

SPRAWOZDANIE

Zajęcia: Grafika Komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium: 1

Data: 26.02.2024

Temat: "Grafika 2D z użyciem HTML Canvas"

Wariant:

Zadanie 1: 5

Zadanie 2: Dodanie koloru i figury

Michał Michalik
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.3a

Zadanie 1

1. Polecenie:

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

2. Wprowadzane dane:

Do zadania wprowadziłem informację od prowadzącego o wyglądzie grafiki. W moim przypadku otrzymałem polecenie wykonania wersji 5.

3. Wykorzystane komendy:

Do wykonania zadania musieliśmy stworzyć kod, który reprezentuje grafika zadaną przez prowadzącego.

Kod potrzebny do utworzenia o 6-kąta:

```
14     var canvas = document.getElementById('canvas');
15     var ctx = canvas.getContext('2d');
16     const a = 2 * Math.PI / 6;
17     const r = 100;
18
19     function drawHexagon(x, y) {
20         ctx.beginPath();
21         for (var i = 0; i < 6; i++) {
22             ctx.lineTo(x + r * Math.cos(a * i), y + r * Math.sin(a * i));
23         }
24         ctx.fillStyle = "#A000FF";
25         ctx.fill();
26         ctx.closePath();
27         ctx.stroke();
28     }
```

Kod do dodania oczu:

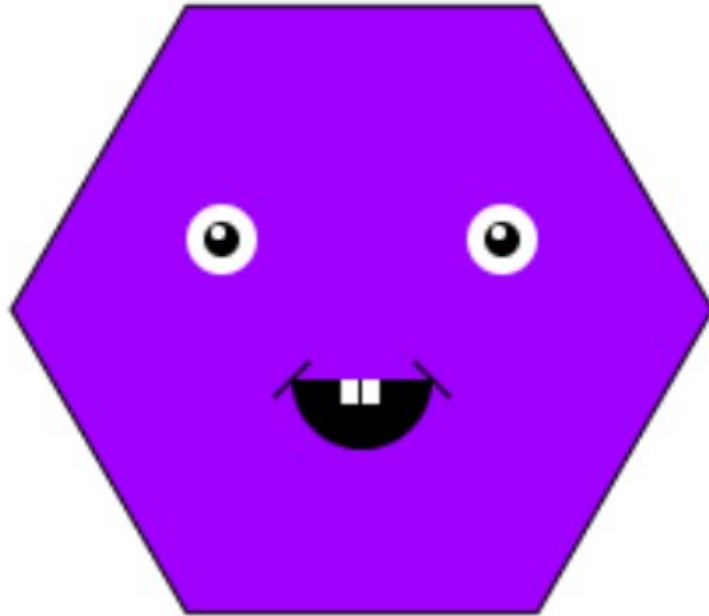
```
function drawEyes(x, y) {  
    ctx.beginPath();  
    ctx.arc(x - 40, y - 20, 10, 0, 2 * Math.PI);  
    ctx.fillStyle = "white";  
    ctx.fill();  
    ctx.closePath();  
  
    ctx.beginPath();  
    ctx.arc(x - 40, y - 20, 5, 0, 2 * Math.PI);  
    ctx.fillStyle = "black";  
    ctx.fill();  
    ctx.closePath();  
  
    ctx.beginPath();  
    ctx.arc(x - 41, y - 22, 2, 0, 2 * Math.PI);  
    ctx.fillStyle = "white";  
    ctx.fill();  
    ctx.closePath();  
  
    ctx.beginPath();  
    ctx.arc(x + 40, y - 20, 10, 0, 2 * Math.PI);  
    ctx.fillStyle = "white";  
    ctx.fill();  
    ctx.closePath();  
  
    ctx.beginPath();  
    ctx.arc(x + 40, y - 20, 5, 0, 2 * Math.PI);  
    ctx.fillStyle = "black";  
    ctx.fill();  
    ctx.closePath();  
}
```

```
60  
61     ctx.beginPath();  
62     ctx.arc(x + 39, y - 22, 2, 0, 2 * Math.PI);  
63     ctx.fillStyle = "white";  
64     ctx.fill();  
65     ctx.closePath();  
66 }  
67
```

