

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: prof. dr hab. Vasyl Martenyuk

Laboratorium 2

14.03.2023

Temat: "Grafika 2D z użyciem HTML Canvas"

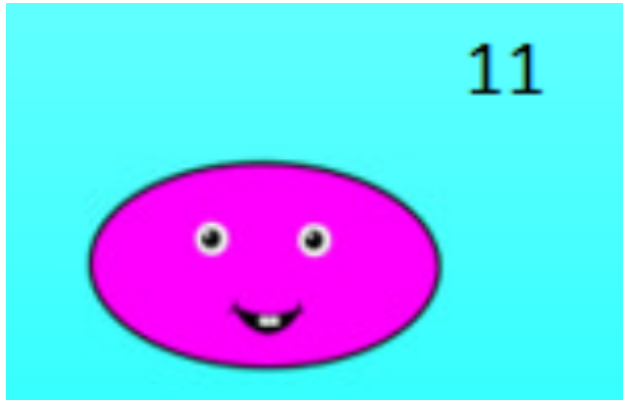
Wariant: 15-kąt, figura 11

Jakub Światłoch
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.2a

Pokazuję tylko kod który został zmodyfikowany, reszta kodu jest taka sama jaka została podana na zajęciach.

ZADANIE 1

1. **Polecenie:** Zadanie polegało na napisaniu kodu JavaScript w HTML za pomocą Canvas'u tak aby powstała określona figura nr 11



2. **Wprowadzane dane:** brak
3. **Wykorzystane komendy wraz z opisem oraz wyniki działania:**

Zostaną pokazane etapy po kolei tworzenia takiej figury (każda komenda zostanie opisana raz, jeśli pojawi się następny raz to nie ma sensu jej kolejny raz tłumaczyć):

graphics.clearRect(0,0,600,600); - wyczyści obszar graficzny o wymiarach 600 na 600 pikseli, zaczynając od pozycji (0, 0) na ekranie

graphics.ellipse(300, 300, 200, 100, 0, 0, 2 * Math.PI); - rysuje elipse na obiekcie Canvas. Parametry są następujące:

- 300, 300 – współrzędne położenia elipsy
- 200, 100 – szerokość i wysokość elipsy
- 0, 0 – kąty początkowy i końcowy okręgu (w radianach)
- 2*Math.PI – określa pełny obrót elipsy (2*pi radianów to 360 stopni).

graphics.beginPath(); - rozpoczyna nową ścieżkę graficzną

graphics.stroke(); - tworzy obrys danej figury

graphics.lineWidth = 3; - ustawia grubość linii na 3

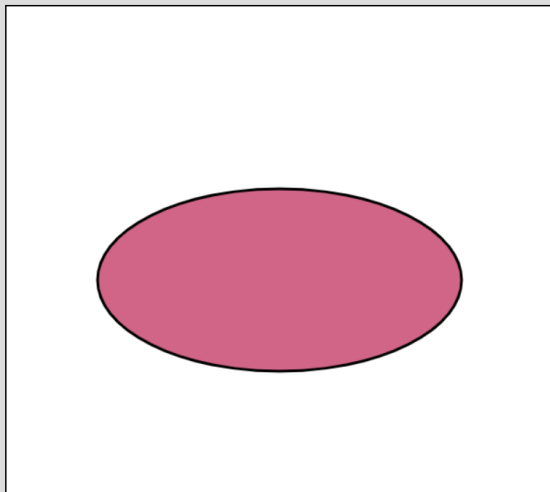
graphics.fillStyle = "#D16587"; - ustawia kolor który wypełni dany obszar

graphics.fill(); - wypełnia dany obszar, wybranym kolorem

graphics.strokeStyle = "black"; ustawia kolor który obramuje wybrany obszar takim kolorem

