

ARCH 4342 ΠΛΗΡΟΦΟΡΙΚΗ:

Υπολογιστικές Προσεγγίσεις στις Δημιουργικές Τέχνες και Επιστήμες

ΑΣΚΗΣΗ 2 ~ΓΕΩΡΓΙΑ ΜΑΡΚΟΣΙΑΝ

Προβλημα 1

LINK: https://editor.p5js.org/GEORGIA_MARKOSIAN/sketches/1g5GdW1Th

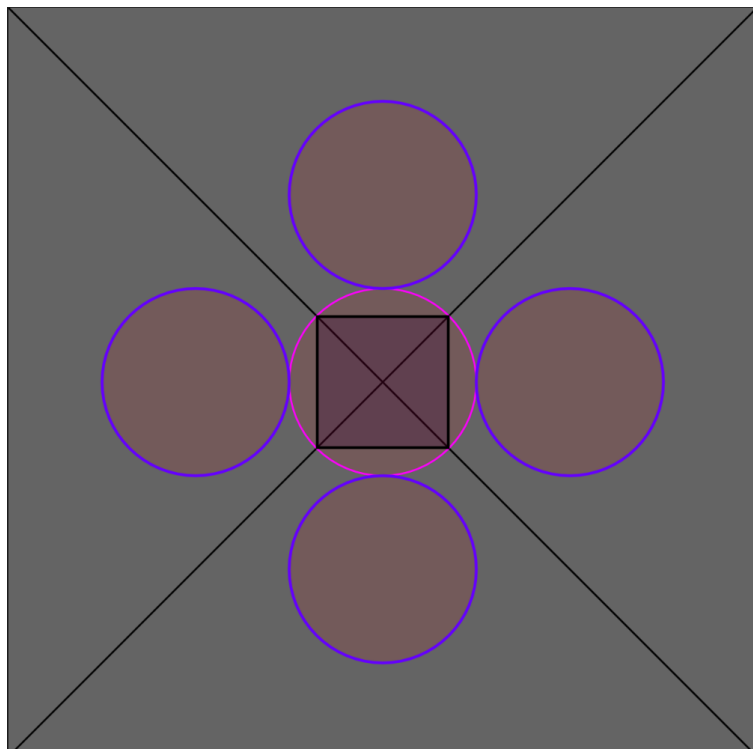
```
//Canvas
function setup() {
  createCanvas(400, 400);
}
function draw() {
  background(100)

  //Διαγωνιες γραμμες
  strokeWeight(1);
  stroke(0,0,0)
  line(0,0,400,400);
  line(400,0,0,400);

  //κεντρικος κυκλος
  stroke(255,0,255);
  ellipse(200,200,100,100);

  //τετραγωνο
  stroke(0,0,0)
  strokeWeight(1.5)
  fill(50,0,50,75)
  rectMode(CENTER);
  rect(200,200,70,70);

  //κυκλοι
  stroke(100,0,255)
  fill(255,0,0,25)
```



```
ellipse(100,200,100,100)
ellipse(200,100,100,100)
ellipse(300,200,100,100)
ellipse(200,300,100,100) }
```

Ψευδοκώδικας

ΑΡΧΗ

Συνάρτηση setup()

Δημιούργησε καμβά 400 * 400 πίξελες

Τέλος συνάρτησης setup()

Συνάρτηση draw()

--Ορισμός χρώματος καμβά σε γκρι

//Σχεδίαση διαγωνίων γραμμών

---ορισμός πάχους γραμμής : 1

---ορισμός χρώματος γραμμών: μαύρο

---σχεδίαση γραμμής από (0,0) σε (400,400)

--- σχεδίαση γραμμής από (400,0) σε (0,400)

