

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: 2

Data: 2.03.2020r

Temat: Grafika 2D z użyciem HTML Canvas

Wariant: figura 13, 17-kąt

Imię Nazwisko: Marek Żyła

Informatyka I stopień,

stacjonarne,

4 semestr, Gr.1b

1. Polecenie:

Plik `Lab2Ex1.html` proponuje rozszerzenia do standardowych funkcji rysowania HTML Canvas.

Narysować obraz zgodnie z wariantem zadania (używając zarówno standardowe jak i niestandardowe funkcje rysowania).

2. W Plik `Lab2Ex2.html` program domyślnie rysuje szereg kwadratów.

Stworzyć narzędzia pozwalające na wykonywanie czynności

- "czyszczenie" canvasu - Clear button:
- dodanie jednego nowego koloru do elementu `<select>`. Implementować nowy kolor przez funkcję `doMouseMove`.
- opracowanie nowego narzędzia - rysowania szeregu wielokątów (zgodnie z wariantem zadania). Opcja ma być dostępna przez nowy element `<select>`

2. Wprowadzone Dane:

Zad.1.

```
//tło
graphics.fillStyle="orange";
graphics.fillRect(200,200,200,200);
//obramowanie
graphics.fillStyle="black";
graphics.lineWidth=10;
graphics.strokeRect(200,200,200,200);
//oko lewe
graphics.fillStyle="white";
graphics.fillCircle(260,260,20);
//----- (poziomo, pionowo, promień)
graphics.fillStyle="black";
graphics.fillCircle(260,260,15);

graphics.fillStyle="white";
graphics.fillCircle(255,255,5);
//oko prawe
graphics.fillStyle="white";
graphics.fillCircle(360,260,20);

graphics.fillStyle="black";
graphics.fillCircle(360,260,15);

graphics.fillStyle="white";
graphics.fillCircle(355,255,5);
//usta

graphics.beginPath()
//graphics.fillStyle="black";
graphics.lineWidth=2;
graphics.moveTo(260,320);
graphics.bezierCurveTo(260, 360, 330, 360, 330,320);
graphics.bezierCurveTo(330,340, 260, 340, 260,320);
//graphics.bezierCurveTo(360, 320, 350, 350, 260,360);
graphics.fillStyle="black";
graphics.fill();
graphics.stroke();
//kaciki_ust
graphics.lineWidth=5;
graphics.moveTo(250,330);
graphics.bezierCurveTo(260, 320, 260, 320, 260,310);
graphics.stroke();
graphics.moveTo(325,310);
graphics.bezierCurveTo(335, 330, 335, 330, 340,330);
graphics.stroke();
//zeby
graphics.lineWidth=1;
graphics.fillStyle="white";
graphics.fillRect(285,333,10,10);
graphics.fillRect(296,333,10,10);
```

Zad.2. zmodyfikowałem pętlę if

```
if ( Math.abs(x-prevX) + Math.abs(y-prevY) < 3 ) {  
    return; // don't draw squares too close together  
}  
  
if (colorChoice == 0) {  
    graphics.fillStyle = randomColorString();  
}  
else if (colorChoice == 1) {  
    graphics.fillStyle = "red";  
}  
else if (colorChoice == 2) {  
    graphics.fillStyle = "green";  
}  
else if (colorChoice == 3) {  
    graphics.fillStyle = "blue";  
}  
else if (colorChoice == 4) {  
    graphics.fillStyle = "pink";  
}
```

