

ARCH 4342 ΠΛΗΡΟΦΟΡΙΚΗ:

Υπολογιστικές Προσεγγίσεις στις Δημιουργικές Τέχνες και Επιστήμες

Δημήτριος Μπίμης - ar24759 - 2ο εξάμηνο

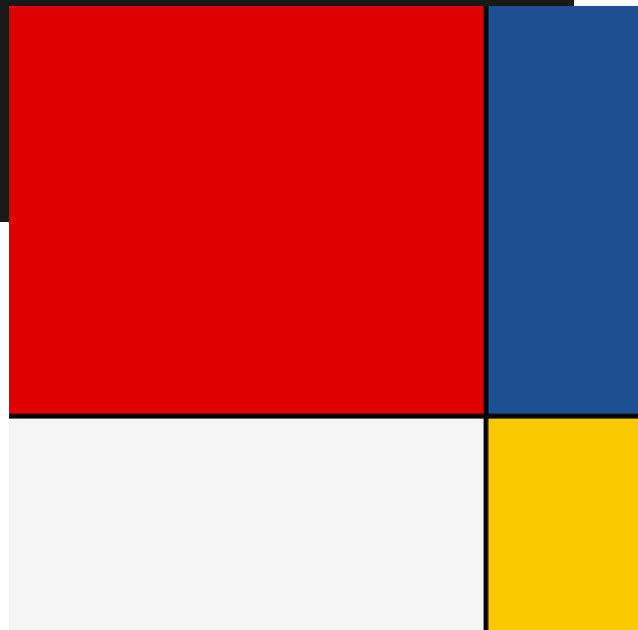
Assignment 02:

Πρόβλημα 2:

<https://editor.p5js.org/mhtsos/sketches/mNC39nz3M>

1. Σχεδιάσε μία γραμμή που αρχή θα έχει ένα παραμετροποιημένο σημείο από κάθε πλευρά του καμβά και τέλος τις συντεταγμένες του κέρσορα του ποντικιού.
2. Όρισε τα παραμετροποιημένα ορθογώνια και βάψε τα με χρώμα της επιλογής σου.

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4 function draw() {  
5  
6   /* I need to add the proper lines to make four rectangles, according to mouse coordinates */  
7   stroke(0);  
8   strokeWeight(3);  
9   line(mouseX, 0, mouseX, mouseY);  
10  line(0, mouseY, mouseX, mouseY);  
11  line(mouseX, width, mouseX, mouseY);  
12  line(height, mouseY, mouseX, mouseY);  
13  
14  /* Now I need to define and fill the four rectangles created with a different color(I will use  
the hues Mondrian used in his paintings) */  
15  fill(225, 1, 0);  
16  rect(0, 0, mouseX, mouseY);  
17  
18  fill(34, 80, 149);  
19  rect(mouseX, 0, width, mouseY);  
20  
21  fill(249, 249, 249);  
22  rect(0, mouseY, mouseX, height);  
23  
24  fill(250, 201, 1);  
25  rect(mouseX, mouseY, width, height);  
26 }  
27
```

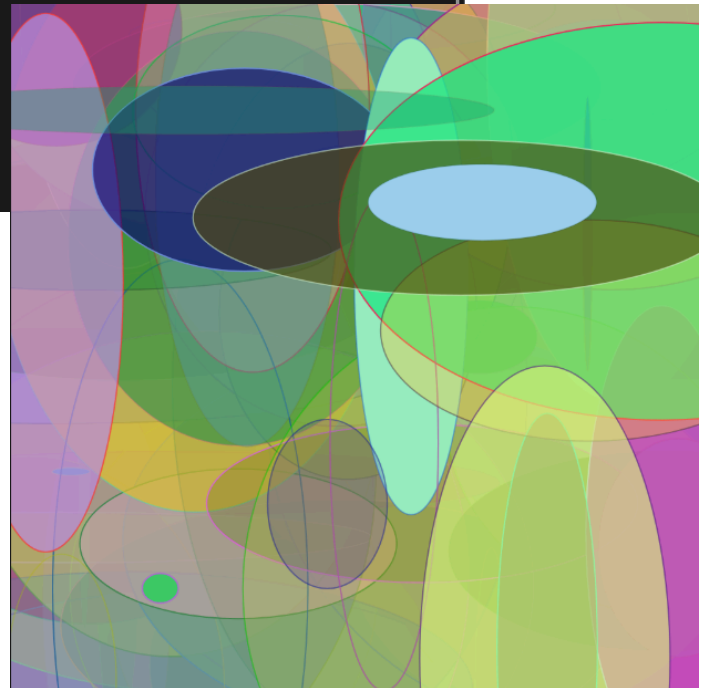


Πρόβλημα 3:

<https://editor.p5js.org/mhtsos/sketches/nDRT50b2J>

1. Όρισε τυχαίες συντεταγμένες για το κέντρο της κάθε έλλειψης.
2. Όρισε τυχαίες παραμέτρους για το μήκος και το πλάτος των ελλείψεων.
3. Όρισε δύο τυχαία χρώματα(ένα για το περίγραμμα και ένα για το εσωτερικό των ελλείψεων).
4. Ζωγράφισε τις ελλείψεις με τα τυχαία χαρακτηριστικά που ορίστηκαν παραπάνω.

