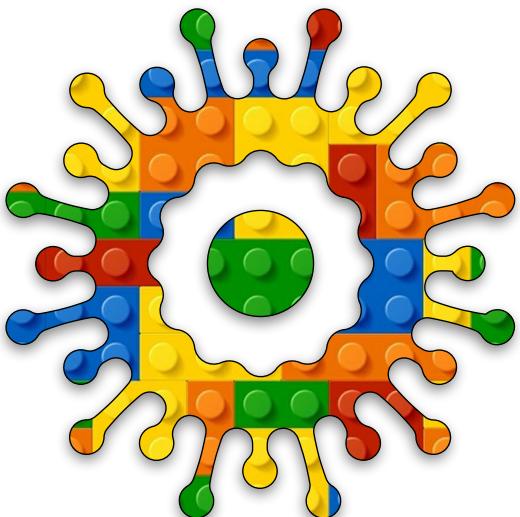


Making Games

Module A

Unit #1

lerp





Section A Unit #1 lerp game

- Sketch A1.1 the enemy
- Sketch A1.2 you are the goody
- Sketch A1.3 seeking you out
- Sketch A1.4 the arrow keys
- Sketch A1.5 the arrow keys
- Sketch A1.6 off the edge
- Sketch A1.7 targets
- Sketch A1.8 measuring the distance
- Sketch A1.9 removing the targets
- Sketch A1.10 error message
- Sketch A1.11 you have won
- Sketch A1.12 you have lost
- Sketch A1.13 increase the speed
- Sketch A1.14 a bit more menacing



Introduction to the lerp game

We are going to make what should be quite a simple game, although it is far from simple as you will see. You will move around the canvas using the arrow keys to collect targets without being caught by a circle following you around.

This uses the `lerp()` function, which measures the distance between two points and incrementally closes that distance in units supplied by you. The bigger the units/steps/increments, the faster it approaches you. It is worked out as a percentage, so as it gets closer, the rate it moves decreases and stops when the distance is zero.



Sketch A1.1 the enemy

Create two variables for the enemy's **x** and **y** coordinates and draw the enemy who will come after you.

```
let xEnemy = 50
let yEnemy = 50

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

function draw()
{
    background(220)
    circle(xEnemy, yEnemy, 20)
}
```

Figure A1.1

The screenshot shows the p5.js web-based editor interface. At the top, there's a red header bar with the p5 logo and navigation links: File ▾, Edit ▾, Sketch ▾, Help ▾, and English ▾. Below the header, there are icons for play, stop, and refresh, followed by the text "Auto-refresh" and "Making Games". The main workspace is titled "sketch.js" and shows the following code:

```
1 let xEnemy = 50
2 let yEnemy = 50
3
4 function setup()
5 {
6   createCanvas(400, 400)
7 }
8
9 function draw()
10 {
11   background(220)
12   circle(xEnemy, yEnemy, 20)
13 }
```

To the right of the code is a "Preview" window showing a gray canvas with a single white circle at the center (at coordinates 50, 50). Below the preview is a "Console" window which is currently empty. At the bottom right of the console is a "Clear" button.



Sketch A1.2 you are the goody

Now we add you to the goody and you are in yellow.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300

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

function draw()
{
    background(220)
    fill('white')
    circle(xEnemy, yEnemy, 20)
    fill('yellow')
    circle(xGood, yGood, 20)
}
```

Figure A1.2

The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for File, Edit, Sketch, Help, and Language (English). Below the toolbar, the title bar says "sketch.js" and "Saved: 15 seconds ago". On the left, the code editor displays the following JavaScript code:

```
1 let xEnemy = 50
2 let yEnemy = 50
3 let xGood = 300
4 let yGood = 300
5
6 function setup()
7 {
8   createCanvas(400, 400)
9 }
10
11 function draw()
12 {
13   background(220)
14   fill('white')
15   circle(xEnemy, yEnemy, 20)
16   fill('yellow')
17   circle(xGood, yGood, 20)
18 }
```

To the right of the code editor is a preview window titled "Preview" which shows a gray square canvas with two circles: a white one at (50, 50) and a yellow one at (300, 300).



Sketch A1.3 seeking you out

We can now demonstrate the `lerp()` function; the enemy will seek you out gradually, slowing down as it gets closer.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300

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

function draw()
{
    background(220)
    fill('white')
    xEnemy = lerp(xEnemy, xGood, 0.01)
    yEnemy = lerp(yEnemy, yGood, 0.01)
    circle(xEnemy, yEnemy, 20)
    fill('yellow')
    circle(xGood, yGood, 20)
}
```

Challenges

1. Play around with the `lerp()` incremental variable, change it from **0.01** to **0.1**.
2. Also, give the **x** and **y** different values.

