

Section E

Unit #2

Pong



Section E Unit #2 pong game

Sketch E2.1	make the pong
Sketch E2.2	moving the pong
Sketch E2.3	bouncing off the edges
Sketch E2.4	adding a paddle
Sketch E2.5	adding paddle B
Sketch E2.6	moving paddle B
Sketch E2.7	moving paddle A
Sketch E2.8	limiting the paddles
Sketch E2.9	pong hits paddle
Sketch E2.10	respawn the pong
Sketch E2.11	widening the canvas
Sketch E2.12	centre line
Sketch E2.13	refactoring
Sketch E2.14	the game function
Sketch E2.15	the winner is...
Sketch E2.16	starting and restarting the game



Introduction to Unit #2 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.



Sketch E2.1 make the pong

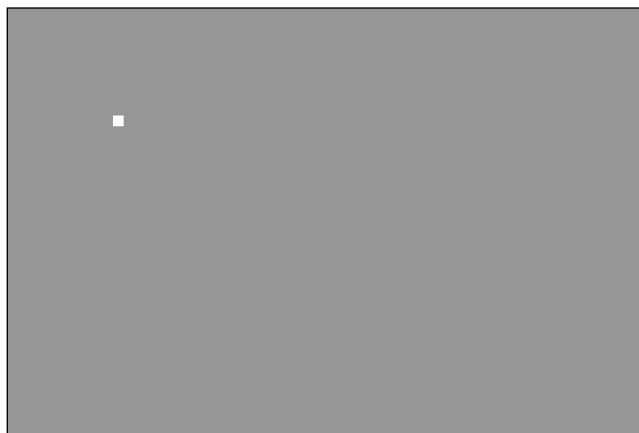
This will be the pong square

```
let x = 100
let y = 100

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

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

The pong





Sketch E2.2 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
}
```



Sketch E2.3 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
}
}
```



Sketch E2.4 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()
}

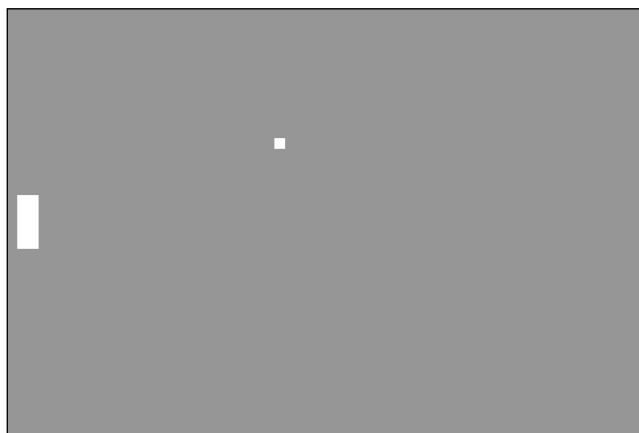
function edges()
{
  if (x > width)
  {
    xvelocity = -xvelocity
  }
}
```



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

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

Paddle A





Sketch E2.5 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)
}
```

Paddle A and paddle B





Sketch E2.6 moving paddle B

We will move paddleB 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



Sketch E2.7 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 and q and a for paddle A



Sketch E2.8 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)
}
```

Top and bottom limiters



Notes

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



Sketch E2.9 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 E2.10 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 2/3 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 E2.11 widening the canvas

This option (2) is the easiest in terms of changing code, se 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
  }
}
```



Sketch E2.12 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)
  }

```

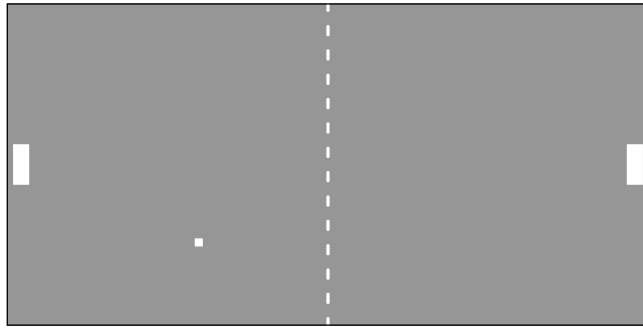
```
}  
}  
  
function edges()  
{  
  if (x > width)  
  {  
    x = width/2  
    y = height/2  
    let direction = random(1)  
    if (direction < 0.5)  
    {  
      xvelocity = xvelocity  
    }  
    else  
    {  
      xvelocity = -xvelocity  
    }  
  }  
  if (y > height)  
  {  
    yvelocity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

Wider and now with a centre line





Sketch E2.13 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 E2.14 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)
  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)
}
```



Sketch E2.15 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() in draw()

```
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)  
    }  
  }  
}  
}
```



Sketch E2.16 starting and restarting the game

Clicking the mouse to start the game and to restart 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)
    {

```

```

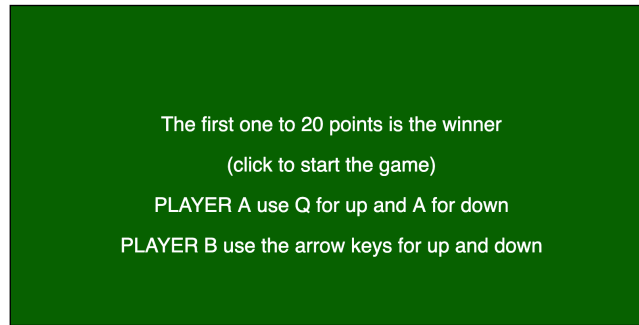
    xvelocity = -xvelocity
  }
  if( keyIsDown(UP_ARROW))
  {
    ypaddleB -= 5
    if (ypaddleB < 25)
    {
      ypaddleB = 25
    }
  }
  if( keyIsDown(DOWN_ARROW))
  {
    ypaddleB += 5
    if (ypaddleB > 375)
    {
      ypaddleB = 375
    }
  }
}

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
}
}
```

Welcome screen



Winning screen



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