

diploma project

a collection of code
written and operates
the different aspects
of the system.

Simaresi
Paraskevi
Dimitra

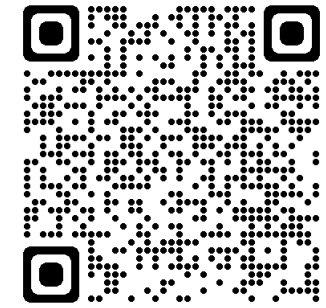
promoter
Oun-
grinis Kon-
stantinos
- Alketas


2023



Arduino

Language
C++ 100.00%



Ο κώδικας είναι διαθέσιμος στο Github 



servo-light-sens [Arduino 1.8.5]

File Edit Sketch Tools Help

servo-light-sens servo_0 servo_1 servo_2 servo_3

```
1 // C++ code
2 //https://www.arduino.cc/en/Tutorial/Knob - servo instructions
3 #include <Servo.h>
4
5 //1st Servo
6 int sensorvalue = 0;
7 int sensorvalue_new = 0;
8 int rotation = 0;
9 Servo myservo;
10
11 //2nd Servo
12 int sensorvalue_2 = 0;
13 int sensorvalue_new_2 = 0;
14 int rotation_2 = 0;
15 Servo myservo2;
16
17 //3rd Servo
18 int sensorvalue_3 = 0;
19 int sensorvalue_new_3 = 0;
20 int rotation_3 = 0;
21 Servo myservo3;
22
23 //4th Servo
24 int sensorvalue_4 = 0;
25 int sensorvalue_new_4 = 0;
26 int rotation_4 = 0;
27 Servo myservo4;
28
29
30
31 int maxL = 500; //max light
32 int minL = 5; //min light
33 int maxR = 180; //max rotation
34 int time = 15; //delay time
35 int lim = 10; // + - limits for if
36
37 void setup()
38 {
39   Serial.begin(9600);
40 }
```

```
41 //1st Servo-////////////////////////////////////
42 myservo.attach(11);
43 pinMode(A0, INPUT);
44
45 //2nd Servo-////////////////////////////////////
46 myservo2.attach(10);
47 pinMode(A1, INPUT);
48
49 //3rd Servo-////////////////////////////////////
50 myservo3.attach(9);
51 pinMode(A2, INPUT);
52
53 //3th Servo-////////////////////////////////////
54 myservo4.attach(6);
55 pinMode(A3, INPUT);
56 }
57
58 void loop()
59 {
60   sensorvalue_new = analogRead(A0); //1st Servo
61   sensorvalue_new_2 = analogRead(A1); //2nd Servo
62   sensorvalue_new_3 = analogRead(A2); //3rd Servo
63   sensorvalue_new_4 = analogRead(A3); //4th Servo
64
65   //Servos ifs-----
66   firstServo();
67   secondServo();
68   thirdServo();
69   fourthServo();
70   //end of servos -----
71
72   myservo.write (rotation); //send the angle of rotatin to 1st servo
73   myservo2.write (rotation_2); //send the angle of rotatin to 2nd servo
74   myservo3.write (rotation_3); //send the angle of rotatin to 3rd servo
75   myservo4.write (rotation_4); //send the angle of rotatin to 4th servo
76 }
```

Ο κώδικας δέχεται ως εισροή τα δεδομένα από τις φωτοαντιστάσεις και τα ανατροφοδοτεί ως κίνηση στα Servo.

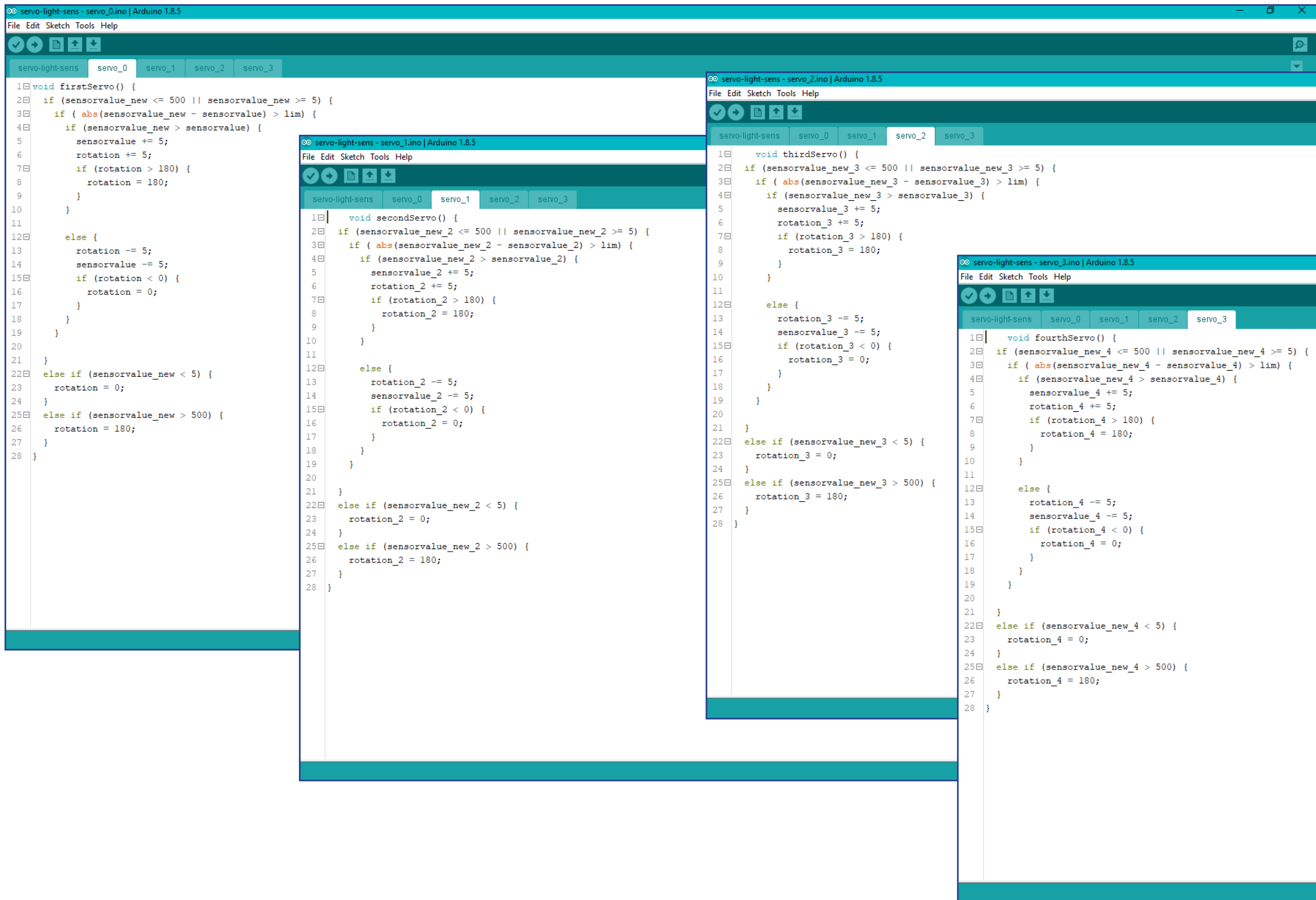
```
77 //1st Servo-----
78 Serial.print ("1st:");
79 Serial.print (sensorvalue_new);
80 Serial.print (":");
81 Serial.println(rotation);
82
83 //2nd Servo-----
84 Serial.print ("2nd:");
85 Serial.print (sensorvalue_new_2);
86 Serial.print (":");
87 Serial.println(rotation_2);
88
89 //3rd Servo-----
90 Serial.print ("3rd:");
91 Serial.print (sensorvalue_new_3);
92 Serial.print (":");
93 Serial.println(rotation_3);
94
95 //4th Servo-----
96 Serial.print ("4th:");
97 Serial.print (sensorvalue_new_4);
98 Serial.print (":");
99 
```

Ο κώδικας δέχεται ως εισροή τα δεδομένα από τις φωτοαντιστάσεις και τα ανατροφοδοτεί ως κίνηση στα Servo.

```

77
78 //1st Servo-----
79 Serial.print ("1st:");
80 Serial.print(sensorvalue_new);
81 Serial.print (":");
82 Serial.println(rotation);
83
84 //2nd Servo-----
85 Serial.print ("2nd:");
86 Serial.print(sensorvalue_new_2);
87 Serial.print (":");
88 Serial.println(rotation_2);
89
90 //3rd Servo-----
91 Serial.print ("3rd:");
92 Serial.print(sensorvalue_new_3);
93 Serial.print (":");
94 Serial.println(rotation_3);
95
96 //4th Servo-----
97 Serial.print ("4th:");
98 Serial.print(sensorvalue_new_4);
99 Serial.print (":");
100 Serial.println(rotation_4);
101
102 //time = rotation / 5000 ; //delay time
103 }
104
105
106 □ /*void monitor_print_info () {
118
119