Oraz rysowanie kształtu

```
if (colorChoice < 5){  
    graphics.fillRect(x-20,y-20,40,40);  
    graphics.strokeRect(x-20,y-20,40,40);  
}  
else {  
    graphics.fillStyle = "orange";  
  
    graphics.fillPoly(  
        x-20*Math.cos(2*2*Math.PI /17),y-20*Math.sin(2*2*Math.PI /17),  
        x-20*Math.cos(2*3*Math.PI /17),y-20*Math.sin(2*3*Math.PI /17),  
        x-20*Math.cos(2*4*Math.PI /17),y-20*Math.sin(2*4*Math.PI /17),  
        x-20*Math.cos(2*5*Math.PI /17),y-20*Math.sin(2*5*Math.PI /17),  
        x-20*Math.cos(2*6*Math.PI /17),y-20*Math.sin(2*6*Math.PI /17),  
        x-20*Math.cos(2*7*Math.PI /17),y-20*Math.sin(2*7*Math.PI /17),  
        x-20*Math.cos(2*8*Math.PI /17),y-20*Math.sin(2*8*Math.PI /17),  
        x-20*Math.cos(2*9*Math.PI /17),y-20*Math.sin(2*9*Math.PI /17),  
        x-20*Math.cos(2*10*Math.PI /17),y-20*Math.sin(2*10*Math.PI /17),  
        x-20*Math.cos(2*11*Math.PI /17),y-20*Math.sin(2*11*Math.PI /17),  
        x-20*Math.cos(2*12*Math.PI /17),y-20*Math.sin(2*12*Math.PI /17),  
        x-20*Math.cos(2*13*Math.PI /17),y-20*Math.sin(2*13*Math.PI /17),  
        x-20*Math.cos(2*14*Math.PI /17),y-20*Math.sin(2*14*Math.PI /17),  
        x-20*Math.cos(2*15*Math.PI /17),y-20*Math.sin(2*15*Math.PI /17),  
        x-20*Math.cos(2*16*Math.PI /17),y-20*Math.sin(2*16*Math.PI /17),  
        x-20*Math.cos(2*17*Math.PI /17),y-20*Math.sin(2*17*Math.PI /17),  
        x-20*Math.cos(2*18*Math.PI /17),y-20*Math.sin(2*18*Math.PI /17)  
    );  
    graphics.strokePoly(  
        x-20*Math.cos(2*2*Math.PI /17),y-20*Math.sin(2*2*Math.PI /17),  
        x-20*Math.cos(2*3*Math.PI /17),y-20*Math.sin(2*3*Math.PI /17),  
        x-20*Math.cos(2*4*Math.PI /17),y-20*Math.sin(2*4*Math.PI /17),  
        x-20*Math.cos(2*5*Math.PI /17),y-20*Math.sin(2*5*Math.PI /17),  
        x-20*Math.cos(2*6*Math.PI /17),y-20*Math.sin(2*6*Math.PI /17),  
        x-20*Math.cos(2*7*Math.PI /17),y-20*Math.sin(2*7*Math.PI /17),  
        x-20*Math.cos(2*8*Math.PI /17),y-20*Math.sin(2*8*Math.PI /17),  
        x-20*Math.cos(2*9*Math.PI /17),y-20*Math.sin(2*9*Math.PI /17),  
        x-20*Math.cos(2*10*Math.PI /17),y-20*Math.sin(2*10*Math.PI /17),  
        x-20*Math.cos(2*11*Math.PI /17),y-20*Math.sin(2*11*Math.PI /17),  
        x-20*Math.cos(2*12*Math.PI /17),y-20*Math.sin(2*12*Math.PI /17),  
        x-20*Math.cos(2*13*Math.PI /17),y-20*Math.sin(2*13*Math.PI /17),  
        x-20*Math.cos(2*14*Math.PI /17),y-20*Math.sin(2*14*Math.PI /17),  
        x-20*Math.cos(2*15*Math.PI /17),y-20*Math.sin(2*15*Math.PI /17),  
        x-20*Math.cos(2*16*Math.PI /17),y-20*Math.sin(2*16*Math.PI /17),  
        x-20*Math.cos(2*17*Math.PI /17),y-20*Math.sin(2*17*Math.PI /17),  
        x-20*Math.cos(2*18*Math.PI /17),y-20*Math.sin(2*18*Math.PI /17)  
    );  
}  
}
```

```

<p><b>Color:</b>
  <select id="colorChoice">
    <option value="0">Random</option>
    <option value="1">Red</option>
    <option value="2">Green</option>
    <option value="3">Blue</option>
    <option value="4">Pink</option>
    <option value="5">siedemnastokat</option>
  </select>
  <button onclick="wyczysc()">Clear</button>
</p>

```

```

function wyczysc(){
  graphics.fillStyle = "white";
  graphics.fillRect(0,0,canvas.width,canvas.height);
}

```

Dodałem guzik clear, oraz 2 dodatkowe opcje rysowania, kolor różowy oraz figurę 17-kątną