Figure A1.3

The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for file operations, sketch settings, and language selection (English). Below the toolbar, the title 'sketch.js' is displayed next to a play/pause button and a refresh button. The status bar indicates the file was saved 2 minutes ago.

The main area contains the JavaScript code for the sketch:

```
1 let xEnemy = 50
2 let yEnemy = 50
3 let xGood = 300
4 let yGood = 300
5
6 function setup()
7 {
8   createCanvas(400, 400)
9 }
10
11 function draw()
12 {
13   background(220)
14   fill('white')
15   xEnemy = lerp(xEnemy, xGood, 0.01)
16   yEnemy = lerp(yEnemy, yGood, 0.01)
17   circle(xEnemy, yEnemy, 20)
18   fill('yellow')
19   circle(xGood, yGood, 20)
20 }
```

To the right of the code editor is a preview window showing the current state of the sketch. It features a light gray background with two circles: a white one at the top-left and a yellow one at the bottom-right.



Sketch A1.4 the arrow keys

However, we want to escape the clutches of our arch-enemy, so we are going to use the arrow keys to move you around. We will start with the left arrow key. Try it (first click on the canvas and then press the left arrow key).

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300

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

function draw()
{
    background(220)
    fill('white')
    xEnemy = lerp(xEnemy, xGood, 0.01)
    yEnemy = lerp(yEnemy, yGood, 0.01)
    circle(xEnemy, yEnemy, 20)
    fill('yellow')
    circle(xGood, yGood, 20)

    if (keyIsDown(LEFT_ARROW))
    {
        xGood -= 5
    }
}
```

Figure A1.4

The screenshot shows the p5.js IDE interface. At the top, there's a toolbar with icons for File, Edit, Sketch, Help, and English. Below the toolbar, the title bar says "p5* sketch.js" and "Saved: 1 minute ago". There's also a checkbox for "Auto-refresh" and a link to "Making Games". The main area contains the code for "sketch.js". The code initializes variables for enemy and good positions, sets up a canvas, and handles key input for movement. The preview window on the right shows a yellow circle moving towards a white circle.

```
> sketch.js
1 let yEnemy = 50
2 let xGood = 300
3 let yGood = 300
4
5
6 function setup()
7 {
8   createCanvas(400, 400)
9 }
10
11 function draw()
12 {
13   background(220)
14   fill('white')
15   xEnemy = lerp(xEnemy, xGood, 0.01)
16   yEnemy = lerp(yEnemy, yGood, 0.01)
17   circle(xEnemy, yEnemy, 20)
18   fill('yellow')
19   circle(xGood, yGood, 20)
20
21   if (keyIsDown(LEFT_ARROW))
22   {
23     xGood -= 5
24   }
25 }
```



Sketch A1.5 more arrow keys

Now let us add the other keys and keep away from the enemy.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300

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

function draw()
{
    background(220)
    fill('white')
    xEnemy = lerp(xEnemy, xGood, 0.01)
    yEnemy = lerp(yEnemy, yGood, 0.01)
    circle(xEnemy, yEnemy, 20)
    fill('yellow')
    circle(xGood, yGood, 20)

    if (keyIsDown(LEFT_ARROW))
    {
        xGood -= 5
    }
    if (keyIsDown(RIGHT_ARROW))
    {
        xGood += 5
    }
}
```

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

Figure A1.5

The screenshot shows the p5.js IDE interface. The top bar includes the p5 logo, File, Edit, Sketch, Help, and English options. Below the menu is a toolbar with a play button, a stop button, and an auto-refresh checkbox labeled "Auto-refresh Making Games". The main area has a left sidebar with a file tree showing "sketch.js" and a preview window on the right.

