

ARCH 4342 ΠΛΗΡΟΦΟΡΙΚΗ:

Υπολογιστικές Προσεγγίσεις στις Δημιουργικές Τέχνες και Επιστήμες

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 3

ΓΕΩΡΓΙΑ ΜΑΡΚΟΣΙΑΝ

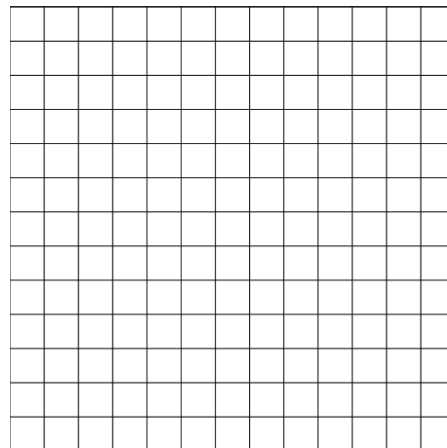
Πρόβλημα 1 / Arrays & Iteration

Βήμα 1: Χρησιμοποιώντας ένα διπλό βρόχο, χωρίζω τον καμβά σε δεδομένο αριθμό γραμμών και στηλών δημιουργώντας κελιά.

```
var seires=13
var sthles=13
var megethosKeliou=45

function setup() {
  createCanvas(seires * megethosKeliou, sthles *
megethosKeliou);
  background(255)
  stroke(0)
  strokeWeight(1)
  for (var a=0; a<seires; a++){
    for (var b=0; b<sthles; b++){
      let x=a*megethosKeliou
      let y=b*megethosKeliou

      rect(x,y,megethosKeliou,megethosKeliou)
    }
  }
}
```



Βήμα 2: Για κάθε κελί του κανάβου ξεχωριστά, ορίζω τυχαίο αριθμό υπό-γραμμών και υπό-στηλών και στρογγυλοποιώ τους αριθμούς ώστε να είναι ακέραιοι.

```

var seires=13
var sthles=13
var megethosKeliou=40

let yposeires=[]
let yposthles=[]

function setup() {
  createCanvas(seires * megethosKeliou, sthles * megethosKeliou);
  background(255)
  stroke(0)
  strokeWeight(1)

  for (var a=0; a<seires; a++){
    yposeires[a]=[]
    yposthles[a]=[]
    for (var b=0; b<sthles; b++){
      yposeires[a][b] = round(random(3, 10))
      yposthles[a][b] = round(random(3, 10))
    }
  }
  for (var a=0; a<seires; a++){
    yposeires[a]=[]
    yposthles[a]=[]
    for (var b=0; b<sthles; b++){
      let x=a*megethosKeliou
      let y=b*megethosKeliou

      rect(x,y,megethosKeliou,megethosKeliou)
    }
  }
}

```

Βήμα 3: Για κάθε κελί, χρησιμοποιώντας έναν διπλό βρόχο, χωρίζω το κελί σε υπογραμμές και υποστήλες σύμφωνα με τους τυχαίους αριθμούς που δημιουργήθηκαν, δημιουργώντας υποκελιά. **Βήμα 4:** Για κάθε υποκελί, σχεδιάζω ένα ορθογώνιο σύμφωνα με τις διαστάσεις του υποκελιού, δημιουργώ τυχαία απόχρωση του ίδιου χρώματος, και χρωματίζω το υπό-κελί.

```
var seires=13
```

```
var sthles=13
```

```
var megethosKeliou=40
```

```
let yposeires=[]
```

```
let yposthles=[]
```

```
function setup() {
```

```
  createCanvas(seires
```

```
*
```

```
  megethosKeliou,      sthles
```

```
*
```

```
  megethosKeliou);
```

```
  background(255)
```

```
  stroke(0)
```

```
  strokeWeight(0.2)
```

```
  for (var a=0; a<seires; a++){
```

```
    yposeires[a]=[]
```

```
    yposthles[a]=[]
```

```
    for (var b=0; b<sthles; b++){
```

```
      yposeires[a][b] = round(random(3, 10))
```

```
      yposthles[a][b] = round(random(3, 10))
```

```
    }
```

```
  }
```