// σχεδίαση κεντρικού κύκλου

---ορισμός χρώματος περιγράμματος :μωβ

---σχεδίαση κύκλου με κέντρο (200,200) και διάμετρο 100

//σχεδίαση τετραγώνου μέσα στον κύκλο

--ορισμός χρώματος στο περίγραμμα: μαύρο

--ορισμός πάχους γραμμής: 1,5

-- ορισμός γεμίσματος μωβ με διαφάνεια 75

-- ορισμός σχεδίασης τετραγώνου στο κέντρο

-- σχεδίαση τετραγώνου με κέντρο (200,200) και πλευρά 70

//σχεδίαση 4 εξωτερικών κύκλων

--ορισμός χρώματος περιγράμματος μπλε

--ορισμός γεμίσματος κόκκινο με διαφάνεια 25

--σχεδίαση κύκλου με κέντρο (100,200) και διάμετρο 100

-- σχεδίαση κύκλου με κέντρο (200,100) και διάμετρο 100

--σχεδίαση κύκλου με κέντρο (300,200) και διάμετρο 100

--σχεδίαση κύκλου με κέντρο (200,300) και διαμετρο 100

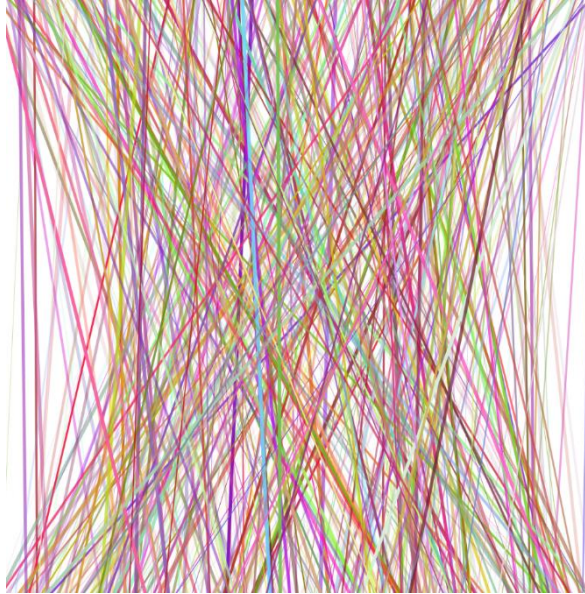
Τέλος συνάρτησης draw()

ΤΕΛΟΣ

Πρόβλημα 4

link; https://editor.p5js.org/GEORGIA_MARKOSIAN/sketches/OXuq-bRKL

```
function setup() {  
  createCanvas(400, 400);  
  
  background(255);}  
function draw() {  
  // Τυχαίο σημείο στην πάνω πλευρά  
  let x1 = random(0, width);  
  // Τυχαίο σημείο στην κάτω πλευρά  
  let x2 = random(0, width);  
  //μήκος γραμμής  
  let y1 = 0;  
  let y2 = height;  
  //τυχαίο χρώμα με τυχαία διαφάνεια  
  stroke(random(100,255), random(0,255), random(0,255), random(0, 255));  
  // Τυχαίο πάχος γραμμής  
  strokeWeight(random(0, 2.5));  
  // Σχεδίαση γραμμής  
  line(x1, y1, x2, y2);  
  
}
```



Βήματα

----Για το function setup()

Ορίζουμε τον καμβά σε 400 * 400 πίξελ και ορίζουμε το φόντο σε λευκό

---Για το function draw()

Ορίζουμε τυχαίο σημείο της πάνω πλευράς του καμβά και τυχαίο σημείο της κάτω πλευράς του καμβά μέσω των τυχαίων τιμών των οριζόντιων συντεταγμένων X1 και X2. Ορίζουμε Y1 να είναι 0 (πάνω πλευρά) και y2 να είναι ίσο με height δηλαδή στην κάτω πλευρά.

Επιλέγουμε ένα τυχαίο χρώμα με τυχαίο επίπεδο διαφάνειας και τυχαίο πάχος γραμμής.

Κάθε φορά που εκτελείται η draw(), σχεδιάζεται μία γραμμή από το (x1, y1) στο (x2, y2).

Έτσι δημιουργείται ένα μοτίβο με τυχαίες γραμμές διαφορετικού χρώματος, έντασης και διαφάνειας όπως φαίνεται στην εικόνα.