graphics.stroke(); - obramowuje obszar

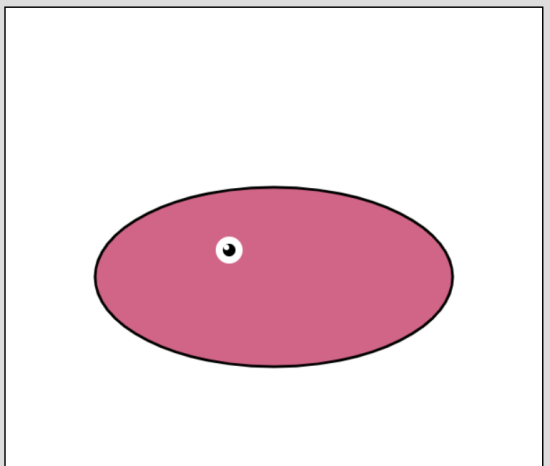
zad a



```
22
23
24 var canvas;
25 var graphics;
26 var pixelSize;
27
28 function draw() {
29     graphics.clearRect(0, 0, 600, 600);
30     //rysunek 11
31
32     graphics.beginPath();
33     graphics.ellipse(300, 300, 200,
34         100, 0, 0, 2 * Math.PI);
35     graphics.stroke();
36
37     graphics.lineWidth = 3;
38     graphics.fillStyle = "#D16587";
39     graphics.fill();
40     graphics.strokeStyle = "black";
41     graphics.stroke();
42
43     graphics.lineWidth = 1;
```

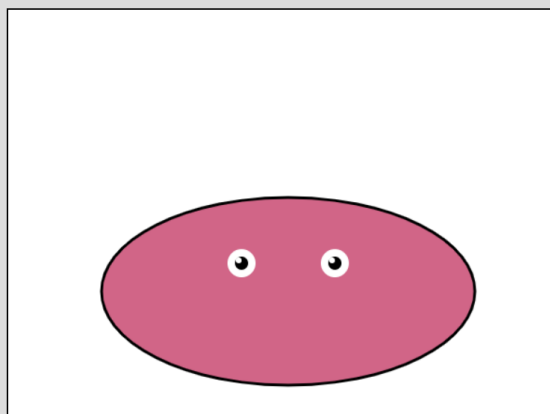
graphics.fillCircle(250, 270, 15); - powoduje narysowanie okręgu o środku w punkcie (250, 270) i promieniu 15

zad a



```
32     graphics.ellipse(300, 300, 200, 100, 0, 0, 2
33     graphics.stroke();
34
35     graphics.lineWidth = 3;
36     graphics.fillStyle = "#D16587";
37     graphics.fill();
38     graphics.strokeStyle = "black";
39     graphics.stroke();
40
41     graphics.lineWidth = 1;
42
43     //lewe oko
44     graphics.fillStyle = "white";
45     graphics.strokeStyle = "white";
46     graphics.fillCircle(250, 270, 15);
47     graphics.fillStyle = "black";
48     graphics.fillCircle(250, 270, 7);
49     graphics.fillStyle = "white";
50     graphics.strokeStyle = "white";
51     graphics.fillCircle(247, 267, 3);
52
```

zad a



```
44     graphics.fillStyle = "white";
45     graphics.strokeStyle = "white";
46     graphics.fillCircle(250, 270, 15);
47     graphics.fillStyle = "black";
48     graphics.fillCircle(250, 270, 7);
49     graphics.fillStyle = "white";
50     graphics.strokeStyle = "white";
51     graphics.fillCircle(247, 267, 3);
52
53     //prawe oko
54     graphics.fillStyle = "white";
55     graphics.strokeStyle = "white";
56     graphics.fillCircle(350, 270, 15);
57     graphics.fillStyle = "black";
58     graphics.fillCircle(350, 270, 7);
59     graphics.fillStyle = "white";
60     graphics.strokeStyle = "white";
61     graphics.fillCircle(347, 267, 3);
```

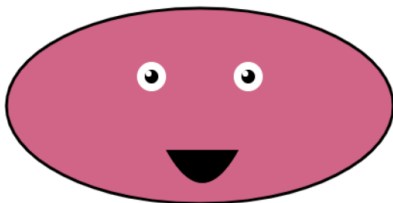
graphics.moveTo(265, 345); - przesuwa aktualny kontekst rysowania do współrzędnych 265, 345. Oznacza to, że wszystkie kolejne linie, które są rysowane, będą rozpoczynać się od tego punktu

gdyby połączyć **graphics.moveTo(265, 345)** oraz **graphics.bezierCurveTo(290, 390, 315, 390, 340, 345);** to można to tak opisać

Argumenty są podzielone na trzy grupy:

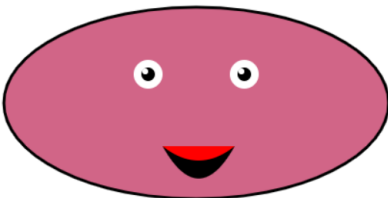
1. Pierwsza grupa (265, 345) przedstawia punkt początkowy na którym ma zacząć się rysowanie krzywej.
2. Druga grupa (290, 390, 315, 390) przedstawia punkty kontrolne, które określają kształt krzywej.
3. Trzecia grupa (340, 345) przedstawia punkt końcowy, w którym ma się zakończyć rysowanie krzywej.

