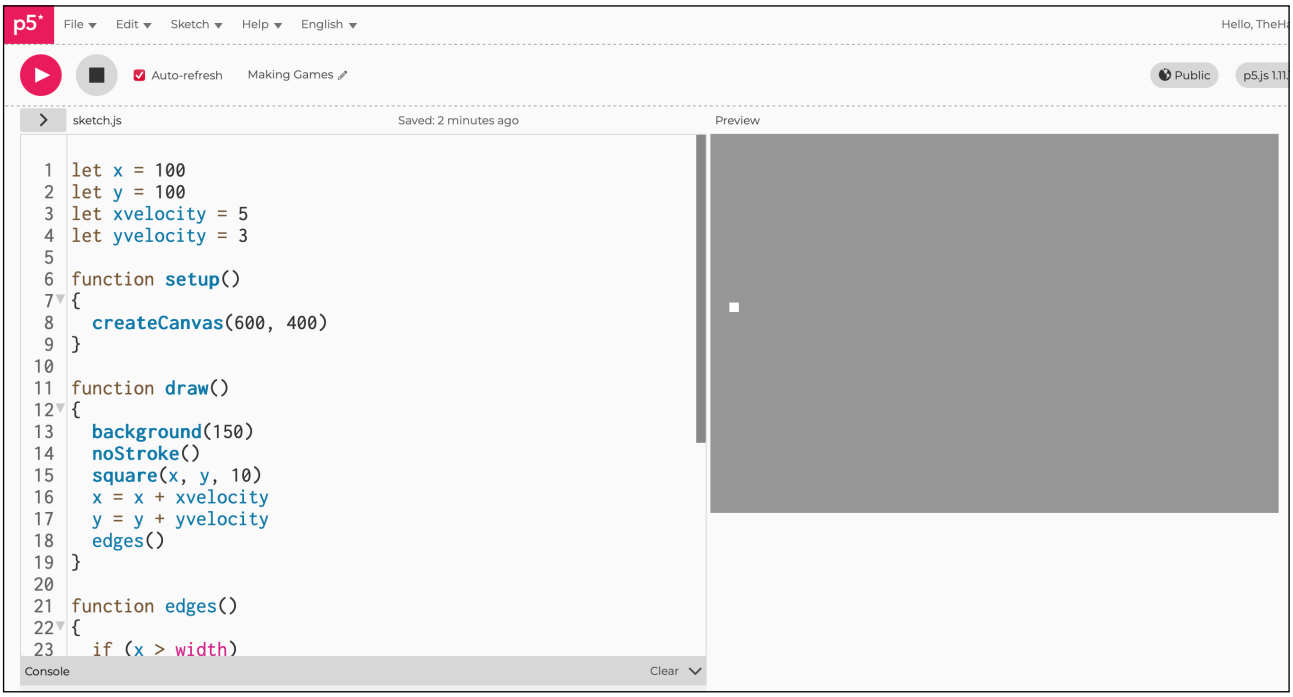
Sketch A2.3 bouncing off the edges

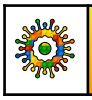
Now to bounce off the walls.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3

function setup()
{
  createCanvas(600, 400)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
}

function edges()
{
  if (x > width)
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
```

```
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}
```

Figure A2.3


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 2 minutes ago
Preview
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
function setup()
{
createCanvas(600, 400)
}
function draw()
{
background(150)
noStroke()
square(x, y, 10)
x = x + xvelocity
y = y + yvelocity
edges()
}
function edges()
{
if (x > width)
Console
Clear



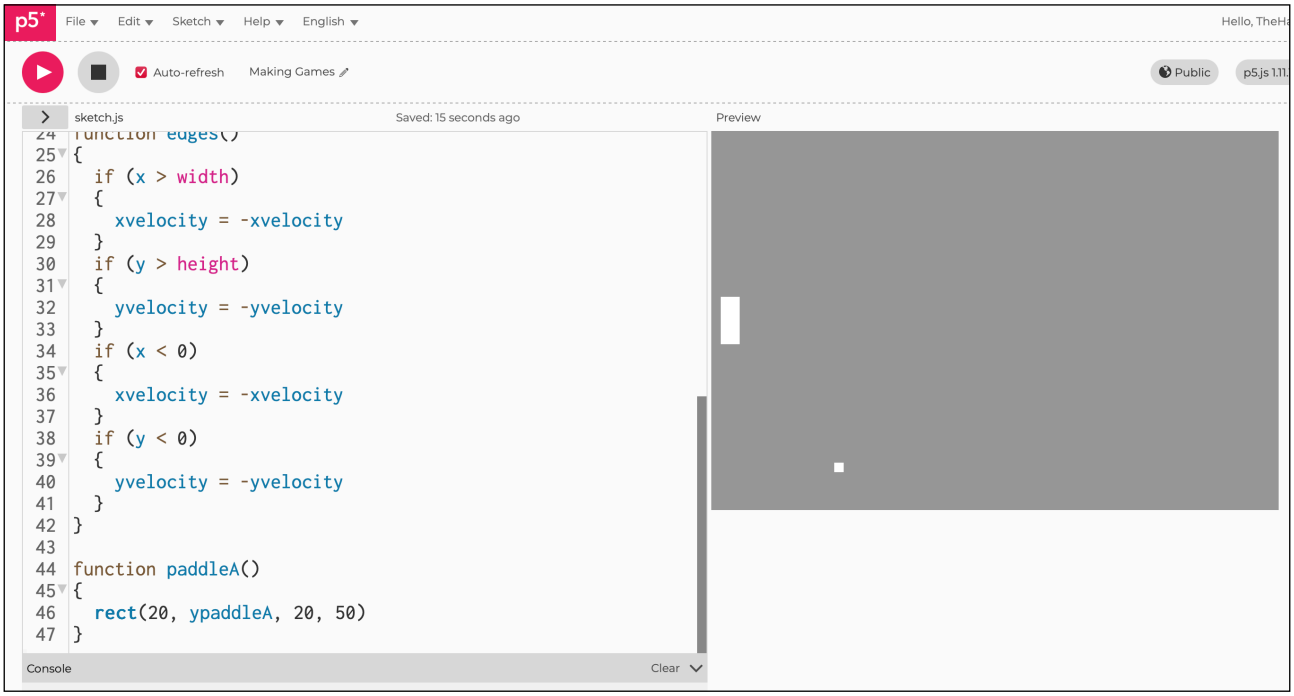
Sketch A2.4 adding a paddle

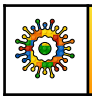
Let's add a paddle.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
}

function edges()
{
  if (x > width)
  {
    xvelocity = -xvelocity
```

```
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
}
```