```



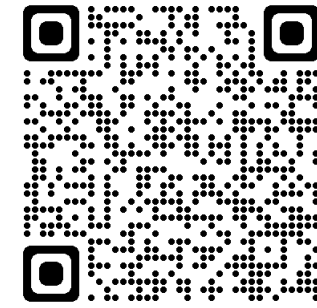
percentage-calculator

Language

JavaScript 77.1%

HTML 19.5%

CSS 3.4%



Ο κώδικας είναι
διαθέσιμος στο
Github



```
File Edit Selection View Go Run Terminal Help
sketchjs - percentage-calculator_coding - Visual Studio Code

EXPLORER
OPEN EDITORS
GROUP 1
  JS sketchjs M
  Settings
GROUP 2
  JS sketchjs M
PRECENTAGE-CALCULATOR_CODING
  images
    im1.png U
    im1G.png U
    im1Va.png U
    im2.png U
    im2Va.png U
    im3.png U
    im4.png U
    im4G.png U
    im9G.png U
    im13G.png U
    im14G.png U
    imV.png U
  libraries
    .gitattributes
    index.html
    jsconfig.json
    README.md
  JS sketchjs M
  # style.css

  > OUTLINE
  > TIMELINE

JS sketchjs > preload
1 let img;
2 let patternCounts = {
3   red: 0,
4   green: 0,
5   blue: 0;
6 };
7
8 function preload() {
9   // Load the image
10  img = loadImage('images/imV.png');
11  console.log("image loaded");
12 }
13
14 function setup() {
15   // Create a canvas that matches the image dimensions
16   createCanvas(img.width, img.height);
17
18   // Display the image on the canvas
19   image(img, 0, 0);
20
21   // Loop through every pixel in the image
22   for (let x = 0; x < img.width; x++) {
23     for (let y = 0; y < img.height; y++) {
24       // Get the color of the current pixel
25       let pixelColor = get(x, y);
26
27       // Check if the color matches any of the patterns
28       if (colorMatchesPattern(pixelColor, color(255, 0, 0))) {
29         patternCounts.red++;
30       } else if (colorMatchesPattern(pixelColor, color(0, 255, 0))) {
31         patternCounts.green++;
32       } else if (colorMatchesPattern(pixelColor, color(0, 0, 255))) {
33         patternCounts.blue++;
34       }
35     }
36   }
37
38   // Calculate the percentages
39   let totalRGB = patternCounts.red + patternCounts.green + patternCounts.blue;
40   let redPercentage = (patternCounts.red / totalRGB) * 100;
41   let greenPercentage = (patternCounts.green / totalRGB) * 100;
42   let bluePercentage = (patternCounts.blue / totalRGB) * 100;
43
44   // Print the percentages
45   console.log('Red percentage:', redPercentage);
46   console.log('Green percentage:', greenPercentage);
47   console.log('Blue percentage:', bluePercentage);
48 }
49
50 function colorMatchesPattern(color1, color2) {
51   // Compare the RGB values of two colors for an exact match
52   return (red(color1) === red(color2) &&
53     green(color1) === green(color2) &&
54     blue(color1) === blue(color2));
55 }
```

Ο κώδικας δέχεται ως εισροή ένα
μοτίβο των τριών βασικών χρωμάτων
(RGB) σε μπρόχη pixels και υπολογίζει
το ποσοστό εμφάνισης κάθε
χρώματος.