sketch.js code:

```
14  function setup ()  
15  var xEnemy = 100;  
16  var yEnemy = 100;  
17  var xGood = 500;  
18  var yGood = 500;  
19  background(255);  
20  fill('yellow');  
21  circle(xGood, yGood, 20);  
22  fill('black');  
23  circle(xEnemy, yEnemy, 20);  
24  if (keyIsDown(LEFT_ARROW))  
25  {  
26    xGood -= 5;  
27  }  
28  if (keyIsDown(RIGHT_ARROW))  
29  {  
30    xGood += 5;  
31  }  
32  if (keyIsDown(UP_ARROW))  
33  {  
34    yGood -= 5;  
35  }  
36  if (keyIsDown(DOWN_ARROW))  
37  {  
38    yGood += 5;  
39  }  
40  
41  frameRate(30);  
42  
43  function draw ()  
44  {  
45    background(255);  
46    fill('black');  
47    circle(xEnemy, yEnemy, 20);  
48    fill('yellow');  
49    circle(xGood, yGood, 20);  
50  }
```

The preview window shows a black circle at the bottom-left and a yellow circle at the center-right of the canvas.



Sketch A1.6 off the edge

The problem is that you can disappear off the edge. We could do one of two things:

1. Put a barrier so you can't escape beyond the canvas, or
2. Reappear on the opposite side. We will go with the second option because it is easier to play.

We add another function called `edges()` to facilitate this. We call it in `draw()`.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300

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

function draw()
{
    background(220)
    fill('white')
    xEnemy = lerp(xEnemy, xGood, 0.01)
    yEnemy = lerp(yEnemy, yGood, 0.01)
    circle(xEnemy, yEnemy, 20)
    fill('yellow')
    circle(xGood, yGood, 20)

    if (keyIsDown(LEFT_ARROW))
    {
        xGood -= 5
    }
}
```

```
}

if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()
}
```

```
function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {
        xGood = width
    }
    if (yGood > height)
    {
        yGood = 0
    }
    if (yGood < 0)
    {
        yGood = height
    }
}
```

```
}
```

```
}
```

Figure A1.6

The screenshot shows the p5.js IDE interface. At the top, there's a toolbar with icons for play, stop, and refresh, followed by menu items: File, Edit, Sketch, Help, and English. Below the toolbar, the title bar says "sketch.js" and "Saved: 35 seconds ago". To the right of the title bar is a "Preview" button.

The main area contains the following JavaScript code:

```
sketch.js
yGood = 0
edges()
}
function edges()
{
  if (xGood > width)
  {
    xGood = 0
  }
  if (xGood < 0)
  {
    xGood = width
  }
  if (yGood > height)
  {
    yGood = 0
  }
  if (yGood < 0)
  {
    yGood = height
  }
}
```

Below the code editor is a "Console" section with a "Clear" button. In the preview window, there is a yellow circle at the top left and a white circle at the bottom right.



Sketch A1.7 targets

We need targets for you to accumulate as part of the game without being caught by the enemy. We will store them in two arrays for their x and y values. Although we will create 20 of them in `setup()`, we will only display 5 of them at a time in `draw()`.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        fill('red')
        circle(xTarget[i], yTarget[i], 20)
    }
    fill('white')
```

```

xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()
}

function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {

```

```
xGood = width  
}  
if (yGood > height)  
{  
    yGood = 0  
}  
if (yGood < 0)  
{  
    yGood = height  
}  
}
```

Notes

The idea is that you collect them before you get caught, so we need to work out when you have caught one of the targets. We use the **dist()** function.

Figure A1.7

The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play, stop, and refresh, followed by menu items: File ▾, Edit ▾, Sketch ▾, Help ▾, and English ▾. Below the toolbar, a status bar indicates "Saved: 15 seconds ago". The main area is divided into two sections: "sketch.js" on the left and "Preview" on the right.

sketch.js:

```
47 }  
48 }  
49 edges()  
50 }  
51  
52 function edges()  
53 {  
54   if (xGood > width)  
55   {  
56     xGood = 0  
57   }  
58   if (xGood < 0)  
59   {  
60     xGood = width  
61   }  
62   if (yGood > height)  
63   {  
64     yGood = 0  
65   }  
66   if (yGood < 0)  
67   {  
68     yGood = height  
69   }  
70 }
```

Preview:

The preview window shows a light gray background. It contains several red circular shapes of varying sizes, some with black outlines. There is also one yellow circular shape with a black outline. The shapes appear to be representing game objects or particles.