```
1 function setup() {  
2   createCanvas(500, 500);  
3   background(255);  
4 }  
5  
6 function draw() {  
7   /* I need to define an element for every aspect that is going to be randomized */  
8   let x = random(0, width);  
9   let b = random(0, height);  
10  let n = random(0, width);  
11  let m = random(0, height);  
12  /* x and b are made for the center and n and m for the width and height of the ellipse */  
13  let y = random(0, 255);  
14  let z = random(0, 255);  
15  let c = random(0, 255);  
16  let v = random(0, 255);  
17  // y,z,c,v are made for colour hue and the transparency of the ellipses//  
18  
19  /*I need to set two colours(One for the fill and the other for the stroke of the ellipses). For that reason I need  
20  another four elements, to define a second random colour*/  
21  let q = random(0, 255);  
22  let w = random(0, 255);  
23  let t = random(0, 255);  
24  let r = random(0, 255);  
25  
26  let e = color(y, z, c, v);  
27  let f = color(q, w, t, r);  
28  
29  //I can also manage the tempo the ellipses generate//  
30  frameRate(20);  
31  
32  fill(e);  
33  stroke(f);  
34  ellipse(x, b, n, m);  
35 }
```

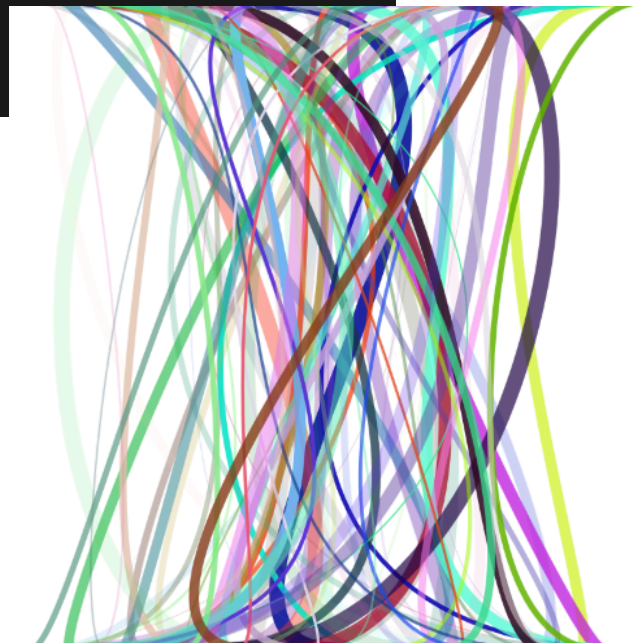


Πρόβλημα 4:

<https://editor.p5js.org/mhtsos/sketches/xE-9flmB5>

1. Όρισε τυχαίες παραμέτρους για τέσσερα τυχαία σημεία στη πάνω και κάτω πλευρά του καμβά
2. Όρισε τυχαίο χρώμα για τις καμπύλες Bezier.
3. Όρισε τυχαία παράμετρο για το πάχος των καμπυλών
4. Ζωγράφισε τις καμπύλες Bezier με τις παραπάνω παραμέτρους.

