

物聯網實務

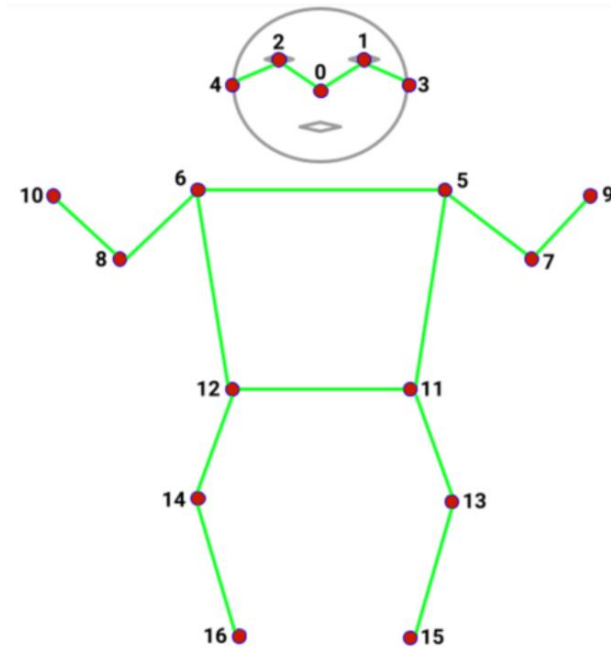
12_21

Project demo

- Motion Sensing Game

Introduction

- This project aims to create an application that uses a facial feature tracker and converts the status of individual facial features to input for a ping-pong game.



TensorFlow.js

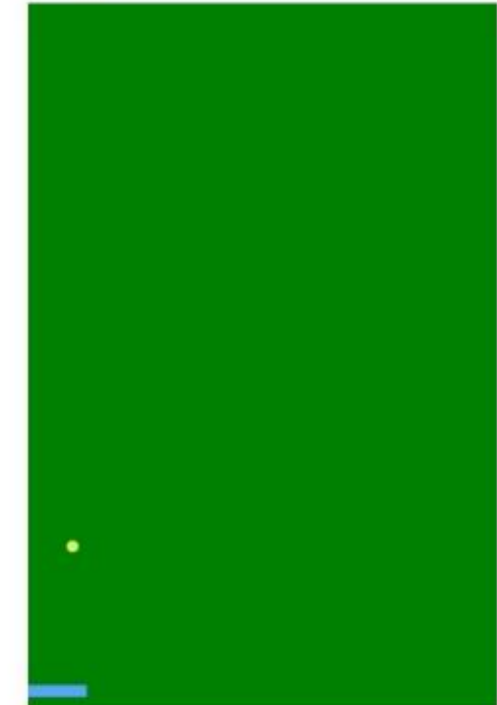
Score: 0

countdown: 5

Game is running

Bestscore:
17

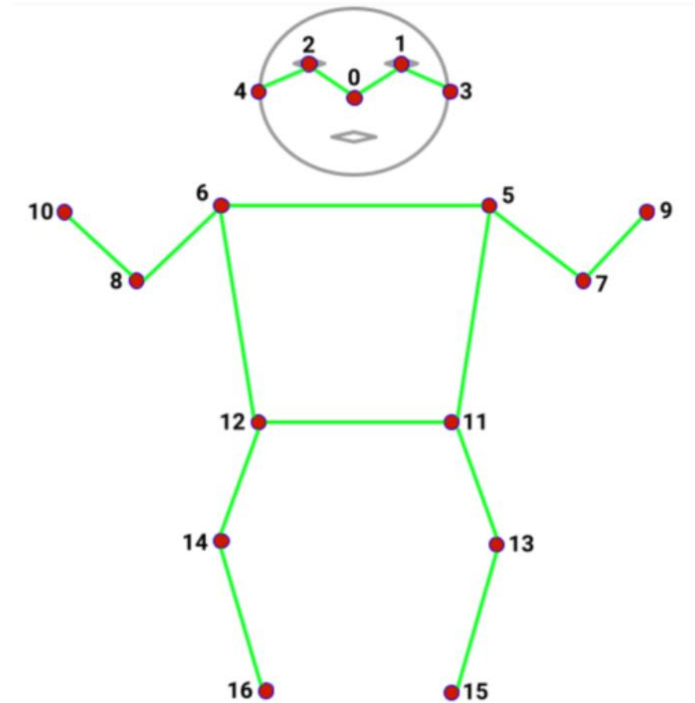
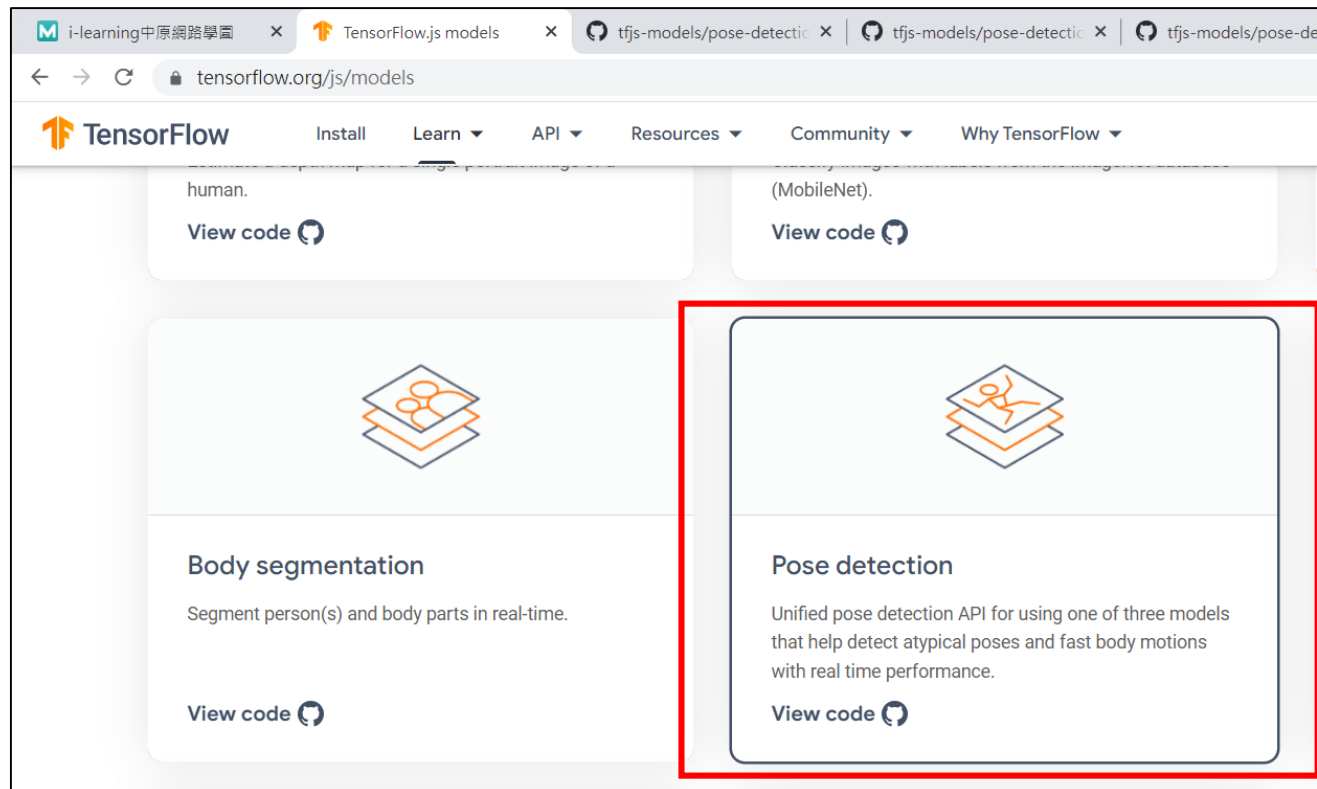
Loading model...



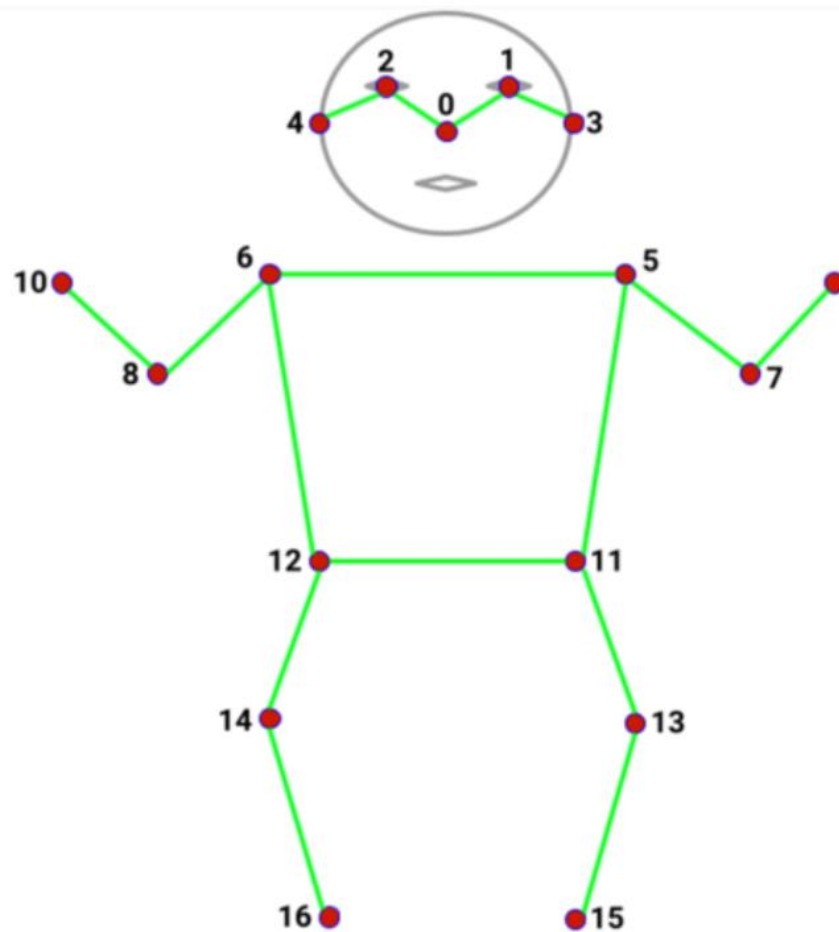
PoseNet

- PoseNet is a vision model that can be used to estimate the pose of a person in an image or video by estimating where key body joints are.

<https://www.tensorflow.org/js/models>



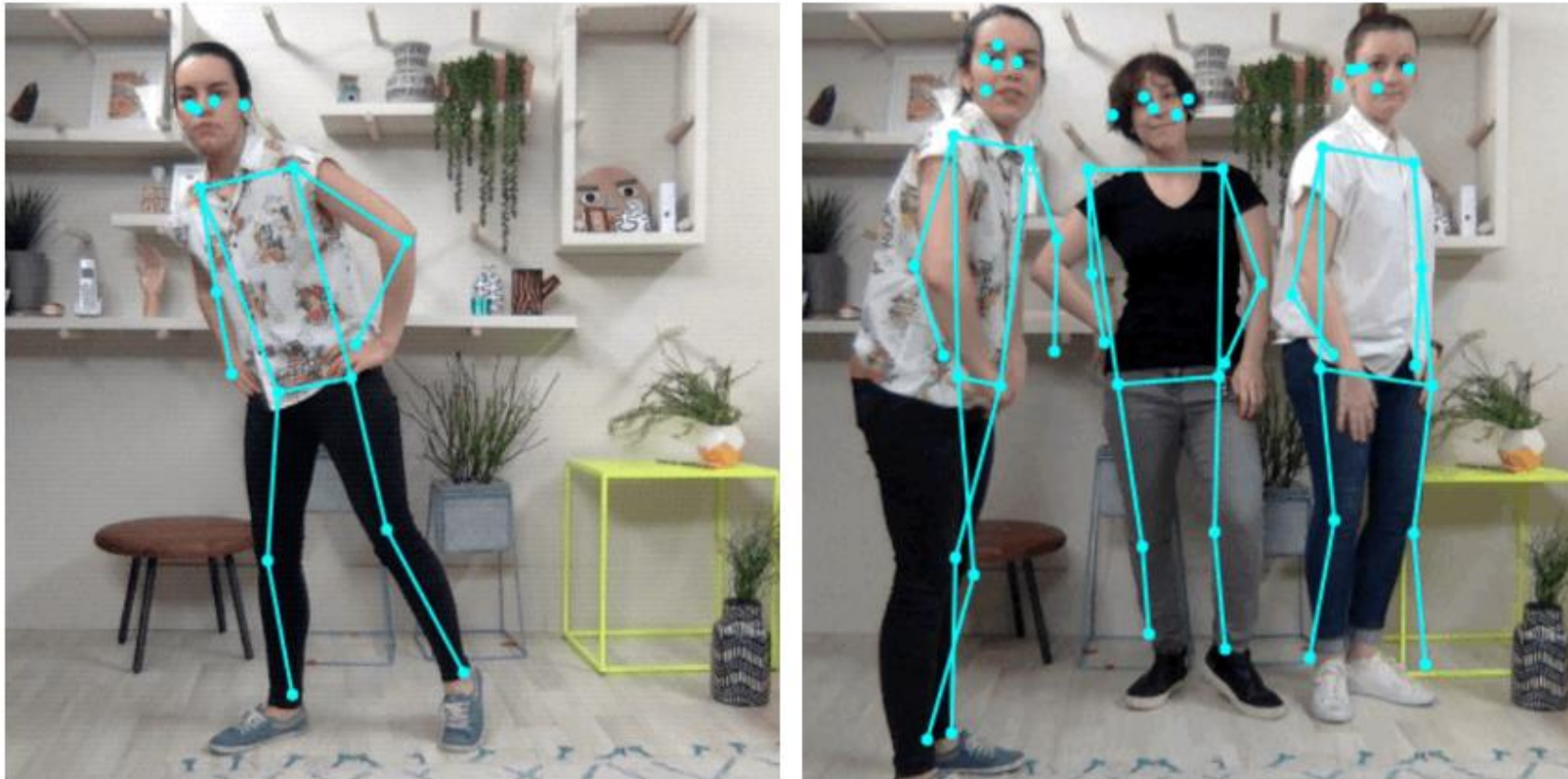
COCO Keypoints: Used in MoveNet and PoseNet



