

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 3

Data 09.03.2022

Temat: „Modelowanie hierarchiczne w grafice 2D”

Wariant -

Wojciech Biegun
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr. 2b

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,

na dwa sposoby:

- (a) używając hierarchię funkcje (sposób subroutinowy)
- (b) tworząc graf sceny (sposób obiektowy).

2. Wprowadzane dane:

Brak podanych wariantów (zamiast wielokątów użyłem kół takich jak na obrazku)

3. Wykorzystane komendy:

a) Użyte funkcje:

scale()

rotate()

translate()

save()

restore()

line()

filledRect()

filledTriangle()

drawWheel()

drawTower()

b) Kod źródłowy:

Zadanie 1:

```
// TODO: Draw the content of the scene.  
  
drawTower("blue");  
graphics.save();  
graphics.scale(0.8,0.8);  
graphics.translate(-3,2.7);  
drawTower("purple");  
graphics.restore();
```

```
graphics.save();
graphics.scale(0.6,0.6);
graphics.translate(3.5,3.5);
drawTower("green");
graphics.restore();
```

Dodatkowe funkcje:

```
function drawTower(triangleColor){
    graphics.save();
    graphics.scale(1.3,1.3);
    graphics.translate(-0.95,-0.35);
    drawWheel();
    graphics.restore();

    graphics.save();
    graphics.scale(1.3,1.3);
    graphics.translate(1.1,-1.1);
    drawWheel();
    graphics.restore();
    graphics.save();
    graphics.translate(-0.7,-0.3);
    graphics.rotate(-Math.PI/9);
    graphics.fillStyle = "red";
    filledRect();
    graphics.restore();

    graphics.save();
    graphics.translate(0,-2.9);
    graphics.fillStyle = triangleColor;
    filledTriangle(graphics);
    graphics.restore();

}
function drawWheel(){
    graphics.rotate((frameNumber*0.75) * Math.PI/180);
    circle();
    line();
    graphics.rotate(Math.PI/6);
    line();
    graphics.rotate(Math.PI/6);
    line();
    graphics.rotate(Math.PI/6);
    line();
    graphics.rotate(Math.PI/6);
    line();
    graphics.rotate(Math.PI/6);
    line();
}
}
```

Zadanie 2:

```
// TODO: Create objects and add them to the scene graph.
//First tower
w1 = new TransformedObject(wheel);
w1.setScale(1.3,1.3);
w1.setTranslation(-1.15,-0.5);
world.add(w1);
w2 = new TransformedObject(wheel);
w2.setScale(1.3,1.3);
w2.setTranslation(1.5,-1.4);
world.add(w2);
var p1 = new TransformedObject(filledRect);
p1.setScale(3,0.25);
p1.setColor("red");
p1.setRotation(-20);
p1.setTranslation(0.2,-1);
world.add(p1);
var t1 = new TransformedObject(filledTriangle);
t1.setScale(0.7,2);
t1.setColor("blue");
t1.setTranslation(0,-2.9);
world.add(t1);
//Second tower
w3 = new TransformedObject(wheel);
w3.setScale(1,1);
w3.setTranslation(-3.35,1.8);
world.add(w3);
w4 = new TransformedObject(wheel);
w4.setScale(1,1);
w4.setTranslation(-1.1,1);
world.add(w4);
var p2 = new TransformedObject(filledRect);
p2.setScale(2.6,0.2);
p2.setColor("red");
p2.setRotation(-20);
p2.setTranslation(-2.2,1.4);
world.add(p2);
var t2 = new TransformedObject(filledTriangle);
t2.setScale(0.6,1.7);
t2.setColor("purple");
t2.setTranslation(-2.4,-0.2);
world.add(t2);
//Third tower
w5 = new TransformedObject(wheel);
w5.setScale(0.8,0.8);
w5.setTranslation(1.3,1.8); //2.45,2.3
world.add(w5);
w6 = new TransformedObject(wheel);
w6.setScale(0.8,0.8);
w6.setTranslation(3,1.2);
```

```
world.add(w6);
var p3 = new TransformedObject(filledRect);
p3.setScale(2,0.15);
p3.setColor("red");
p3.setRotation(-20);
p3.setTranslation(2.15,1.5);
world.add(p3);
var t3 = new TransformedObject(filledTriangle);
t3.setScale(0.5,1.3);
t3.setColor("green");
t3.setTranslation(2.15,0.2);
world.add(t3);
```