Sketch A1.8 measuring the distance

We create an empty array to store the distances between you and the targets. It is constantly updated as you move around the canvas.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        fill('red')
        circle(xTarget[i], yTarget[i], 20)
        tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
    }
    fill('white')
```

```

xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()
}

function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {

```

```
xGood = width  
}  
if (yGood > height)  
{  
    yGood = 0  
}  
if (yGood < 0)  
{  
    yGood = height  
}  
}
```

Notes

Nothing new to see. We can now see if at any time you are within **20** pixels from centre to centre to a target.



Sketch A1.9 removing the targets

We are then going to remove that target from the array. You will notice that it replaces it with another target from the array. When you run out of targets, you should get an error message.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        fill('red')
        circle(xTarget[i], yTarget[i], 20)
        tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
        if(tDist[i] < 20)
```

```

    {
        xTarget.splice([i], 1)
        yTarget.splice([i], 1)
    }
}

fill('white')
xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()
}

function edges()
{

```

```
if (xGood > width)
{
    xGood = 0
}
if (xGood < 0)
{
    xGood = width
}
if (yGood > height)
{
    yGood = 0
}
if (yGood < 0)
{
    yGood = height
}
```

Notes

Eventually, you get an error message; just keep finding the targets.

Figure A1.9

The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play/pause, stop, and refresh, followed by menu items: File ▾, Edit ▾, Sketch ▾, Help ▾, and English ▾. A checkbox labeled "Auto-refresh" is checked, and the text "Making Games" is displayed.

The main area shows a code editor with the file "sketch.js" open. The code is as follows:

```
> sketch.js
Saved: 15 seconds ago
Preview

54 } yGood = 0
55 }
56 edges()
57 }
58
59 function edges()
60 {
61   if (xGood > width)
62   {
63     xGood = 0
64   }
65   if (xGood < 0)
66   {
67     xGood = width
68   }
69   if (yGood > height)
70   {
71     yGood = 0
72   }
73   if (yGood < 0)
74   {
75     yGood = height
76   }
77 }
```

Below the code editor is a "Console" section with the message "scope." and a "Clear" button. At the bottom, there's a link: "+ More info: <https://p5js.org/examples/data-variable-scope.html> (<https://p5js.org/reference/p5/dist>)".

The preview window on the right shows a yellow circle at the top center and a smaller white circle below it.



Sketch A1.10 error message

To get round the error message because it is trying to draw a circle that has been deleted from that index, we have a check if the **xTarget** or **yTarget** value is null. Also, stop the loop on the last circle.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        if (xTarget[i] != null || yTarget[i] != null)
        {
            fill('red')
            circle(xTarget[i], yTarget[i], 20)
        }
    }
}
```

```

tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
if(tDist[i] < 20)
{
    xTarget.splice([i], 1)
    yTarget.splice([i], 1)
}
}

fill('white')
xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()

```

```
if (xTarget.length <= 0)
{
    noLoop()
}
}
```

```
function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {
        xGood = width
    }
    if (yGood > height)
    {
        yGood = 0
    }
    if (yGood < 0)
    {
        yGood = height
    }
}
```

Notes

No error messages, it just stops because of the **noLoop()** function.

Figure A1.10

p5*

File ▾ Edit ▾ Sketch ▾ Help ▾ English ▾

Auto-refresh Making Games

sketch.js

Saved: just now

Preview