- 0: nose
- 1: left_eye
- 2: right_eye
- 3: left_ear
- 4: right_ear
- 5: left_shoulder
- 6: right_shoulder
- 7: left_elbow
- 8: right_elbow
- 9: left_wrist
- 10: right_wrist
- 11: left_hip
- 12: right_hip
- 13: left_knee
- 14: right_knee
- 15: left_ankle
- 16: right_ankle



Real-time Human Pose Estimation in the Browser with TensorFlow.js



ml5.js

ml5.js - Friendly Machine Learning for the Web

Welcome to the ml5.js documentation. Here you'll find everything you need to get up and started with ml5.

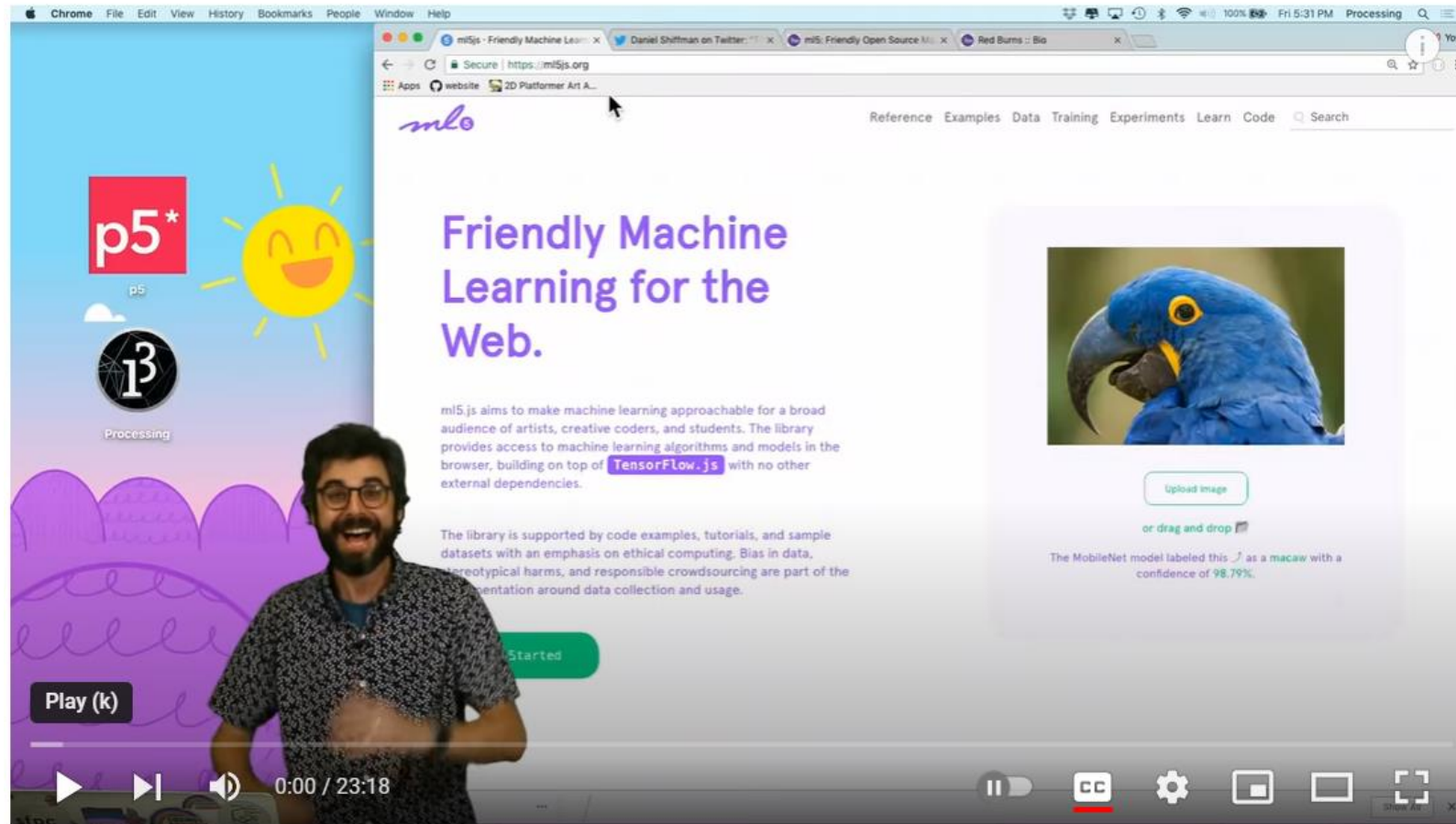
Getting Started

Take a ride on the Coding Train to watch Dan Shiffman's ["A Beginner's Guide to Machine Learning with ml5.js"](#). Here Dan explains what ml5.js is and where it all comes from.

ml5.js is machine learning *for the web* in your web browser. Through some clever and exciting advancements, the folks building [TensorFlow.js](#) figured out that it is possible to use the web browser's built in graphics processing unit (GPU) to do calculations that would otherwise run very slowly using central processing unit (CPU). A really nice explanation of what is happening with GPUs can be found [here - Why are shaders fast?](#). ml5 strives to make all these new developments in machine learning on the web more approachable for everyone.

<https://learn.ml5js.org/#/>

A Beginner's Guide to Machine Learning with ml5.js



<https://p5js.org/get-started/>

p5.js

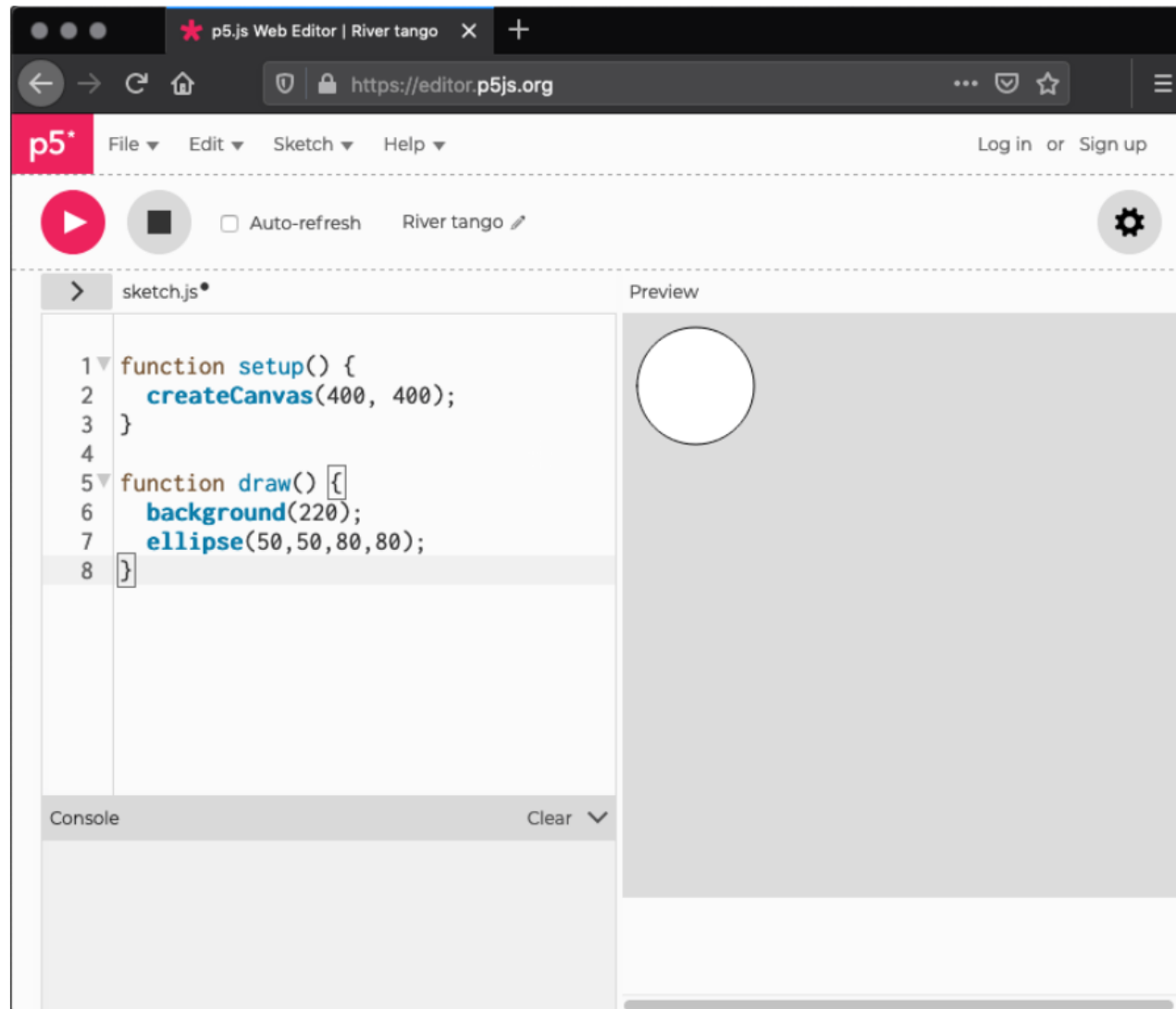
Hello!

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

Using the metaphor of a sketch, p5.js has a full set of drawing functionality. However, you're not limited to your drawing canvas. You can think of your whole browser page as your sketch, including HTML5 objects for text, input, video, webcam, and sound.

Start creating with the p5 Editor!

<https://p5js.org/get-started/>



Using ml5.js & p5.js

- `<script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript">`
- `<script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js">
</script>`

<https://learn.ml5js.org/#/>

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Getting Started with ml5.js</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- p5 -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>
    <!-- ml5 -->
    <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
  </head>

  <body>
    <script>
      // Your code will go here
      // open up your console - if everything loaded properly you should see the latest ml5 version
      console.log('ml5 version:', ml5.version);

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

      function draw() {
        background(200);
      }
    </script>
  </body>
</html>
```

Copy to clipboard

Copy

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

The image shows a screenshot of the W3Schools HTML Tryit editor. The left pane contains HTML and JavaScript code for a p5.js sketch. The right pane shows the rendered output: a black rectangle on a white background. Blue dimension lines indicate the rectangle's width is 100 and its height is 200. Three blue callout boxes point to specific values in the code: (100,200) points to the createCanvas function arguments, 5 points to the background function argument, and a 'Paste' box points to the console.log statement.