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

    // TODO: Update state in preparation for drawing the next frame.
    w1.setRotation(frameNumber*0.75);
    w2.setRotation(frameNumber*0.75);
    w3.setRotation(frameNumber*0.75);
    w4.setRotation(frameNumber*0.75);
    w5.setRotation(frameNumber*0.75);
    w6.setRotation(frameNumber*0.75);
}
```

Link do zdalnego repozytorium:

<https://github.com/WojciechBiegun/GK>

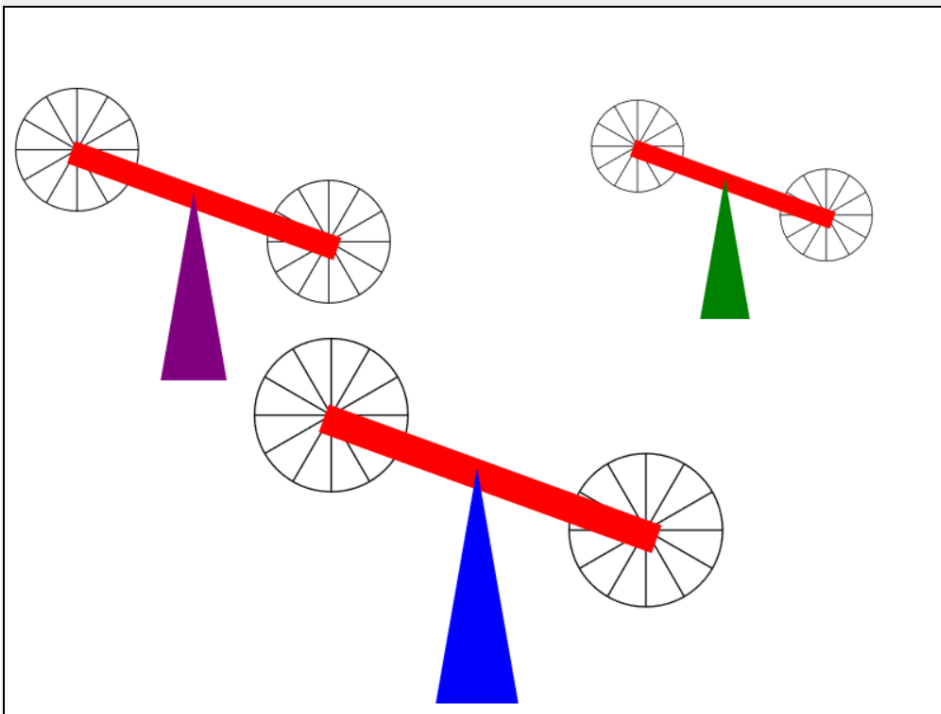
Zadanie zostało umieszczone w folderze LAB3.

4. Wynik działania:

Zadanie 1:

Subroutine Hierarchy

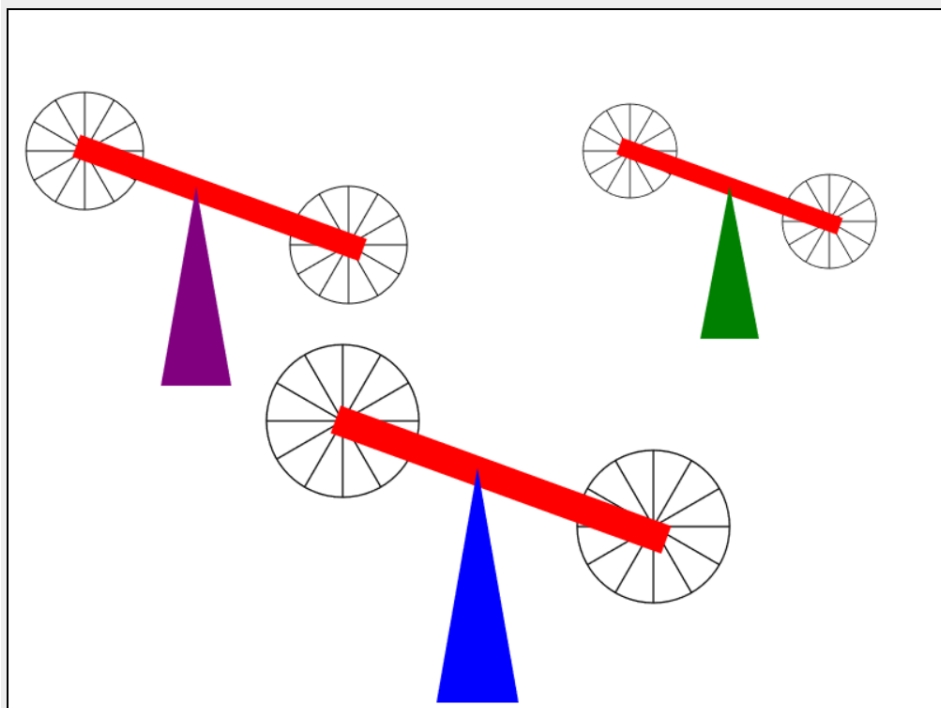
☐ Run the Animation



Zadanie 2:

Scene Graph 2D

☐ Run the Animation



Wnioski:

Powyższe wyniki pokazują że do tworzenia grafiki w JS możemy wykorzystywać różne sposoby. Podejście obiektowe pozwala pisać klasy na podstawie których możemy tworzyć obiekty graficzne, a podejście subroutinowe pozwala tworzyć skomplikowane obiekty, które możemy skalować, przesuwać i obracać.