

Transformări grafice 2D

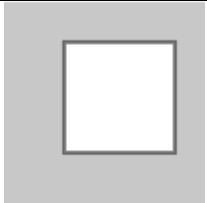
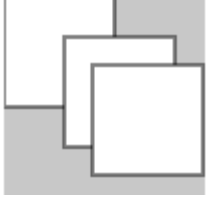
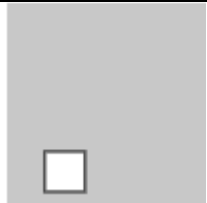
Transformări grafice 2D în biblioteca grafică p5jc:

- (translate)
- (scale)
- (rotate)

Translarea

Funcția **translate()** permite mutarea obiectelor în orice locație din fereastră. În **Tabelul 2.1** este indicate sintaxa translației.

Tabelul 2.1

	<pre>translate(30, 20); rect(0, 0, 55, 55);</pre>
	<pre>rect(0, 0, 55, 55); // Draw rect at original 0,0 translate(30, 20); rect(0, 0, 55, 55); // Draw rect at new 0,0 translate(14, 14); rect(0, 0, 55, 55); // Draw rect at new 0,0</pre>
	<pre>function draw() { background(200); rectMode(CENTER); translate(width / 2, height / 2); translate(p5.Vector.fromAngle(millis() / 1000, 40)); rect(0, 0, 20, 20); }</pre>

În exemplul de mai jos vedeți metoda de executare a translației în cod:

```
let x = 0;  
let y = 0;  
let dim = 80.0;  
  
function setup() {  
  createCanvas(720, 400);  
  noStroke();  
}
```

```

function draw() {
  background(102);

  // Animează creșterea valorii x

  x = x + 0.8;

  // Dacă forma iese din grilă, resetați poziția

  if (x > width + dim) {
    x = -dim;
  }

  // Chiar dacă comanda noastră „rect” desenează forma cu centrul său la
  // origine, translația o mută în noua poziție x și y

  translate(x, height / 2 - dim / 2);
  fill(255);
  rect(-dim / 2, -dim / 2, dim, dim);

  // Transformările se acumulează. Observați cum acest „rect”
  // se mișcă de două ori mai repede decât celălalt, dar are
  // același parametru pentru valoarea axei x

  translate(x, dim);
  fill(0);
  rect(-dim / 2, -dim / 2, dim, dim);
}

```

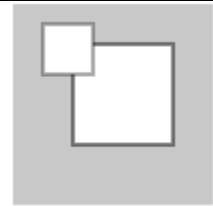
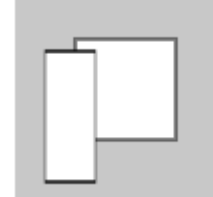


Figura 2.1 Translarea

Scalarea

Funcția **scale()** transformare care redimensionează un element în planul 2D. Deoarece cantitatea de scalare este definită de un vector, acesta poate redimensiona dimensiunile orizontale și verticale la scări diferite. În **Tabelul 2.2** este indicate sintaxa scalării.

Tabelul 2.2

	<pre>rect(30, 20, 50, 50); scale(0.5); rect(30, 20, 50, 50);</pre>
	<pre>rect(30, 20, 50, 50); scale(0.5, 1.3); rect(30, 20, 50, 50);</pre>