```
<html lang="en">
<head>
  <title>Getting Started with ml5.js</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- p5 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js">
</script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js">
</script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
</head>

<body>
  <script>
    // Your code will go here
    // open up your console - if everything loaded properly you should see the latest
ml5 version
    console.log('ml5 version:', ml5.version);

    function setup() {
      createCanvas(100, 200);
    }

    function draw() {
      background(5);
    }
  </script>
</body>
</html>
```

Paste

(100,200)

5

100

200

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

The screenshot shows the W3Schools HTML Tryit editor interface. The left pane contains the following HTML code:

```
<html lang="en">
<head>
  <title>Getting Started with ml5.js</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- p5 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js">
</script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js">
</script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
</head>

<body>
  <script>
    // Your code will go here
    // open up your console - if everything loaded properly you should see the latest
ml5 version
    console.log('ml5 version:', ml5.version);

    function setup() {
      createCanvas(300, 200);
    }

    function draw() {
      background(195);
    }
  </script>
</body>
</html>
```

The right pane displays the rendered output: a gray square with a width of 300 and a height of 200. Two blue callout boxes provide additional context: one points to the values 300 and 200 in the `createCanvas()` function, and the other points to the value 195 in the `background()` function.

PoseNet



<https://learn.ml5js.org/#/reference/posenet>

Examples

p5.js

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

p5 web editor

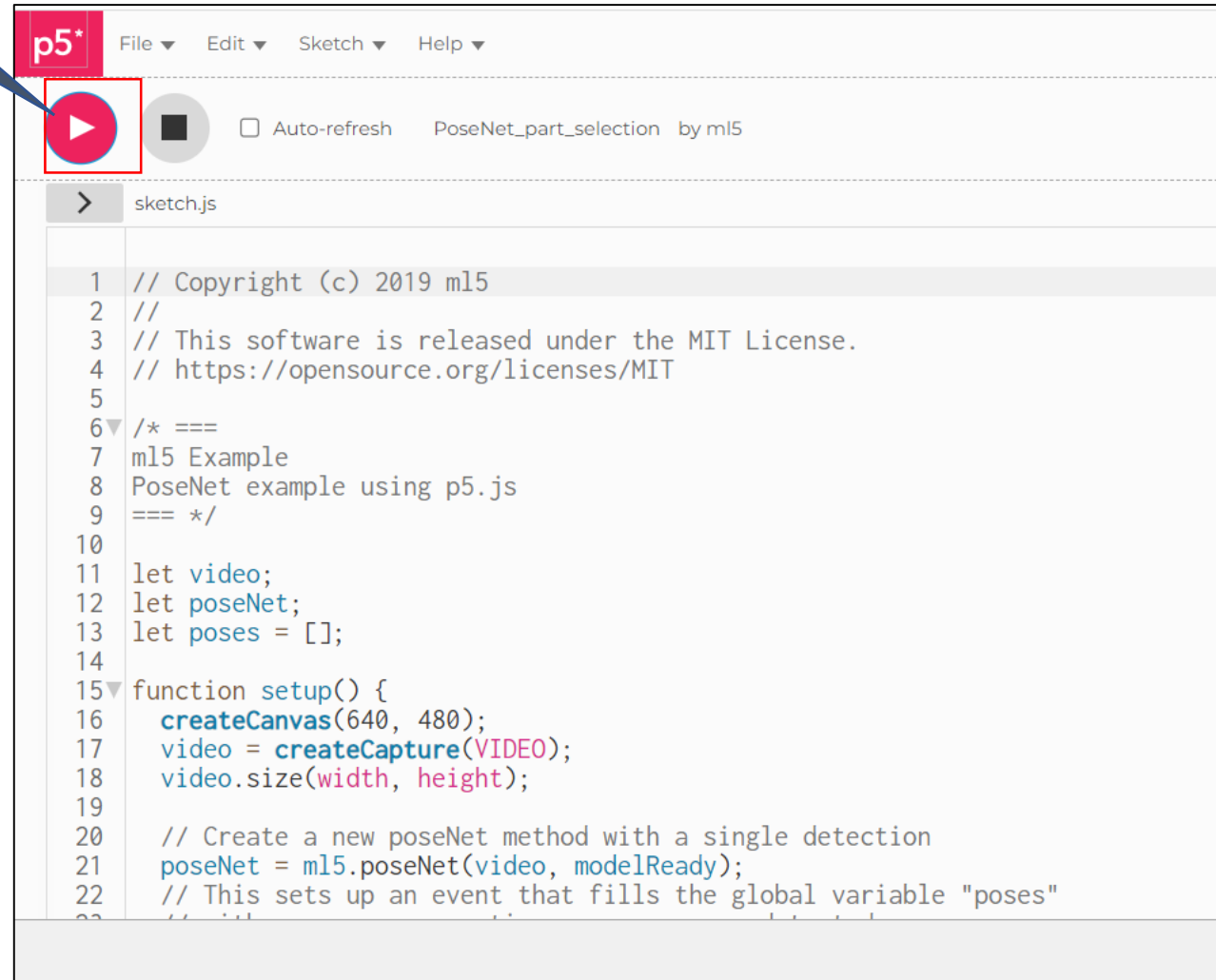
- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

plain javascript

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

Run

Select



The screenshot shows the p5.js web editor interface. At the top, there is a menu bar with 'File', 'Edit', 'Sketch', and 'Help'. Below the menu bar, there is a toolbar with a red 'p5*' logo, a play button (highlighted with a red box and a 'Run' callout), a square button, and a checkbox labeled 'Auto-refresh'. The main area displays a code editor with a file named 'sketch.js'. The code in the editor is as follows:

```
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet example using p5.js
9 === */
10
11 let video;
12 let poseNet;
13 let poses = [];
14
15 function setup() {
16   createCanvas(640, 480);
17   video = createCapture(VIDEO);
18   video.size(width, height);
19
20   // Create a new poseNet method with a single detection
21   poseNet = ml5.poseNet(video, modelReady);
22   // This sets up an event that fills the global variable "poses"
```

☐ Auto-refresh

PoseNet_part_selection by ml5



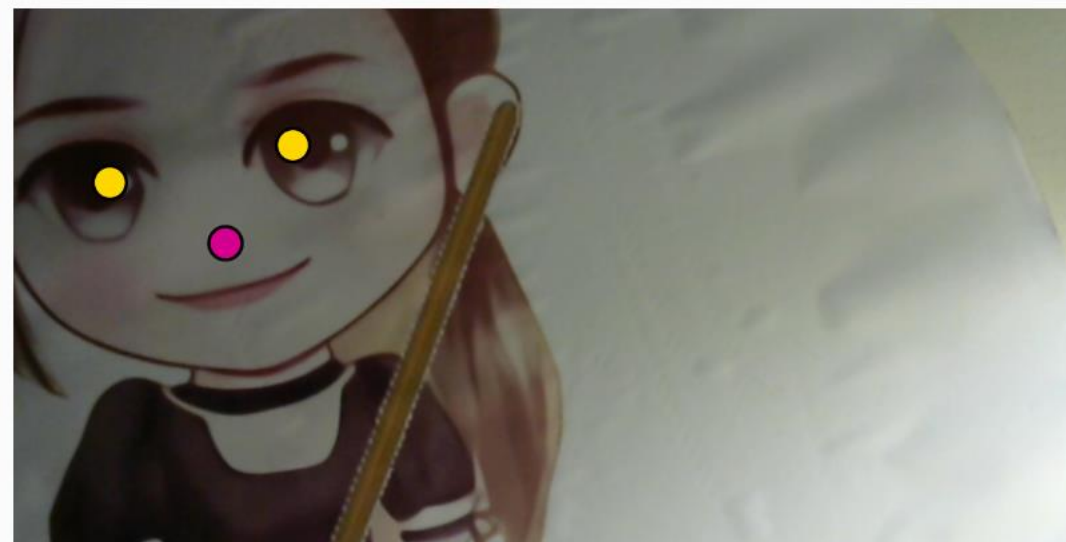
sketch.js

```
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet example using p5.js
9 === */
10
11 let video;
12 let poseNet;
13 let poses = [];
14
15 function setup() {
16   createCanvas(640, 480);
17   video = createCapture(VIDEO);
18   video.size(width, height);
19
20   // Create a new poseNet method with a single detection
21   poseNet = ml5.poseNet(video, modelReady);
22   // This sets up an event that fills the global variable "poses"
23   // with the current list of detected poses when ml5.poseNet is
```

Preview

PoseNet example with feature selection using p5.js

Model Loaded



Examples

p5.js

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

p5 web editor

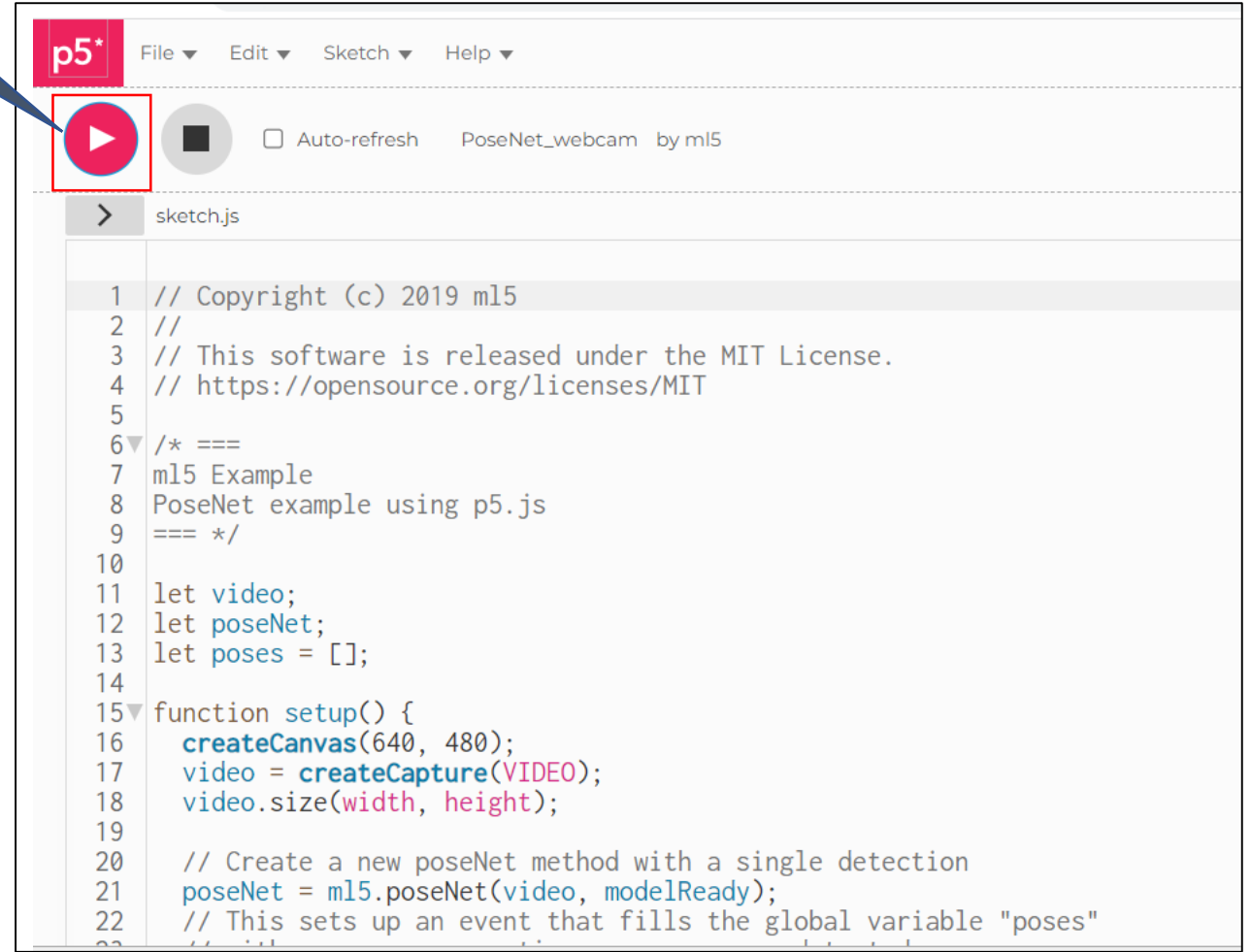
- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

plain javascript

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

Run

Select



```
p5* File Edit Sketch Help
p5
Auto-refresh PoseNet_webcam by ml5
sketch.js
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet example using p5.js
9 === */
10
11 let video;
12 let poseNet;
13 let poses = [];
14
15 function setup() {
16   createCanvas(640, 480);
17   video = createCapture(VIDEO);
18   video.size(width, height);
19
20   // Create a new poseNet method with a single detection
21   poseNet = ml5.poseNet(video, modelReady);
22   // This sets up an event that fills the global variable "poses"
```

editor.p5js.org/ml5/sketches/PoseNet_webcam

p5*

File Edit Sketch Help

English Log in

▶

Auto-refresh

PoseNet_webcam by ml5

>

sketch.js

```
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet example using p5.js
9 === */
10
11 let video;
12 let poseNet;
13 let poses = [];
14
15 function setup() {
16   createCanvas(640, 480);
17   video = createCapture(VIDEO);
18   video.size(width, height);
19
20   // Create a new poseNet method with a single detection
21   poseNet = ml5.poseNet(video, modelReady);
22   // This sets up an event that fills the global variable "poses"
23   // with the current list of detected poses when ml5.poseNet
```

Preview

PoseNet example using p5.js

Model Loaded



← → ↺

editor.p5js.org/ml5/sketches/PoseNet_webcam

English Log in

p5*

File Edit Sketch Help

▶

Auto-refresh

PoseNet_webcam by ml5


sketch.js