Figure A2.4


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 15 seconds ago
Preview
function edges()
{
if (x > width)
{
xvelocity = -xvelocity
}
if (y > height)
{
yvelocity = -yvelocity
}
if (x < 0)
{
xvelocity = -xvelocity
}
if (y < 0)
{
yvelocity = -yvelocity
}
}
function paddleA()
{
rect(20, ypaddleA, 20, 50)
}
Console
Clear



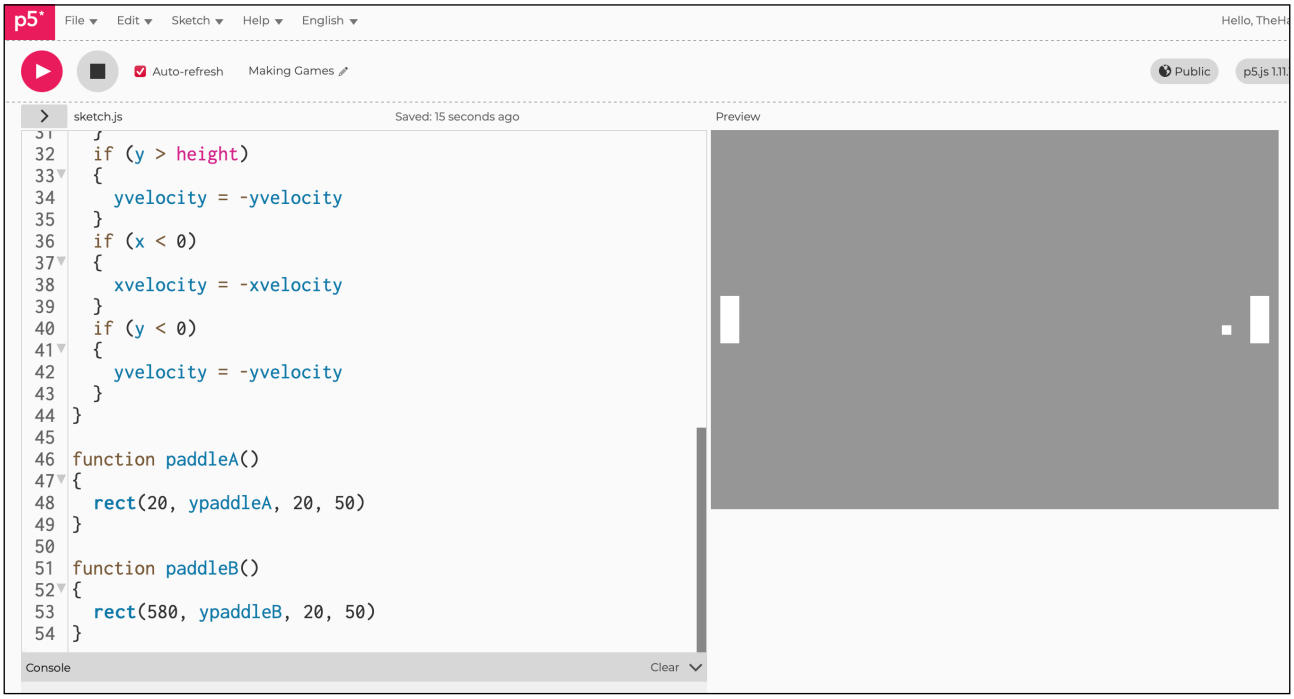
Sketch A2.5 adding paddle B

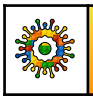
Let's make the other paddle.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()
}

function edges()
{
  if (x > width)
```

```
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
}
```

Figure A1.5


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 15 seconds ago
Preview
if (y > height)
{
yvelocity = -yvelocity
}
if (x < 0)
{
xvelocity = -xvelocity
}
if (y < 0)
{
yvelocity = -yvelocity
}
}
function paddleA()
{
rect(20, ypaddleA, 20, 50)
}
function paddleB()
{
rect(580, ypaddleB, 20, 50)
}
Console
Clear



Sketch A2.6 moving paddle B

We will move paddle up and down with the up and down arrow keys.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
```

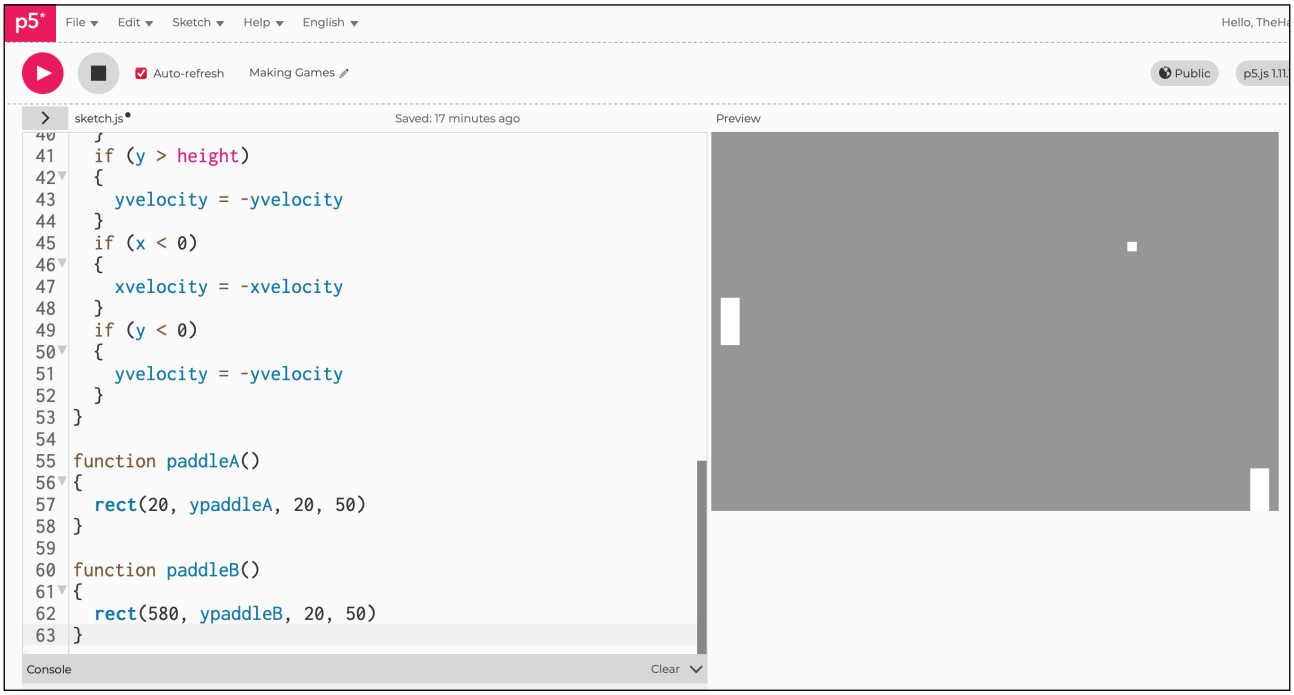
```
  {
    ypaddleB -= 5
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
  }
}

function edges()
{
  if (x > width)
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}
```

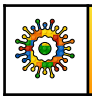
```
function paddleA()
{
  rect(20, ypaddleA, 20, 50)
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
}
```

🗒️ Notes

Notice that the paddle does not stop at the bottom. We will address that later.

Figure A2.6


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 17 minutes ago
Preview
41 if (y > height)
42 {
43 yvelocity = -yvelocity
44 }
45 if (x < 0)
46 {
47 xvelocity = -xvelocity
48 }
49 if (y < 0)
50 {
51 yvelocity = -yvelocity
52 }
53 }
54
55 function paddleA()
56 {
57 rect(20, ypaddleA, 20, 50)
58 }
59
60 function paddleB()
61 {
62 rect(580, ypaddleB, 20, 50)
63 }
Console
Clear



Sketch A2.7 moving paddle A

Now for paddle A with the q and a keys for up and down. To make life easier, you can get the code for every key on the keyboard from this website: https://www.toptal.com/developers/keycode
So q is 81 and a is 65.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
```

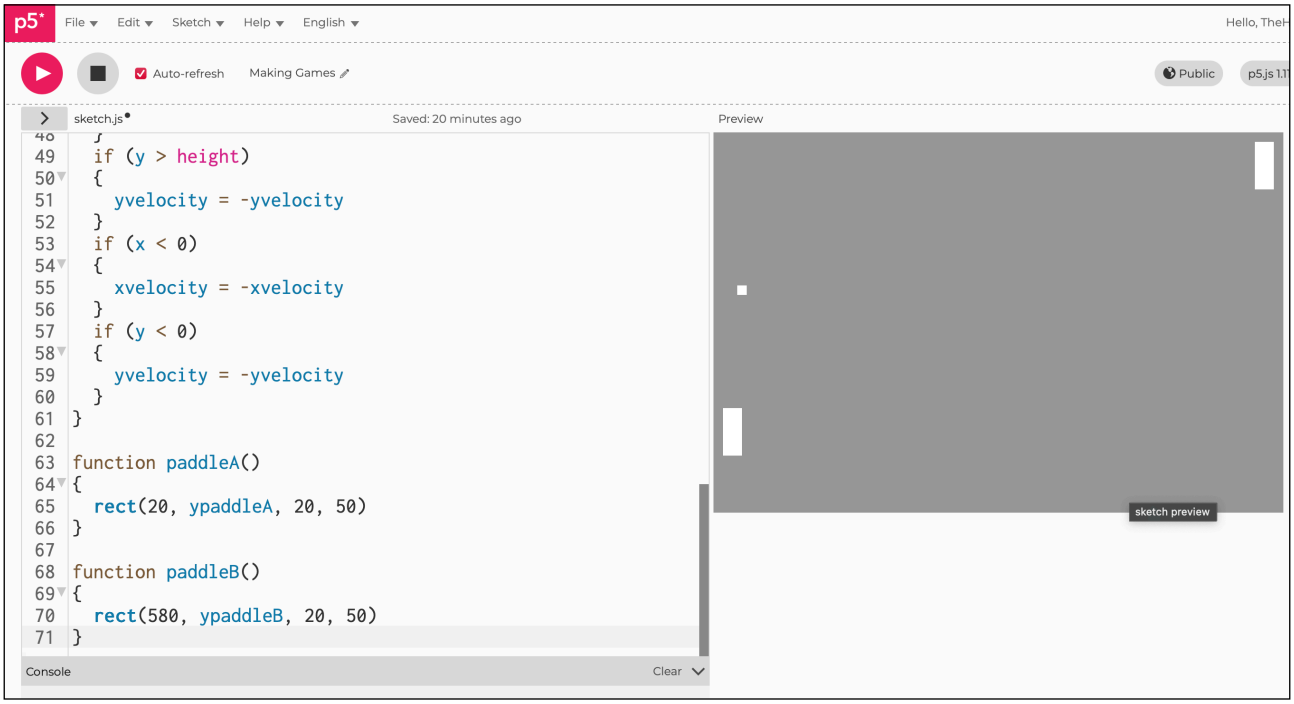
```
    ypaddleB -= 5
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
  }
}

function edges()
{
  if (x > width)
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
```

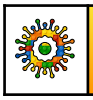
```
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
}
```

🗒️ Notes

Now you should be able to move each paddle independently with the arrow keys for paddle B. We use keyboard keys q and a for paddle A.

Figure A2.7


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 20 minutes ago
Preview
if (y > height)
{
yvelocity = -yvelocity
}
if (x < 0)
{
xvelocity = -xvelocity
}
if (y < 0)
{
yvelocity = -yvelocity
}
}
function paddleA()
{
rect(20, ypaddleA, 20, 50)
}
function paddleB()
{
rect(580, ypaddleB, 20, 50)
}
Console
Clear
sketch preview



Sketch A2.8 limiting the paddles

Need to stop them going off the top and bottom.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
```

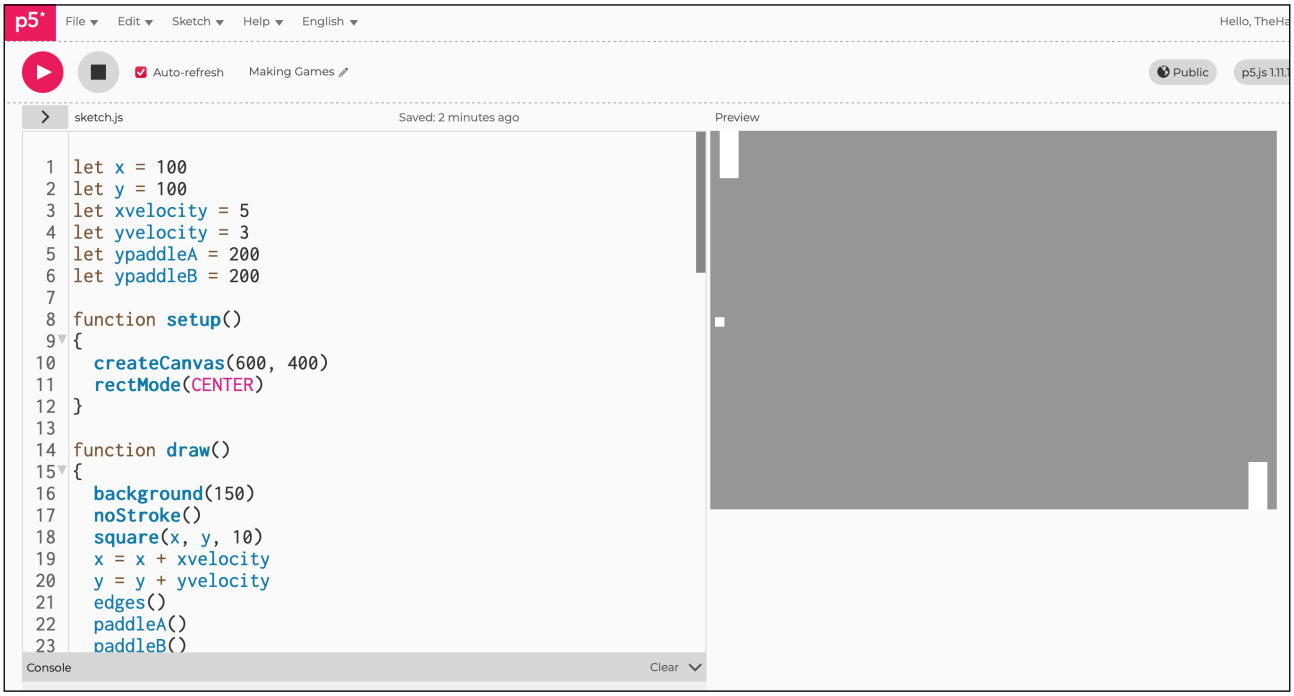
```
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function edges()
{
  if (x > width)
```