Πρόβλημα 3

LINK: https://editor.p5js.org/GEORGIA_MARKOSIAN/sketches/OXuq-bRKL

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
}  
  
function draw() {  
  //ημιδιαφανές φόντο  
  background(255, 255, 255, 5);  
  // Τυχαία θέση x  
  let x = random(width);  
  // Τυχαία θέση y  
  let y = random(height);  
  // Τυχαίο πλάτος έλλειψης  
  let w = random(0, 150);  
  // Τυχαίο ύψος έλλειψης  
  let h = random(0, 100);  
  //τυχαίο γέμισμα και περίγραμμα  
  fill(random(255), random(255), random(255), random(0, 255));  
  stroke(random(255), random(255), random(255), random(100, 255));  
  strokeWeight(random(0, 6));  
  // Σχεδίαση της έλλειψης  
  ellipse(x, y, w, h);  
}
```



Βήματα

Σκοπός είναι να δημιουργήσουμε ένα κώδικα που να σχεδιάζει τυχαίες ελλείψεις με διαφορετικά χρώματα ,μεγέθη και πάχη περιγράμματος . Οι ελλείψεις δεν παραμένουν στην οθόνη, αλλά σβήνουν σταδιακά και δημιουργούν ένα ίχνος από την κίνηση τους.

ΑΡΧΗ

1. Δημιούργησε έναν καμβά με πλάτος και ύψος 400x400 pixels
2. Όρισε το φόντο σε λευκό

ΕΠΑΝΑΛΑΒΕ ΓΙΑ ΠΑΝΤΑ (συνάρτηση draw)

3. ορισμός φόντου με μικρή διαφάνεια (alpha 5)
4. Επιλογή τυχαίας θέσης (x, y) στον καμβά
5. Επιλογή τυχαίου μεγέθους έλλειψης(πλάτος w, ύψος h)
6. Επιλογή τυχαίου χρώματος γεμίσματος (RGB + τυχαία διαφάνεια)
7. Επιλογή τυχαίου χρώματος περιγράμματος (RGB + τυχαία διαφάνεια)
8. Επιλογή τυχαίου πάχους περιγράμματος
9. Σχεδίαση έλλειψης στη θέση (x, y) με τις καθορισμένες παραμέτρους

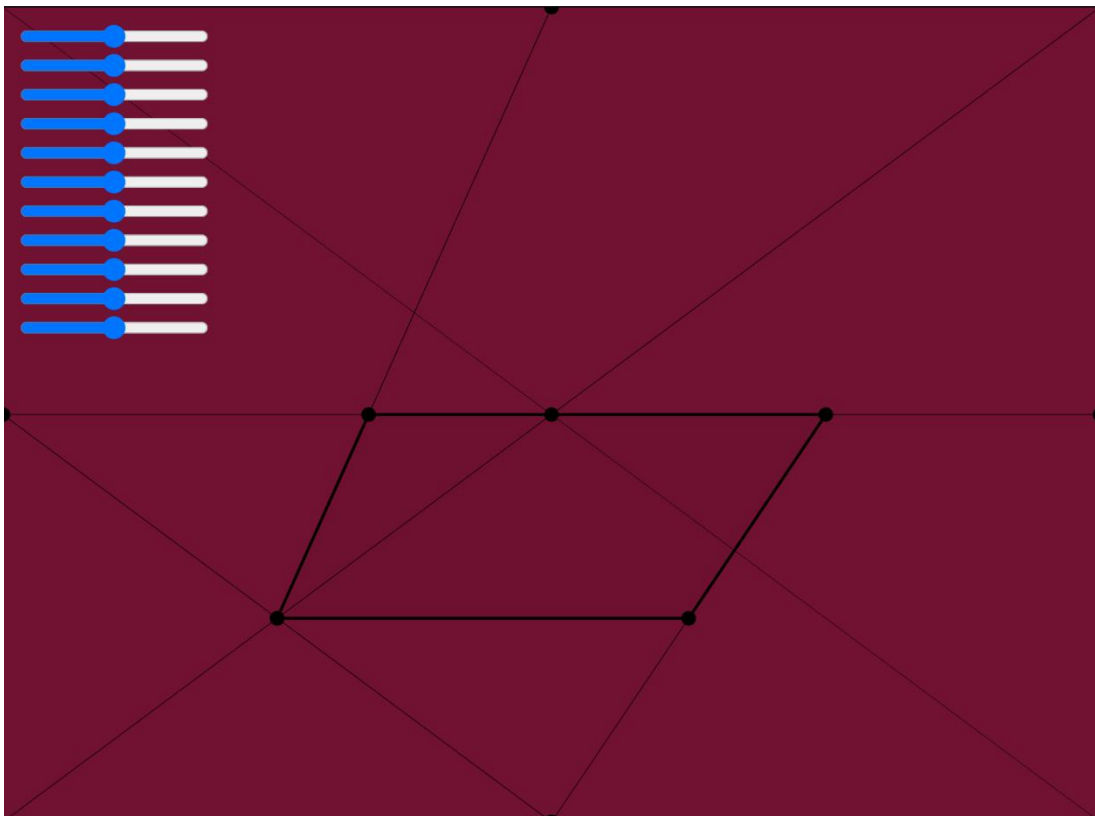
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ

ΜΕΡΟΣ ΤΡΙΤΟ

link : https://editor.p5js.org/GEORGIA_MARKOSIAN/sketches/8POH_ZBns

Σκοπός είναι να δημιουργήσουμε ένα τετράπλευρο σύμφωνα με τις οδηγίες του Sole w it, όπου οι πλευρές του μεταβάλλονται παραμετρικά μέσω slider. Επιπλέον , μέσω της κίνησης του ποντικιού μεταβάλλεται το χρώμα του φόντου .