dzięki temu powstaje takie coś:



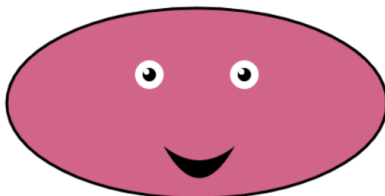
```
64
65     graphics.beginPath();
66     graphics.moveTo(265, 345);
67
68     //dolna warga
69     graphics.bezierCurveTo(290, 390, 315, 390, 340, 345);
70     graphics.fillStyle = "black";
71     graphics.fill();
72
```

ustawiono na czerwono górną wargę aby było lepiej widać co się dzieje



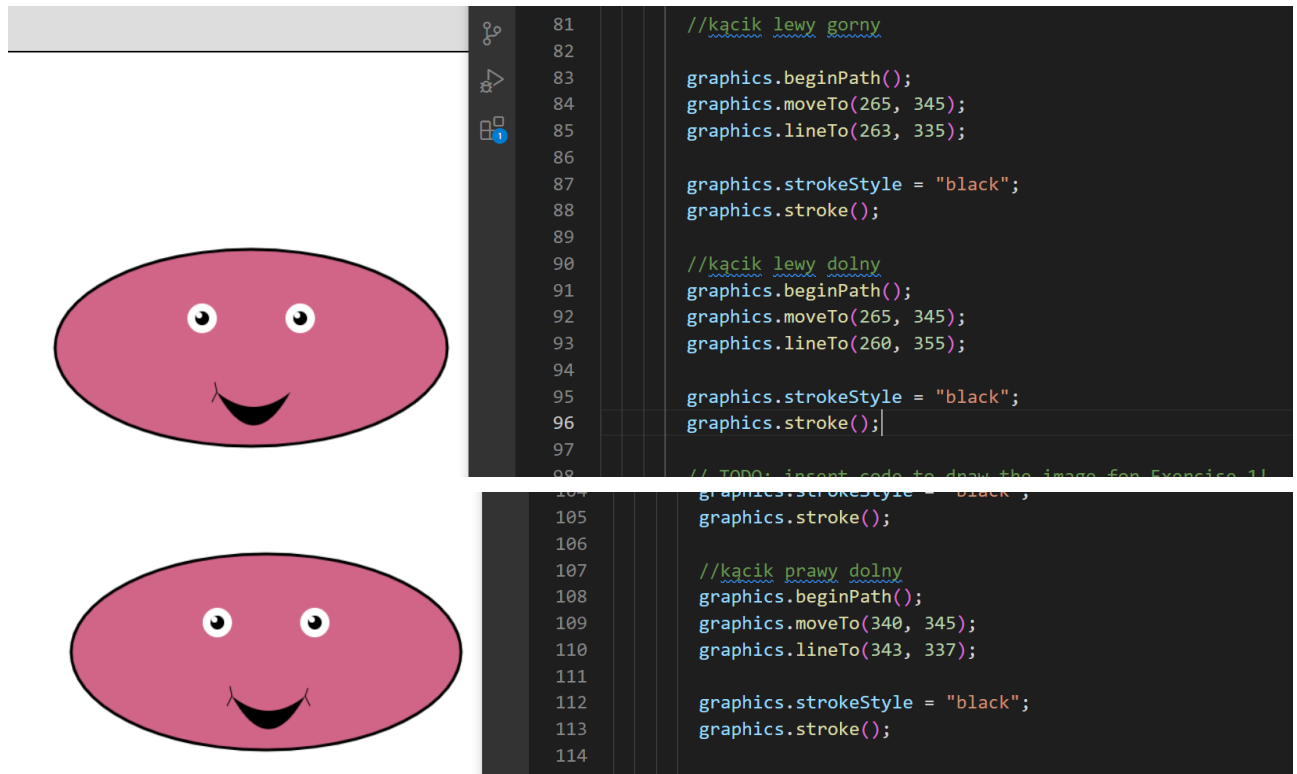
```
67
68     //dolna warga
69     graphics.bezierCurveTo(290, 390, 315, 390, 340, 345);
70     graphics.fillStyle = "black";
71     graphics.fill();
72
73     //gorna warga
74     graphics.beginPath();
75     graphics.moveTo(265, 345);
76     graphics.bezierCurveTo(290, 365, 315, 365, 340, 345);
77
78     graphics.fillStyle = "red";
```

