

# **SPRAWOZDANIE**

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Labolatorium: 1**

28.02.2025

**Temat:** Grafika 2D z użyciem HTML Canvas

**Wariant: 4**

Jakub Początek,  
Informatyka I stopień,  
Stacjonarne,  
4 semestr,  
Gr. 1a

## Spis treści

Zadanie 1 .....	3
1. Polecenie .....	3
2. Wprowadzane dane .....	3
3. Wykorzystane komendy .....	4
4. Wynik działania .....	5
5. Wnioski .....	5
Zadanie 2 .....	6
1. Polecenie .....	6
2. Wprowadzane dane .....	6
3. Wykorzystane komendy .....	6
4. Wynik działania .....	8
5. Wnioski .....	8

# Zadanie 1

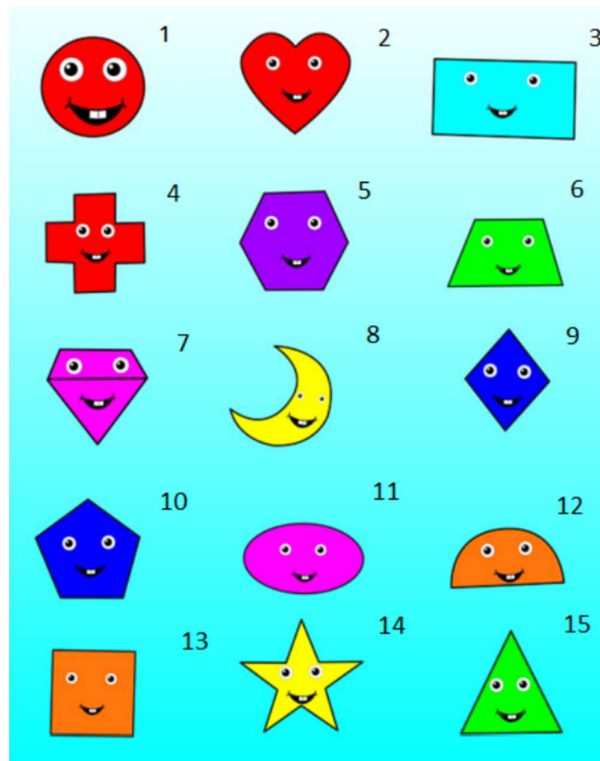
## 1. Polecenie

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. Wprowadzane dane

Zadanie polegało na utworzeniu wybranego kształtu z różnego rodzaju figur geometrycznych. Z puli wyboru kształtów do narysowania, wybrałem krzyż z buźką (4).



W celu utworzenia krzyża najlepiej połączyć dwa prostokąty. Ja użyłem trzech prostokątów, ułożonych poziomo element za elementem, w celu łatwiejszego późniejszego użycia obramowania. Oczy ze źrenicami i gradientowym błyskiem utworzyłem korzystając ze standardowych poleceń wytwarzania koła. Skręcona buźka powstała przy użyciu dwóch elips oraz wykluczenia, zęby powstały w taki sposób, że utworzony został poziomy prostokąt i została nałożona czarna granica, a więc kolejny prostokąt, tylko z takim kolorem, jaki ma tło ust. Końcówki ust utworzone symetrycznie po obu stronach poprzez krzywe linie.

### 3. Wykorzystane komendy

Komendy wytwarzające całość kształt były wprowadzane w funkcji *draw()* wytwarzania kształtu w obszarze Canvas.

Krzyż:

```
function draw() {
    /*
     * Making a cross of rectangles
     */
    graphics.fillStyle = "red";

    /*
     * First rectangle
     */
    graphics.fillRect(150, 100, 100, 100);

    graphics.strokeStyle = "black";
    graphics.lineWidth = 2.5;

    graphics.beginPath();
    graphics.moveTo(150, 100);
    graphics.lineTo(250, 100);

    graphics.lineTo(250, 200);

    graphics.moveTo(150, 100);
    graphics.lineTo(150, 200);

    graphics.stroke();

    /*
     * Second rectangle
     */
    graphics.fillRect(50, 200, 300, 100);

    graphics.beginPath();

    // Top border (with a gap in the center)
    graphics.moveTo(50, 200);
    graphics.lineTo(150, 200);
    graphics.moveTo(250, 200);
    graphics.lineTo(350, 200);

    // Bottom border (with a gap in the center)
    graphics.moveTo(50, 300);
    graphics.lineTo(150, 300);
    graphics.moveTo(250, 300);
    graphics.lineTo(350, 300);

    // Left border (full)
    graphics.moveTo(50, 200);
    graphics.lineTo(50, 300);

    // Right border (full)
    graphics.moveTo(350, 200);
    graphics.lineTo(350, 300);

    graphics.stroke();

    /*
     * Third rectangle
     */
    graphics.fillRect(150, 300, 100, 100);

    graphics.beginPath();

    // Left border
    graphics.moveTo(150, 300);
    graphics.lineTo(150, 400);

    // Right border
    graphics.moveTo(250, 300);
    graphics.lineTo(250, 400);

    // Bottom border
    graphics.moveTo(150, 400);
    graphics.lineTo(250, 400);

    graphics.stroke();
}
```

