SPRAWOZDANIE
Zajęcia: Grafika komputerowa
Prowadzący: prof. dr. hab. Vasyl Martsenyuk

Laboratorium 2
4 III 2021 r.
Temat: „Grafika 2D z użyciem HTML Canvas”
Wariant:
Liczba kątów:5
Figura:7

Robert Laszczak
Informatyka I stopień
Stacjonarne, 4 semestr
Grupa 2B

1. Polecenie

a) Narysować obraz zgodnie z wariantem zadania (używając zarówno standardowych jak i niestandardowych funkcji rysowania)

b) Stworzyć narzędzia pozwalające na wykonanie 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:

Wariant zadania: figura nr 7

Liczba kątów n = 5

3. Wykorzystane komendy:

a) Kod źródłowy:

```
        <!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="600" height="600" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>
let canvas = document.querySelector("#myCanvas");
let context = canvas.getContext("2d");

function drawTriangle() {
  context.clearRect(0, 0, canvas.width, canvas.height);

  context.fillStyle = "#FFFFFF";
  context.fillRect(0, 0, canvas.width, canvas.height);

  let height = 200 * Math.cos(Math.PI / 6);

  context.beginPath();
  context.moveTo(100, 200);
  context.lineTo(300, 200);
  context.lineTo(200, 200 + height);
  context.closePath();


  // the outline
  context.lineWidth = 10;
  context.strokeStyle = '#666666';
  context.stroke();

  // the fill color
  context.fillStyle = "#FFCC00";
  context.fill();




    // górna część
  context.beginPath();
  context.lineTo(100, 200)
  context.lineTo(300, 200);
  context.lineTo(275, 150);
  context.lineTo(125, 150);
  context.lineTo(100, 200);
  context.lineTo(200, 200 + height);
  context.lineTo(300, 200);
  context.fillStyle = "#FF1493";
  context.fill();

  //szczegóły

context.beginPath();
 context.moveTo(175, 250);
 context.bezierCurveTo(175, 275, 225, 275, 225, 250);
 context.stroke();
 context.strokeStyle = "black";
                context.stroke();


   context.beginPath();
 context.moveTo(175, 250);
 context.bezierCurveTo(175, 265, 225, 265, 225, 250);
 context.stroke();
 context.strokeStyle = "black";
                context.stroke();


  // zęby

  context.beginPath();
context.moveTo(195, 260);
context.lineTo(205, 260);
context.stroke();
 context.strokeStyle = "white";
                context.stroke();


  //oczy

  context.beginPath();
context.arc(150, 175, 12, 0, 1.99*Math.PI);
   context.fillStyle = "white";
  context.fill();

   context.beginPath();
context.arc(150, 175, 7, 0, 1.99*Math.PI);
   context.fillStyle = "black";
  context.fill();


    context.beginPath();
context.arc(250, 175, 12, 0, 1.99*Math.PI);
   context.fillStyle = "white";
  context.fill();

   context.beginPath();
context.arc(250, 175, 7, 0, 1.99*Math.PI);
   context.fillStyle = "black";
  context.fill();
 //obramówka
 context.beginPath();
 context.lineTo(100, 200)
 context.lineTo(300, 200);
 context.lineTo(275, 150);
 context.lineTo(125, 150);
 context.lineTo(100, 200);
 context.lineTo(200, 200 + height);
 context.lineTo(300, 200);
 context.strokeStyle = "black";
         context.stroke();








}
drawTriangle();


</script>

</body>
</html>
```