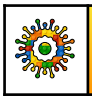
```
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
}
```

🗒️ Notes

The paddles will now stop when they reach the top and bottom.

Figure A2.8


File
Edit
Sketch
Help
English
Auto-refresh
Making Games
Public
sketch.js
Saved: 2 minutes ago
Preview
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200
function setup()
{
createCanvas(600, 400)
rectMode(CENTER)
}
function draw()
{
background(150)
noStroke()
square(x, y, 10)
x = x + xvelocity
y = y + yvelocity
edges()
paddleA()
paddleB()
Console
Clear



Sketch A2.9 pong hits paddle

We need to bounce off the paddles for paddle A and B.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
```

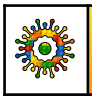
```
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function edges()
{
  if (x > width)
```

```
  {
    xvelocity = -xvelocity
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    xvelocity = -xvelocity
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x == 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
  if (x == 570 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
```

```
}
```

Notes

You should see the pong bounce off the paddles, but you will also note that it can bounce behind the paddles.



Sketch A2.10 respawn the pong

Every time the pong misses the paddle, it respawns in the centre of the canvas and will randomly go left or right.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(600, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
```

```
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function edges()
{
  if (x > width)
```

```
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
```

```
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
}

function paddleB()
{
  rect(580, ypaddleB, 20, 50)
  if (x == 570 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
}
```

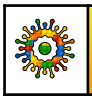
🗒️ Notes

This makes it very difficult to play because the pong is coming very quickly.

🌻 Challenges

There are a number of things to make it a bit more playable.

1. The respawn pong is coming from two-thirds of the canvas, so it has longer to travel.
2. Make the canvas wider, e.g. 800, but you will need to change some of the figures.
3. Slow the pong down a little bit.



Sketch A2.11 widening the canvas

This option is the easiest in terms of changing code, as we simply alter the width of the canvas to 800.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
```

