SPRAWOZDANIE

Zajęcia: Grafika Komputerowa
Prowadzący: mgr inż. Mikołaj Grygiel

Laboratorium: 4
Data: 12.03.2025
Temat: "Modelowanie hierarchiczne w grafice 2D "

Wariant: 14

Illia Bryka,
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.1a

Zadanie 1

1. Polecenie:

Opracować scenę hierarchiczną zgodnie z obrazem używając zamiast kół wielokąty obracające się (animacja!) według wariantu. Opracowanie powinno być w jednym z języków: Java lub JavaScript,
używając hierarchię funkcje (sposób subroutinowy)

2. Wprowadzane dane:

Do zadania wykorzystałem informację od prowadzącego ile wierzchołków ma mieć wykorzystany przeze mnie wielokąt. W moim wariancie jest to 14-kąt.

3. Wykorzystane komendy:

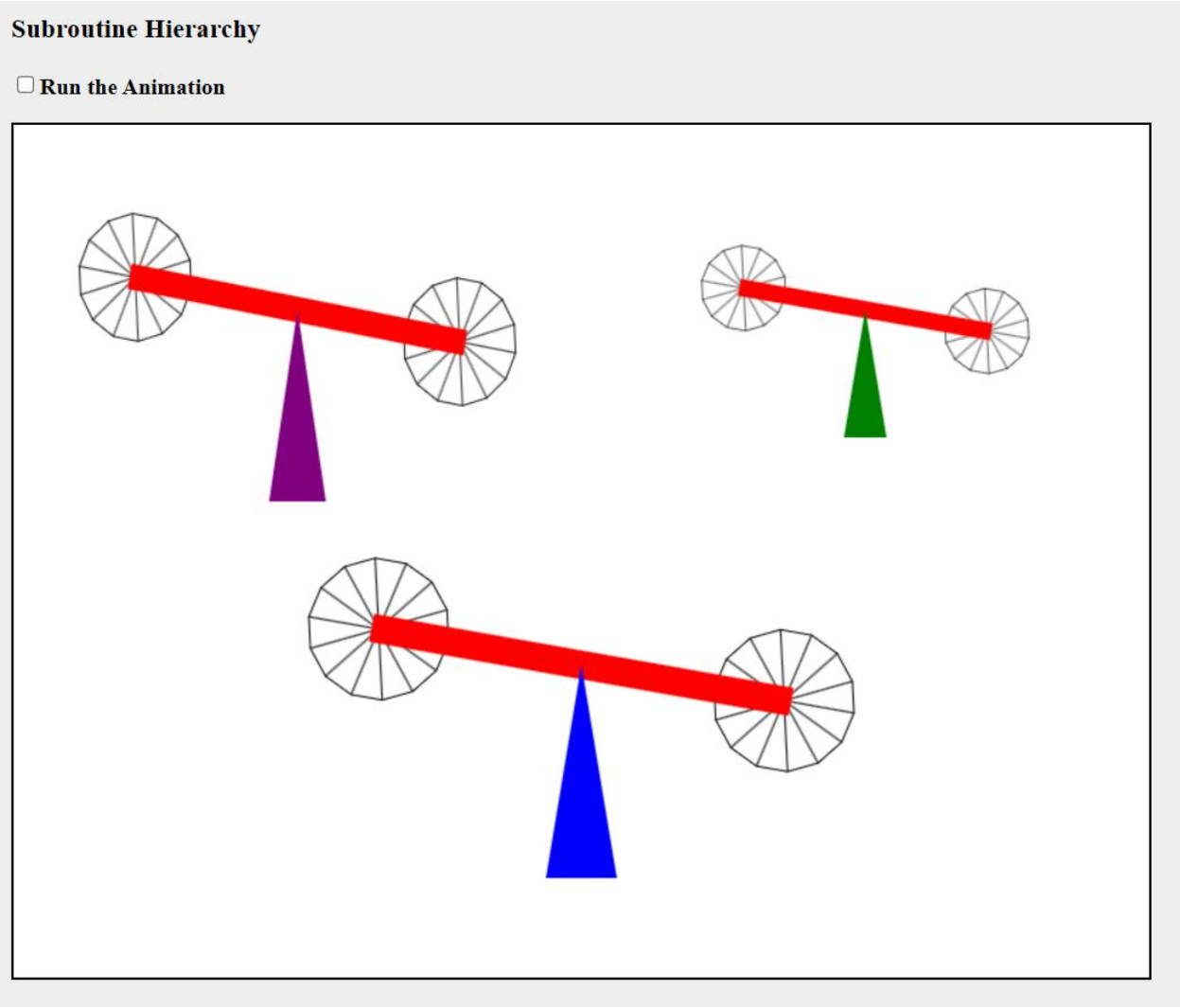
Żeby wykonać zadanie należało zmodyfikować kod w wyznaczonych miejscach.

```
function drawWorld() {

    // TODO: Draw the content of the scene.
    graphics.translate(0,-0.8);
    swing("blue");


    graphics.save();
    graphics.translate(-2,2.5);
    graphics.scale(0.8,0.9);
    swing("purple");
    graphics.restore();


    graphics.translate(2,2.5);
    graphics.scale(0.6,0.6);
    swing("green");

}
```

```
function rotatingShape(vertices, x, y, direction) {
    graphics.save();
    graphics.translate(x, y);
    graphics.rotate(direction * (frameNumber * 0.75) * Math.PI / 180 );

    // Rysowanie obwodu wielokąta (czternastokąt)
    graphics.beginPath();
    for (var i = 0; i <= vertices; i++) {
        var angle = i * 2 * Math.PI / vertices;
        var vx = 0.5 * Math.cos(angle);
        var vy = 0.5 * Math.sin(angle);
        if (i === 0) {
            graphics.moveTo(vx, vy);
        } else {
            graphics.lineTo(vx, vy);
        }
    }
    graphics.closePath();
    graphics.stroke();

    // Rysowanie linii od środka do każdego wierzchołka
    for (var i = 0; i < vertices; i++) {
        var angle = i * 2 * Math.PI / vertices;
        var vx = 0.5 * Math.cos(angle);
        var vy = 0.5 * Math.sin(angle);
        graphics.beginPath();
        graphics.moveTo(0, 0);
        graphics.lineTo(vx, vy);
        graphics.stroke();
    }
    graphics.restore();
}
```