```
1 function setup() {  
2   createCanvas(400, 400);  
3   background(255,100,100);  
4 }  
5  
6 function draw() {  
7   let x = random(0, width);  
8   let y = random(0, width);  
9   let q = random(0, width);  
10  let e = random(0, width);  
11  //x, y, q, e will be used as the random start and end of the curves//  
12  
13  //I also need to define the colour of these curves//  
14  let z = random(0, 255);  
15  let c = random(0, 255);  
16  let v = random(0, 255);  
17  let b = random(0, 255);  
18  
19  let w = color(z, c, v, b);  
20  
21  stroke(w);  
22  //I can customize the thickness of the curves//  
23  let s = random(0, 10);  
24  strokeWeight(s);  
25  noFill();  
26  frameRate(10);  
27  bezier(x, 0, y, 0, q, height, e, height);  
28 }
```





ΜΕΡΟΣ ΤΡΙΤΟ:

https://editor.p5js.org/mhtsos/sketches/5_AAmHmvE

Επέλεξα να αναλύσω το έργο του Sol LeWitt: *The location of a trapezoid*

Πρώτο βήμα ήταν να διαιρέσω το κείμενο σε υποκατηγορίες με σκοπό την διευκόλυνση της κατασκευής του τραπεζίου. Η ανάλυση μου φαίνεται παρακάτω:

- A trapezoid whose top side is half as long as its bottom side and whose left side is one and a half times as long as the top side

and is located where **a line** drawn

from **a point** halfway between

a point halfway between the center of the square and the upper left corner

and a point halfway between the midpoint of the top side and the upper left corner

to **a point** halfway between the midpoint of the right side

and the upper right corner **is crossed by two lines,**

the first of which is drawn **from**

a point halfway between

the midpoint of the top side and a point halfway between the midpoint of the top side and the upper left corner

to a point halfway between

a point halfway between the center of the square and the midpoint of the bottom side

and a point halfway between the midpoint of the bottom side and the lower left corner

the second line is drawn **from** a point halfway between

the midpoint of the top side and the upper right corner

to the center of the square

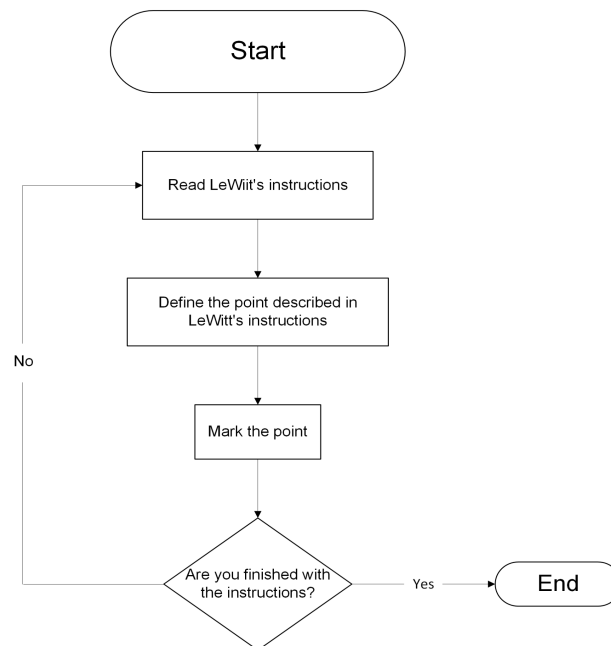
the left side is drawn to a point halfway between

a point halfway between the center of the square and the lower left corner

and a point halfway between the midpoint of the left side and the lower left corner.

Με αυτό τον τρόπο ξεκίνησα να γράφω τον κώδικα μου, ακολουθώντας τις οδηγίες του LeWitt. Η λογική μου φαίνεται και στο διπλανό flowchart.

Όσο ακολουθούσα τη συγκεκριμένη λογική χρειάστηκε να ορίσω το function `lineIntersection`, το οποίο δεν ήταν διαθέσιμο στην βιβλιοθήκη της `p5`. Για αυτό με την βοήθεια της `TN` στις σειρές 5 έως 14 του κώδικα το όρισα. (Αυτές οι γραμμές κώδικα φαίνονται και παρακάτω).