```
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet example using p5.js
9 === */
10
11 let video;
12 let poseNet;
13 let poses = [];
14
15 function setup() {
16   createCanvas(640, 480);
17   video = createCapture(VIDEO);
18   video.size(width, height);
19
20   // Create a new poseNet method with a single detection
21   poseNet = ml5.poseNet(video, modelReady);
22   // This sets up an event that fills the global variable "poses"
```

Preview

PoseNet example using p5.js

Model Loaded



Examples

p5.js

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

p5 web editor

- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)


plain javascript


- [PoseNet_image_single](#)
- [PoseNet_part_selection](#)
- [PoseNet_webcam](#)

1.Select

The screenshot shows the GitHub interface for the `ml5js / ml5-library` repository. The repository is public and has 221 issues, 64 pull requests, and 11 projects. The breadcrumb navigation shows the path: `ml5-library / examples / javascript / PoseNet / PoseNet_webcam /`. A commit by `bomanimc` titled "Fix j++ unary operators" is visible. Below the commit, there is a list of files. The file `index.html` is highlighted with a red box, and a callout bubble labeled "2.Select" points to it. The file `sketch.js` is also listed. The commit message "Fix j++ unary operators" is visible next to `sketch.js`.

main ml5-library / examples / javascript / PoseNet / PoseNet_webcam / index.html

 **bomanimc** Add the entire examples folder taken from the development branch of m... [...](#)

 1 contributor

Executable File | 19 lines (13 sloc) | 394 Bytes

```
1 <html>
2
3 <head>
4   <meta charset="UTF-8">
5   <title>PoseNet Example</title>
6
7   <script src="http://localhost:8080/ml5.js" type="text/javascript"></script>
8
9 </head>
10
11 <body>
12   <h1>PoseNet Example</h1>
13   <canvas id="canvas" width="640" height="480"></canvas>
14   <video id="video" width="640" height="480" autoplay style="display: none"></video>
15
16   <script src="sketch.js"></script>
17 </body>
18
19 </html>
```

Copy

Replace



The image shows a web editor interface with a sidebar containing 'Home', 'HTML', and 'Try HTML' tabs. Below the tabs are icons for a menu, a file, a refresh, and a moon, followed by a green 'Run' button with a right arrow. The main editor area contains HTML code for a 'PoseNet Example'. A red rectangular box highlights the following line of code:

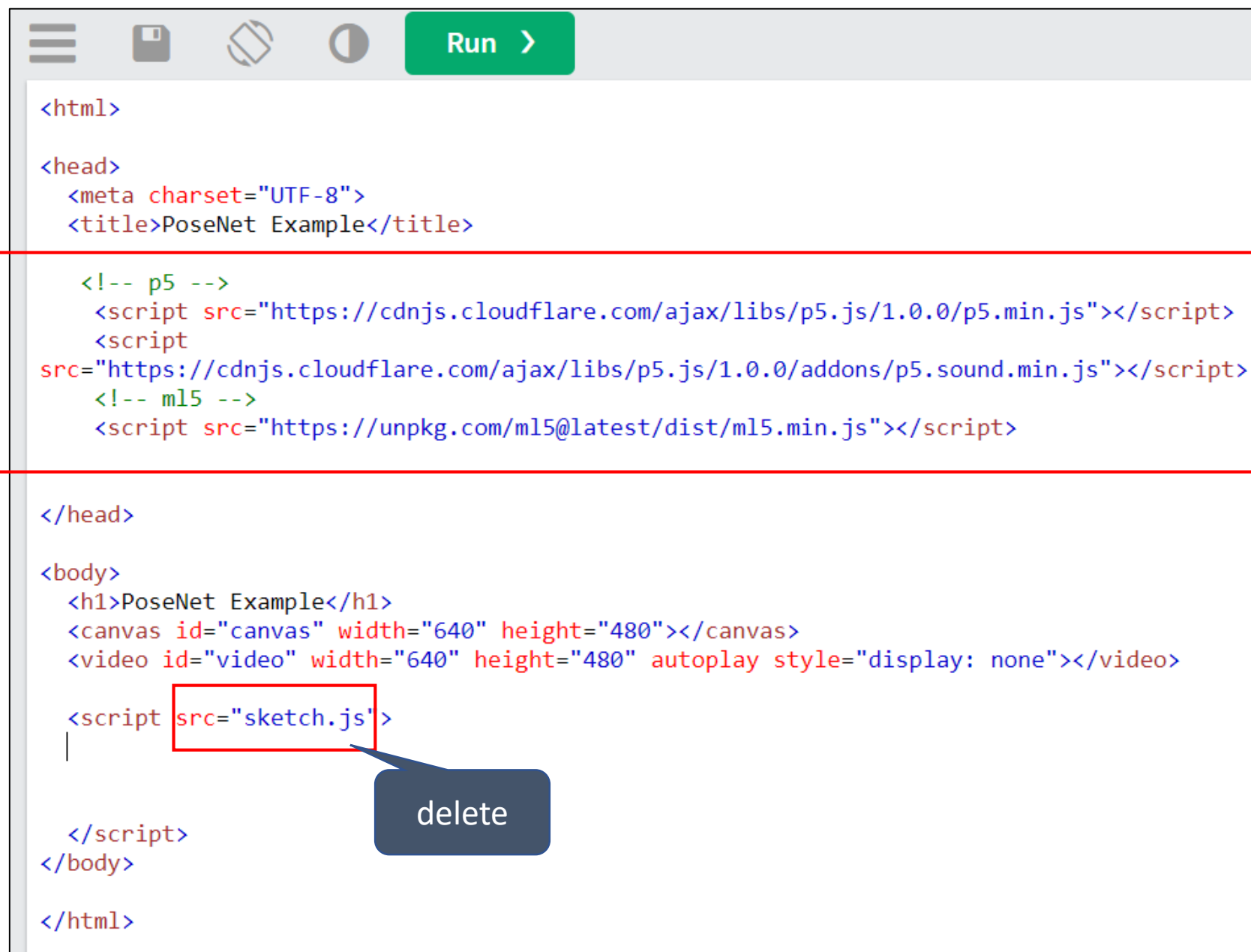
```
<script src="http://localhost:8080/ml5.js" type="text/javascript"></script>
```

A dark blue speech bubble points to this line, containing the replacement code:

```
<!-- p5 -->  
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>  
<!-- ml5 -->  
<script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
```

The full HTML code in the editor is as follows:

```
<html>  
  
<head>  
  <meta charset="UTF-8">  
  <title>PoseNet Example</title>  
  
  <script src="http://localhost:8080/ml5.js" type="text/javascript"></script>  
</head>  
  
<body>  
  <h1>PoseNet Example</h1>  
  <canvas id="canvas" width="640" height="480"></canvas>  
  <video id="video" width="640" height="480" autoplay style="display: none"></video>  
  
  <script src="sketch.js"></script>  
</body>  
  
</html>
```



```
<html>

<head>
  <meta charset="UTF-8">
  <title>PoseNet Example</title>

  <!-- p5 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>

</head>

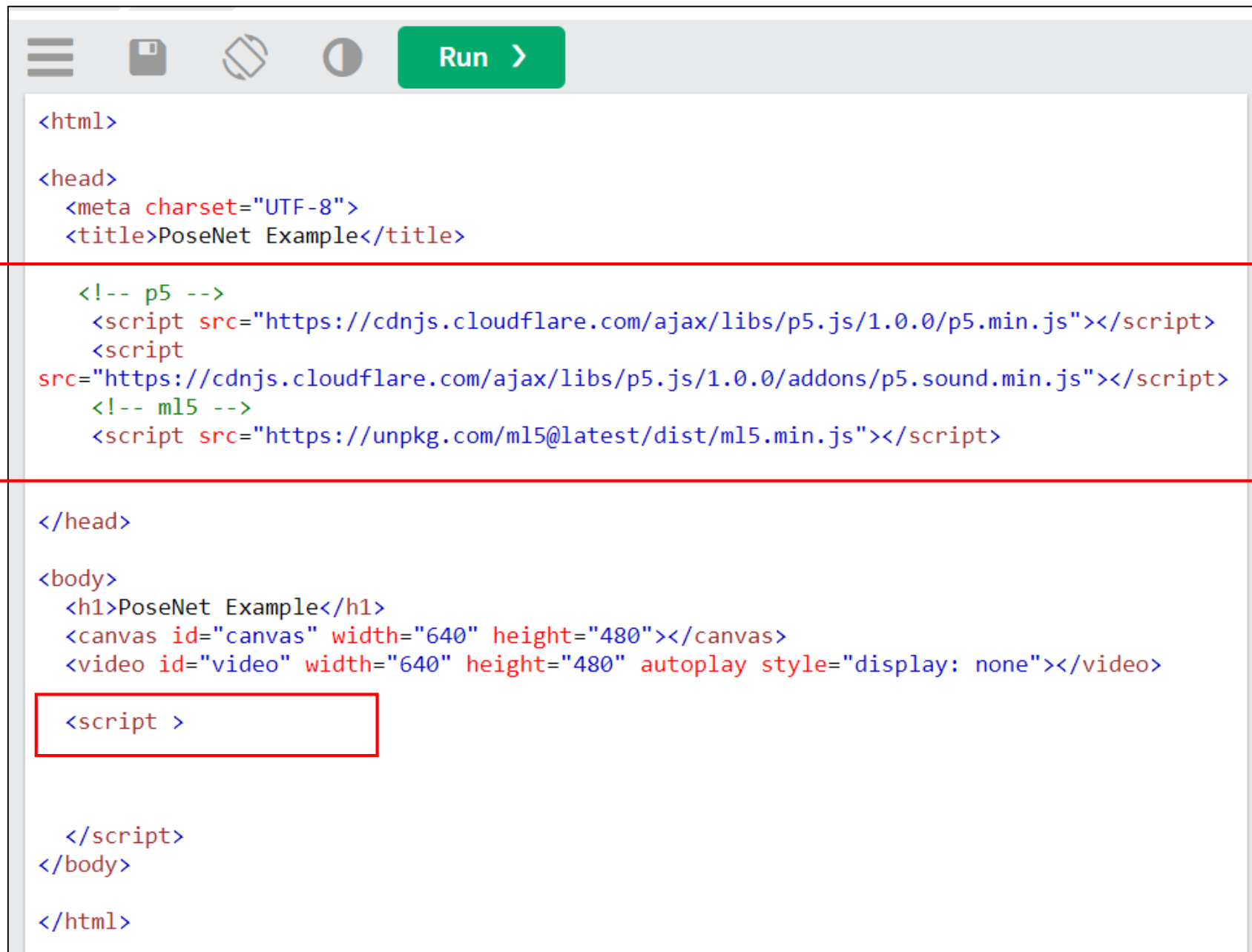
<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

  <script src="sketch.js">
|

  </script>
</body>

</html>
```

delete



The image shows a web editor interface. At the top, there is a toolbar with icons for a menu, saving, undo, and a 'Run' button. Below the toolbar is a code editor containing HTML and JavaScript code. A red rectangle highlights a section of the code in the head of the document, and another red rectangle highlights a script tag in the body.