b) Kod źródłowy:

```
<!DOCTYPE html>
<html>
<!--
  This web page does the minimal setup for using mouse events along
  with 2D canvas graphics.
-->
<head>
<meta charset="UTF-8">
<title>Grafika Komputerowa | 04 III 2021 | Zadanie 2</title>
<style>
   /* This style section is here to make the canvas more obvious on the
      page.  It is white on a light gray page background, with a thin
      black border.  Also, turn off text selection to avoid having
      selection interfere with mouse action. */
   body {
      background-color: #DDDDDD;
      -webkit-user-select: none; /* turn off text selection  / Webkit */
      -moz-user-select: none;   /* Firefox */
      -ms-user-select: none;    /* IE 10  */
      -o-user-select: none;     /* Opera */
      user-select: none;
   }
   canvas {
      background-color: white;
      display: block;
   }
   #canvasholder {
        border:2px solid black;
        float: left; /* This makes the border exactly fit the canvas. */
      }
    </style>
    <script>

      "use strict";

      var canvas;
      var graphics;

      function randomColorString() {
        var r = Math.floor(256*Math.random());
        var g = Math.floor(256*Math.random());
        var b = Math.floor(256*Math.random());
        return "rgb(" + r + "," + g + "," + b + ")";
      }

      function installMouseHandler() {

        var dragging = false;
        var startX, startY;
        var prevX, prevY;
        var colorChoice;
        var PolyChoice;

        function doMouseDown(evt) {
          if (dragging) {
            return;
          }
          if (evt.button != 0) {
            return;
          }
          var x,y;
          var r = canvas.getBoundingClientRect();
          x = Math.round(evt.clientX - r.left);
          y = Math.round(evt.clientY - r.top);
          dragging = true;
          if (dragging) {
            startX = prevX = x;
            startY = prevY = y;
            document.addEventListener("mousemove", doMouseMove, false);
            document.addEventListener("mouseup", doMouseUp, false);
          }
          colorChoice = Number(document.getElementById("colorChoice").value);
          PolyChoice = Number(document.getElementById("PolyChoice").value);

        }

        function doMouseMove(evt) {

          if (!dragging) {
            return;
          }
          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);
         var xPoints =[];
      var yPoints =[];
      for(var i=0;i<5;i++){
      xPoints[i] = 30*Math.cos((2*Math.PI*i)/5) + x;
      yPoints[i] = 30*Math.sin((2*Math.PI*i)/5) + y;
```