```
63  noLoop()
64 }
65 }
66
67 function edges()
68 {
69   if (xGood > width)
70   {
71     xGood = 0
72   }
73   if (xGood < 0)
74   {
75     xGood = width
76   }
77   if (yGood > height)
78   {
79     yGood = 0
80   }
81   if (yGood < 0)
82   {
83     yGood = height
84   }
85 }
```

Console

Clear



Sketch A1.11 you have won

When we reach the point where you have caught all the targets you want, a sign that says you have won. So instead of just a `noLoop()`, we change the canvas.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        if (xTarget[i] != null || yTarget[i] != null)
        {
            fill('red')
            circle(xTarget[i], yTarget[i], 20)
        }
    }
}
```

```

tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
if(tDist[i] < 20)
{
    xTarget.splice([i], 1)
    yTarget.splice([i], 1)
}
}

fill('white')
xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()

```

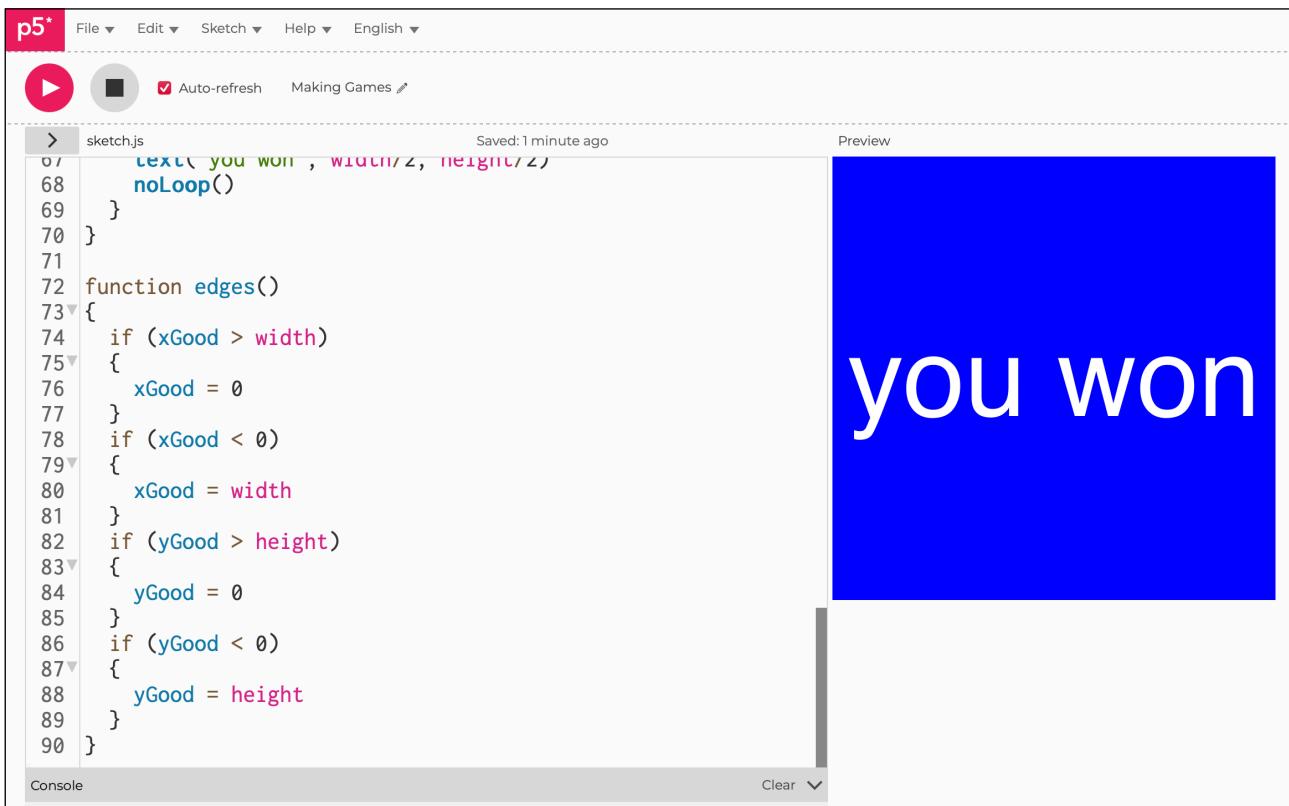
```

if (xTarget.length <= 0)
{
    background('blue')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you won', width/2, height/2)
    noLoop()
}
}

function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {
        xGood = width
    }
    if (yGood > height)
    {
        yGood = 0
    }
    if (yGood < 0)
    {
        yGood = height
    }
}