```
<html>

<head>
  <meta charset="UTF-8">
  <title>PoseNet Example</title>

  <!-- p5 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>

</head>

<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

  <script >

  </script>
</body>

</html>
```

https://github.com/ml5js/ml5-library/tree/main/examples/javascript/PoseNet/PoseNet_webcam

The screenshot shows the GitHub interface for the `ml5js / ml5-library` repository, which is marked as `Public`. The navigation bar includes links for `Code`, `Issues` (221), `Pull requests` (64), `Actions`, `Projects` (11), `Wiki`, `Security`, and `Insights`. The breadcrumb path is `main` / `ml5-library` / `examples` / `javascript` / `PoseNet` / `PoseNet_webcam` /.

A commit by `bomanimc` with the message "Fix j++ unary operators" is displayed. Below the commit, a file list shows `index.html` and `sketch.js`. The `sketch.js` file is highlighted with a red rectangle, and a dark blue callout bubble with the text "Select" points to it. The description for `sketch.js` is "Fix j++ unary operators".

bomanimc Fix j++ unary operators

Latest commit 0dda94b on May 19, 2020 History

1 contributor

88 lines (78 sloc) | 2.62 KB

Raw

Blame



```
1 // Copyright (c) 2019 ml5
2 //
3 // This software is released under the MIT License.
4 // https://opensource.org/licenses/MIT
5
6 /* ===
7 ml5 Example
8 PoseNet using p5.js
9 === */
10 /* eslint-disable */
11
12 // Grab elements, create settings, etc.
13 var video = document.getElementById("video");
14 var canvas = document.getElementById("canvas");
15 var ctx = canvas.getContext("2d");
16
17 // The detected positions will be inside an array
18 let poses = [];
19
20 // Create a webcam capture
21 if (navigator.mediaDevices && navigator.mediaDevices.getUserMedia) {
22   navigator.mediaDevices.getUserMedia({ video: true }).then(function(stream) {
23     video.srcObject = stream;
24     video.play();
25   });
26 }
```

Copy



Run >

```
<html>

<head>
  <meta charset="UTF-8">
  <title>PoseNet Example</title>

  <!-- p5 -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>

</head>

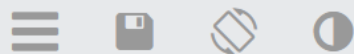
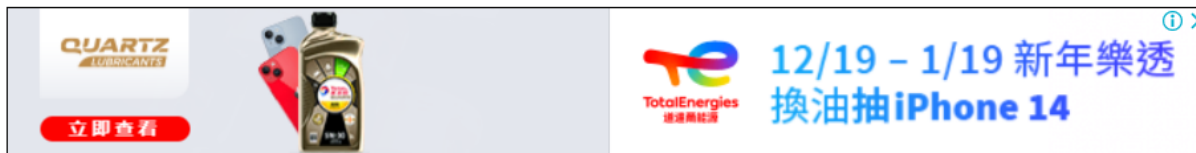
<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

  <script >

  </script>
</body>

</html>
```

Paste



Run >

Result Size: 753 x 568

Get your own website

```
</script>
<!-- m15 -->
<script src="https://unpkg.com/m15@latest/dist/m15.min.js"></script>

</head>

<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

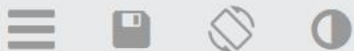
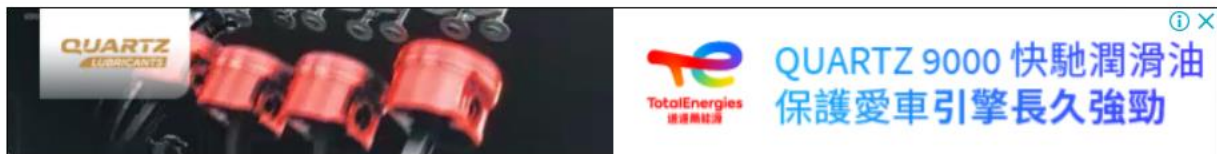
  <script >
    // Copyright (c) 2019 m15
    //
    // This software is released under the MIT License.
    // https://opensource.org/licenses/MIT

    /* ===
    m15 Example
    PoseNet using p5.js
    === */
    /* eslint-disable */

    // Grab elements, create settings, etc.
    var video = document.getElementById("video");
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");

    // The detected positions will be inside an array
    let poses = [];
```

PoseNet Example



Run >

Result Size: 753 x 568

Get your own website

```
</script>
<!-- m15 -->
<script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>

</head>

<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

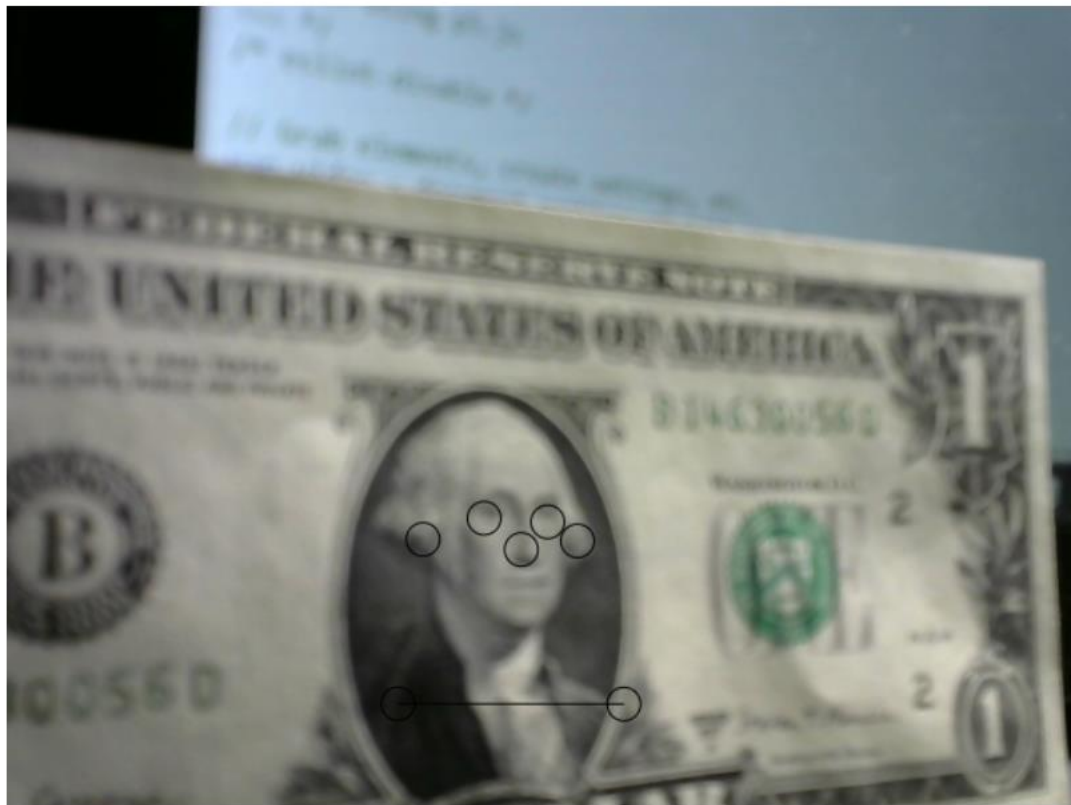
  <script >
    // Copyright (c) 2019 ml5
    //
    // This software is released under the MIT License.
    // https://opensource.org/licenses/MIT

    /* ===
    ml5 Example
    PoseNet using p5.js
    === */
    /* eslint-disable */

    // Grab elements, create settings, etc.
    var video = document.getElementById("video");
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");

    // The detected positions will be inside an array
    let poses = [];
```

PoseNet Example





```
</script>
  <!-- ml5 -->
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>

</head>

<body>
  <h1>PoseNet Example</h1>
  <canvas id="canvas" width="640" height="480"></canvas>
  <video id="video" width="640" height="480" autoplay style="display: none"></video>

  <script >
    // Copyright (c) 2019 ml5
    //
    // This software is released under the MIT License.
    // https://opensource.org/licenses/MIT

    /* ===
    ml5 Example
    PoseNet using p5.js
    === */
    /* eslint-disable */

    // Grab elements, create settings, etc.
    var video = document.getElementById("video");
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");

    // The detected positions will be inside an array
    let poses = [];
```

Copy

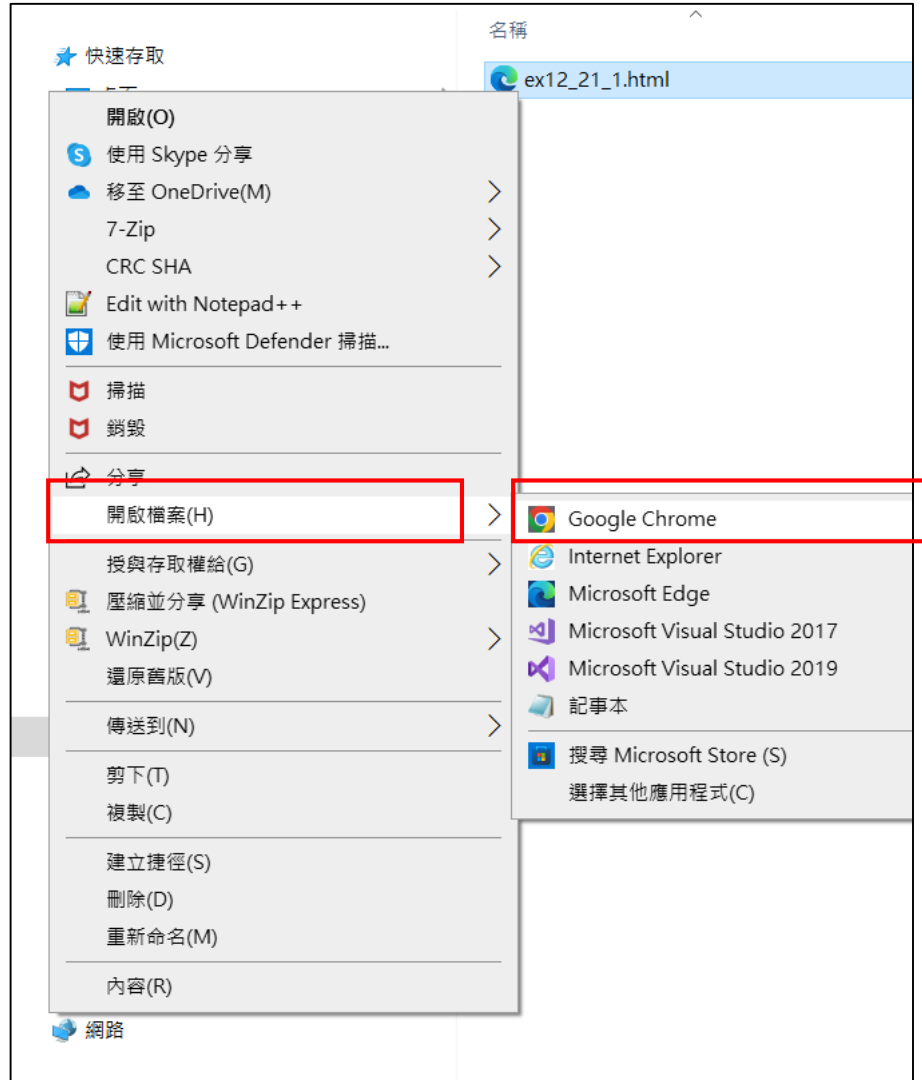
```
new 15 x new 16 x new 17 x index.html x new 18 x ex12_21_1.html x
1 <html>
2
3 <head>
4   <meta charset="UTF-8">
5   <title>PoseNet Example</title>
6
7   <!-- p5 -->
8   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"
9   ></script>
10  <script src="
11  https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"
12  ></script>
13  <!-- ml5 -->
14  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
15
16 </head>
17
18 <body>
19   <h1>PoseNet Example</h1>
20   <canvas id="canvas" width="640" height="480"></canvas>
```

Paste

Save