Link do Repozytorium: https://github.com/bebrabimba/Grafika-Komputerowa/blob/main/Lab4/Lab4Ex1.html

4. Wynik działania:



Zadanie 2

1. Polecenie:

Opracować scenę hierarchiczną zgodnie z obrazem używając zamiast kół wielokąty obracające się (animacja!) według wariantu. Opracowanie powinno być w jednym z języków: Java lub JavaScript,

tworząc graf sceny (sposób obiektowy).

2. Wprowadzane dane:

Do zadania wykorzystałem informację od prowadzącego ile wierzchołków ma mieć wykorzystany przeze mnie wielokąt. W moim wariancie jest to 7-kąt, tak samo jak w zadaniu poprzednim.

3. Wykorzystane komendy:

Żeby wykonać zadanie należało zmodyfikować kod w wyznaczonych miejscach.

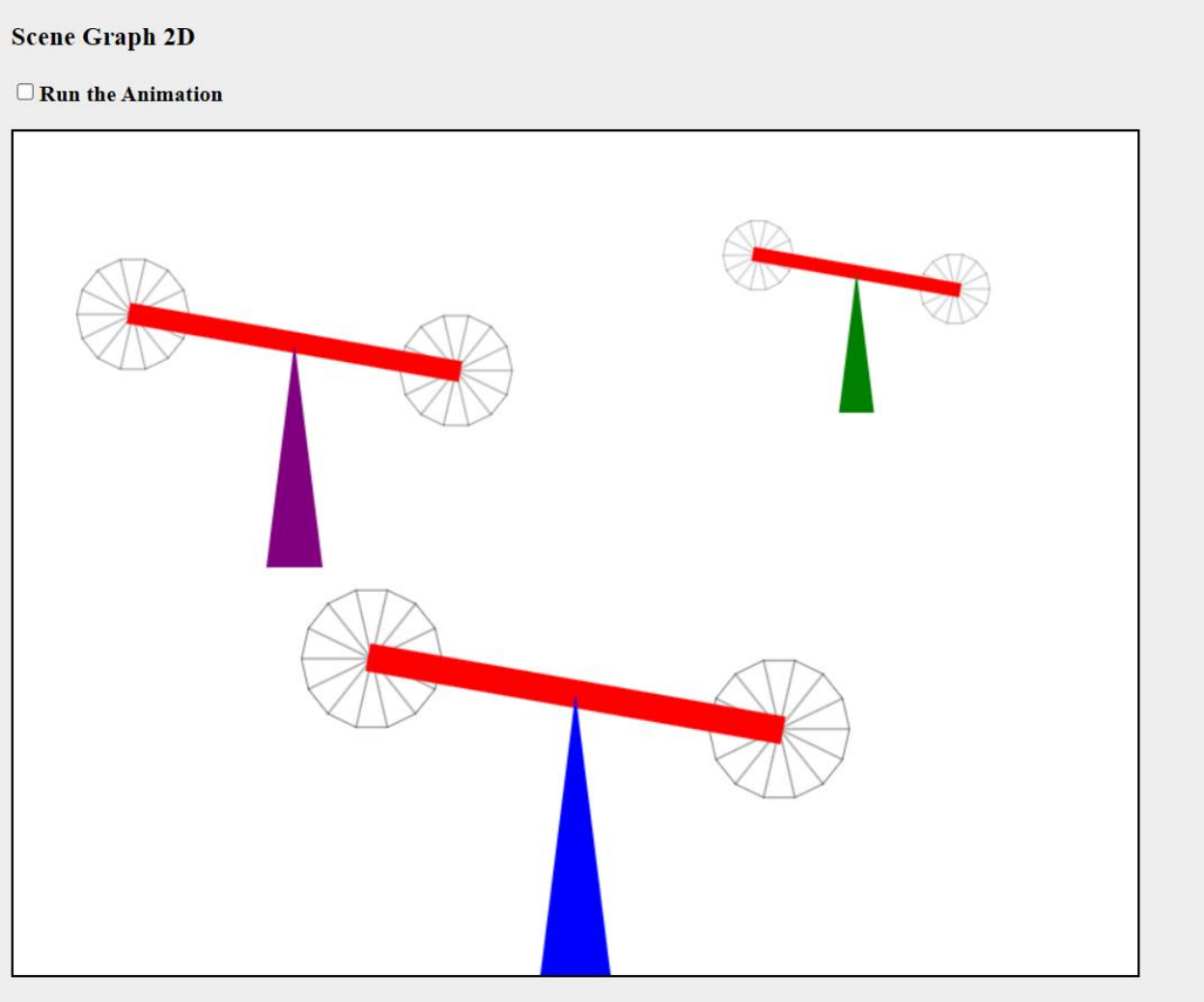
```
41      function createWorld() {
42
43          world = new CompoundObject();  // Root node for the scene graph.
44
45
46
47          violetLeft = new TransformedObject(shape);
48          violetLeft.setTranslation(-3.15, 1.7).setColor("black").setScale(0.4,0.4);
49          world.add(violetLeft);
50          violetRight = new TransformedObject(shape);
51          violetRight.setTranslation(-0.85, 1.3).setColor("black").setScale(0.4,0.4);
52          world.add(violetRight);
53          violetSwing = new TransformedObject(filledRect);
54          violetSwing.setTranslation(-2, 1.5).setColor("red").setScale(2.4,0.15).setRotation(-10);
55          world.add(violetSwing);
56          violetBaseTriangle = new TransformedObject(filledTriangle);
57          violetBaseTriangle.setTranslation(-2.0, -0.1).setColor("purple").setScale(0.4,1.6);
58          world.add(violetBaseTriangle);
59
60          blueLeft = new TransformedObject(shape);
61          blueLeft.setTranslation(-1.45, -0.75).setColor("black").setScale(0.5,0.5);
62          world.add(blueLeft);
63          blueRight = new TransformedObject(shape);
64          blueRight.setTranslation(1.45, -1.25).setColor("black").setScale(0.5,0.5);
65          world.add(blueRight);
66          blueSwing = new TransformedObject(filledRect);
67          blueSwing.setTranslation(0, -1).setColor("red").setScale(3,0.2).setRotation(-10);
68          world.add(blueSwing);
69          blueBaseTriangle = new TransformedObject(filledTriangle);
70          blueBaseTriangle.setTranslation(0, -3).setColor("blue").setScale(0.5,2.0);