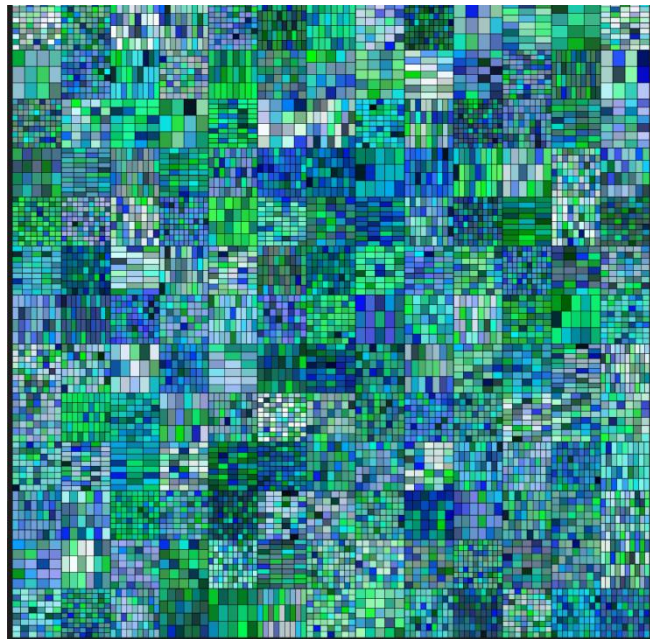
```
  for (var a=0; a<seires; a++){
```

```
    for (var b=0; b<sthles; b++){
```

```
      let x=a*megethosKeliou
```

```
      let y=b*megethosKeliou
```

```
      rect(x,y,megethosKeliou,megethosKeliou);
```



```

let yposeires1=yposeires[a][b]
let yposthles1=yposthles[a][b]

let platosYpokeliou=megethosKeliou/yposthles1;
let ypsosYpokeliou=megethosKeliou/yposeires1;

for (var i=0; i<yposeires1; i++){
  for (var j=0; j<yposthles1; j++){
    let subx=x+j* platosYpokeliou;
    let suby=y+i*ypsosYpokeliou;

    fill(random(0),random(0,255),random(0,255),random(0,255))
    rect(subx,suby,platosYpokeliou,ypsosYpokeliou)
  }}}

```

ΤΕΛΙΚΗ ΜΟΡΦΗ ΚΩΔΙΚΑ ΜΕ ΣΧΟΛΙΑ ΚΑΙ ΠΡΟΣΘΗΚΗ ΔΙΑΔΡΑΣΤΙΚΟΤΗΤΑΣ

https://editor.p5js.org/GEORGIA_MARKOSIAN/sketches/wR0DzBTNV

```

//ορίζω τις σειρες , τις στηλες του καναβου και το μεγεθος των μεγαλων κελιων
var seires=13
var sthles=13
var megethosKeliou=40
//δημιουργω array για υποσειρες και υποστηλες
let yposeires=[]
let yposthles=[]

//φτιαχνω καμβα
function setup() {
  createCanvas(seires * megethosKeliou, sthles * megethosKeliou);

  stroke(0)
  strokeWeight(0.2)

```

//με διπλο βροχο χωριζω καθε κελι σε υπογραμμες και υποστηλες και στρογγυλοποιω ωστε να
ειναι ακεραιοι

```
for (var a=0; a<seires; a++){  
  yposeires[a]=[]  
  yposthles[a]=[]  
  for (var b=0; b<sthles; b++){  
    yposeires[a][b] = round(random(3, 10))  
    yposthles[a][b] = round(random(3, 10))  
  }  
} drawGrid() // // σχεδιαζω τον αρχικο κάνναβο με τη συνάρτηση drawGrid()  
}
```