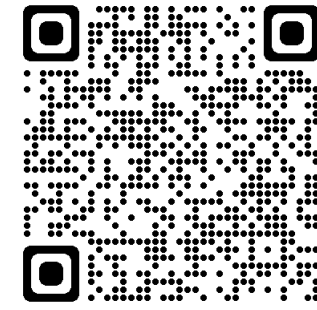
p5.js-Save_JSON_interval

Language

JavaScript 86.6%

HTML 12.2%

CSS 1.2%



Ο κώδικας είναι διαθέσιμος στο Github



8

EXPLORER

OPEN EDITORS

GROUP 1

JS serial_comm.js

GROUP 2

JS serial_comm.js

Settings

P5.JS-SAVE_JSON_INTERVAL

.gitattributes

index.html

README.md

JS serial_comm.js

sketch.js

splitData.js

style.css

OUTLINE

serial

latestData

setup

serverConnected

gotList

i

gotOpen

gotClose

gotError

gotData

currentString

serial_comm.js > gotData

1 // serial communication between a microcontroller with a switch on pin 2

2 // arduino code can be found here : <https://gist.github.com/shfitz/7fd206b7db4e0e64>

3

4 let serial; // variable for the serial object

5 let latestData = "waiting for data"; // variable to hold the data

6

7 function setup() {

8 createCanvas(800, 800);

9 background(255, 255, 255);

10 // serial constructor

11 serial = new p5.SerialPort();

12 // get a list of all connected serial devices

13 serial.list();

14 // serial port to use - you'll need to change this

15 serial.open("COM3");

16 // callback for when the sketches connects to the server

17 serial.on("connected", serverConnected);

18 // callback to print the list of serial devices

19 serial.on("list", gotList);

20 // what to do when we get serial data

21 serial.on("data", gotData);

22 // what to do when there's an error

23 serial.on("error", gotError);

24 // when to do when the serial port opens

25 serial.on("open", gotOpen);

26 // what to do when the port closes

27 serial.on("close", gotClose);

28 initiateIntervals(); //initiate the intervalls for date and saving

29 }

30

31 function serverConnected() {

32 console.log("Connected to Server");

33 }

34

35 // list the ports

36 function gotList(thelist) {

37 console.log("List of Serial Ports:");

38

39 for (let i = 0; i < thelist.length; i++) {

40 console.log(i + " " + thelist[i]);

41 }

42 }

43

serial_comm.js > gotData

44 function gotOpen() {

45 console.log("Serial Port is Open");

46 }

47

48 function gotClose() {

49 console.log("Serial Port is Closed");

50 latestData = "Serial Port is Closed";

51 }

52

53 function gotError(theerror) {

54 console.log(theerror);

55 }

56

57 // when data is received in the serial buffer

58

59 function gotData() {

60 let currentString = serial.readLine(); // store the data in a variable

61 trim(currentString); // get rid of whitespace

62 if (!currentString) return; // if there's nothing in there, ignore it

63 // console.log(currentString); // print it out

64 latestData = currentString; // save it to the global variable

65 splitData();

66 }

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

SERIAL MONITOR

Filter (e.g. text, **/*.ts, !**/node_modules/**)

No problems have been detected in the workspace.

current_changes

0 0 0

Συλλέγει δεδομένα από το Serial monitor του Arduino.

9

Visual Studio Code interface showing two JavaScript files: `splitData.js` and `splitData.js`.

Left Panel (EXPLORER):

- GROUP 1: `splitData.js`
- GROUP 2: `splitData.js`, `Settings`
- P5JS-SAVE_JSON_INTERVAL: `.gitattributes`, `index.html`, `README.md`, `serial_comm.js`, `sketch.js`, `splitData.js`, `style.css`
- OUTLINE: `DateToday`, `data`, `splitString`, `initiateIntervals`, `currentdate`, `currentdate`, `splitData`, `dataTrack`, `group`, `lightValue`, `rotationValue`, `time`, `timer`, `saveDataToFile`, `cleararray`

Right Panel (CODE EDITOR):

