

Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatica



Raport

Lucrarea de laborator nr. 2

Grafică pe Calculator

Varianta 3

A efectuat:

Student grupa TI-231 FR

Apareci Aurica

A verificat:

asistent universitar

Ursu Adriana

Chișinău

2024

Cuprins

1.	Cadrul teoretic	3
2.	Rezumat succint la temă.....	4
3.	Listingul programului.....	5
4.	Testarea aplicației.....	5
5.	Concluzii	6

1. Cadrul teoretic

Tema: Transformări grafice 2D

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

Sarcina (conform variantei): 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.

Varianta	Unghiul de rotire	Coeficientul de scalare	Deplasarea X și Y
3	30	2.1	15, 10



Laborator nr. 1 Varianta 3

2. Rezumat succint la temă

Funcția **translate()** permite mutarea obiectelor în orice locație din fereastră. Aceasta va primi ca parametri, doua numere, coordonatele noi ale obiectului pe axa Ox si axa Oy. Specifică o cantitate pentru a deplasa obiectele în fereastra de afișare. Parametrul x specifică translația stânga/dreapta, parametrul y specifică translația sus/jos. Transformările sunt cumulative și se aplică la tot ceea ce se întâmplă după, iar apelurile ulterioare ale funcției acumulează efectul. De exemplu, apelarea `translate(50, 0)` și apoi `translate(20, 0)` este aceeași cu `translate(70, 0)`. Dacă `translate()` este apelată în `draw()`, transformarea este resetată când bucla începe din nou.

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. 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 `scale()` și interschimbarea în funcție de ordinea lor. Mărește sau micșorează dimensiunea unei forme prin extinderea sau contractarea vârfurilor. Obiectele se scalează întotdeauna de la originea lor relativă la sistemul de coordonate. Valorile scalei sunt specificate ca procente zecimale. De exemplu, apelul funcției `scale(2.0)` mărește dimensiunea unei forme cu 200%.

Transformările se aplică la tot ceea ce se întâmplă după, iar apelurile ulterioare la funcție multiplică efectul. De exemplu, apelarea `scale(2.0)` și apoi `scale(1.5)` este aceeași cu `scale(3.0)`. Dacă `scale()` este apelată în `draw()`, transformarea este resetată când bucla începe din nou.

Funcția **rotate()** definește o transformare care rotește un element în jurul unui punct fix pe planul 2D, fără a-l deforma. 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. Rotește o formă cu cantitatea specificată de parametrul unghi. Această funcție ține cont de `angleMode`, astfel încât unghiurile pot fi introduse fie în `RADIANS`, fie în `GRADE`. Obiectele sunt întotdeauna rotite în jurul poziției lor relativ la origine, iar numerele pozitive rotesc obiectele în sensul acelor de ceasornic. Transformările se aplică la tot ceea ce se întâmplă după, iar apelurile ulterioare la funcție acumulează efectul. De exemplu, apelarea `rotate(HALF_PI)` și apoi `rotate(HALF_PI)` este aceeași cu `rotate(PI)`. Toate transformările sunt resetate când `draw()` începe din nou.