Parametrii pentru funcția **scale()** sunt valori specificate ca procente zecimale. De exemplu, metoda `scale(2.0)` va crește dimensiunea formei cu 200 la sută. Obiectele se întind întotdeauna de la origine. Acest exemplu arată cum se acumulează transformările și, de asemenea, cum interacționează scala și interschimbarea în funcție de ordinea lor. În exemplul de mai jos vedeți o metodă de executare a translației în cod:

```
let a = 0.0;  
let s = 0.0;  
  
function setup() {  
  createCanvas(720, 400);  
  noStroke();  
  
  // Desenați toate dreptunghiurile din centrul lor  
  
  rectMode(CENTER);
```

```

}

function draw() {
  background(102);

  // Creșteți „a” și apoi animați „s” cu
  // o mișcare ciclică prin găsirea cosinusului „a”

  a = a + 0.04;
  s = cos(a) * 2;

  // Translați dreptunghiul de la origine la mijlocul
  // grlei, apoi scalați-l cu „s”

  translate(width / 2, height / 2);
  scale(s);
  fill(51);
  rect(0, 0, 50, 50);

  // Translarea și scala se acumulează, această translare
  // deplasează al doilea dreptunghi mai la dreapta decât primul
  // și scara se dublează. Rețineți că cosinusul este
  // făcând „s” atât negativ, cât și pozitiv, astfel se ciclează
  // de la stanga la dreapta.

  translate(75, 0);
  fill(255);
  scale(s);
  rect(0, 0, 50, 50);
}

```


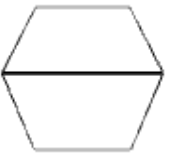
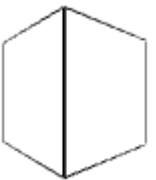
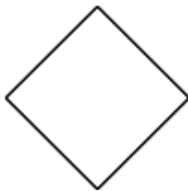


Figura 2.2 Translarea

Rotire

Funcția de rotire definește o transformare care rotește un element în jurul unui punct fix pe planul 2D, fără a-l deforma. Rezultatul său este un tip de date `rotate()`

Tabelul 2.3

	<pre>translate(width / 2, height / 2); rotate(PI / 3.0); rect(-26, -26, 52, 52);</pre>
	<pre>function setup() { createCanvas(100, 100, WEBGL); } function draw() { background(255); rotateX(millis() / 1000); box(); }</pre>
	<pre>function setup() { createCanvas(100, 100, WEBGL); } function draw() { background(255); rotateY(millis() / 1000); box(); }</pre>
	<pre>function setup() { createCanvas(100, 100, WEBGL); } function draw() { background(255); rotateZ(millis() / 1000); box(); }</pre>

Rotirea unui pătrat în jurul axei Z. Pentru a obține rezultatele pe care le așteptați, trimiteți parametrii unghiului funcției de rotație care sunt valori cuprinse între 0 și $PI * 2$ (`TWO_PI` care este de aproximativ 6,28). Dacă preferați să vă gândiți la unghiuri ca grade (0-360), puteți utiliza metoda `radians()` pentru a vă converti valorile. De exemplu: `rotate(radians(90))` este identic cu declarația `rotate`

($\pi / 2$). În acest exemplu, la fiecare secundă numerotată pară se adaugă o rotație la rotație. În secunde impare, rotația deplasează CW și CCW la viteza determinată de ultima valoare a jitterului.

```
let angle = 0.0;
let jitter = 0.0;

function setup() {
  createCanvas(720, 400);
  noStroke();
  fill(255);

  // Desenați dreptunghiul din centru care va fi și
  // rotația în jurul aceluia centru

  rectMode(CENTER);
}

function draw() {
  background(51);

  // în secunde par (0, 2, 4, 6...) adăugați jitter la rotație

  if (second() % 2 === 0) {
    jitter = random(-0.1, 0.1);
  }

  // măriți valoarea unghiului folosind
  // cea mai recentă valoare de jitter

  angle = angle + jitter;

  // utilizați cosinusul pentru a obține
  // o mișcare lină CW și CCW atunci când nu vibrați

  let c = cos(angle);

  // mutați forma în centrul grilei

  translate(width / 2, height / 2);

  // aplicați rotația finală

  rotate(c);
  rect(0, 0, 180, 180);
}
```

