**SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: mgr Mikołaj Grygiel

**Laboratorium 2**

09.03.2024

**Temat:** Grafika 2D z u˙zyciem HTML Canvas

**Wariant**: 1

Radosław Skrzypczyński

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr. 1B

1. **Polecenie:**
2. Plik Lab2Ex1.html proponuje rozszeżenia do standardowych funkcji rysowania HTML Canvas. Narysować obraz zgodnie z wariantem zadania (używając zarówno standardowe jak i niestandardowe funkcje rysowania).
3. 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. Wprowadzane dane:**

a) brak wprowadzanych danych

b)

* kliknięcie przycisku Clear w celu wyczyszczenia obszaru roboczego
* wybór opcji rysowania przez kliknięcie i wybór z listy
* kliknięcie lewego przycisk myszy w celu rysowania (wprowadzenie współrzędnych poprzez położenie kursora myszy)

**3. Wykorzystane komendy:**

a)

function draw() {

        graphics.clearRect(0,0,600,600);

        drawFaceShape(100,5);

        drawEye(80, 40, 25, 15, 4, 10, "left");

        drawEye(80, 40, 25, 15, 4, 10, "right");

        drawSmile(100, 30, 5, 30);

    }

    function drawFaceShape(size, strokeSize){

        graphics.lineWidth = strokeSize;

        graphics.strokeCircle(canvas.width/2,canvas.height/2,size,size);

        graphics.fillStyle = "red";

        graphics.fillCircle(canvas.width/2,canvas.height/2,size,size);

        graphics.lineWidth = 1;

    }

    function drawEye(spacing, offsetY, eyeSize, pupilSize, sparkSize, sparkAngle, leftOrRight){

        let x = leftOrRight == "left" ? canvas.width/2 - spacing/2 : canvas.width/2 + spacing/2;

        let y = canvas.height/2 - offsetY;

        graphics.fillStyle = "white";

        graphics.fillCircle(x,y,eyeSize,eyeSize);

        graphics.fillStyle = "black";

        graphics.fillCircle(x,y,pupilSize,pupilSize);

        graphics.fillStyle = "white";

        graphics.fillCircle(x-5,y-8,sparkSize,sparkSize);

    }

    function drawSmile(width, height, offsetX, offsetY) {

        let startX = canvas.width/2 - width/2 + offsetX;

        let endX = startX + width + offsetX;

        let startY = canvas.height/2 + offsetY;

        graphics.beginPath()

        graphics.moveTo(startX, startY);

        graphics.bezierCurveTo(startX, startY, startX + width/2, startY + height, endX, startY)

        graphics.bezierCurveTo(endX, startY, startX + width/2, startY + height \* 3, startX, startY)

        graphics.fillStyle = "black";

        graphics.fill();

        graphics.fillStyle = "white";

        graphics.fillRect(startX + width/2 - 10, startY + height/2.1, 10,10);

        graphics.fillRect(startX + width/2 + 1, startY + height/2.1, 10,10);

        graphics.fillStyle = "red";

        graphics.fillCircle(startX, startY, 10,10);

        graphics.fillCircle(endX, startY, 10,10);

        graphics.lineWidth = 3;

        graphics.beginPath();

        graphics.arc(startX, startY, 10, 0, Math.PI/2);

        graphics.stroke();

        graphics.beginPath();

        graphics.arc(endX, startY, 10, Math.PI/2, Math.PI);

        graphics.stroke();

    }

b) kod źródłowy

function doMouseMove(evt) {

                // This function is called when the user moves the mouse during a drag.

            if (!dragging) {

                return;  // (shouldn't be possible)

            }

            var x,y;  // mouse position in canvas coordinates

            var r = canvas.getBoundingClientRect();

            x = Math.round(evt.clientX - r.left);

            y = Math.round(evt.clientY - r.top);

            /\*------------------------------------------------------------\*/

            /\* TODO: Add support for more drawing tools. \*/

            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 = "yellow";

            }

            else if (colorChoice == 5) {

                graphics.fillStyle = "violet";

            }

            let shape = (document).getElementById("shapeChoice").value;

            let size = 40;

            let numberOfSides = Number(shape) + 4;

            console.log(numberOfSides);

            let step = 2 \* Math.PI / numberOfSides;

            let shiftAngle = -45;

            if (numberOfSides == 5){

                shiftAngle = -18;

            }else if(numberOfSides == 6){

                shiftAngle = -30;

            }

            let shift = (Math.PI / 180.0) \* shiftAngle;

            graphics.beginPath();

            for (var i = 0; i <= numberOfSides;i++) {

                var curStep = i \* step + shift;

                graphics.lineTo (x + size \* Math.cos(curStep), y + size \* Math.sin(curStep));

            }

            graphics.stroke();

            graphics.fill();

            /\*------------------------------------------------------------\*/

            prevX = x;  // update prevX,prevY to prepare for next call to doMouseMove

            prevY = y;