Buzia:

```
/*
 * Face
 */

// Open Mouth (black semi-circle for bottom)
graphics.fillStyle = "black";
graphics.beginPath();
graphics.ellipse(200, 230, 50, 60, 0, 0, Math.PI);
graphics.fill();

graphics.beginPath();
graphics.arc(200, 230, 50, 0.15 * Math.PI, 0.85 * Math.PI,
false); // Bottom curve
graphics.fill();

// Teeth (white rectangle inside the mouth)
graphics.fillStyle = "white";
graphics.fillRect(190, 270, 20, 8);

// Tooth Separation Line (middle black line)
graphics.strokeStyle = "black";
graphics.lineWidth = 2.5;
graphics.beginPath();
graphics.moveTo(200, 230);
graphics.lineTo(200, 278);
graphics.stroke();

graphics.fillStyle = "red";
graphics.beginPath();
graphics.ellipse(200, 225, 60, 45, 0, 0, 2 * Math.PI);
graphics.fill();

// Eyes
graphics.fillCircle(180, 230, 15); // Left eye
graphics.lineWidth = 1;
graphics.stroke();
graphics.strokeStyle = "black";
graphics.fillCircle(220, 230, 15); // Right eye
graphics.lineWidth = 1;
graphics.stroke();
graphics.strokeStyle = "black";

// Pupils
graphics.fillStyle = "black";
graphics.fillCircle(180, 230, 5);
graphics.fillCircle(220, 230, 5);

// Glows
graphics.fillStyle = "rgba(250, 250, 250, 0.89)";
graphics.fillCircle(178, 228, 2);
graphics.fillCircle(218, 228, 2);
}
```

```
// Mouth lines at the angles
// Left angle
graphics.beginPath();
graphics.moveTo(156, 256); // Start Point
graphics.lineTo(150, 262); // End point
graphics.lineWidth = 2;
graphics.stroke();

graphics.beginPath();
graphics.moveTo(156, 256); // Start Point
graphics.lineTo(157, 248); // End point
graphics.lineWidth = 2;
graphics.stroke();

// Right angle
graphics.beginPath();
graphics.moveTo(245, 256); // Start Point
graphics.lineTo(251, 262); // End point
graphics.lineWidth = 2;
graphics.stroke();

graphics.beginPath();
graphics.moveTo(245, 256); // Start Point
graphics.lineTo(244, 248); // End point
graphics.lineWidth = 2;
graphics.stroke();
```

Github: <https://github.com/Rorschach-IT/Computer-Graphics-Labs/blob/lab1/zad1/canvasDrawing.js>

## 4. Wynik działania



## 5. Wnioski

Dzięki narzędziu Canvas, przy użyciu różnego typu komend javascript, jesteśmy w stanie wytworzyć różne kształty na stronie internetowej. Użyteczne narzędzie, przydatne w sytuacji, gdy stricte na stronie człowiek tworzący stronę chciałby wytworzyć jakieś elementy, najczęściej bardziej zaawansowane (przy użyciu współrzędnych).

# Zadanie 2

## 1. Polecenie

W pliku 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

Po utworzeniu przycisku czyszczącego ekran Canva user wywołuje funkcje czyszczącą ekran w javascript. Do opcji <select> został dodany nowy kolor żółty, użytkownik, po wybraniu go, może rysować kształty na stronie o takim kolorze. Domyślnie w javascript jest utworzony mechanizm tworzący prostokąty. W HTML został dodany segment <select> z dwoma opcjami <option>, a więc user może wybrać sobie kształt w którym będzie rysować po Canvie z puli: {kwadrat, wielokąt}.