```
15 x new 16 x new 17 x index.html x new 18 x ex12_21_1.html x
1 <html>
2
3 <head>
4   <meta charset="UTF-8">
5   <title>PoseNet Example</title>
6
7   <!-- p5 -->
8   <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/p5.min.js"
9   ></script>
10  <script src="
11  https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"
12  ></script>
13  <!-- ml5 -->
14  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
15
16 </head>
17
18 <body>
19   <h1>PoseNet Example</h1>
20   <canvas id="canvas" width="640" height="480"></canvas>
21   <video id="video" width="640" height="480" autoplay style="display: none"
```

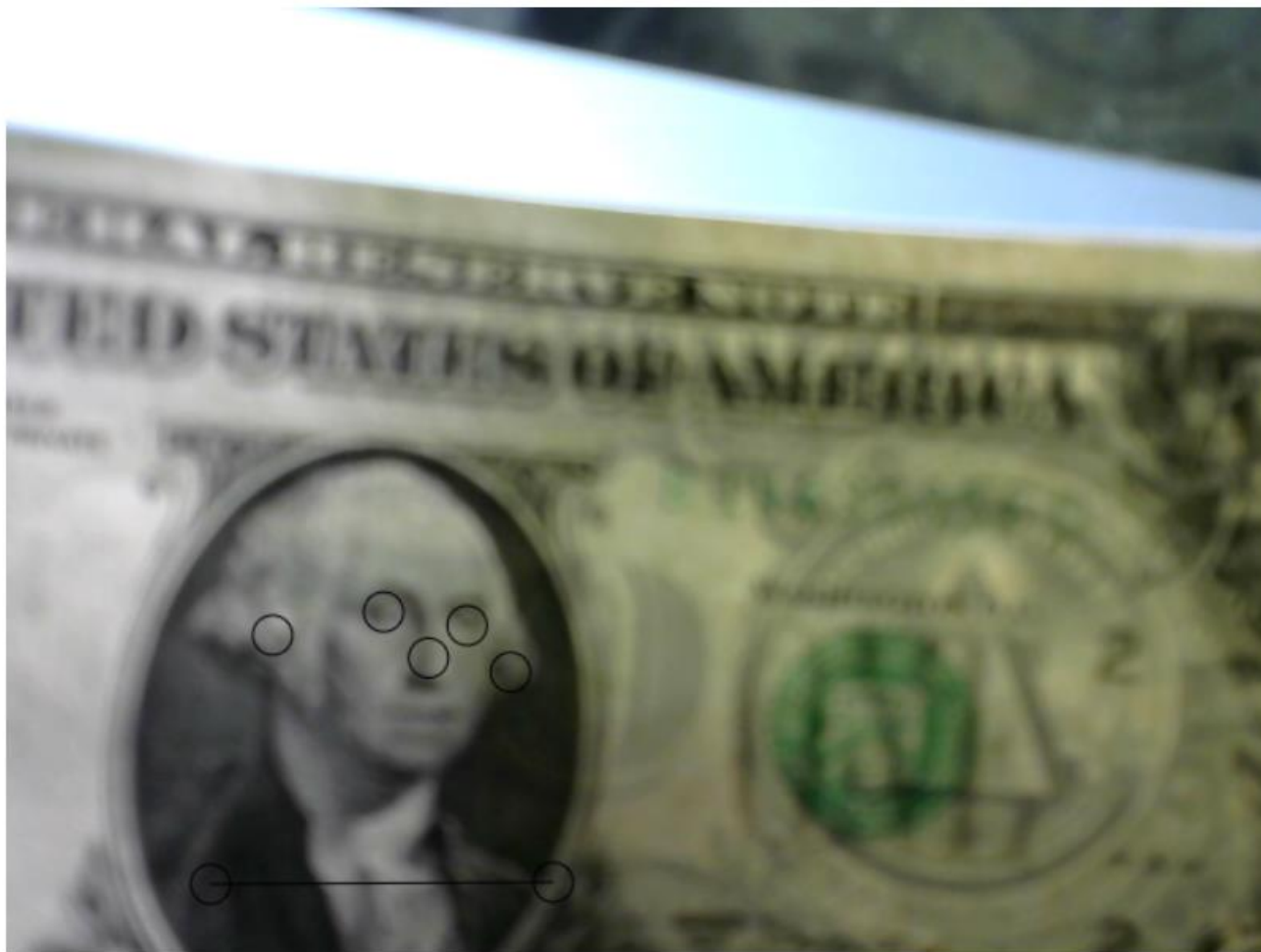
Open ex12_21_1.html with Goggle Chrome



Allow



PoseNet Example

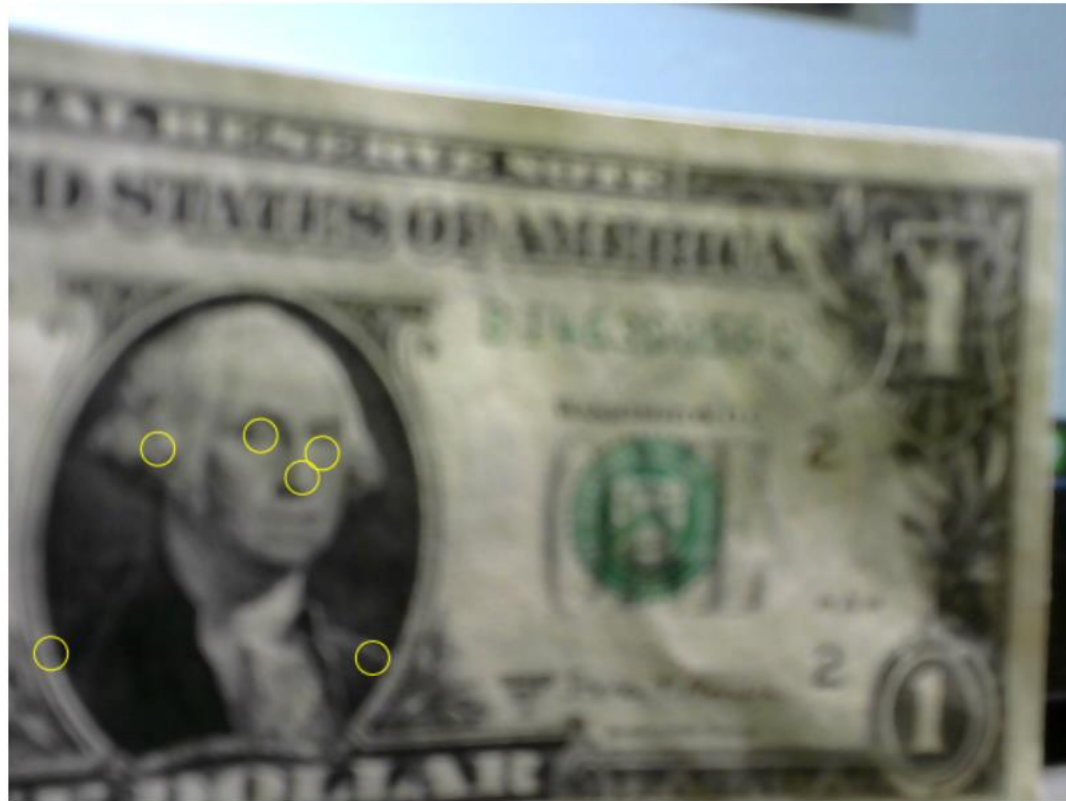


Exercise 14-1

- Change circle color

← → ↺ Ⓜ File | C:/Users/88697/Downloads/2022物聯網實務/12_21/ex12_21_1.html

PoseNet Example



https://www.w3schools.com/jsref/canvas_strokestyle.asp

HTML canvas strokeStyle Property

[← Canvas Object](#)

Example

Draw a rectangle. Use a stroke color of red:



JavaScript:

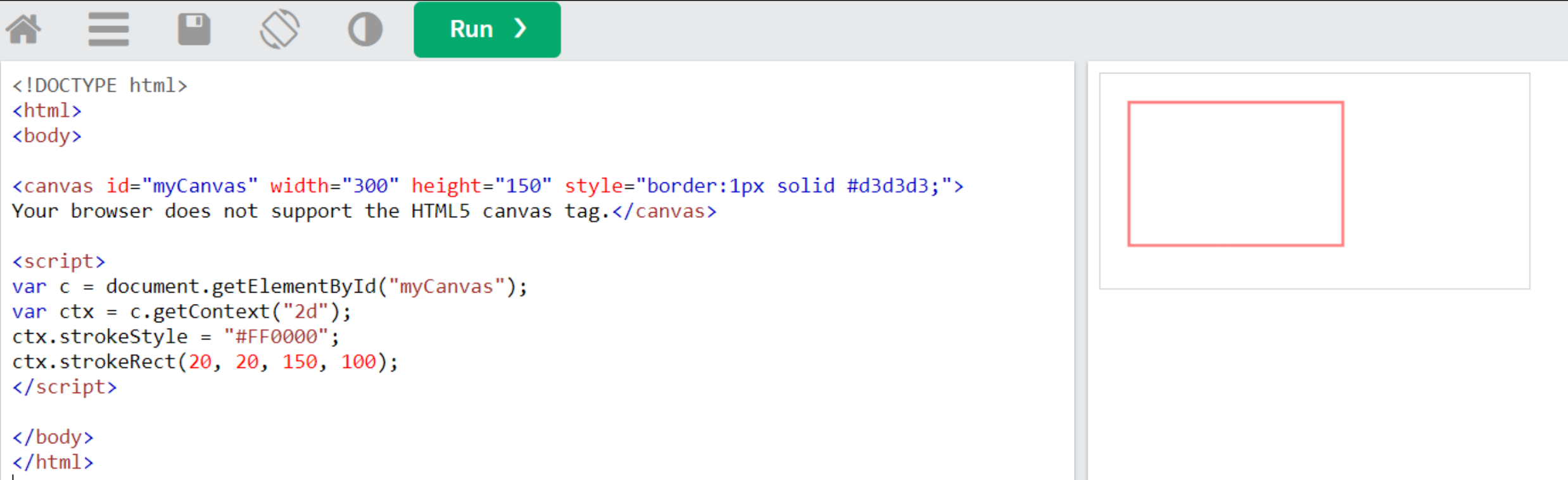
```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.strokeStyle = "#FF0000";  
ctx.strokeRect(20, 20, 150, 100);
```

Try it Yourself

Try it Yourself »

#FF0000

red



#00FF00

green



```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.strokeStyle = "#00FF00";
ctx.strokeRect(20, 20, 150, 100);
</script>

</body>
</html>
```









#00FF00

blue



yellow




```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.strokeStyle = "yellow";
ctx.strokeRect(20, 20, 150, 100);
</script>

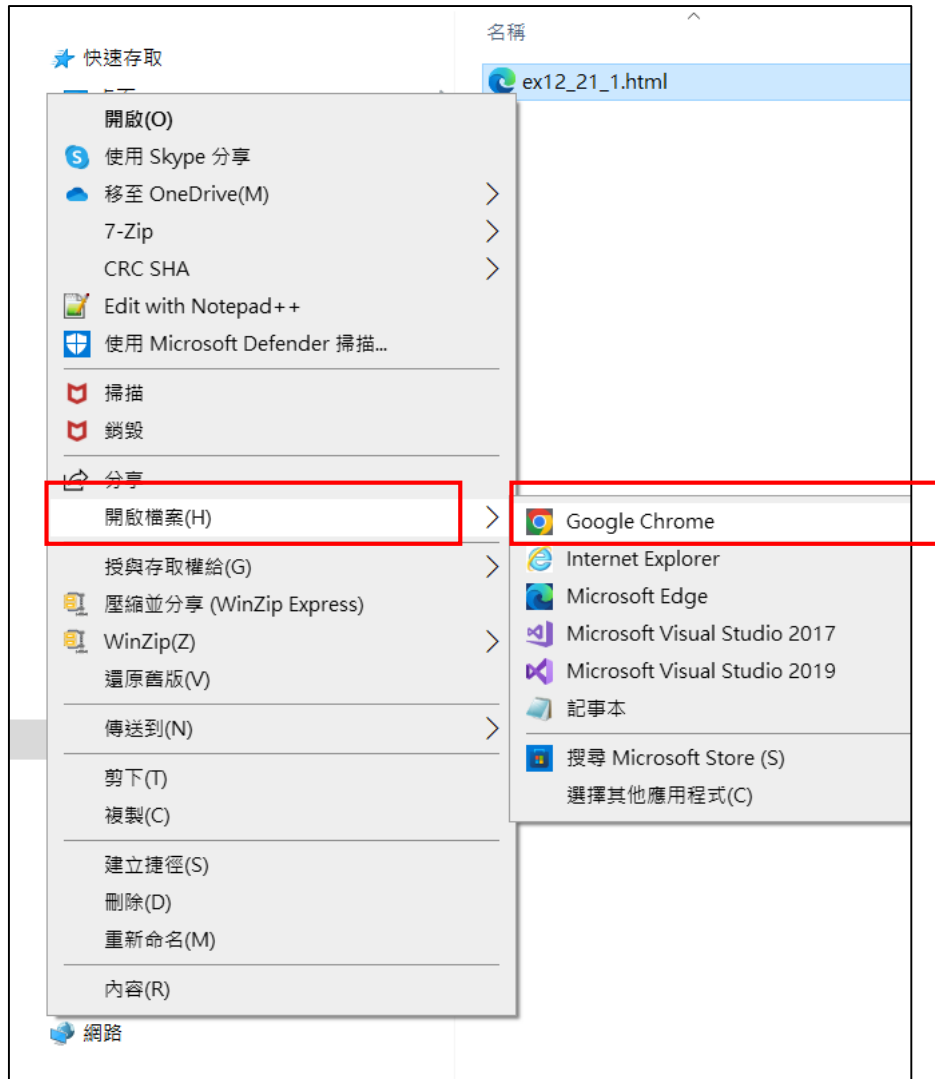
</body>
</html>
```



Edit ex12_21_1.html

```
77
78 // A function to draw ellipses over the detected keypoints
79 function drawKeypoints() {
80     // Loop through all the poses detected
81     for (let i = 0; i < poses.length; i += 1) {
82         // For each pose detected, loop through all the keypoints
83         for (let j = 0; j < poses[i].pose.keypoints.length; j += 1) {
84             let keypoint = poses[i].pose.keypoints[j];
85             // Only draw an ellipse if the pose probability is bigger than 0.2
86             if (keypoint.score > 0.2) {
87                 ctx.beginPath();
88                 ctx.strokeStyle="yellow";
89                 ctx.arc(keypoint.position.x, keypoint.position.y, 10, 0, 2 * Math.PI);
90                 ctx.stroke();
91             }
92         }
93     }
94 }
```

Run ex12_21_1.html



File | C:/Users/88697/Downloads/2022物聯網實務/12_21/ex12_21_1.html

PoseNet Example



Initialize ml5.poseNet

```
// Initialize with video, type and callback
const poseNet = ml5.poseNet(?video, ?type, ?callback);
// OR Initialize with video, options and callback
const poseNet = ml5.poseNet(?video, ?options, ?callback);
// OR Initialize WITHOUT video. Just options and callback here
const poseNet = ml5.poseNet(?callback, ?options);
```

Parameters

- **video**: OPTIONAL. Optional HTMLVideoElement input to run poses on.
- **type**: OPTIONAL. A String value to run `single` or `multiple` estimation. Changes the `detectionType` property of the options. Default is `multiple`.
- **callback**: OPTIONAL. A function that is called when the model is loaded.
- **options**: OPTIONAL. A object that contains properties that effect the posenet model accuracy, results, etc.

Methods

`.on('pose', ...)`

An event listener that returns the results when a pose is detected. You can use this with `.singlePose()` or `.multiPose()` or just listen for poses if you pass in a `video` into the constructor.

```
poseNet.on('pose', callback);
```

js

.on

Inputs

- **callback:** REQUIRED. A callback function to handle the results when a pose is detected. For example.

```
poseNet.on('pose', (results) => {  
  // do something with the results  
  console.log(results);  
});
```

Outputs

- **Array:** Returns an array of objects. See documentation for `.singlePose()` and `.multiPose()`
-

.singlePose()

```
poseNet.singlePose(?input);
```

Inputs

- **input:** Optional. A HTML video or image element or a p5 image or video element. If no input is provided, the default is to use the video given in the constructor.

Outputs

- **Array:** Returns an array of objects. A sample is included below.

```
[
  {
    pose: {
      keypoints: [{ position: { x, y }, score, part }, ...],
      leftAngle: { x, y, confidence },
      leftEar: { x, y, confidence },
      leftElbow: { x, y, confidence },
      ...
    },
  },
];
```

js

Exercise 14-2



PoseNet Moel

PoseNet example using p5.js

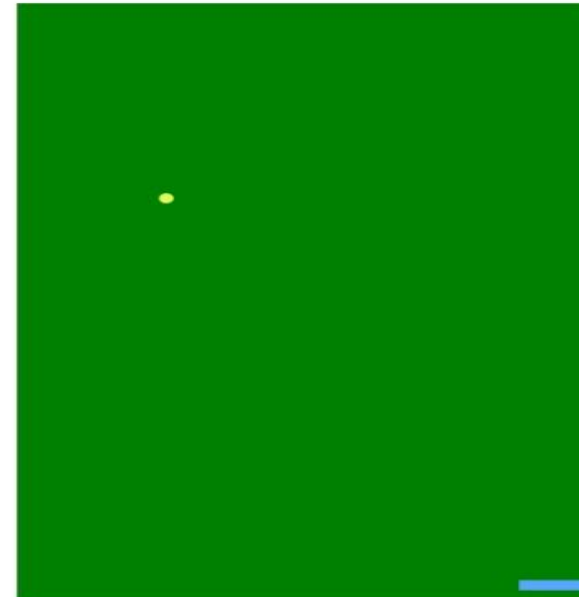
Static

Score:17

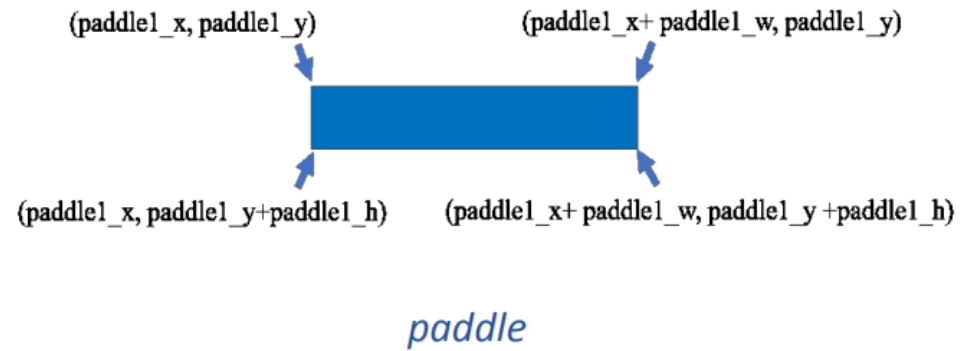
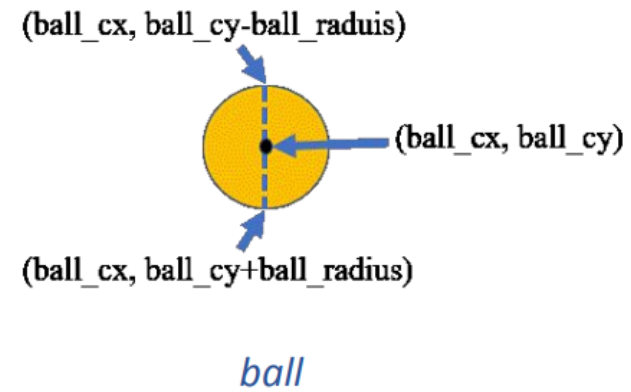
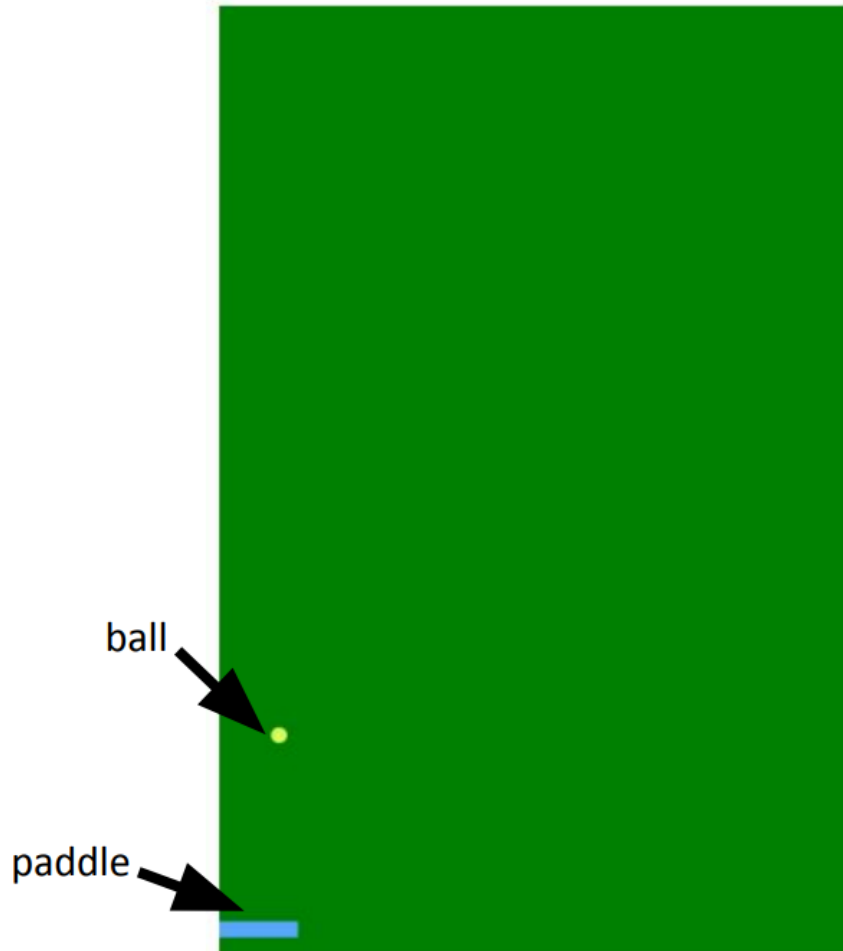
countdown: 5