        }

…

<p>JavaScript is required to use this page.</p>

</noscript>

<div style="display: flex; align-items: center; margin-bottom: 5px">

    <button type="button" onclick="clearCanvas()" style="margin-right: 10px; height:30px">Clear</button>

    <div style="margin-right:10px"><b>Color:</b>

        <select id="colorChoice">

            <option value="0" selected="selected">Random</option>

            <option value="1">Red</option>

            <option value="2">Green</option>

            <option value="3">Blue</option>

            <option value="4">Yellow</option>

            <option value="5">Violet</option>

        </select>

    </div>

    <div><b>Shape:</b>

        <select id="shapeChoice">

            <option value="0" selected="selected">Rectangle</option>

            <option value="1">Pentagon</option>

            <option value="2">Hexagon</option>

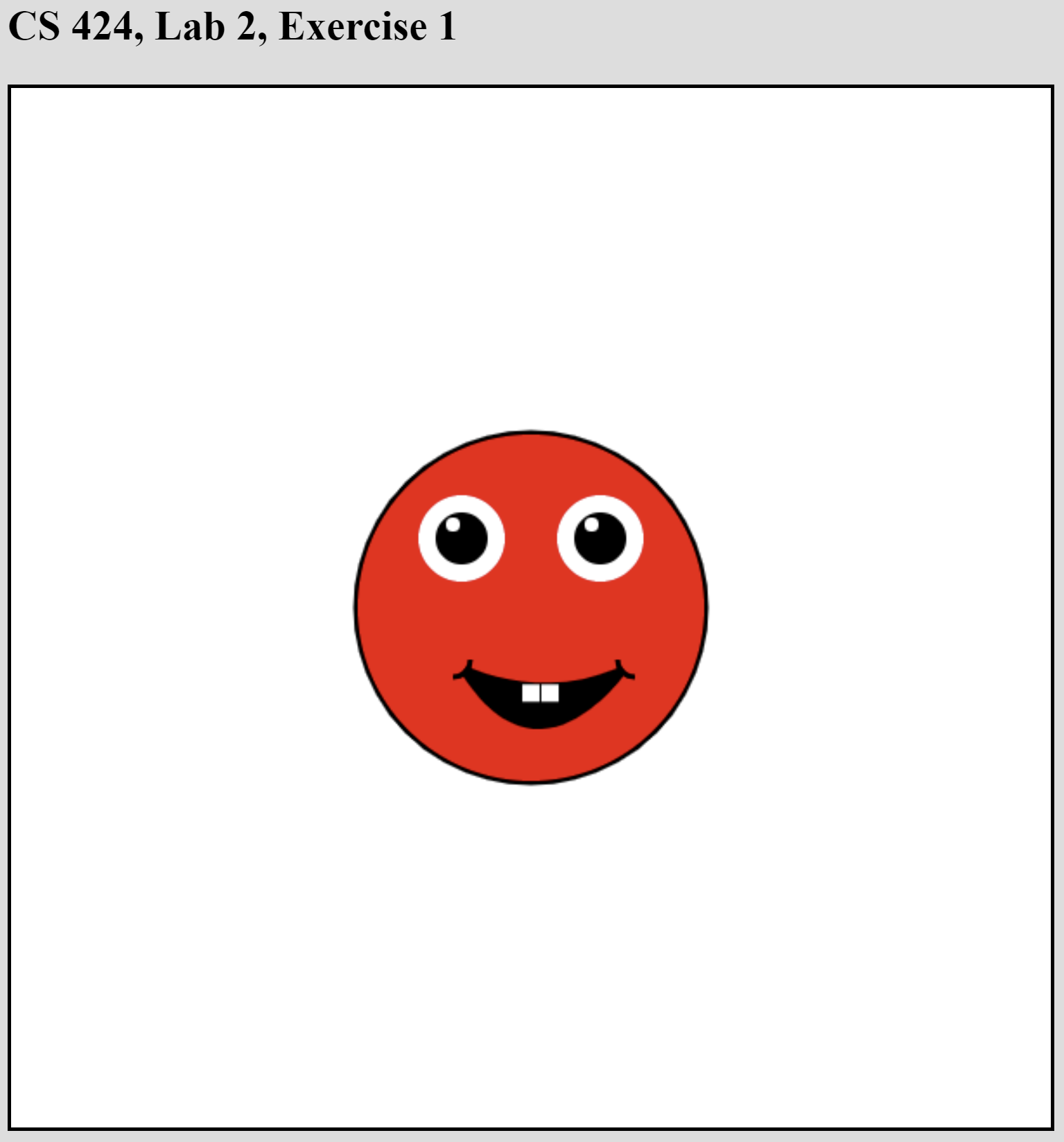