```

Figure A1.11



The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for file operations, sketch settings, and help. Below the toolbar, the title bar says "sketch.js" and "Saved: 1 minute ago". To the right of the title bar is a "Preview" window displaying a solid blue background with the white text "you won" centered.

```
> sketch.js
 67  text('you won', width/2, height/2)
 68  noLoop()
 69 }
70 }
71
72 function edges(){
73 {
74   if (xGood > width)
75   {
76     xGood = 0
77   }
78   if (xGood < 0)
79   {
80     xGood = width
81   }
82   if (yGood > height)
83   {
84     yGood = 0
85   }
86   if (yGood < 0)
87   {
88     yGood = height
89   }
90 }
```

At the bottom of the editor, there's a "Console" tab and a "Clear" button.



Sketch A1.12 you have lost

But what if you get caught by the enemy? We create a variable to measure that distance called **eDist** for short.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []
let eDist

function setup()
{
    createCanvas(400, 400)
    for (let i = 0; i < 20; i++)
    {
        xTarget[i] = random(width)
        yTarget[i] = random(height)
    }
}

function draw()
{
    background(220)
    for (let i = 0; i < 5; i++)
    {
        if (xTarget[i] != null || yTarget[i] != null)
        {
            fill('red')
            circle(xTarget[i], yTarget[i], 20)
```

```

tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
if(tDist[i] < 20)
{
    xTarget.splice([i], 1)
    yTarget.splice([i], 1)
}
}

fill('white')
xEnemy = lerp(xEnemy, xGood, 0.01)
yEnemy = lerp(yEnemy, yGood, 0.01)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()

```

```

if (xTarget.length <= 0)
{
    background('blue')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you won', width/2, height/2)
    noLoop()
}

eDist = dist(xEnemy, yEnemy, xGood, yGood)
if (eDist < 20)
{
    background('green')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you lost', width/2, height/2)
    noLoop()
}

function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {
        xGood = width
    }
    if (yGood > height)
    {

```

```
yGood = 0
}
if (yGood < 0)
{
    yGood = height
}
}
```

Figure A1.12

The screenshot shows the p5.js code editor interface. At the top, there's a toolbar with icons for play, stop, and refresh, followed by menu items: File, Edit, Sketch, Help, and English. Below the toolbar, the title "sketch.js" is displayed along with the status "Saved: just now". To the right of the code area is a preview window showing a dark green background with the white text "you lost" centered.

```
10  text('you lost', width/2, height/2)
11  noLoop()
12 }
13
14 function edges()
15 {
16   if (xGood > width)
17   {
18     xGood = 0
19   }
20   if (xGood < 0)
21   {
22     xGood = width
23   }
24   if (yGood > height)
25   {
26     yGood = 0
27   }
28   if (yGood < 0)
29   {
30     yGood = height
31   }
32 }
```

At the bottom left is a "Console" tab, and at the bottom right is a "Clear" button.



Sketch A1.13 increase the speed

We are going to give the enemy's speed a variable name `enemySpeed` and set it to `0.01`, but during the game that speed will increase incrementally each time you catch a target. You will notice that it speeds up quite a lot by the end.

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []
let eDist
let enemySpeed = 0.01

function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 20; i++)
  {
    xTarget[i] = random(width)
    yTarget[i] = random(height)
  }
}

function draw()
{
  background(220)
  for (let i = 0; i < 5; i++)
  {
    if (xTarget[i] != null || yTarget[i] != null)
```

```

{
    fill('red')
    circle(xTarget[i], yTarget[i], 20)
    tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
    if(tDist[i] < 20)
    {
        xTarget.splice([i], 1)
        yTarget.splice([i], 1)
        enemySpeed += 0.001
    }
}

fill('white')
xEnemy = lerp(xEnemy, xGood, enemySpeed)
yEnemy = lerp(yEnemy, yGood, enemySpeed)
circle(xEnemy, yEnemy, 20)
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{
    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))

```

```

{
    yGood += 5
}
edges()
if (xTarget.length <= 0)
{
    background('blue')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you won', width/2, height/2)
    noLoop()
}
eDist = dist(xEnemy, yEnemy, xGood, yGood)
if (eDist < 20)
{
    background('green')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you lost', width/2, height/2)
    noLoop()
}
}

function edges()
{
    if (xGood > width)
    {
        xGood = 0
    }
    if (xGood < 0)
    {

```

```
xGood = width  
}  
if (yGood > height)  
{  
    yGood = 0  
}  
if (yGood < 0)  
{  
    yGood = height  
}  
}
```

Notes

There is nothing new to show you here; it just gets faster and faster.