71          world.add(blueBaseTriangle);
```

```
        greenLeft = new TransformedObject(shape);
        greenLeft.setTranslation(1.3, 2.12).setColor("black").setScale(0.25,0.25);
        world.add(greenLeft);
        greenRight = new TransformedObject(shape);
        greenRight.setTranslation(2.7, 1.88).setColor("black").setScale(0.25,0.25);
        world.add(greenRight);
        greenSwing = new TransformedObject(filledRect);
        greenSwing.setTranslation(2, 2).setColor("red").setScale(1.5,0.1).setRotation(-10);
        world.add(greenSwing);
        greenBaseTriangle = new TransformedObject(filledTriangle);
        greenBaseTriangle.setTranslation(2, 1).setColor("green").setScale(0.25,1);
        world.add(greenBaseTriangle);
    }

    var shape = new SceneGraphNode();
        shape.doDraw = function (g) {
            var vertices = 7;
            g.beginPath();
            for (i = 0; i <= vertices; i++) {
                graphics.lineTo((1 * Math.cos(i * 2 * Math.PI / vertices)),
                                (1 * Math.sin(i * 2 * Math.PI / vertices)));
                graphics.lineTo(0,0);
                graphics.lineTo((1 * Math.cos(i * 2 * Math.PI / vertices)),
                                (1 * Math.sin(i * 2 * Math.PI / vertices)));
            }
            g.closePath();
            g.stroke();
        }
```

```
    function updateFrame() {
        frameNumber++;

        blueLeft.setRotation(-frameNumber * 0.8);
        blueRight.setRotation(frameNumber * 0.8);
        violetLeft.setRotation(-frameNumber * 0.8);
        violetRight.setRotation(frameNumber * 0.8);
        greenLeft.setRotation(-frameNumber * 0.8);
        greenRight.setRotation(frameNumber * 0.8);
    }
```

Link do Repozytorium: https://github.com/bebrabimba/Grafika-Komputerowa/blob/main/Lab4/Lab4Ex2.html

4. Wynik działania:



Wnioski:

Na podstawie otrzymanych wyników można stwierdzić, że używając odpowiednich funkcji i odpowiedniej metodologii w języku Javascript, możemy manipulować grafiką na różne sposoby (Subrutinowy czy Obiektowy) i nie jest to, aż tak skomplikowane.

Dzięki użyciu odpowiednich funkcji oraz możliwości języka JavaScript możemy manipulować tworzoną przez nas grafiką na różne sposoby, min: Obiektowy czy Podprogramowy.