        </select>

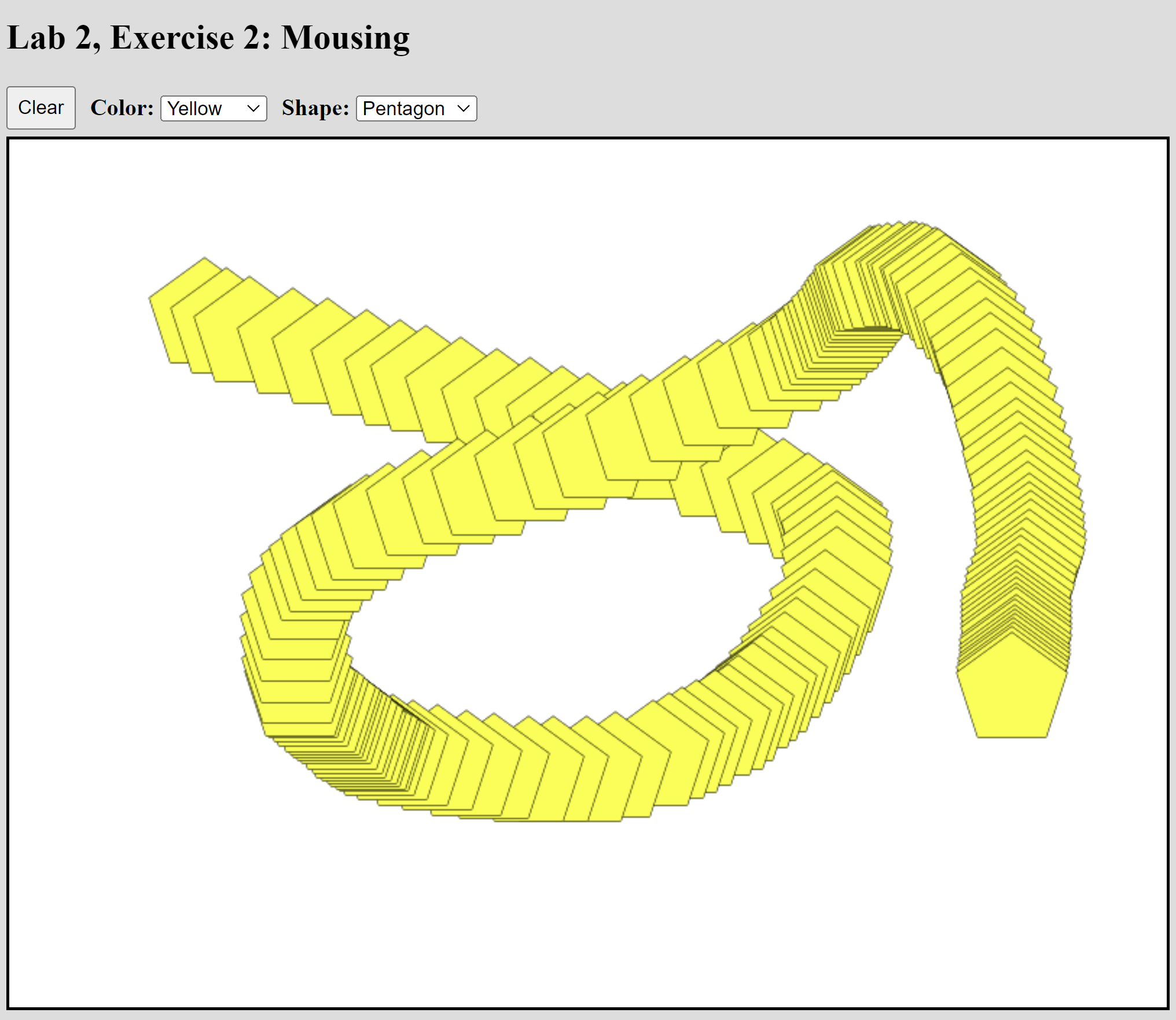
    </div>

</div>

Link do zdalnego repozytorium: https://github.com/Slayzerus/UBB\_GrafikaKomputerowa/tree/main/Lab%202

**4. Wynik działania:**



**5. Wnioski:**

Na podstawie otrzymanego wyniku można stwierdzić, że w języku JavaScript na powierzchniach elementów języka HTML zwanych Canvas można tworzyć zarówno proste jak i bardziej złożone grafiki 2D i 2.5D a nawet grafiki 3D.

Grafiki tworzone w ten sposób mogą być zarówno statyczne jak i dynamiczne. Mogą się zmieniać i przesuwać w czasie. Do dyspozycji użytkownika jest bardzo wiele funkcji umożliwiających rysowanie konturów i wypełnianie zarówno bardzo prostych (tj. np. linia, koło, prostokąt) jak i złożonych kształtów (tj. łuk, krzywa).

Z wykorzystaniem elementu Canvas można tworzyć np. interfejsy użytkownika, prezentacje, programy graficzne albo gry przeglądarkowe.