`splitData.js` (Left Editor):

```
1 //Storing minutes and seconds in data array and save them in a Json file, the name
2 //And after saving the array is cleared
3 let DateToday = "current date";
4 let data = [];
5 //let dataTrack;
6
7 let splitString;
8
9 function initiateIntervals() {
10   setInterval(timer, 60000); //to save the file every x milliseconds, 60000=60second
11   // setInterval(splitData(), 2000); // to update the array with current time every
12   setInterval(saveImage, 500); //to save the file every x milliseconds, 0.5seconds
13 }
14
15 //its beeing used for the JSON name, so it can be a unique every 1 second
16 function currentdate() {
17   let currentdate = year() + " " + month() + " " + day() + " " + hour() + " " + minute()
18   return currentdate;
19 }
20
21 //seperate data according to number of servo and store them in array
22 function splitData() {
23   splitString = split(latestData, ":");
24   let dataTrack = { //create this object with the following 3 propertiew
25     group: splitString[0],
26     lightValue: splitString[1],
27     rotationValue: splitString[2],
28     time: hour()+":" +minute()+":" +second()
29   };
30   drawing (dataTrack);
31   // GlobaldataTrack = dataTrack; //Store the dataTrack object in a global variable
32   data.push(dataTrack);
33   return dataTrack;
34 }
35
36 //when its called the procedure of saving is triggered
37 function timer() {
38   console.log("saving...");
39   DateToday = currentdate();
40   console.log(DateToday);
41   saveDataToFile();
42   cleararray();
43 }
```

`splitData.js` (Right Editor):

```
44
45 //when its called the data array is been saved in the JSON file
46 function saveDataToFile() {
47   saveJSON(data, DateToday);
48   console.log('Data saved to file.');
```

Bottom Panel (PROBLEMS):

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR

No problems have been detected in the workspace.

Filter (e.g. text, **/*.ts, !**/*.ts)

Διαχειρίζεται τα δεδομένα που ελήφθησαν. Τα κατηγοριοποιεί και τα αποθηκεύει σε ένα αρχείο .JSON για μελλοντική ανάλυση.

12

13

sketchjs - p5.js-Save_JSON_interval - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

JS sketchjs

OPEN EDITORS

GROUP 1

JS sketchjs

GROUP 2

JS sketchjs

Settings

P5.JS-SAVE_JSON_INTERVAL

.gitattributes

index.html

README.md

serial_comm.js

JS sketchjs

JS splitData.js

style.css

OUTLINE

draw

drawing

c

c

c

c

makeLine

makereellipse

maketext

saveimage

JS sketchjs > saveimage

```

1 function draw() {
2   // background(255, 255, 255);
3   //ellipse(width / 2, height / 2, 100, 100);
4 }
5
6 function drawing(dataTrack) {
7   // if (second() % 5 == 0) {
8   //   background(255);
9   // }
10  //maketext(dataTrack.lightValue)
11  if (dataTrack.group == "1st") {
12    let c = color(242, 77, 22, 10); //orange, port 11
13    makereellipse(200, 200, dataTrack.lightValue, dataTrack.rotationValue, c);
14    makeLine(200, 200, 300, 500);
15    // makebullet(200, 200, dataTrack.lightValue);
16  } else if (dataTrack.group == "2nd") {
17    let c = color(74, 36, 24, 10); //brown port 10
18    makereellipse(300, 500, dataTrack.lightValue, dataTrack.rotationValue, c);
19    // makeLine(width / 2, height / 2, dataTrack.lightValue, dataTrack.rotation
20    makeLine(300, 500, 550, 180);
21    // makebullet(300, 500, dataTrack.lightValue);
22  } else if (dataTrack.group == "3rd") {
23    let c = color(23, 224, 227, 10); //cyan port 9
24    makereellipse(550, 180, dataTrack.lightValue, dataTrack.rotationValue, c);
25    // makeLine(550, 180, dataTrack.lightValue, dataTrack.rotationValue);
26    makeLine(550, 180, 550, 550);
27    // makebulletReverse(550,180, dataTrack.lightValue);
28  } else if (dataTrack.group == "4th") {
29    let c = color(23, 227, 173, 10); //cyan-green port 9
30    makereellipse(550, 550, dataTrack.lightValue, dataTrack.rotationValue, c);
31    // makeLine(550, 550, dataTrack.lightValue, dataTrack.rotationValue);
32    makeLine(550, 550, 200, 200);
33  } else {
34    console.error("Label '" + dataTrack.group + "' doesn't much");
35  }
36 }
37
38
39 function makeLine(x1, x2, y1, y2) {
40   strokeWeight(1);
41   stroke(240);
42   line(x1, x2, y1, y2);
43 }