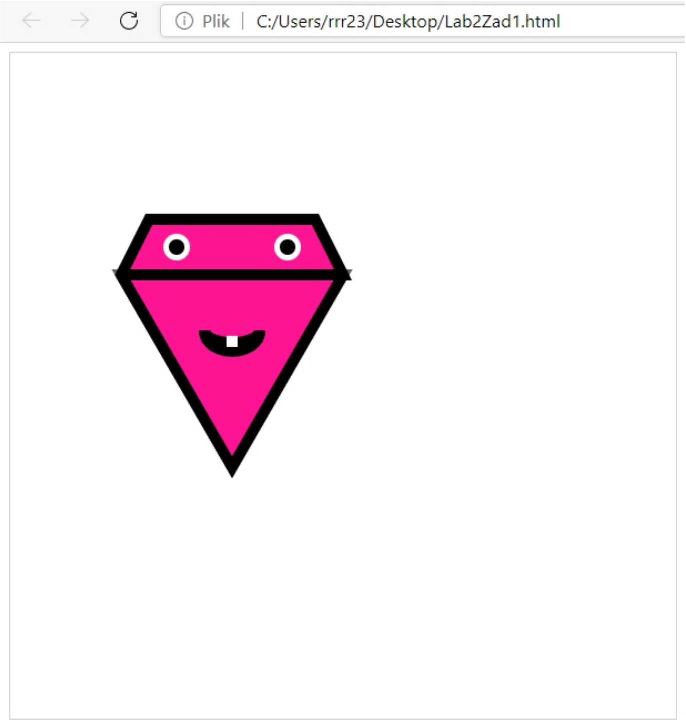
```
   }

         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 = "magenta";
         }

         if (PolyChoice == 0) {
            graphics.fillRect(x-20,y-20,40,40);
            graphics.strokeRect(x-20,y-20,40,40);
         }
         else if (PolyChoice == 1)
         {

graphics.fillPoly(xPoints[0],yPoints[0],xPoints[1],yPoints[1],xPoints[2],yPoints[2],xPoints[3],yPoints[3],xPoints[4],
yPoints[4],xPoints[5],yPoints[5]);

graphics.strokePoly(xPoints[0],yPoints[0],xPoints[1],yPoints[1],xPoints[2],yPoints[2],xPoints[3],yPoints[3],xPoints
[4],yPoints[4],xPoints[5],yPoints[5]);
         }

         /*------------------------------------------------------------*/

         prevX = x;
         prevY = y;
      }

      function doMouseUp(evt) {
         if (!dragging) {
            return;
         }
         dragging = false;
         document.removeEventListener("mousemove", doMouseMove, false);
         document.removeEventListener("mouseup", doMouseMove, false);
       }

       canvas.addEventListener("mousedown", doMouseDown, false);

  }

   function addGraphicsContextExtras(graphics) {
      graphics.strokeLine = function(x1,y1,x2,y2) {
        this.beginPath();
        this.moveTo(x1,y1);
        this.lineTo(x2,y2);
```

```
     this.stroke();
   }
   graphics.fillCircle = function(x,y,r) {
     this.beginPath();
     this.arc(x,y,r,0,2*Math.PI,false);
     this.fill();
   }
   graphics.strokeCircle = function(x,y,radius) {
     this.beginPath();
     this.arc(x,y,radius,0,2*Math.PI,false);
     this.stroke();
   }
   graphics.fillPoly = function() {
      if (arguments.length < 6)
        return;
      this.beginPath();
      this.moveTo(arguments[0],arguments[1]);
      for (var i = 2; i+1 < arguments.length; i = i + 2) {
        this.lineTo(arguments[i],arguments[i+1]);
      }
      this.closePath();
      this.fill();
   }
   graphics.strokePoly = function() {
      if (arguments.length < 4)
        return;
      this.beginPath();
      this.moveTo(arguments[0],arguments[1]);
      for (var i = 2; i+1 < arguments.length; i = i + 2) {
        this.lineTo(arguments[i],arguments[i+1]);
      }
      this.closePath();
      this.stroke();
   }
   graphics.fillOval = function(x,y,horizontalRadius,verticalRadius) {
     this.save();
     this.translate(x,y);
     this.scale(horizontalRadius,verticalRadius);
     this.beginPath();
     this.arc(0,0,1,0,2*Math.PI,false);
     this.restore();
     this.fill();
   }
   graphics.strokeOval = function(x,y,horizontalRadius,verticalRadius) {
     this.save();
     this.translate(x,y);
     this.scale(horizontalRadius,verticalRadius);
     this.beginPath();
     this.arc(0,0,1,0,2*Math.PI,false);
     this.restore();
     this.stroke();
   }
   graphics.getRGB = function(x,y) {
      var color = this.getImageData(x,y,1,1);
      return color.data;
   }
}

function doClear(){
graphics.fillStyle = "white";
graphics.fillRect(0,0,canvas.width,canvas.height);
document.getElementById("clearButton").onclick = doClear;
}
```