3. Wykorzystane komendy

Zad.1.

```

fillStyle="orange";
fillRect(200,200,200,200);
fillCircle(260,260,20);
beginPath();
lineWidth=2;
moveTo(260,320);
bezierCurveTo(260, 360, 330, 360, 330,320);
fill();
stroke();

```

zad.2.

```

<button onclick="function()">text</button>

. fillStyle = "pink";

.fillPoly();

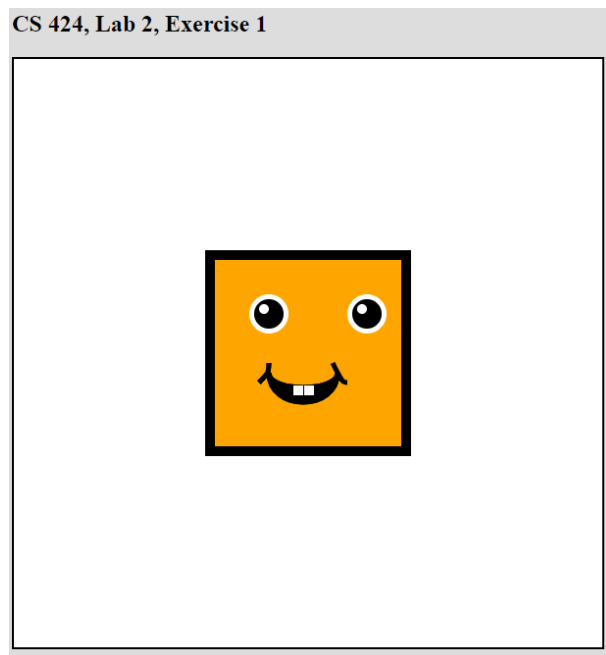
.strokePoly();

```

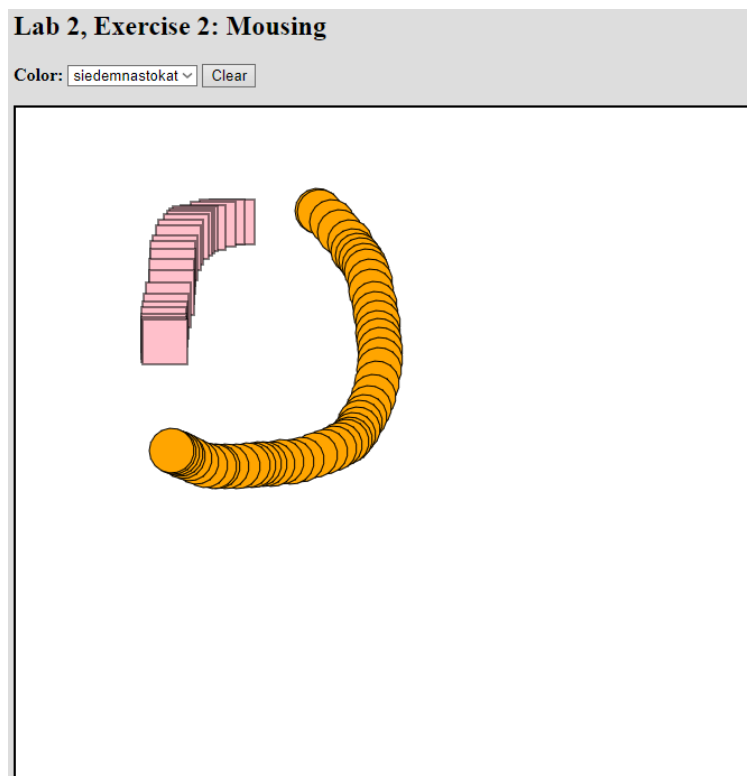
https://github.com/Samo276/GK_lab_2

4. Wynik Działania:

Zad.1.



Zad.2.



Oraz guzik czyszczenia

5. Wnioski:

Przy rysowaniu kwadratu zauważyłem że polecenia rysowania są wykonywane po kolei a atrybuty koloru i grubości linii są zapamiętywane aż do momentu ich zmiany.

Zauważyłem że w zadaniu 2 zamiast tworzyć własną funkcję czyszczącą można podpiąć pod guzik „clear” funkcję `init()` i wtedy powinniśmy uzyskać ten sam efekt czyszczenia canvasu.