## 3. Wykorzystane komendy

Kod HTML pod nowe opcje:

```
<p><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> <!--New color option added-->
  </select>
</p>

<!--Which polygon will be drawn?-->
<select id="toolChoice">
  <option value="0">Draw Squares</option>
  <option value="1">Draw Polygons</option>
  <!--New option-->
</select>
<p>
  <button onclick="clearCanvas()">Clear</button>
</p>
```

Czyszczenie ekranu:

```
/*
   This function clears the entire canvas when being evoked
*/
function clearCanvas() {
    graphics.clearRect(0, 0, canvas.width, canvas.height);
    graphics.fillStyle = "white";
    graphics.fillRect(0, 0, canvas.width, canvas.height);
}
```

Dodanie nowego koloru do wyboru (modyfikacja istniejącej funkcji)

```
function doMouseMove(evt) {

    // Sprawdzamy wybrany kolor
    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";
    }
    // Added new color to options
    else if (colorChoice == 4) {
        graphics.fillStyle = "yellow";
    }
}
```

Użytkownik ma do wyboru dwa kształty:

```
function doMouseMove(evt) {

    var toolChoice = document.getElementById
    ("toolChoice").value;
    if (toolChoice == 0) {
        // Square drawing
        graphics.fillRect(x - 20, y - 20, 40, 40);
        graphics.strokeRect(x - 20, y - 20, 40, 40);
    }
    else if (toolChoice == 1) {
        // Polygon drawing
        var sides = 5;
        var radius = 20;
        var angle = Math.PI * 2 / sides;
        var points = [];

        for (var i = 0; i < sides; i++) {
            var px = x + radius * Math.cos(i * angle);
            var py = y + radius * Math.sin(i * angle);
            points.push(px, py);
        }

        graphics.fillPoly(...points);
        graphics.strokePoly(...points);
    }

    prevX = x;
    prevY = y;
}
```

Github: <https://github.com/Rorschach-IT/Computer-Graphics-Labs/tree/lab1/zad2>

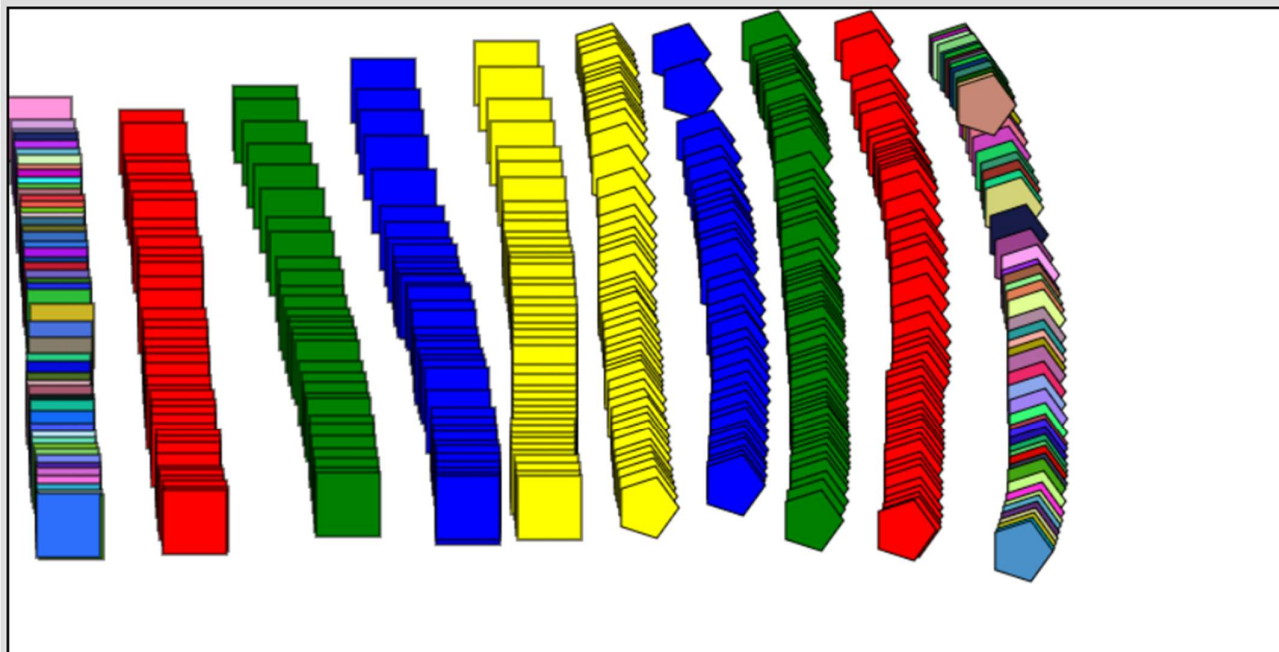
## 4. Wynik działania

### Lab 2, Exercise 2: Mousing

Color: Random ▾

Draw Polygons ▾

Clear



## 5. Wnioski

Narzędzie Canvas oprócz ręcznego wytwarzania kształtów statycznych, umożliwia również wdrażanie mechanizmów umożliwiających użytkownikowi strony rysowanie kształtów lub innych komponentów związanych z grafiką.