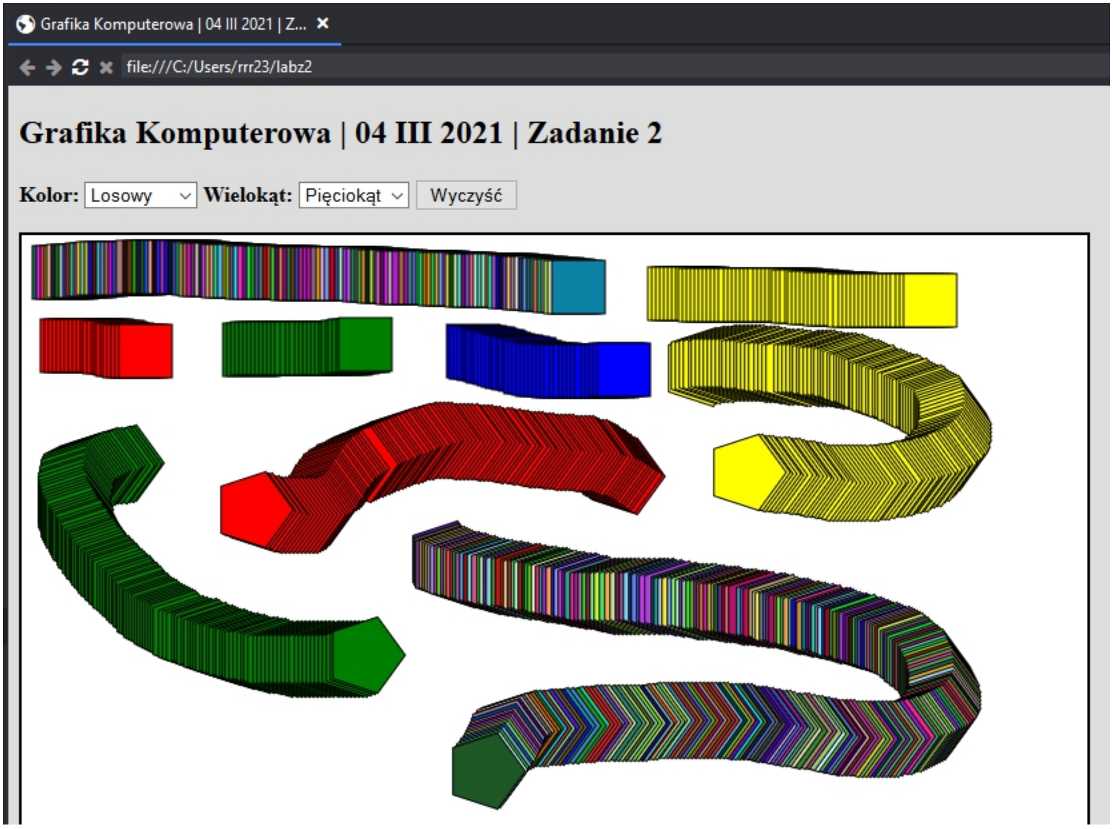
```
   function init() {
      try {
         canvas = document.getElementById("canvas");
         graphics = canvas.getContext("2d");
      } catch(e) {
         document.getElementById("canvasholder").innerHTML =
           "<p>Canvas graphics is not supported.<br>" +
           "An error occurred while initializing graphics.</p>";
           return;
      }
      addGraphicsContextExtras(graphics);
      installMouseHandler();
      graphics.fillStyle = "white";
      graphics.fillRect(0,0,canvas.width,canvas.height);
      doClear();
   }
</script>
</head>
<body onload="init()">

<h2>Grafika Komputerowa | 04 III 2021 | Zadanie 2</h2>

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

<p><b>Kolor:</b>
   <select id="colorChoice">
      <option value="0">Losowy</option>
      <option value="1">Czerwony</option>
      <option value="2">Zielony</option>
      <option value="3">Niebieski</option>
      <option value="4">Żółty</option>
             <option value="5">Magenta</option>
   </select>
   <b>Wielokąt:</b>
   <select id="PolyChoice">
      <option value="0">Kwadrat</option>
      <option value="1">Pięciokąt</option>
   </select>
   <button id="clearButton">Wyczyść</button>
</p>

<div id="canvasholder">
<canvas id="canvas" width="800" height="600">
</canvas>

</div>
</body>
</html>
```

4. Wyniki działania:

a)



b)


Grafika Komputerowa | 04 III 2021 | Z...
file:///C:/Users/rrr23/labz2

Grafika Komputerowa | 04 III 2021 | Zadanie 2

Kolor: Losowy **Wielokąt:** Pięciokąt

5. Wnioski

Na podstawie otrzymanych wyników można stwierdzić, że:
A) JavaScript ma funkcje pozwalające na swobodną iterację z użytkownikiem za pomocą okien dialogowych
B) Canvas pozwala urozmaicić widok strony internetowej o nowe elementy graficzne i dostarcza nam sporo możliwości
C) Duża część przeglądarek obsługuje interfejs API grafiki 2D pozwalający na tworzenie różnych grafik na stronach internetowych