// η drawGrid() σχεδιάζει όλα τα κελιά και τις υποδιαιρέσεις τους

//με διπλο βροχο σχεδιαζω ορθογωνια ως κελια

```
function drawGrid(){  
  for (var a=0; a<seires; a++){  
    for (var b=0; b<sthles; b++){  
      let x=a*megethosKeliou  
      let y=b*megethosKeliou  
      rect(x,y,megethosKeliou,megethosKeliou);
```

//παιρνω ποσες υποσειρες και ποσες υποστηλες εχει το συγκεκριμενο κελι

```
let yposeires1=yposeires[a][b]
```

```
let yposthles1=yposthles[a][b]
```

//βρισκω το μεγεθος καθε υποκελιου

```
let platosYpokeliou=megethosKeliou/yposthles1;
```

```
let ypsosYpokeliou=megethosKeliou/yposeires1;
```

//με διπλο βροχο σχεδιαζω τα μικρα υποκελια με τυχαιο χρωμα

```
for (var i=0; i<yposeires1; i++){
```

```
  for (var j=0; j<yposthles1; j++){
```

```

    let subx=x+j* platosYpokeliou; // x θέση του υποκελίου
    let suby=y+i*ypsosYpokeliou;
    //τυχαία χρώματα
    fill(random(0),random(0,255),random(0,255),random(0,255))
    rect(subx,suby,platosYpokeliou,ypsosYpokeliou) //υποκελί
  }
}
}
}
}

//με κλικ στο ποντίκι μέσα στον καμβά ξανασχεδιάζεται ο κανάβος
function mousePressed() {
  if (
    mouseX >= 0 &&
    mouseX <= width &&
    mouseY >= 0 &&
    mouseY <= height
  ){
    drawGrid();
  }
}

```

Αρχικά, ορίζονται τρεις βασικές μεταβλητές: `seires` και `sthles`, που καθορίζουν πόσες σειρές και στήλες θα έχει ο κάνναβος, καθώς και `megethosKeliou`, που δηλώνει το μέγεθος κάθε μεγάλου κελιού. Παράλληλα, δημιουργούνται δύο δισδιάστατοι πίνακες (`yposeires` και `yposthles`) που αποθηκεύουν για κάθε κελί τον αριθμό των υποδιαιρέσεων που θα έχει οριζόντια και κάθετα αντίστοιχα.

Στη συνέχεια, δημιουργώ τη συνάρτηση `setup()`, η οποία δημιουργεί τον καμβά και αρχικοποιεί τους πίνακες με τυχαίες ακέραιες τιμές από 3 έως 10 για κάθε κελί. Οι τιμές αυτές καθορίζουν πόσα μικρότερα υπο-κελιά θα περιέχει κάθε μεγάλο κελί, δίνοντας έτσι διαφορετική πυκνότητα ανά περιοχή του κανάβου. Τέλος,, η συνάρτηση `drawGrid()` σχεδιάζει ολόκληρο τον κανάβο.

Η `drawGrid()` αποτελεί τον πυρήνα του σχεδίου. Μέσω δύο βρόχων `for`, περνάει από κάθε κελί, υπολογίζει τις διαστάσεις του και σχεδιάζει πρώτα το μεγάλο κελί με μαύρο περίγραμμα. Έπειτα,

κάθε μεγάλο κελί χωρίζεται σε υπο-κελιά με βάση τον αριθμό υποσειρών και υποστηλών που του έχουν αποδοθεί. Τα υπο-κελιά σχεδιάζονται γεμάτα με χρώματα που επιλέγονται τυχαία, δημιουργώντας ένα ποικιλόχρωμο, γεωμετρικά οργανωμένο αποτέλεσμα.

Η διαδραστικότητα επιτυγχάνεται με τη χρήση της `mousePressed()` συνάρτησης. Κάθε φορά που κάποιος θα πατήσει κλικ με το ποντίκι, αν βρίσκεται εντός των ορίων του καμβά, τότε καλείται η `drawGrid()`, επανασχεδιάζοντας ολόκληρο τον κάνναβο με νέο, τυχαίο χρωματισμό για όλα τα υπο-κελιά.