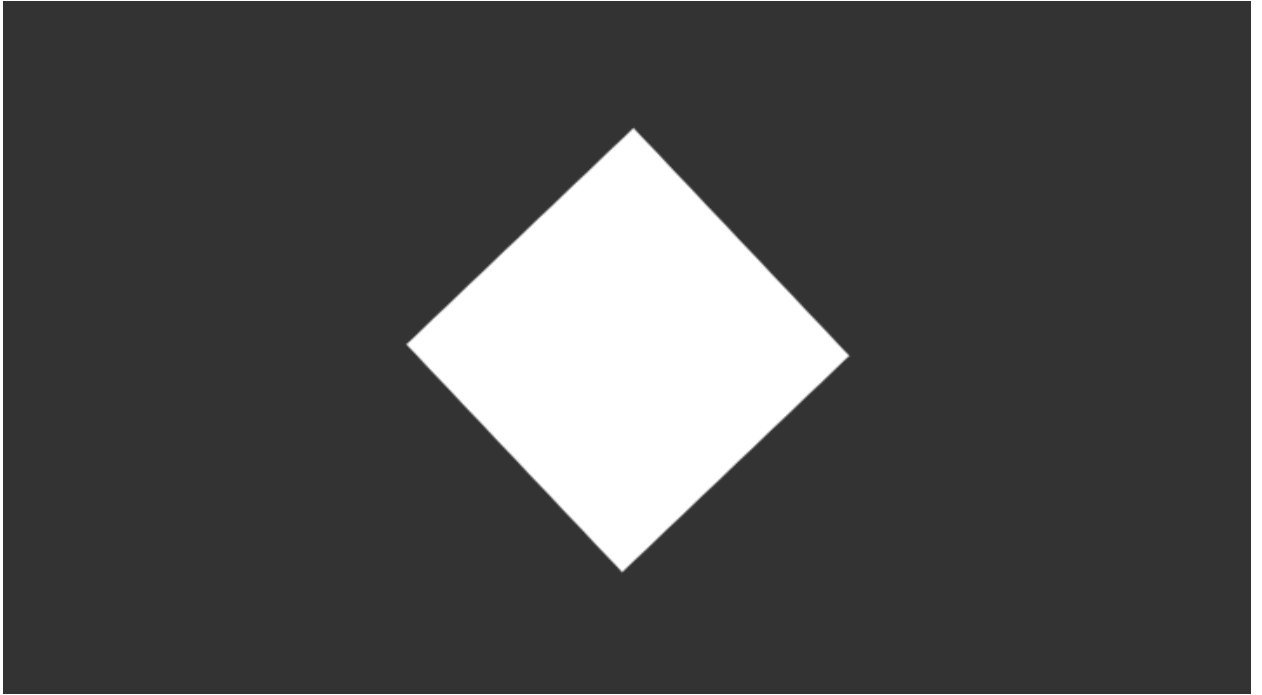


Figura 2.3 Rotire

Lucrare de laborator nr.2

Tema: Transformări grafice 2D

Scopul: Implimentarea transformărilor grafice asupra unei scene 2D utilizând setul de funcționalități a bibliotecii (JavaScript) p5.js.

Sarcina: Elaborați un program pentru efectuarea transformărilor grafice 2D utilizând **rotate()**, **scale()**, **translate()**.

Transformările v-or fi aplicate asupra scenei 2D create la lucrare de laborator nr.1.

Exemplu de program pentru fiecare transformare le puteți găsi mai sus în Tabelul 2.1, Tabelul 2.2 și Tabelul 2.3.

Tabelul 2.4 Variantele pentru pentru realizarea lucrării de laborator

Varianta	Unghiul de rotire	Coefficientul de scalare	Deplasarea X și Y
1	320	0.5	10,10
2	70	1.2	20,10
3	30	2.1	15,10
4	25	0.8	100,100
5	100	1.3	90,85
6	110	1.4	15,15
7	20	0.6	80,80
8	49	1.1	45,40
9	95	0.8	20,25
10	64	1.9	70,70
11	38	0.2	35,35
12	128	2.3	40,45
13	300	1.47	100,110
14	256	0.3	90,95
15	49	1.3	20,10
16	60	1.29	10,20
17	83	1.7	40,20
18	39	0.4	85,100
19	26	1.23	30,35
20	74	1.97	20,30
21	91	2.3	105,125
22	90	1.32	110,130
23	95	1.90	100,90
24	44	1.5	120,115
25	79	0.9	20,40