```

// Ορισμός των μεταβλητών για τα sliders t και τα σημεία f
let t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11;
let f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16;

function setup() {
  createCanvas(windowWidth, windowHeight);
  // Ορισμός των κορυφών του τετραπλεύρου
  f1 = createVector(0, 0);
  f2 = createVector(width, 0);
  f3 = createVector(width,height);
  f4 = createVector(0, height);

  //αρχικές μεταβλητές θέσεων
  f5 = createVector();
  f6 = createVector();
  f7 = createVector();
  f8 = createVector();
  f9 = createVector();
  f10 = createVector();
  f11 = createVector();
  f12 = createVector();
  f13 = createVector();
  f14 = createVector();

  // Δημιουργία sliders για παραμετρικό έλεγχο
  t1 = createSlider(0, 100).position(10, 10);
  t2 = createSlider(0, 100).position(10, 30);
  t3 = createSlider(0, 100).position(10, 50);
  t4 = createSlider(0, 100).position(10, 70);
  t5 = createSlider(0, 100).position(10, 90);
  t6 = createSlider(0, 100).position(10, 110);
  t7 = createSlider(0, 100).position(10, 130);
  t8 = createSlider(0, 100).position(10, 150);
  t9 = createSlider(0, 100).position(10, 170);
  t10 = createSlider(0, 100).position(10, 190);
  t11 = createSlider(0, 100).position(10, 210);
}

```

```

function draw() {
  // Ορισμός χρώματος ανάλογα με τη θέση του ποντικιού
  background(mouseX % 255, mouseY % 255, 50);
  stroke(0);
  noFill();
  strokeWeight(0.5);

  // Σχεδίαση του βασικού τετραπλεύρου
  quad(f1.x,f1.y,f2.x,f2.y, f3.x, f3.y, f4.x,f4.y);
  line(f2.x,f2.y,f4.x,f4.y);

  // Υπολογισμός ενδιαμέσων σημείων μέσω παραμετρικών συντεταγμένων
  f5 = interpolatePoint(f2,f4, t1.value()/100);
  f6 = interpolatePoint(f2,f3, t2.value()/100);
  f7 = interpolatePoint(f5,f6, t3.value()/100);
  f8 = interpolatePoint(f3,f4, t4.value()/100);
  f9 = interpolatePoint(f7,f8, t5.value()/100);
  f10 = interpolatePoint(f3,f4, t6.value()/100);
  f11 = interpolatePoint(f1,f4, t7.value()/100);
  f12 = interpolatePoint(f10,f11, t8.value()/100);
  f13 = interpolatePoint(f1,f4, t9.value()/100);
  f14 = interpolatePoint(f10,f11, t10.value()/100);
  f15 = interpolatePoint(f1,f2, t11.value()/100);
  f16 = intersectLines(f14,f15,f12,f9);
  f17 = intersectLines(f14,f15,f13,f7);

  // Σχεδίαση των γραμμών
  line(f5.x,f5.y,f6.x,f6.y);
  line(f7.x,f7.y,f8.x,f8.y);
  line(f10.x,f10.y,f11.x,f11.y);
  line(f12.x,f12.y,f9.x,f9.y);
  line(f13.x,f13.y,f7.x,f7.y);
  line(f14.x,f14.y,f15.x,f15.y);
  strokeWeight(10);
  point(f5.x,f5.y);
  point(f6.x,f6.y);

```

```

point(f7.x,f7.y);
point(f8.x,f8.y);
point(f9.x,f9.y);
point(f10.x,f10.y);
point(f11.x,f11.y);
point(f12.x,f12.y);
point(f13.x,f13.y);
point(f14.x,f14.y);
point(f15.x,f15.y);
point(f16.x,f16.y);
point(f17.x,f17.y);

strokeWeight(0.3);
line(f1.x,f1.y,f3.x,f3.y);
fill(0,10);
strokeWeight(2);
quad(f17.x, f17.y, f7.x, f7.y, f9.x, f9.y, f16.x, f16.y)