Game is running

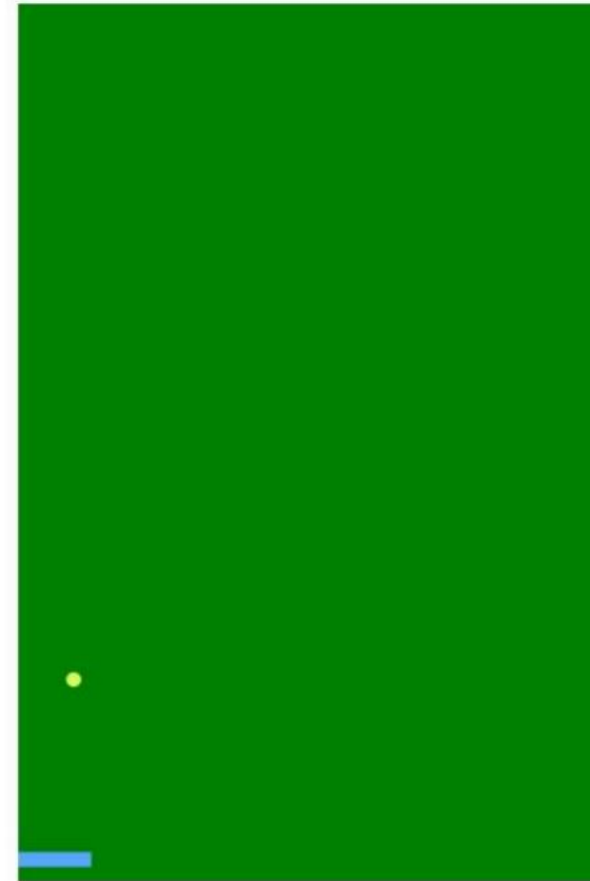
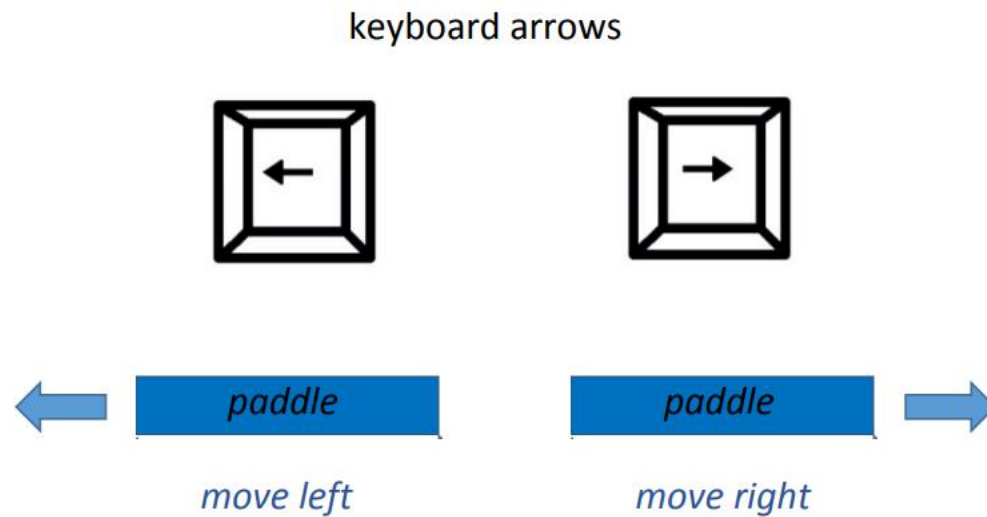
Model Loaded



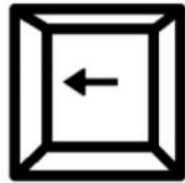
Ping-Pong Game



Ping-Pong Game-Paddle Control



Ping-Pong Game-Paddle Control



event

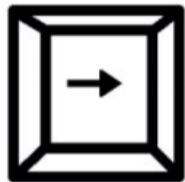
action

keydown

keysDown={"37":true}

keyup

keysDown={ }



event

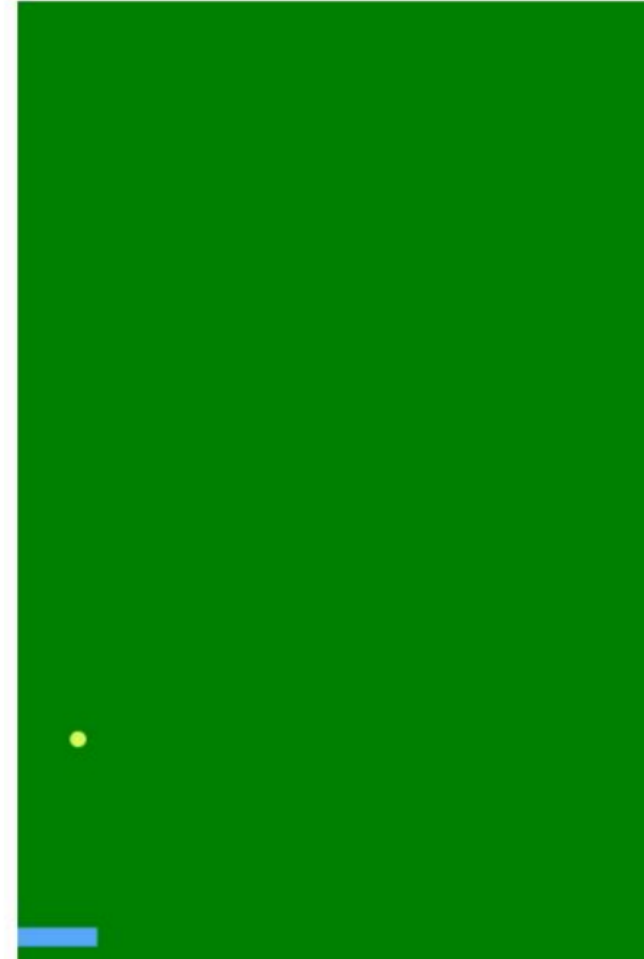
action

keydown

keysDown={"39":true}

keyup

keysDown={ }



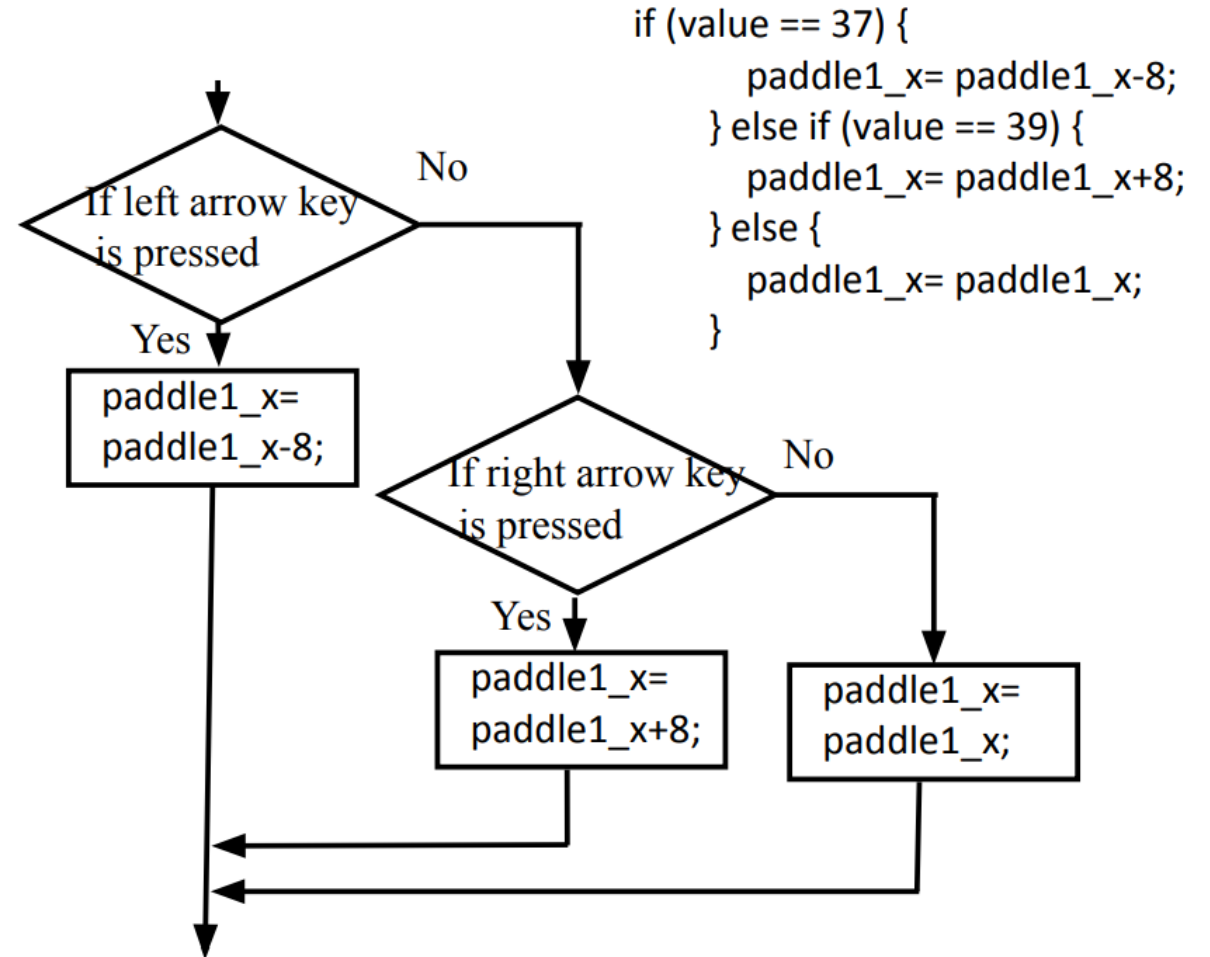
Ping-Pong Game-Paddle Control



event	action
keydown	keysDown={"37":true}
keyup	keysDown={ }

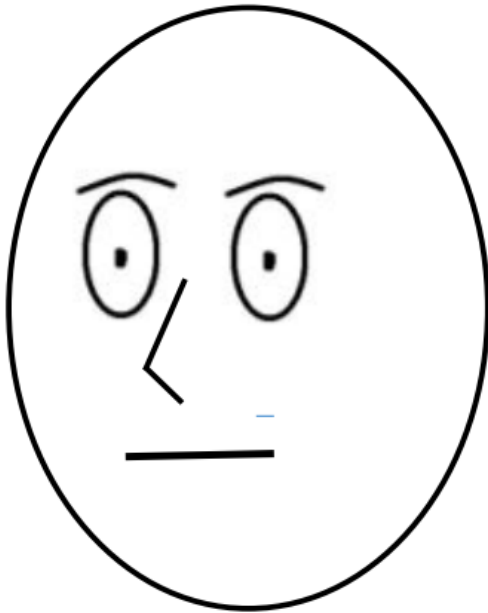


event	action
keydown	keysDown={"39":true}
keyup	keysDown={ }

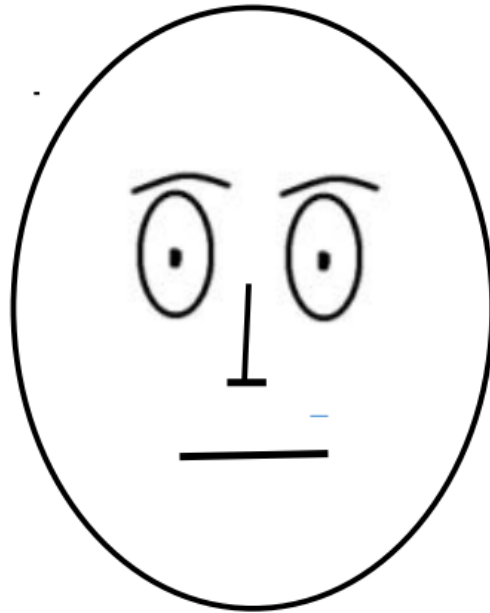


Ping-Pong Game-Paddle Control by Head Pose

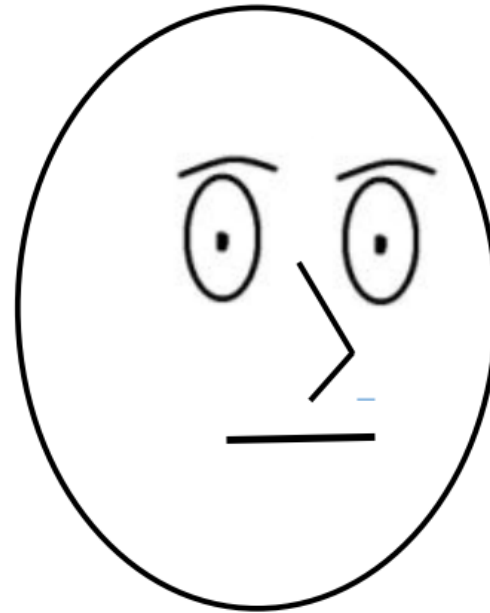
Left



Static



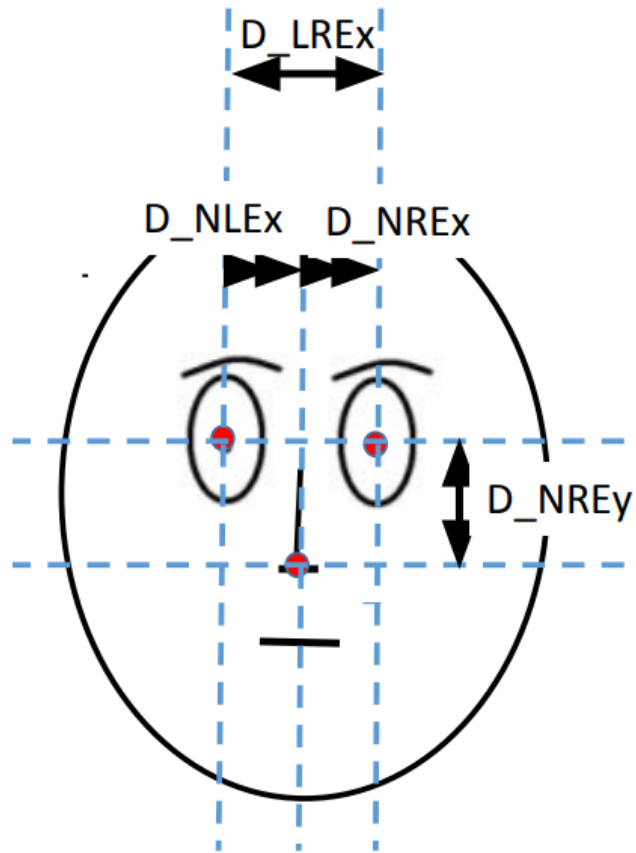
Right



Up

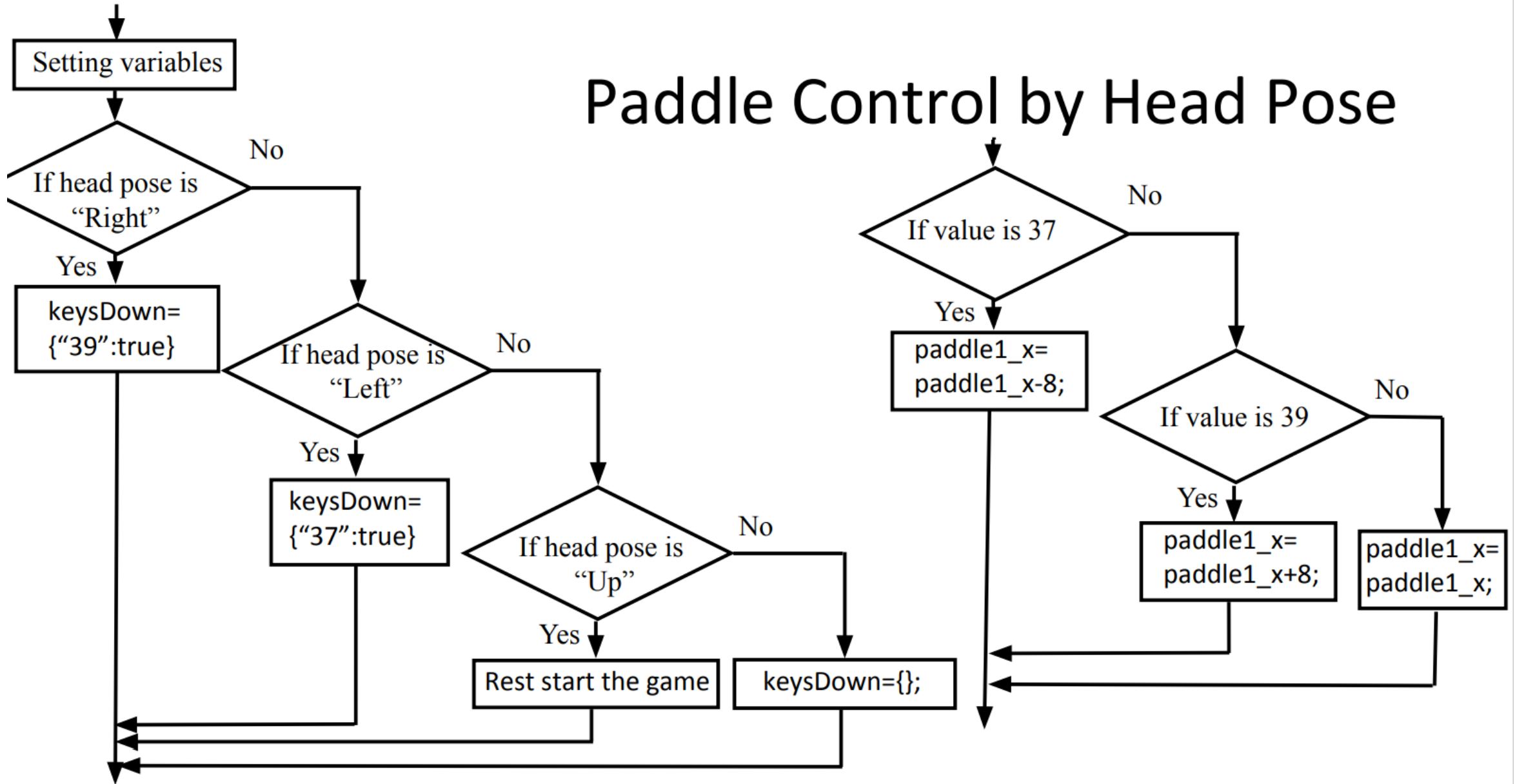


Head Pose estimation



Conditions	Classification
$D_{NLEx} < D_{LREx}/4$	Left
$D_{NREx} < D_{LREx}/4$	Right
$D_{NREy} < D_{LREx}/3$	Up
Noe of the above	Static

Paddle Control by Head Pose



ex12_21_2.html (copy ex12_21_2.txt)