```
1 let p1, p2, p3, p4, p6, p7, p8, p9, p10, p11, p12;
2 let a;
3 let lengthSlider;
4
5 function lineIntersection(x1, y1, x2, y2, x3, y3, x4, y4) {
6   let denom = (x1 - x2) * (y3 - y4) - (y1 - y2) * (x3 - x4);
7   if (denom == 0) return null; // Lines are parallel
8
9   let px =
10    ((x1 * y2 - y1 * x2) * (x3 - x4) - (x1 - x2) * (x3 * y4 - y3 * x4)) / denom;
11   let py =
12    ((x1 * y2 - y1 * x2) * (y3 - y4) - (y1 - y2) * (x3 * y4 - y3 * x4)) / denom;
13
14   return createVector(px, py); // I used the help of AI for these lines of code
15 }
16 function setup() {
17   let a = 600;
18   createCanvas(a, a);
```

Ακολουθώντας τις οδηγίες έφτασα στο σημείο στο οποίο είχα σχεδιάσει την πάνω πλευρά του τραπέζιου και είχα την ευθεία στην οποία εδράζει η αριστερή πλευρά του. Αποφάσισα λοιπόν να ορίσω παραμετρικά την κάτω πλευρά, την οποία θα έλεγχα με τις συντεταγμένες του κέρσσορα του ποντικιού μου. Παρακάτω φαίνεται η διαδικασία που ακολούθησα σε μορφή ψευδοκώδικα, χωρισμένη σε δύο κομμάτια. Στο πρώτο ασχολούμαι με τον ορισμό του αριστερού σημείου της κάτω ευθείας. Στο δεύτερο με τον ορισμό ολόκληρης της ευθείας.

Πρώτο μέρος:

1. Δημιουργώ ένα διάνυσμα που ακολουθεί τον κέρσσορα του ποντικιού

2. Φτιάχνω ένα διάνυσμα από το αριστερό σημείο στο δεξί σημείο της πάνω ευθείας του τραπεζίου.
3. Φτιάχνω ένα άλλο διάνυσμα από το αριστερό σημείο της πάνω ευθείας του τραπεζίου στον κέρσορα του ποντικιού.
4. Προβάλλω το AM στο AB και περιορίζω τον παράγοντα t έτσι ώστε το σημείο να παραμένει εντός του τμήματος AB.
5. Βρίσκω το σημείο που θέλω ως το προβαλλόμενο σημείο από το διάνυσμα AM στο AB.

(Παρακάτω φαίνεται και η διαδικασία σε μορφή κώδικα)

```
156
157 //Firstly I want to set up the left endpoint//
158
159 let mouse = createVector(mouseX, mouseY);
160
161 let AB = p5.Vector.sub(m13, intersection1);
162 let AM = p5.Vector.sub(mouse, intersection1);
163
164 /* I am sketching a vector the same as the line (AB) and one that starts as one point of
the line and ends in the coordinates of my mouse (AM).That way I can find the
perpendicular vector each time from AM (my mouse) to AB (the line) */
165 let t = constrain(AM.dot(AB) / AB.magSq(), 0, 1);
166 //To make sure keeping the point within the line coordinates I use the constrain
function
167
168 movingPoint = p5.Vector.lerp(intersection1, m13, t);
169 //And now, I set up the point I want, as the closest point to the line from my mouse
coordinates//
170
```

Δεύτερο Μέρος:

1. Φτιάχνω ένα διάνυσμα με κατεύθυνση ίδια με την πάνω ευθεία του τραπεζίου.
2. Συνδέω το μήκος του διανύσματος με ένα slider.
3. Ενώνω τα δύο νέα σημεία με μία ευθεία.
4. Σχεδιάζω ελλείψεις σε κάθε άκρο της ευθείας.
5. Σχεδιάζω δύο ευθείες από κάθε πάνω γωνία προς τις αντίστοιχες κάτω γωνίες για να ολοκληρώσω το σχήμα του τραπεζίου.

(Παρακάτω φαίνεται και η διαδικασία σε μορφή κώδικα)

```
170
171 ▾ /* Now I have to setup the other point of the line in a way so its parallel to the top
line. */
172
173     let direction = p5.Vector.sub(intersection2, intersection1).normalize();
174     //By doing that I am setting up a vector with the same direction as the top line but not
with a fixed length yet//
175
176     let len = lengthSlider.value();
177     let newEndpoint = p5.Vector.add(movingPoint, direction.mult(len));
178     // Now I have set up the right point of the line I want to draw, which length is now
connected to the slider I have created.//
179
180     strokeWeight(3);
181     line(movingPoint.x, movingPoint.y, newEndpoint.x, newEndpoint.y);
182
183     ellipse(newEndpoint.x, newEndpoint.y, 8, 8);
184     ellipse(movingPoint.x, movingPoint.y, 8, 8);
185
186     //Lastly I am setting up the last lines to form the trapezoid//
187
188     line(intersection1.x, intersection1.y, movingPoint.x, movingPoint.y)
189     line(intersection2.x, intersection2.y, newEndpoint.x, newEndpoint.y)
190 }
191
```

Και το τελικό αποτέλεσμα του κώδικα: (μέσω του περιβάλλοντος της p5 είναι δυνατές και οι παραμετροποιήσεις που έχουν αναφερθεί).