Challenge

How would you put a limit on its maximum speed?



Sketch A1.14 a bit more menacing

Let's make the enemy a little more menacing

```
let xEnemy = 50
let yEnemy = 50
let xGood = 300
let yGood = 300
let xTarget = []
let yTarget = []
let tDist = []
let eDist
let enemySpeed = 0.01
let r = 0

function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 20; i++)
  {
    xTarget[i] = random(width)
    yTarget[i] = random(height)
  }
}

function draw()
{
  background(220)
  for (let i = 0; i < 5; i++)
  {
    if (xTarget[i] != null || yTarget[i] != null)
    {
```

```

    fill('red')
    circle(xTarget[i], yTarget[i], 20)
    tDist[i] = dist(xGood, yGood, xTarget[i], yTarget[i])
    if(tDist[i] < 20)
    {
        xTarget.splice([i], 1)
        yTarget.splice([i], 1)
        enemySpeed += 0.001
    }
}

fill('black')
xEnemy = lerp(xEnemy, xGood, enemySpeed)
yEnemy = lerp(yEnemy, yGood, enemySpeed)
push()
strokeWeight(2)
translate(xEnemy, yEnemy)
rotate(r)
circle(0, 0, 10)
line(-10, 0, 10, 0)
line(0, -10, 0, 10)
r += 0.9
pop()
fill('yellow')
circle(xGood, yGood, 20)

if (keyIsDown(LEFT_ARROW))
{
    xGood -= 5
}
if (keyIsDown(RIGHT_ARROW))
{

```

```

    xGood += 5
}
if (keyIsDown(UP_ARROW))
{
    yGood -= 5
}
if (keyIsDown(DOWN_ARROW))
{
    yGood += 5
}
edges()
if (xTarget.length <= 0)
{
    background('blue')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you won', width/2, height/2)
    noLoop()
}
eDist = dist(xEnemy, yEnemy, xGood, yGood)
if (eDist < 20)
{
    background('green')
    textSize(100)
    fill('white')
    textAlign(CENTER, CENTER)
    text('you lost', width/2, height/2)
    noLoop()
}
}

function edges()

```

```
{  
    if (xGood > width)  
    {  
        xGood = 0  
    }  
    if (xGood < 0)  
    {  
        xGood = width  
    }  
    if (yGood > height)  
    {  
        yGood = 0  
    }  
    if (yGood < 0)  
    {  
        yGood = height  
    }  
}
```

Challenges

1. Change the background.
2. Make the enemy move increasingly faster.
3. Have the targets move around.
4. Add a PNG image for the targets, you and the enemy.
5. You could have a score at the top.
6. You could have lives.
7. Instead of reappearing on the opposite side, you could stop at the edge.

Figure A1.14

The screenshot shows the p5.js IDE interface. At the top, there's a toolbar with icons for play, stop, and refresh, followed by menu items: File, Edit, Sketch, Help, and English. A status bar indicates "Saved: 15 seconds ago" and "Auto-refresh" is checked. The main area has tabs for "sketch.js" and "Preview". The code editor contains the following JavaScript code:

```
> sketch.js
Saved: 15 seconds ago
Auto-refresh Making Games

89 text("you lost", width/2, height/2)
90 noLoop()
91 }
92 }
93
94 function edges(){
95 {
96   if (xGood > width)
97   {
98     xGood = 0
99   }
100  if (xGood < 0)
101  {
102    xGood = width
103  }
104  if (yGood > height)
105  {
106    yGood = 0
107  }
108  if (yGood < 0)
109  {
110    yGood = height
111  }
112 }
```

The preview window shows a black bird-like character in the center of a gray rectangular area. There are several red circular obstacles scattered around the perimeter. One yellow circular object is also visible.