Kod do rysowania ust:

```
function drawMouth(x, y) {  
  ctx.beginPath();  
  ctx.arc(x, y + 20, 20, 0, Math.PI);  
  ctx.fillStyle = "black";  
  ctx.fill();  
  ctx.closePath();  
  
  ctx.beginPath();  
  ctx.rect(x - 6, y + 20, 5, 7);  
  ctx.fillStyle = "white";  
  ctx.fill();  
  ctx.closePath();  
  
  ctx.beginPath();  
  ctx.rect(x, y + 20, 5, 7);  
  ctx.fillStyle = "white";  
  ctx.fill();  
  ctx.closePath();  
  
  ctx.beginPath();  
  ctx.moveTo(x - 15, y + 15);  
  ctx.lineTo(x - 25, y + 25);  
  ctx.moveTo(x + 15, y + 15);  
  ctx.lineTo(x + 25, y + 25);  
  ctx.strokeStyle = "black";  
  ctx.stroke();  
  ctx.closePath();  
}
```

4. Wynik działania:



5. Wnioski:

Dzięki użyciu canvas html i języka skryptowego JavaScript, znając odpowiednie metody możemy stworzyć wielokąty o dowolnej ilości wierzchołków a ponadto dokonywać różnorodnych przekształceń: obrócenie, przesunięcie, pochylenie.

Zadanie 2

1. Polecenie:

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:

Do zadania wykorzystałem informację od prowadzącego na temat tego co mamy zaimplementować do kodu udostępnionego przez niego.

3. Wykorzystane komendy:

Do wykonania zadania musieliśmy zmodyfikować otrzymany kod dodając do niego odpowiednie metody .

Kod potrzebny do dodania w funkcji „doMouseMove” aby dodać nowy kolor i kształty:

```
function doMouseMove(evt) {  
    // ...  
}  
else if (colorChoice == 4) {  
    graphics.fillStyle = "yellow";  
}  
  
if (sidesChoice < 5) {  
    if (figureChoice == 0) {  
        graphics.fillRect(x - 20, y - 20, 40, 40);  
        graphics.strokeRect(x - 20, y - 20, 40, 40);  
    }  
    else {  
        var circlePath = new Path2D();  
        circlePath.arc(x, y, 50, 0, 2 * Math.PI);  
  
        graphics.stroke(circlePath);  
        graphics.fill(circlePath);  
    }  
}  
else {  
    console.log(sidesChoice)  
    var sides = sidesChoice;  
    drawPolygon(x, y, 50, sides);  
}  
}
```

/*-----*/

```

function drawPolygon(x, y, radius, sides) {
    graphics.beginPath();
    var angle = (2 * Math.PI) / sides;
    for (var i = 0; i < sides; i++) {
        var xPos = x + radius * Math.cos(angle * i);
        var yPos = y + radius * Math.sin(angle * i);
        graphics.lineTo(xPos, yPos);
    }
    graphics.closePath();
    graphics.stroke();
    graphics.fill();
}

```

Kod aby dodać możliwość w selectach:

```

<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>
  </select>
  <select id="figureChoice">
    <option value="0" selected="selected">Rect</option>
    <option value="1">Circle</option>
  </select>

  <select id="newChoice">
    <option value="0"></option>
  </select>
  <script>
    var selectElement = document.getElementById("newChoice");

    for (let i = 5; i <= 20; i++) {
      var optionElement = document.createElement("option");
      optionElement.value = i;
      optionElement.text = i + " - kąt";
      selectElement.appendChild(optionElement);
    }
  </script>

```

Kod do czyszczenia selectów:

```
var figureChoiceInput = document.getElementById("figureChoice");
figureChoiceInput.addEventListener("change", function () {
    document.getElementById("newChoice").selectedIndex = 0;
});

var newChoiceInput = document.getElementById("newChoice");
newChoiceInput.addEventListener("input", function () {
    document.getElementById("figureChoice").selectedIndex = 3;
});
}
```

Kod do czyszczenia canvasa:

```
<button onclick="clear_can()">czyszczenie</button>

<script>
function clear_can() {
    console.log("Znikło wszystko niczym pieniądze z komuni")
    const canvas = document.getElementById("canvas")
    const context = canvas.getContext('2d');
    context.clearRect(0, 0, canvas.width, canvas.height);
}
</script>
```

4. Wynik działania:

