

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący:

prof. dr hab. Vasyl Martsenyuk,

mgr inż. Mikołaj Grygiel

Laboratorium 2

16.03.2025

Temat: Przekształcenia 2D w bibliotece pygame

Wariant 1

Aleksander Dobija
Informatyka I stopień,
niestacjonarne,
4 semestr,
Gr.2a

1. Polecenie:

1. Pokazany jest obraz `shuttle.jpg` w panelu. Narysować zamiast obrazu wielokat według wariantu (liczba n). Okno ma wymiary 600 na 600 pikseli, a wielokat ma promień 150 pikseli. Kolejne zadanie polega na stosowaniu odpowiednich przekształceń do wielokata (lub bedziesz potrzebował kombinacji przekształceń) po naciśnięciu na klawisze od 1 do 9 (patrz Fig. 1).
2. Narysować figure określona wariantem (patrz Fig. 2). Dostępne są trzy podstawowe kształty: koło, kwadrat, trójkąt.

Podstawowe przekształcenia dostępne są przez `pygame.transform`

2. Wprowadzane dane:

-

3. Wykorzystane komendy:

1. kod źródłowy:

```
import pygame
import math

pygame.init()

win = pygame.display.set_mode((600, 600))
pygame.display.set_caption("Przekształcenia")

NIEBIESKI = (0, 255, 0)

oryginalne_punkty = [(300, 100), (400, 200), (350, 350), (250, 350), (200, 200)]
punkty = oryginalne_punkty[:]

def obrot_punkty(kat):
    kat = math.radians(kat)
    return [(300 + (x - 300) * math.cos(kat) - (y - 300) * math.sin(kat),
              300 + (x - 300) * math.sin(kat) + (y - 300) * math.cos(kat))
            for x, y in oryginalne_punkty]

def przekształc_pieciokat(numer):
    global punkty
    if numer == 1:
        punkty = [(x * 0.5 + 150, y * 0.5 + 150) for x, y in oryginalne_punkty]
    elif numer == 2:
        punkty = obrot_punkty(45)
    elif numer == 3:
        punkty = obrot_punkty(90)
```

```

elif numer == 4:
    punkty = [(x + (y - 300) * 0.5, y) for x, y in oryginalne_punkty]
elif numer == 5:
    punkty = [(x, y * 0.7 + 90) for x, y in oryginalne_punkty]
elif numer == 6:
    temp = obroc_punkty(90)
    punkty = [(x + (y - 300) * 0.5, y) for x, y in temp]
elif numer == 7:
    temp = obroc_punkty(90)
    punkty = [(x, y * 0.7 + 90) for x, y in temp]
elif numer == 8:
    temp = obroc_punkty(45)
    punkty = [(x, y * 0.7 + 90) for x, y in temp]
elif numer == 9:
    temp = obroc_punkty(90)
    temp = [(x + (y - 300) * 0.5, y) for x, y in temp]
    punkty = [(x + (y - 300) * 0.5, y) for x, y in temp]
else:
    punkty = oryginalne_punkty[:]

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
        elif event.type == pygame.KEYDOWN:
            if pygame.K_1 <= event.key <= pygame.K_9:
                przekształc_pieciokat(event.key - pygame.K_0)

    win.fill((0, 0, 0))

    pygame.draw.polygon(win, NIEBIESKI, punkty)

    pygame.display.update()

pygame.quit()

```

2. kod źródłowy:

```

import pygame

pygame.init()
win = pygame.display.set_mode((600, 600))
pygame.display.set_caption("Koło z kwadratem")

# Deklarowanie kolorów
CZARNY = (0, 0, 0)
ZOLTY = (255, 255, 0)
BIALY = (255, 255, 255)

```

```
run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

    # Wypełnienie tła
    win.fill(BIALY)

    # Rysowanie czarnego koła
    center = (300, 300)
    radius = 150
    pygame.draw.circle(win, CZARNY, center, radius)

    # Rysowanie żółtego kwadratu w środku koła
    square_size = 100
    square_x = center[0] - square_size // 2
    square_y = center[1] - square_size // 2
    pygame.draw.rect(win, ZOLTY, (square_x, square_y, square_size,
square_size))

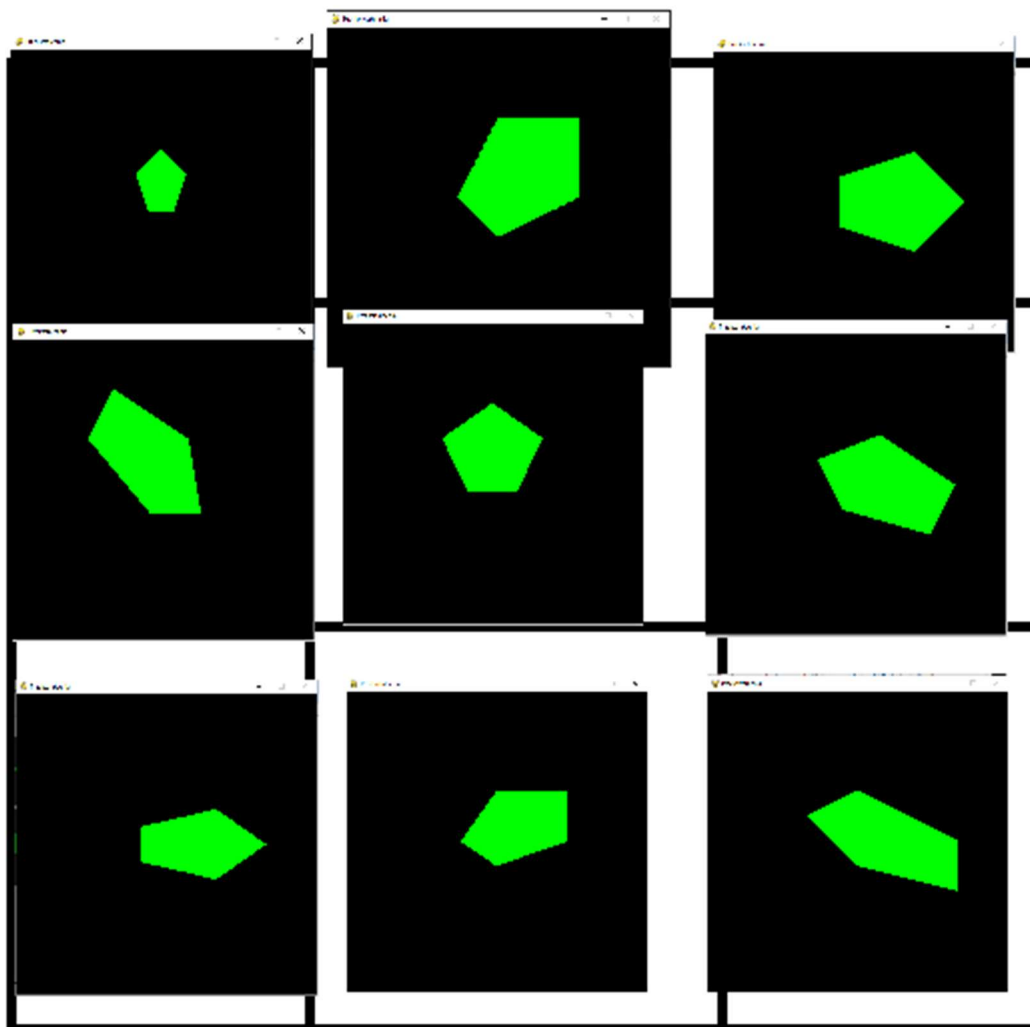
    pygame.display.update()

pygame.quit()
```

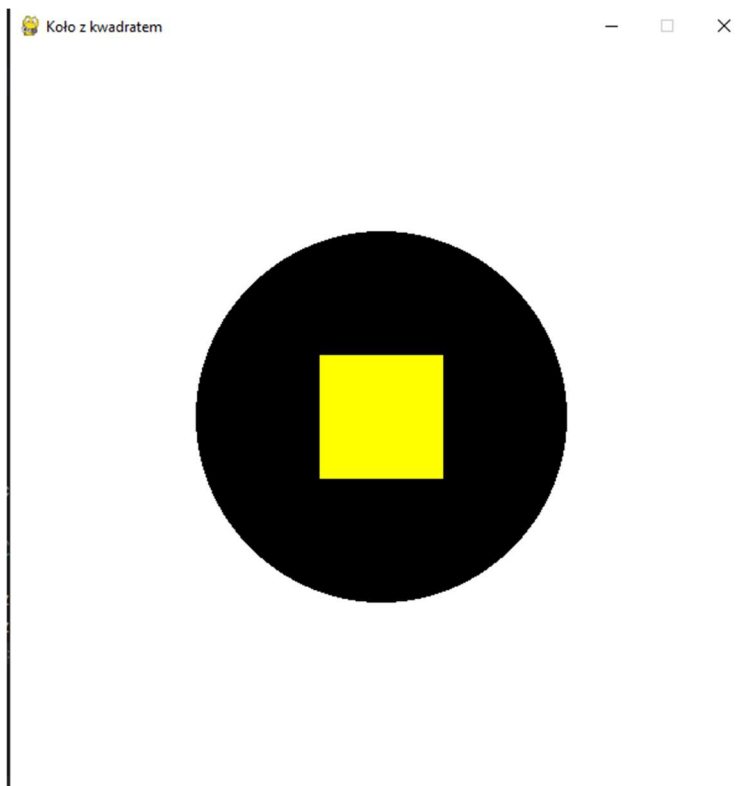
<https://github.com/adobija/GrafikaKomputerowaUBB/tree/main/16.03.2025>

4. Wynik działania:

1.



2.



5. Wnioski:

Na podstawie otrzymanego wyniku można stwierdzić, że na zajęciach skutecznie zapoznałem się z podstawowymi technikami pracy z grafiką 2D w bibliotece Pygame. Udało mi się poprawnie zainicjalizować okno gry, obsłużyć zdarzenia oraz narysować różne figury geometryczne. W szczególności nauczyłem się tworzyć i wyświetlać prostokąty oraz wielokąty, a także stosować ich transformacje przy użyciu funkcji dostępnych w Pygame. Przeprowadzone ćwiczenia pozwoliły mi lepiej zrozumieć zasady manipulacji grafiką komputerową i praktycznie zastosować zdobytą wiedzę.