

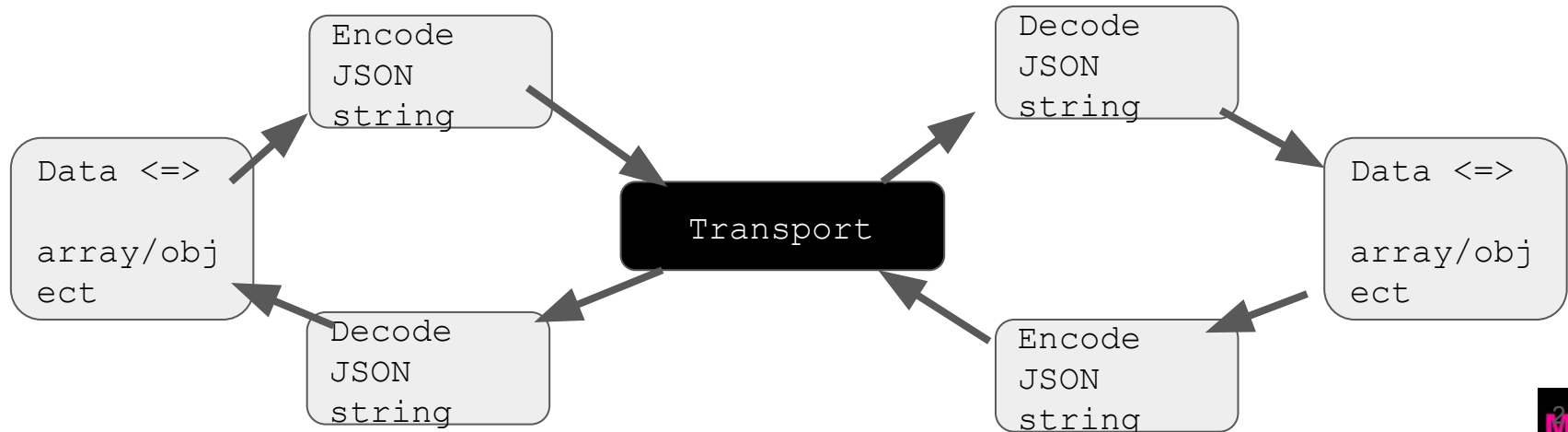
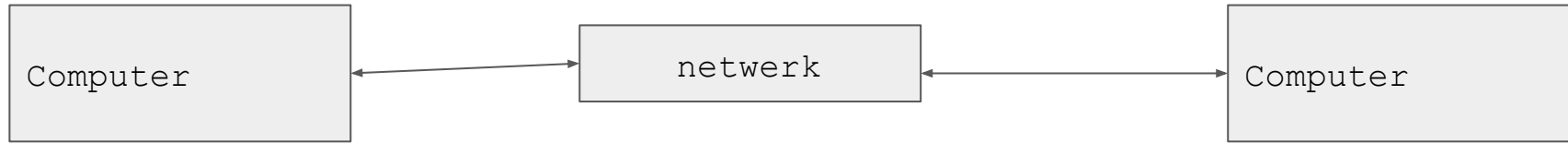
# Paint it Black

JS AJAX JSON PHP

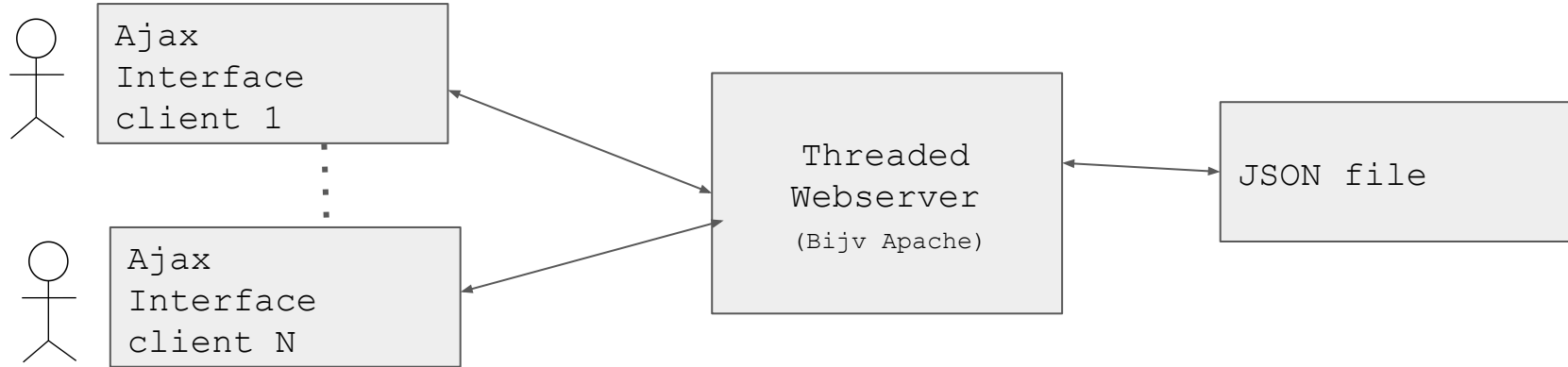
PAINT IT  
BLACK

nov 2022

# Principe data overdracht JSON



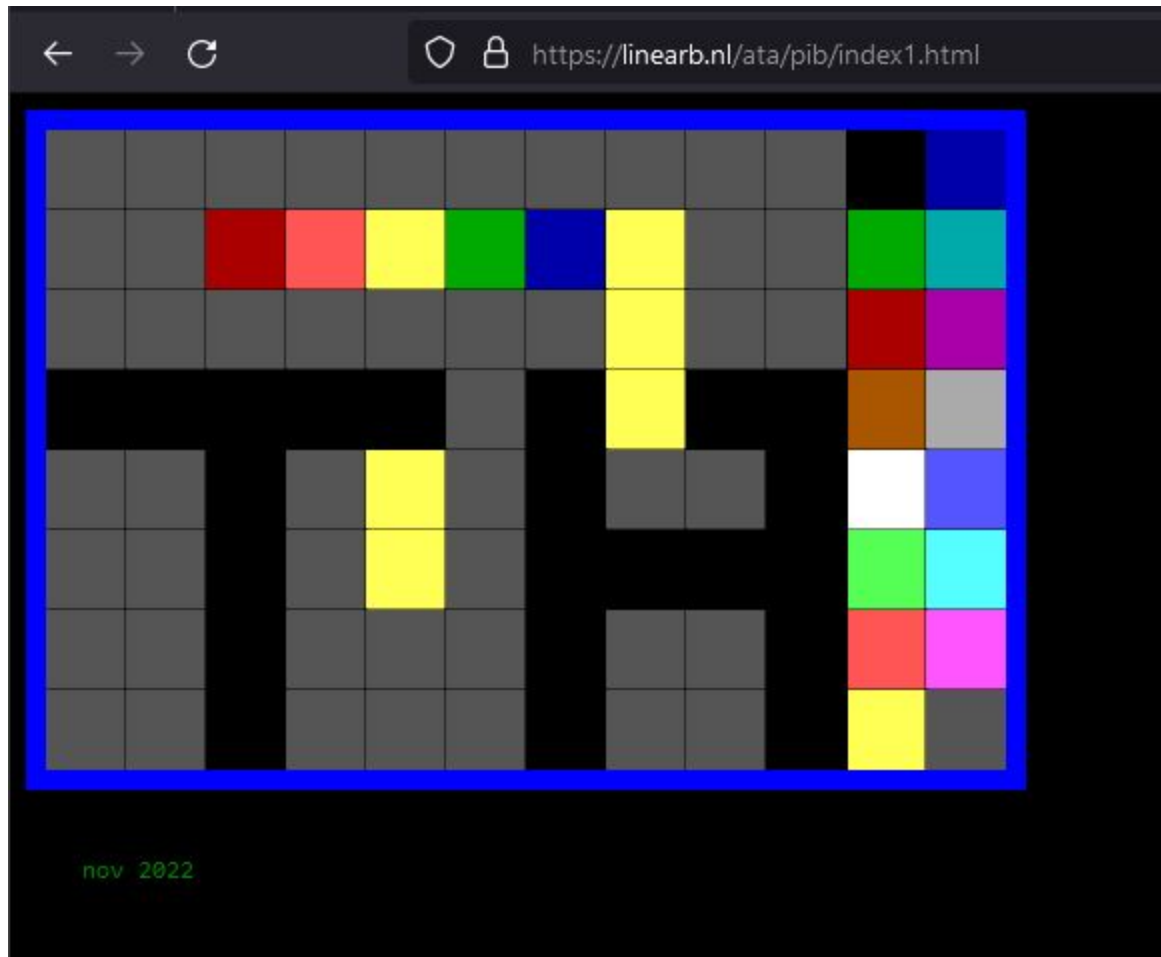
# Paint it Black 1 multiplayer



# Paint it black Polling

*Polling* is the process where the computer or controlling device waits for an external device to check for its readiness or state, often with low-level hardware. For example, when a printer is connected via a parallel port, the computer waits until the printer has received the next character.

[https://en.wikipedia.org/wiki/Polling\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Polling_(computer_science))



Demo :

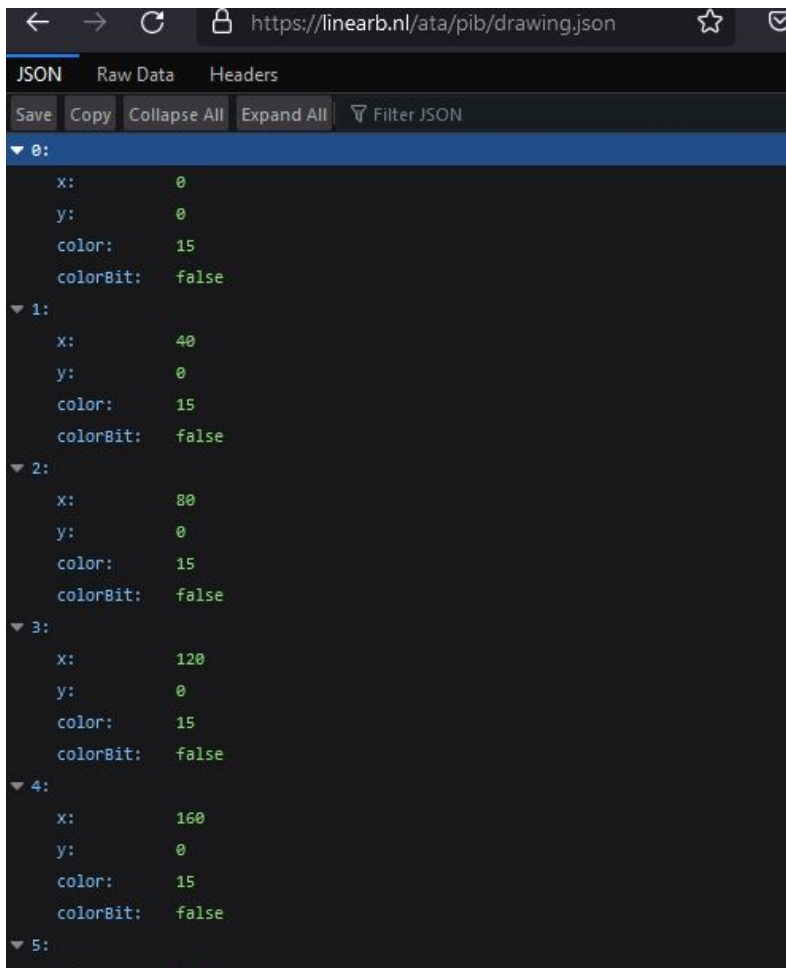
<https://linearb.nl/ata/pib/index1.html>

```
← → ↺ 🔒 https://linearb.nl/ata/pib/drawing.json ☆ 📧 ⚙️ 👤 🔄
JSON Raw Data Headers
Save Copy Pretty Print

[{"x":0,"y":0,"color":15,"colorBit":false}, {"x":40,"y":0,"color":15,"colorBit":false},
{"x":80,"y":0,"color":15,"colorBit":false}, {"x":120,"y":0,"color":15,"colorBit":false},
{"x":160,"y":0,"color":15,"colorBit":false}, {"x":200,"y":0,"color":15,"colorBit":false},
{"x":240,"y":0,"color":15,"colorBit":false}, {"x":280,"y":0,"color":15,"colorBit":false},
{"x":320,"y":0,"color":15,"colorBit":false}, {"x":360,"y":0,"color":15,"colorBit":false},
{"x":0,"y":40,"color":15,"colorBit":false}, {"x":40,"y":40,"color":15,"colorBit":false},
{"x":80,"y":40,"color":4,"colorBit":false}, {"x":120,"y":40,"color":12,"colorBit":false},
{"x":160,"y":40,"color":14,"colorBit":false}, {"x":200,"y":40,"color":2,"colorBit":false},
{"x":240,"y":40,"color":1,"colorBit":false}, {"x":280,"y":40,"color":14,"colorBit":false},
{"x":320,"y":40,"color":15,"colorBit":false}, {"x":360,"y":40,"color":15,"colorBit":false},
{"x":0,"y":80,"color":15,"colorBit":false}, {"x":40,"y":80,"color":15,"colorBit":false},
{"x":80,"y":80,"color":15,"colorBit":false}, {"x":120,"y":80,"color":15,"colorBit":false},
{"x":160,"y":80,"color":15,"colorBit":false}, {"x":200,"y":80,"color":15,"colorBit":false},
{"x":240,"y":80,"color":15,"colorBit":false}, {"x":280,"y":80,"color":14,"colorBit":false},
{"x":320,"y":80,"color":15,"colorBit":false}, {"x":360,"y":80,"color":15,"colorBit":false},
{"x":0,"y":120,"color":0,"colorBit":false}, {"x":40,"y":120,"color":0,"colorBit":false},
{"x":80,"y":120,"color":0,"colorBit":false}, {"x":120,"y":120,"color":0,"colorBit":false},
{"x":160,"y":120,"color":0,"colorBit":false}, {"x":200,"y":120,"color":15,"colorBit":false},