zmieniono górną wargę na kolor wtapiający się w tło aby dało to efekt odcięcia

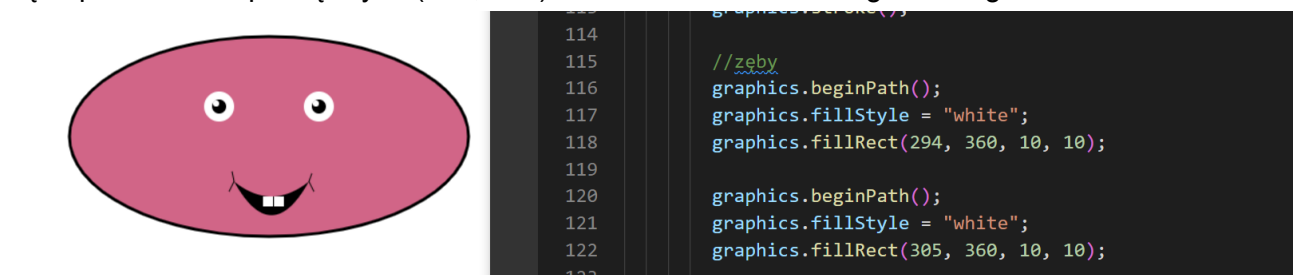


```
68     //dolna warga
69     graphics.bezierCurveTo(290, 390, 315, 390, 340, 345);
70     graphics.fillStyle = "black";
71     graphics.fill();
72
73     //gorna warga
74     graphics.beginPath();
75     graphics.moveTo(265, 345);
76     graphics.bezierCurveTo(290, 365, 315, 365, 340, 345);
77
78     graphics.fillStyle = "#D16587";
79     graphics.fill();
```

graphics.lineTo(260, 355); - rysowanie linii od aktualnego położenia do punktu (260, 355)



graphics.fillRect(294, 360, 10, 10); - rysuje prostokąt o wymiarach 10 x 10 pikseli, który zaczyna się w punkcie o współrzędnych (294, 360) w kontekście kontekstu graficznego canvas