```
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function edges()
{
  if (x > width)
```

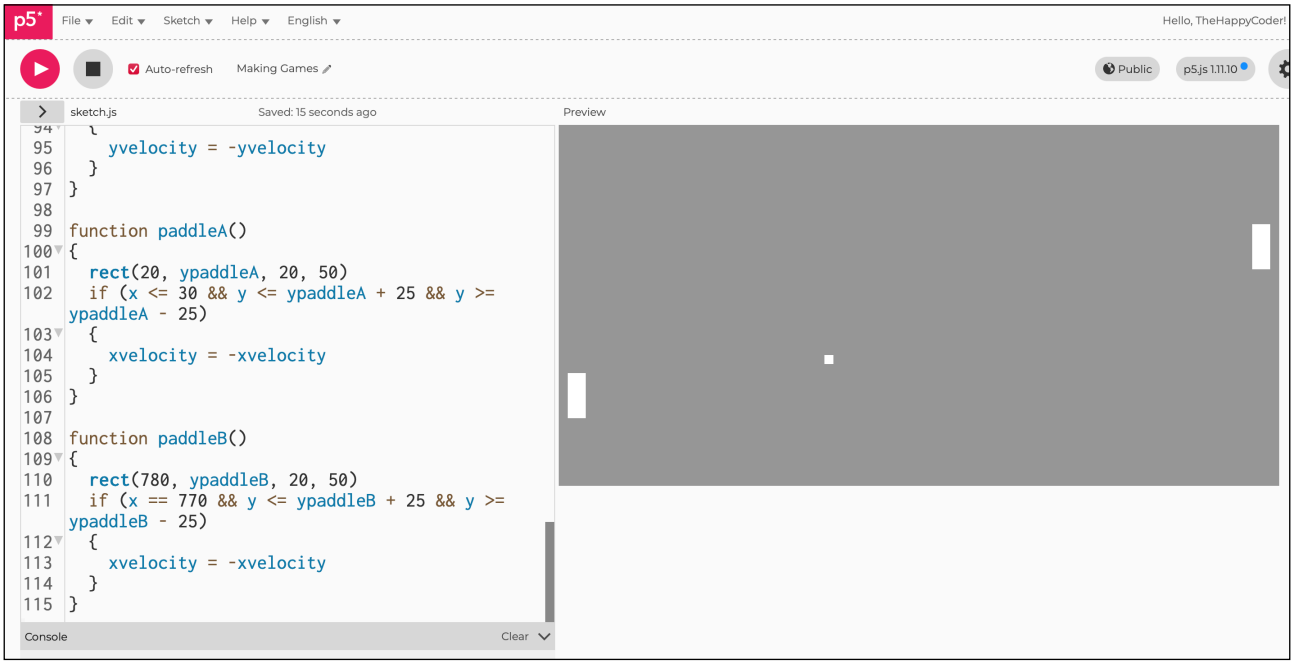
```
{
  x = width/2
  y = height/2
  let direction = random(1)
  if (direction < 0.5)
  {
    xvelocity = xvelocity
  }
  else
  {
    xvelocity = -xvelocity
  }
}
if (y > height)
{
  yvelocity = -yvelocity
}
if (x < 0)
{
  x = width/2
  y = height/2
  let direction = random(1)
  if (direction < 0.5)
  {
    xvelocity = xvelocity
  }
  else
  {
    xvelocity = -xvelocity
  }
}
if (y < 0)
{
```

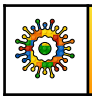
```
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
}
```

🗒️ Notes

This makes it much easier to play or at least less hard.

Figure A2.11


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
yvelocity = -yvelocity
}
}
function paddleA()
{
rect(20, ypaddleA, 20, 50)
if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
{
xvelocity = -xvelocity
}
}
function paddleB()
{
rect(780, ypaddleB, 20, 50)
if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
{
xvelocity = -xvelocity
}
}
Console
Clear



Sketch A2.12 centre line

Adding a centre line.

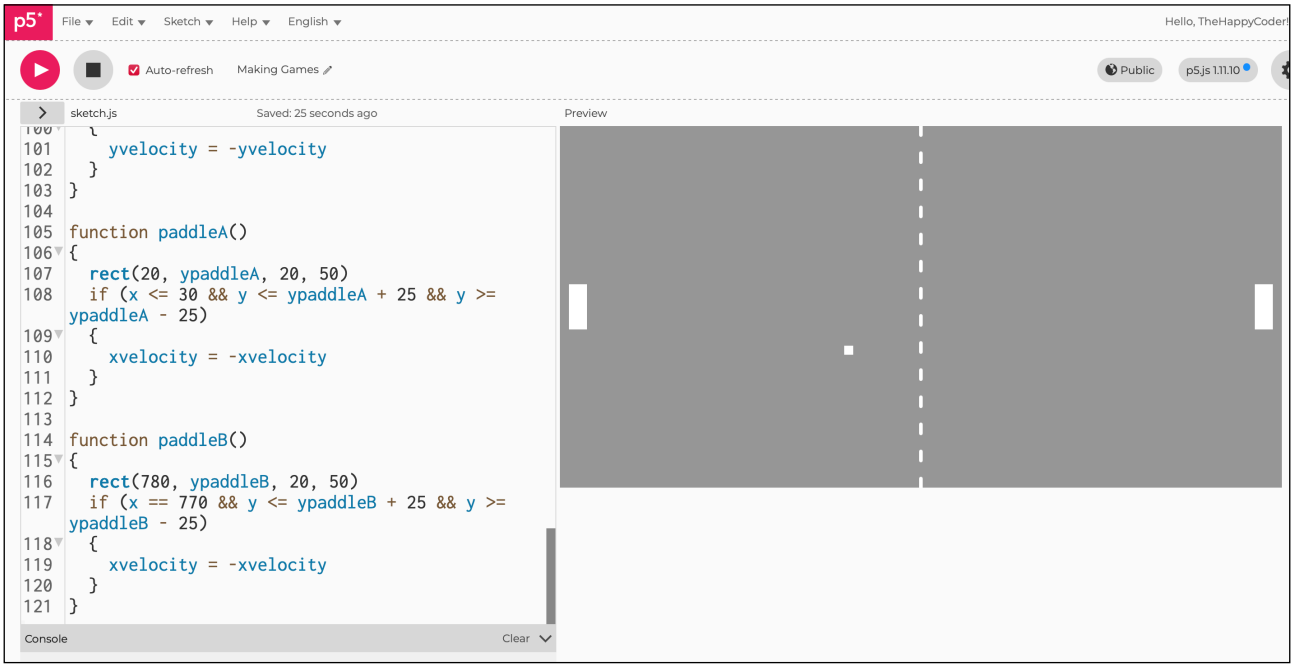
```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()

  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
```

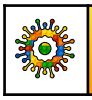
```
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
  strokeWeight(4)
  stroke(255)
  for (let i = 0; i < height; i += 30)
  {
    line(width/2, i, width/2, i + 10)
```

```
  }
}

function edges()
{
  if (x > width)
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
```

```
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
}
```

Figure A2.12


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
yvelocity = -yvelocity
}
}
function paddleA()
{
rect(20, ypaddleA, 20, 50)
if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
{
xvelocity = -xvelocity
}
}
function paddleB()
{
rect(780, ypaddleB, 20, 50)
if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
{
xvelocity = -xvelocity
}
}
Console
Clear



Sketch A2.13 refactoring

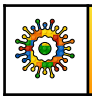
A bit of refactoring to tidy things up before more developments. Move all the key presses into their respective paddle functions and add some variables for keeping the score.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200
let playerA = 0
let playerB = 0
let score = 20

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()
  strokeWeight(4)
```

```
  stroke(255)
  for (let i = 0; i < height; i += 30)
  {
    line(width/2, i, width/2, i + 10)
  }
}

function edges()
{
  if (x > width)
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
```

```
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }

}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
```

```
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
}
```



Sketch A2.14 the game function

Create a game function and add the scores for players. Every time that the pong goes over the edge (misses the paddle), the score is amended.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200
let playerA = 0
let playerB = 0
let score = 20

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
  paddleB()
  strokeWeight(4)
  stroke(255)
```

```
  for (let i = 0; i < height; i += 30)
  {
    line(width/2, i, width/2, i + 10)
  }
  game()
}

function edges()
{
  if (x > width)
  {
    playerA += 1
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    playerB += 1
    x = width/2
    y = height/2
```

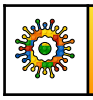
```
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
```

```
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
}
```

```
function game()
{
  textAlign(CENTER, CENTER)
  textSize(25)
  fill(255)
  noStroke()
  text('PLAYER A', width/2 - 100, 20)
  text(playerA, width/2 - 50, 50)
  text('PLAYER B', width/2 + 100, 20)
  text(playerB, width/2 + 50, 50)
}
```

Figure A2.14


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
ypaddleB = 25
}
}
if(keyIsDown(DOWN_ARROW))
{
ypaddleB += 5
if (ypaddleB > 375)
{
ypaddleB = 375
}
}
}
function game()
{
textAlign(CENTER, CENTER)
textSize(25)
fill(255)
noStroke()
text('PLAYER A', width/2 - 100, 20)
text(playerA, width/2 - 50, 50)
text('PLAYER B', width/2 + 100, 20)
text(playerB, width/2 + 50, 50)
}
Console
Clear
PLAYER A
8
PLAYER B
5



Sketch A2.15 the winner is...

We would like a winning screen, so when the score is 20 by either side, then the game stops. We create a boolean variable called `start` and set it to `false`. This is necessary just at the moment but will be shortly. Move `game()` function into the `draw()` function.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200
let playerA = 0
let playerB = 0
let score = 20
let start = false

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
}

function draw()
{
  background(150)
  noStroke()
  square(x, y, 10)
  x = x + xvelocity
  y = y + yvelocity
  edges()
  paddleA()
```

```
  paddleB()
  strokeWeight(4)
  stroke(255)
  for (let i = 0; i < height; i += 30)
  {
    line(width/2, i, width/2, i + 10)
  }
  game()
}

function edges()
{
  if (x > width)
  {
    playerA += 1
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
```

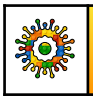
```
    playerB += 1
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
```

```
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
```

```
  }
}

function game()
{
  textAlign(CENTER, CENTER)
  textSize(25)
  fill(255)
  noStroke()
  text('PLAYER A', width/2 - 100, 20)
  text(playerA, width/2 - 50, 50)
  text('PLAYER B', width/2 + 100, 20)
  text(playerB, width/2 + 50, 50)
  if (playerA == score || playerB == score && start == true)
  {
    background(100, 0, 0)
    fill(255)
    if (playerA == score)
    {
      text('PLAYER A won', width/2, height/2)
    }
    if (playerB == score)
    {
      text('PLAYER B won', width/2, height/2)
    }
  }
}
```

🗒️ Notes

You will notice that the player won displays but then the game carries on, we will fix that shortly.