}
// Συνάρτηση για εύρεση ενδιάμεσου σημείου μεταξύ δύο διανυσμάτων
function interpolatePoint(f1,f2, t){
  let x = t * (f2.x-f1.x) + f1.x;
  let y = t * (f2.y-f1.y) + f1.y;
  let f = createVector(x, y);
  return f;
}
// Συνάρτηση για υπολογισμό της τομής δύο ευθειών
function intersectLines(f1,f2,f3,f4){
  var a_12 = (f2.y-f1.y)/(f2.x-f1.x); // Κλίση πρώτης ευθείας
  var b_12 = f1.y - a_12 * f1.x; // Τέμνουσα πρώτης ευθείας
  var a_34 = (f4.y-f3.y)/(f4.x-f3.x); // Κλίση δεύτερης ευθείας
  var b_34 = f3.y - a_34 * f3.x; // Τέμνουσα δεύτερης ευθείας
  var _x = - (b_34-b_12)/(a_34-a_12);
  var _y = a_12 * _x + b_12;
  var _f = createVector(_x,_y);
  return _f;
}

```


ΨΕΥΔΟΚΩΔΙΚΑΣ

ΑΡΧΗ

Δημιουργία καμβά με διαστάσεις ίσες με το παράθυρο

Ορισμός 11 sliders t_1, t_2, \dots, t_{11}

Ορισμός 16 σημείων f_1, f_2, \dots, f_{16} για τις κορυφές και ενδιάμεσα σημεία

ΕΠΑΝΑΛΑΒΕ (Συνάρτηση `draw()`)

ρύθμιση χρώματος του φόντου να αλλάζει ανάλογα με τις συντεταγμένες του ποντικιού:

-Κόκκινο: `map(mouseX, 0, width, 200, 255)`

-Πράσινο: `map(mouseY, 0, height, 200, 255)`

-Μπλε: `map(mouseX + mouseY, 0, width + height, 150, 255)`

Σχεδίαση τετραπλεύρου f_1, f_2, f_3, f_4

Σχεδίαση διαγώνιας $f_2 \rightarrow f_4$

Υπολογισμός ενδιάμεσων σημείων με παρεμβολή

$f_5 \leftarrow \text{interpolate}(f_2, f_4, t_1)$

$f_6 \leftarrow \text{interpolate}(f_2, f_3, t_2)$

$f_7 \leftarrow \text{interpolate}(f_5, f_6, t_3)$

$f_8 \leftarrow \text{interpolate}(f_3, f_4, t_4)$

$f_9 \leftarrow \text{interpolate}(f_7, f_8, t_5)$

$f_{10} \leftarrow \text{interpolate}(f_3, f_4, t_6)$

$f_{11} \leftarrow \text{interpolate}(f_1, f_4, t_7)$

$f_{12} \leftarrow \text{interpolate}(f_{10}, f_{11}, t_8)$

$f_{13} \leftarrow \text{interpolate}(f_1, f_4, t_9)$

$f_{14} \leftarrow \text{interpolate}(f_{10}, f_{11}, t_{10})$

$f_{15} \leftarrow \text{interpolate}(f_1, f_2, t_{11})$

Υπολογισμός σημείων τομής

$f_{16} \leftarrow \text{intersection}(f_{14}, f_{15}, f_{12}, f_9)$

$f_{17} \leftarrow \text{intersection}(f_{14}, f_{15}, f_{13}, f_7)$

Σχεδίαση γραμμών που συνδέουν τα ενδιάμεσα σημεία

Σχεδίαση κύριων σημείων ως μεγάλες κουκκίδες

Σχηματισμός τετραπλεύρου με κορυφές f_{17}, f_7, f_9, f_{16}

ΣΥΝΑΡΤΗΣΗ `interpolate(fA, fB, t)`

-Υπολογισμός νέου σημείου μεταξύ f_A και f_B ως ποσοστό t

ΣΥΝΑΡΤΗΣΗ `intersection(fA, fB, fC, fD)`

-Υπολογισμός τομής δύο ευθειών μέσω εξισώσεων

ΤΕΛΟΣ