Link do GitHub w którym znajduje się kod z zadania oraz wyniki zadania

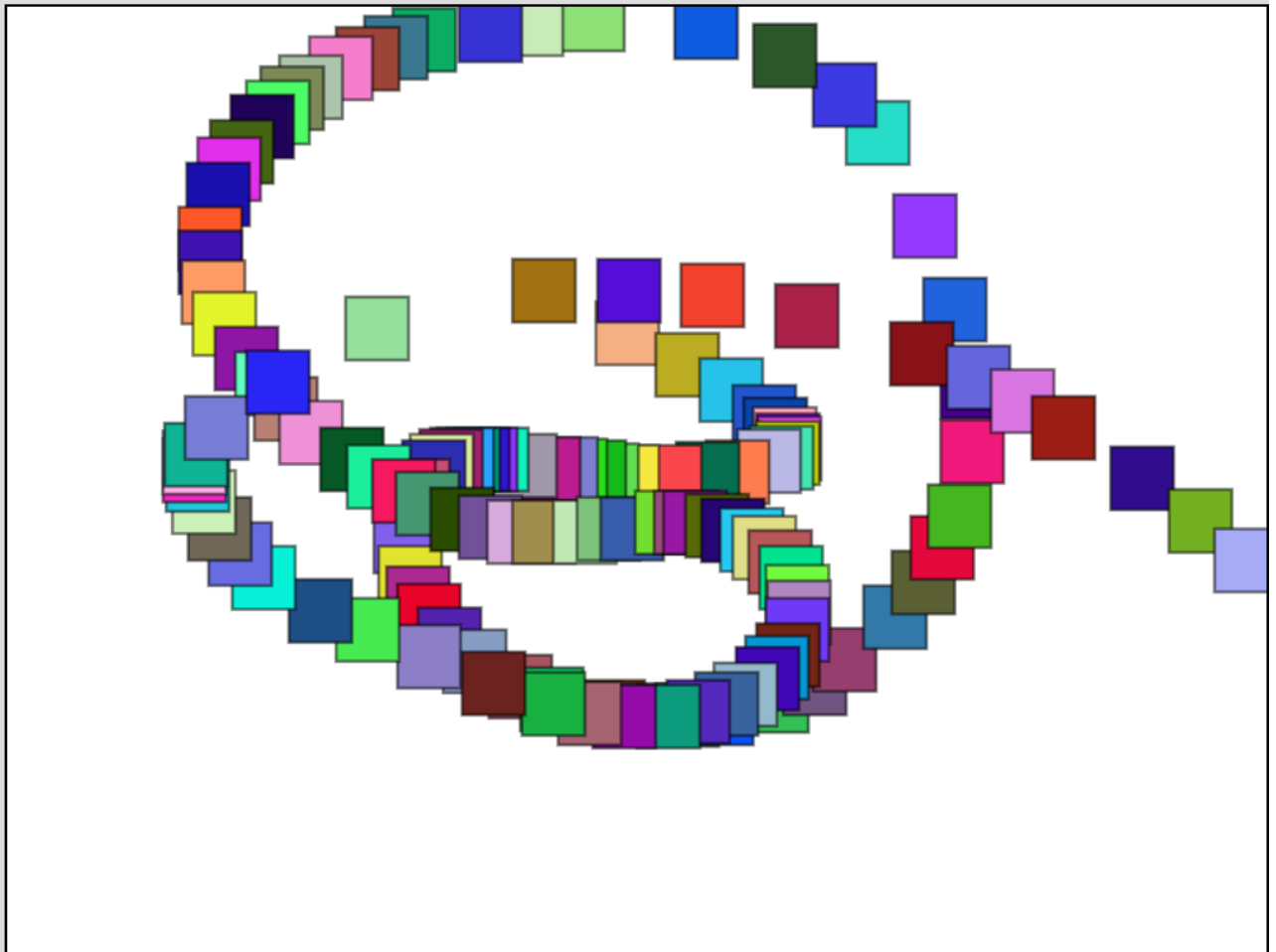
https://github.com/JakubSwiatlochgit/Grafika_Komputerowa/tree/main/cw2

Zadanie 2:

1. Polecenie: Zmodyfikuj istniejący kod tak aby zamiast kwadratu który co chwile zmienia kolor, był piętnastokąt który co chwile zmienia kolor

Lab 2, Exercise 2: Mousing

Color:



2. **Wprowadzane dane:** brak

3. **Wykorzystane komendy wraz z opisem:**

fillHexagon() rysuje wypełniony sześciokąt wybraną kolorem w podanych współrzędnych x i y, a jego średnica wynosi 40.

strokeHexagon() rysuje obrys sześciokąta w podanych współrzędnych x i y, a jego średnica wynosi 40.

```
graphics.fillHexagon(x, y, 40);  
graphics.strokeHexagon(x, y, 40);
```

Ta funkcja rysuje piętnastokąt o określonym położeniu (x, y) i promieniu (r). Przesuwa kursor do położenia (x + r, y) i następnie wykonuje 15 linii, aby utworzyć piętnastokąt. Kiedy wszystkie linie zostaną wykonane, funkcja wypełni piętnastokąt wybranym kolorem (w tym przypadku losowym).

```
function addGraphicsContextExtras(graphics) {
  graphics.fillHexagon = function (x, y, r) {
    var x0 = x;
    var y0 = y;
    this.beginPath();
    this.moveTo(x0 + r, y0);
    for (var i = 0; i < 15; i++) {
      x = x0 + r * Math.cos((i * 2 * Math.PI) / 15);
      y = y0 + r * Math.sin((i * 2 * Math.PI) / 15);
      this.lineTo(x, y);
    }
    this.fill();
  };
};
```

Poniższa funkcja działa na takiej samej zasadzie jak powyższa ale zamiast wypełniać piętnastokąt to go obramowuje