```

JS sketchjs > draw

```

44
45 // function makebullet(x, y, d) {
46 //   fill(0);
47 //   ellipse(x - d, y - d, 5)
48 // }
49
50 function makereellipse(x, y, a, b, c) {
51   stroke(255);
52   strokeWeight(1);
53   fill(c);
54   ellipse(x, y, a, b);
55 }
56
57 function maketext(a) {
58   text(a, width / 8, height / 8); // print the data to the sketch
59 }
60
61 function saveimage(){
62   DateToday = currentdate();
63   save(DateToday + ".jpg");
64   console.log("SavingImage")
65 }

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

SERIAL MONITOR

No problems have been detected in the workspace.

Filter (e.g. text, **/*.ts, !**/...)

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF () JavaScript Go Live

Οπτικοποιεί τα δεδομένα που λαμβάνονται απο το Arduino σε πραγματικό χρόνο και αποθηκεύει screenshots του τελικού αποτελέσματος.

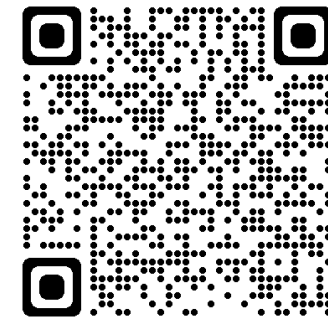
Perlin noise flow field

Language

JavaScript 81.4%

HTML 17.1%

CSS 1.5%



Ο κώδικας είναι
διαθέσιμος στο
Github



```
File Edit Selection View Go Run Terminal Help
sketchjs - perlin_noise_flow_field - Visual Studio Code

EXPLORER
OPEN EDITORS
GROUP 1
  JS sketchjs
GROUP 2
  JS sketchjs
PERLIN_NOISE_FLOW_FIELD
  .gitattributes
  index.html
  README.md
  JS sketchjs
  # style.css