Sketch A2.16 starting and restarting the game

Clicking the mouse to start the game and to restart it after it is finished.

```
let x = 100
let y = 100
let xvelocity = 5
let yvelocity = 3
let ypaddleA = 200
let ypaddleB = 200
let playerA = 0
let playerB = 0
let score = 20
let start = false

function setup()
{
  createCanvas(800, 400)
  rectMode(CENTER)
  background(0, 100, 0)
  fill(255)
  textSize(25)
  textAlign(CENTER, CENTER)
  text('(click to start the game)', width/2, height/2)
  text('PLAYER A use Q for up and A for down', width/2, height/2
+ 50)
  text('PLAYER B use the arrow keys for up and down', width/2,
height/2 + 100)
  text('The first one to 20 points is the winner', width/2,
height/2 - 50)
}

function mousePressed()
{
```

```
  start = true
  playerA = 0
  playerB = 0
}

function draw()
{
  if (start == true)
  {
    background(150)
    noStroke()
    square(x, y, 10)
    x = x + xvelocity
    y = y + yvelocity
    edges()
    paddleA()
    paddleB()
    strokeWeight(4)
    stroke(255)
    for (let i = 0; i < height; i += 30)
    {
      line(width/2, i, width/2, i + 10)
    }
    game()
  }
}

function edges()
{
  if (x > width)
  {
    playerA += 1
    x = width/2
```

```
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y > height)
  {
    yvelocity = -yvelocity
  }
  if (x < 0)
  {
    playerB += 1
    x = width/2
    y = height/2
    let direction = random(1)
    if (direction < 0.5)
    {
      xvelocity = xvelocity
    }
    else
    {
      xvelocity = -xvelocity
    }
  }
  if (y < 0)
  {
    yvelocity = -yvelocity
```

```
  }
}

function paddleA()
{
  rect(20, ypaddleA, 20, 50)
  if (x <= 30 && y <= ypaddleA + 25 && y >= ypaddleA - 25)
  {
    xvelocity = -xvelocity
  }
  if( keyIsDown(81))
  {
    ypaddleA -= 5
    if (ypaddleA < 25)
    {
      ypaddleA = 25
    }
  }
  if( keyIsDown(65))
  {
    ypaddleA += 5
    if (ypaddleA > 375)
    {
      ypaddleA = 375
    }
  }
}

function paddleB()
{
  rect(780, ypaddleB, 20, 50)
  if (x == 770 && y <= ypaddleB + 25 && y >= ypaddleB - 25)
  {
```

```
    xvelocity = -xvelocity
  }
  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
}

function game()
{
  textAlign(CENTER, CENTER)
  textSize(25)
  fill(255)
  noStroke()
  text('PLAYER A', width/2 - 100, 20)
  text(playerA, width/2 - 50, 50)
  text('PLAYER B', width/2 + 100, 20)
  text(playerB, width/2 + 50, 50)
  if (playerA == score || playerB == score && start == true)
  {
    background(100, 0, 0)
```

```
    fill(255)
    if (playerA == score)
    {
      text('PLAYER A won', width/2, height/2)
      text('click to play again', width/2, height/2 + 50)
    }
    if (playerB == score)
    {
      text('PLAYER B won', width/2, height/2)
      text('click to play again', width/2, height/2 + 50)
    }
    if(mousePressed())
    {
      start = true
    }
    start = false
  }
}
```

🌻 Challenges

1. Have the pong speed up after each time a paddle hits it.
2. Make the paddles get smaller as the game goes on.
3. Could you make player A be controlled by the computer, e.g. using lerp()?
4. Add sound each time the pong is hit.

Figure A2.16a

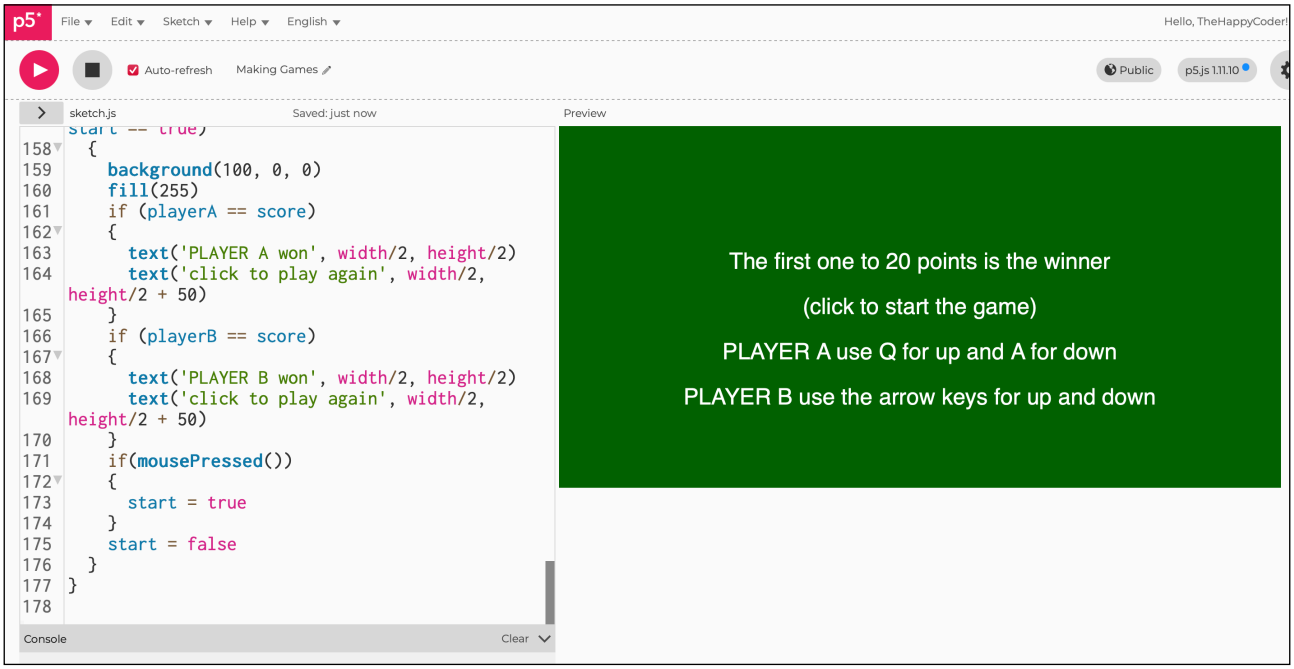


Figure A2.16b