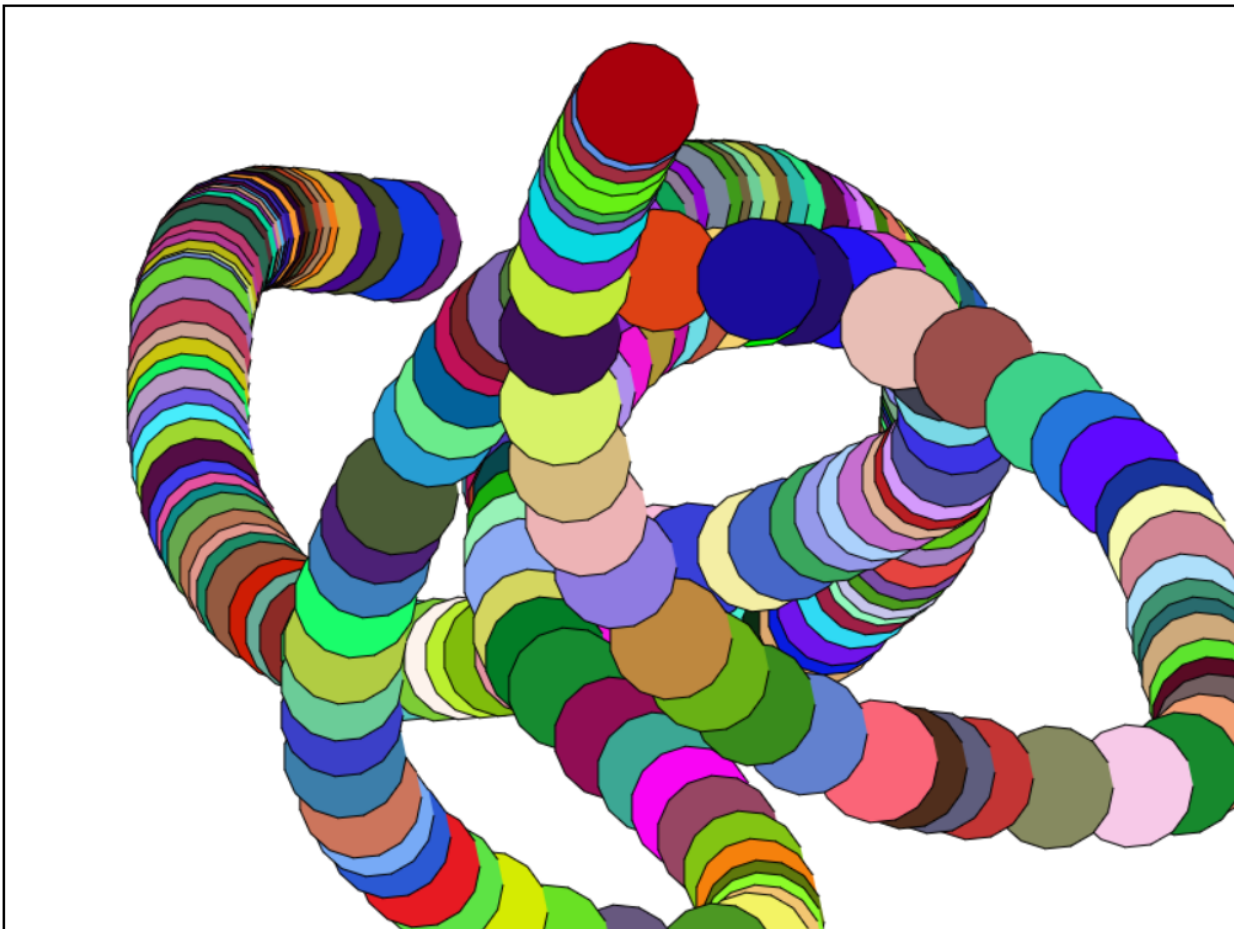
```
graphics.strokeHexagon = function (x, y, r) {
  var x0 = x;
  var y0 = y;
  this.beginPath();
  this.moveTo(x0 + r, y0);
  for (var i = 0; i < 15; i++) {
    x = x0 + r * Math.cos((i * 2 * Math.PI) / 15);
    y = y0 + r * Math.sin((i * 2 * Math.PI) / 15);
    this.lineTo(x, y);
  }
  this.stroke();
};
```

4. Wyniki

Niestety z niewiadomych mi przyczyn funkcja `graphics.stroke Hexagon` nie obramowuje go w całości, jedna ściana nie zostaje pomalowana



Color:



Wnioski:

Istnieje wiele różnych sposobów na wyświetlanie wizualnych elementów w kontekście graficznym. Dzięki funkcji takiej jak `bezierCurveTo` można tworzyć zaawansowane kształty lub kontury, które mogą być używane do tworzenia obrazów lub nawet animacji. Przy odrobinie wprawy i ćwiczenia można z łatwością opanować techniki tworzenia grafiki, które pozwolą na tworzenie atrakcyjnych i wyrafinowanych obrazów.