JS sketchjs > [inc]
JS sketchjs > [inc]
1 > let inc = 0.9; // increment for perlin noise...
2 let cols, rows; // number of columns and rows
3 let zoff = 0; // z offset for perlin noise
4 let particles = []; // array to store particle objects
5
6
7 let flowfield;
8
9 class Particle {
10   constructor() {
11     this.pos = createVector(random(width), random(height)); // initialize particle position
12     this.vel = createVector(0, 0); // initialize particle velocity to zero
13     this.acc = createVector(0, 0); // initialize particle acceleration to zero
14     this.maxSpeed = 2; // set maximum particle speed
15     this.prevPos = this.pos.copy(); // initialize previous position to current position
16   }
17
18   update() {
19     this.vel.add(this.acc); // update velocity based on acceleration
20     this.vel.limit(this.maxSpeed); // limit velocity to maximum speed
21     this.pos.add(this.vel); // update position based on velocity
22     this.acc.mult(0); // reset acceleration to zero
23   }
24
25   applyForce(force) {
26     this.acc.add(force); // apply a force to the particle
27   }
28
29   follow(flowfield) {
30     let x = floor(this.pos.x / scl); // calculate x index in flow field
31     let y = floor(this.pos.y / scl); // calculate y index in flow field
32     let index = x + y * cols; // calculate index of vector in flow field
33     let force = flowfield[index]; // get vector from flow field
34     this.applyForce(force); // apply vector as force to particle
35   }
36
37   show() {
38     stroke(50, 255); // set stroke color to black with 50% opacity
39     strokeWeight(3); // set stroke weight to 1 pixel
40     line(this.pos.x, this.pos.y, this.prevPos.x, this.prevPos.y); // draw line from previous position to current position
41     this.prevPos = this.pos.copy(); // update previous position to current position
42   }
43
44   edges() {
45     if (this.pos.x > width) this.pos.x = 0; // wrap particle around if it goes off the right edge
46     if (this.pos.x < 0) this.pos.x = width; // wrap particle around if it goes off the left edge
47     if (this.pos.y > height) this.pos.y = 0; // wrap particle around if it goes off the bottom edge
48     if (this.pos.y < 0) this.pos.y = height; // wrap particle around if it goes off the top edge
49   }
50 }
51
```

Πειραματικά διαδικαστικά γραφικά, με έμφαση στην απεικόνιση της ιδέας του “καμβά” για της διαδραστικής εγκατάστασης. Ο κώδικας παρήχθη μετά από βρόχο ανατροφοδότησης με chatGPT.

Μελλοντικές εργασίες:

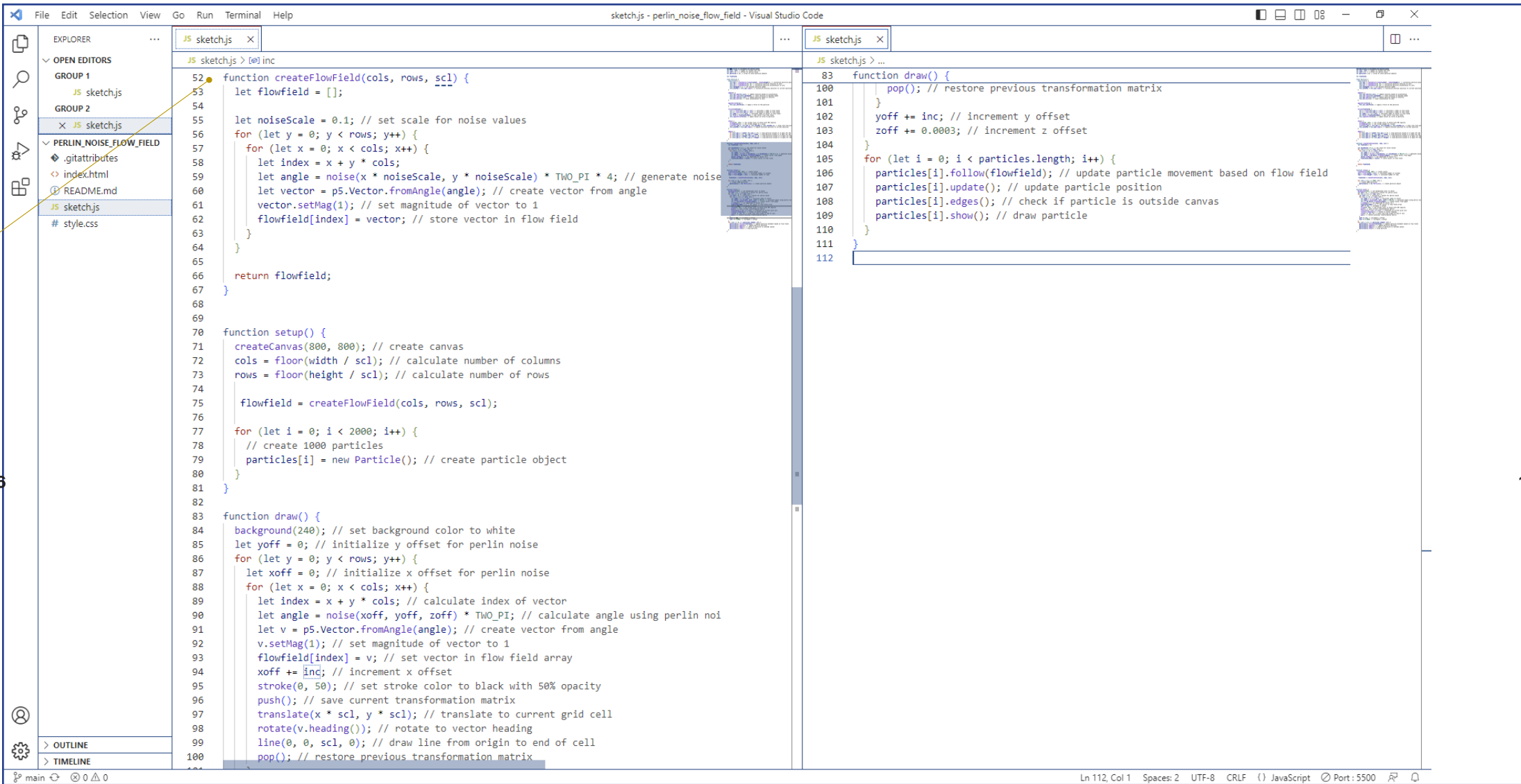
- εισαγωγή αλληλεπίδρασης με την πραγματική εγκατάσταση και τον χρήστη μέσω των διαδικαστικών γραφικών.
- βελτίωση του κώδικα

Μελλοντικές εργασίες:

- εισαγωγή αλληλεπίδρασης με την πραγματική εγκατάσταση και τον χρήστη μέσω των διαδικαστικών γραφικών.
- βελτίωση του κώδικα

14

15



[illegible]