3. Listingul programului

```
function setup() {
  createCanvas(1000, 1000);
  background(233);

  let brown = color(125, 0, 0);
  let yellow = color(255, 190, 0);
  let black = color(0);

  translate(15, 10);
  rotate(radians(30));
  scale(2.1);

  drawFlower(200, 210, 100, brown);

  fill(brown);
  ellipse(200, 210, 220, 220);

  // Urechi mari
  fill(brown);
  stroke(black)
  ellipse(170, 170, 50, 50);
  ellipse(230, 170, 50, 50);

  // Urechi mici
  fill(200, 100, 100);
  ellipse(170, 170, 25, 25);
  ellipse(230, 170, 25, 25);

  // Fata
  fill(yellow);
  stroke(black);
  strokeWeight(3);
  triangle(125, 290, 200, 120, 275, 290);

  // Ochi
  fill(black);
  ellipse(185, 210, 25, 25);
  ellipse(215, 210, 25, 25);

  // Reflexii ochi
  fill(255);
  ellipse(180, 205, 8, 8);
  ellipse(210, 205, 8, 8);

  // Nas
  fill(black);
  triangle(195, 240, 205, 240, 200, 250);

  // Gura
  line(200, 250, 200, 260);
  line(200, 260, 190, 265);
  line(200, 260, 210, 265);

  // Mustăți
  line(190, 260, 170, 255);
  line(190, 265, 170, 265);
  line(190, 270, 170, 275);
```

```

    line(210, 260, 230, 255);
    line(210, 265, 230, 265);
    line(210, 270, 230, 275);
}

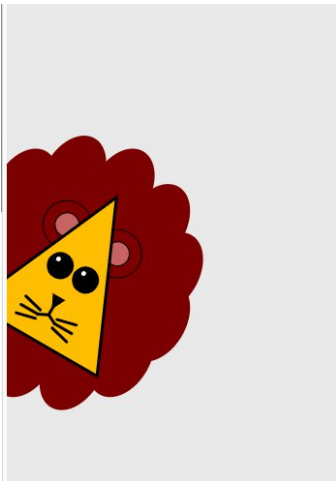
function drawFlower(x, y, radius, petalColor) {
    let numPetals = 12;
    let angleStep = TWO_PI / numPetals;

    fill(petalColor);
    noStroke();

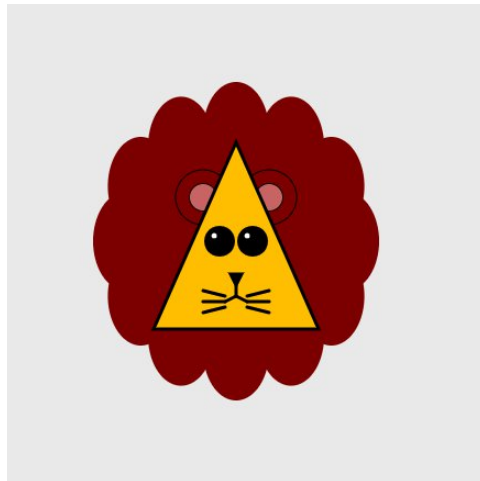
    for (let i = 0; i < numPetals; i++) {
        let angle = i * angleStep;
        let petalX = x + cos(angle) * radius;
        let petalY = y + sin(angle) * radius;
        ellipse(petalX, petalY, 60, 90);
    }
}

```

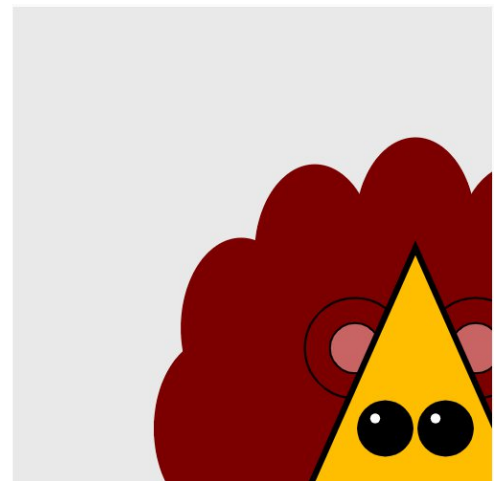
4. Testarea aplicației



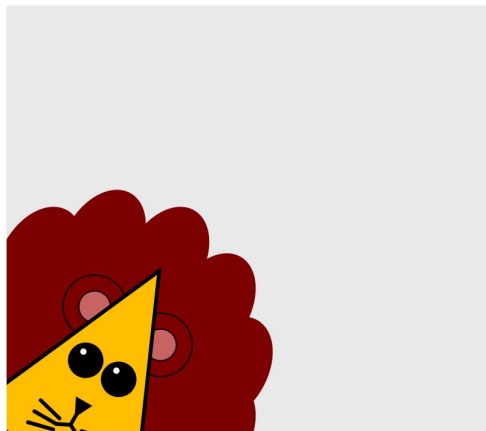
`rotate(radians(30));`



`translate(15, 10);`



`scale(2.1);`



5. Concluzii

JavaScript (JS) este un limbaj de scripting, folosit în principal pe Web. Este folosit pentru a îmbunătăți paginile HTML și este de obicei găsit încorporat în codul HTML. JavaScript este un limbaj interpretat. Astfel, nu trebuie să fie compilat. JavaScript redă paginile web într-un mod interactiv și dinamic. Acest lucru permite paginilor să reacționeze la evenimente, să prezinte efecte speciale, să accepte text variabil, să valideze date, etc.

P5.js este o bibliotecă JavaScript pentru un cod creativ. O colecție de metode predefinite, ce ne oferă instrumente care simplifică procesul de creare a imaginilor interactive prin intermediul codului, în browserul web.

La elaborarea am folosit drept sursa de informație suportul de curs, cat si documentația publică a bibliotecii P5.js. Din motivul ca dețin deja o anumita experiență de lucru in JS, nu am întâlnit dificultăți la elaborarea acestei lucrări. Desenul atribuit variantei primite a fost realizat cu ajutorul primitivelor grafice.