{"x":240,"y":120,"color":0,"colorBit":false}, {"x":280,"y":120,"color":14,"colorBit":false},
{"x":320,"y":120,"color":0,"colorBit":false}, {"x":360,"y":120,"color":0,"colorBit":false},
{"x":0,"y":160,"color":15,"colorBit":false}, {"x":40,"y":160,"color":15,"colorBit":false},
{"x":80,"y":160,"color":0,"colorBit":false}, {"x":120,"y":160,"color":15,"colorBit":false},
{"x":160,"y":160,"color":14,"colorBit":false}, {"x":200,"y":160,"color":15,"colorBit":false},
{"x":240,"y":160,"color":0,"colorBit":false}, {"x":280,"y":160,"color":15,"colorBit":false},
{"x":320,"y":160,"color":15,"colorBit":false}, {"x":360,"y":160,"color":0,"colorBit":false},
{"x":0,"y":200,"color":15,"colorBit":false}, {"x":40,"y":200,"color":15,"colorBit":false},
{"x":80,"y":200,"color":0,"colorBit":false}, {"x":120,"y":200,"color":15,"colorBit":false},
{"x":160,"y":200,"color":14,"colorBit":false}, {"x":200,"y":200,"color":15,"colorBit":false},
{"x":240,"y":200,"color":0,"colorBit":false}, {"x":280,"y":200,"color":15,"colorBit":false},
{"x":320,"y":200,"color":0,"colorBit":false}, {"x":360,"y":200,"color":0,"colorBit":false},
{"x":0,"y":240,"color":15,"colorBit":false}, {"x":40,"y":240,"color":15,"colorBit":false},
{"x":80,"y":240,"color":0,"colorBit":false}, {"x":120,"y":240,"color":15,"colorBit":false},
{"x":160,"y":240,"color":15,"colorBit":false}, {"x":200,"y":240,"color":15,"colorBit":false},
{"x":240,"y":240,"color":0,"colorBit":false}, {"x":280,"y":240,"color":15,"colorBit":false},
{"x":320,"y":240,"color":15,"colorBit":false}, {"x":360,"y":240,"color":0,"colorBit":false},
{"x":0,"y":280,"color":15,"colorBit":false}, {"x":40,"y":280,"color":15,"colorBit":false},
{"x":80,"y":280,"color":0,"colorBit":false}, {"x":120,"y":280,"color":15,"colorBit":false},
{"x":160,"y":280,"color":15,"colorBit":false}, {"x":200,"y":280,"color":15,"colorBit":false},
{"x":240,"y":280,"color":0,"colorBit":false}, {"x":280,"y":280,"color":15,"colorBit":false},
{"x":320,"y":280,"color":15,"colorBit":false}, {"x":360,"y":280,"color":0,"colorBit":false}]
```

<https://linearb.nl/ata/pib/drawing.json>

# JSON Formatted Data



`https://linearb.nl/  
ata/pib/drawing.jso  
n`

# PHP

```
jsonWrite.php x
1  <?php
2      $control = ""; // mandatory for PHP 5.4
3      $filePath = "drawing.json"; // path filename json control file
4      if (!empty($_GET['put'])) {
5          $control = $_GET['put'];
6      }
7      $file = fopen($filePath, "w") or die("can't open file");
8      fwrite($file, $control); // overwrite existing file
9      fclose($file);
10 ?>
```



ajax\_callback.js

X

```
1 let controlScript = "jsonWrite.php";
2 let jsonFile = "drawing.json";
3 let preloadJsonFile = "preload.json";
4
5 > function makeAjaxCall(url, methodType){ ... }
30
31 > function errorHandler(statusCode){ ... }
34
35 > function serverWriteJson(data){ ... }
41
42 > function serverGetJson(){ ... }
45
46 > function preload() { ... }
56
```

Structuur  
ajax\_callback.js

startscript

deze  
code  
zelf  
maken

Structuur  
pib.js

startscript

```
script.js
1 let canvas = document.getElementById('canvas');
2 let context = canvas.getContext('2d');
3 let colorSelection = 0xe; // startvalue color selection
4 let jsonObj = "";
5 let ega = ["#000000", "#0000aa", "#00aa00", "#00aaaa", "#aa0000",
• "#aa00aa", "#aa5500", "#aaaaaa", "#ffffff", "#5555ff", "#55ff55",
• "#55ffff", "#ff5555", "#ff55ff", "#ffff55", "#555555"]; // colors
6 let pallet = [];
7 let drawing = [];
8 let boxWidth = boxHeight = 80;
9 let refreshTimer = window.setInterval(serverGetJson, 2000); //
• timer get data from server
10
11 > class Bit {}
41
42 > function init() {}
66
67 > function readJson(jsonString) {}
75
76 init();
```

deze  
code  
zelf  
maken

startscript

```
11 class Bit {
12     constructor(x, y, color) {
13         this.x = x;
14         this.y = y;
15         this.color = color;
16         this.colorBit = false;
17         addEventListener('mousedown', (e) => {
18             let box = canvas.getBoundingClientRect();
19             let mouseX = e.clientX - box.left; // is mouse in the box?
20             let mouseY = e.clientY - box.top; // is mouse in the box?
21             if (mouseX > this.x && mouseX < this.x + boxWidth && mouseY >
22                 * this.y && mouseY < this.y + boxHeight) {
23                 if (this.colorBit) {
24                     colorSelection = this.color;
25                 } else {
26                     this.color = colorSelection;
27                     this.draw(context);
28                 }
29             }
30         })
31     }
```

```
32     draw(context) {
33         context.beginPath();
34         context.fillStyle = ega[this.color];
35         context.rect(this.x, this.y, boxWidth, boxHeight);
36         context.stroke();
37         context.fill();
38         context.closePath();
39     }
40 }
```

```

42 function init() {
43     context.canvas.width = 12 * boxWidth; // set canvas width
44     context.canvas.height = 8 * boxHeight; // set canvas height
45     for (let i = 0; i < 0x10; i++) {
46         let numOnRow = 2;
47         let bitWidth = boxWidth;
48         let x = 800 + (i % numOnRow) * bitWidth;
49         let y = Math.floor(i / numOnRow) * bitWidth;
50         let bit = new Bit(x, y, i);
51         bit.colorBit = true;
52         bit.draw(context);
53         pallet.push(bit);
54     }
55     for (i = 0; i < 80; i++) {
56         //grid 80 col x 10 row
57         let numOnRow = 10;
58         let bitWidth = boxWidth;
59         let x = (i % numOnRow) * bitWidth;
60         let y = Math.floor(i / numOnRow) * bitWidth;
61         let bit = new Bit(x, y, 0xf); // 0xf background color from ega
        * array
62         bit.draw(context);
63         drawing[i] = bit; //opslag van de tekening tbv export naar json
64     }
65 }
66

```

startscript



# Paint it Black Start versie

Start code:

<https://github.com/amsterdam-tech-academy/Paint-it-black>

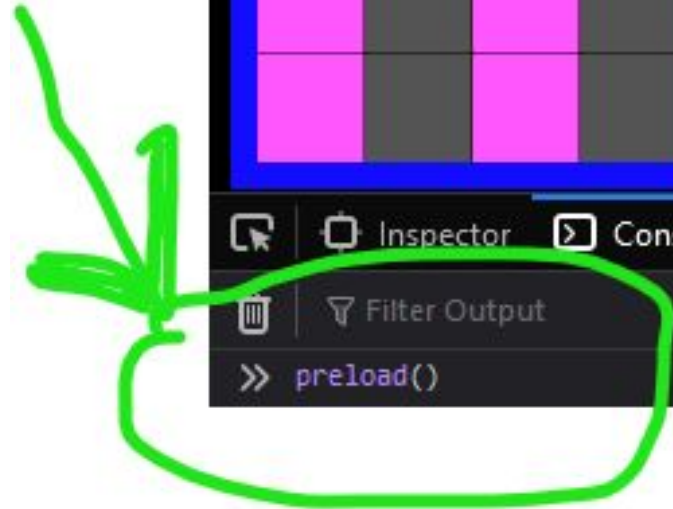
## Opdracht

Startopdracht: maak de single player versie werkend. Probeer te bedenken wat nodig is voor een multiplayer versie.



# PiB challenge 1

Maak een  
function  
`preload()`  
waarmee een  
vooraf  
geprogrammeerd  
patroon geladen  
wordt



## PiB challenge 2

In de demo wordt  
de aangeklikte  
rechthoek  
groter. Bedenk  
een manier omdat  
te doen

## PiB challenge 3

Deze multiplayer versie van Paint it Black werkt met een PHP server. Hoe zou jij dit oplossen met een NodeJS server?

