SPRAWOZDANIE

Zajęcia: Grafika Komputerowa
Prowadzący: mgr inż. Mikołaj Grygiel

Laboratorium: 3
Data: 05.03.2025
Temat: "Przekształcenia 2D w bibliotece pygame"

Wariant:
Zadanie 1: 14+4=18-kąt
Zadanie 2: 2

Illia Bryka,
Informatyka I stopień,
stacjonarne,
4 semestr,
Gr.1a

Zadanie 1

1. Polecenie:

Pokazany jest obraz shuttle.jpg w panelu. Narysowa´c zamiast obrazu wielokat wed lug wariantu (liczba n). Okno ma wymiary 600 na 600 pikseli, a wielokat ma promie´n 150 pikseli. Kolejne zadanie polega na stosowaniu odpowiednich przekszta lce´n do wielokata (lub bedziesz potrzebowa l kombinacji przekszta lce´n) po naci´snieciu na klawisze od 1 do 9 (patrz Fig. 1).

2. Wprowadzane dane:

Do zadania wprowadziłem informację od prowadzącego o ilości wierzchołków, które ma posiadać stworzony przeze mnie wielokąt. W moim przypadku otrzymałem polecenie wykonania 18-kąta.

3. Wykorzystane komendy:

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

Aby wykonać zadanie trzeba było zmodyfikować podany kod, poprzez dodanie do niego nowych metod, tworzących interesujący nas wielokąt

Kod potrzebny do utworzenia 18-kąta:

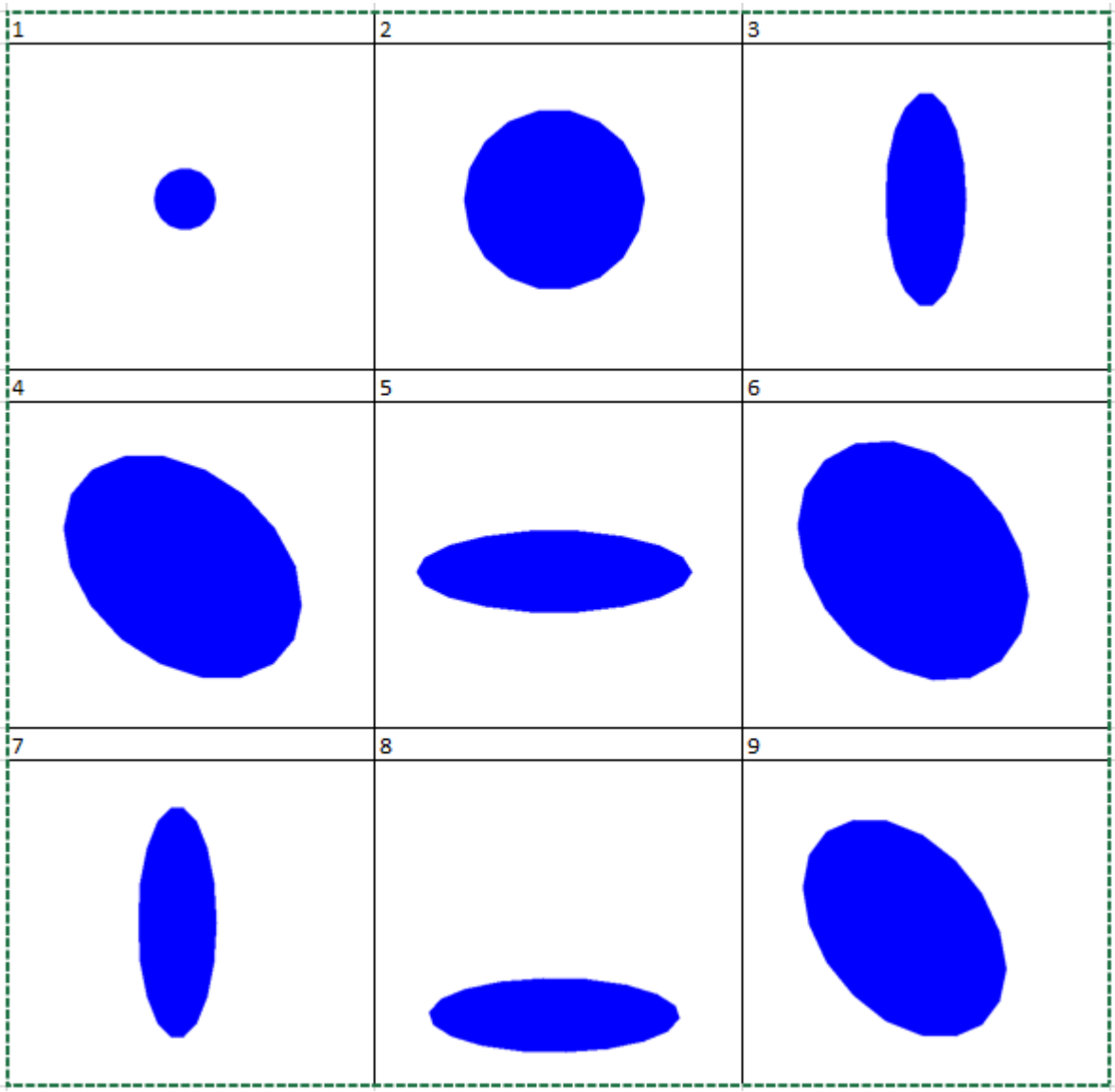
```
n_sides = 18
radius = 150
polygon = []
for i in range(1, n_sides + 1):
    angle = 2 * math.pi * i / n_sides
    x = radius * math.cos(angle)
    y = radius * math.sin(angle)
    polygon.append((x, y))
```

Kod do przekształceń wielokąta, tak jak w poleceniu:

```python
def identity(x, y):
    return x, y

def trans1(x, y):
    return 0.35 * x, 0.35 * y

def trans2(x, y):
    theta = math.radians(60)
    return x * math.cos(theta) - y * math.sin(theta), x * math.sin(theta) + y * math.cos(theta)

def trans3(x, y):
    sx, sy = -0.3 * x, 0.8 * y
    return -sx, -sy  # obrót 180°

def trans4(x, y):
    return x + 0.35 * y, y

def trans5(x, y):
    sx, sy = x, 0.3 * y
    return sx, sy

def trans6(x, y):
    sh_x, sh_y = x, y - 0.3 * x
    return sh_x, -sh_y

def trans7(x, y):
    sx, sy = 0.3 * x, 0.9 * y
    return -sx, -sy

def trans8(x, y):
    theta = math.radians(45)
    rx = x * math.cos(theta) - y * math.sin(theta)
    ry = x * math.sin(theta) + y * math.cos(theta)
    tx, ty = rx, ry + 300
    return tx, 0.3 * ty

def trans9(x, y):
    rx, ry = -x, -y  # obrót 180°
    sx = rx
    sy = 0.4 * rx + ry
    return sx - 100, sy
```

Link do Repozytorium: https://github.com/bebrabimba/Grafika-Komputerowa/blob/main/Lab3/Lab3Ex1.py

4. Wynik działania:



Zadanie 2

1. Polecenie:

Narysowa´c figure okre´slona wariantem (patrz Fig. 2). Dostepne sa trzy podstawowe kszta lty: ko lo, kwadrat, tr´ojkat.
Podstawowe przekszta lcenia dostepne sa przez pygame.transform

2. Wprowadzane dane:

Do zadania wykorzystałem informację od prowadzącego na temat tego który wariant powinienem wybrać. Mój wariant to nr 2

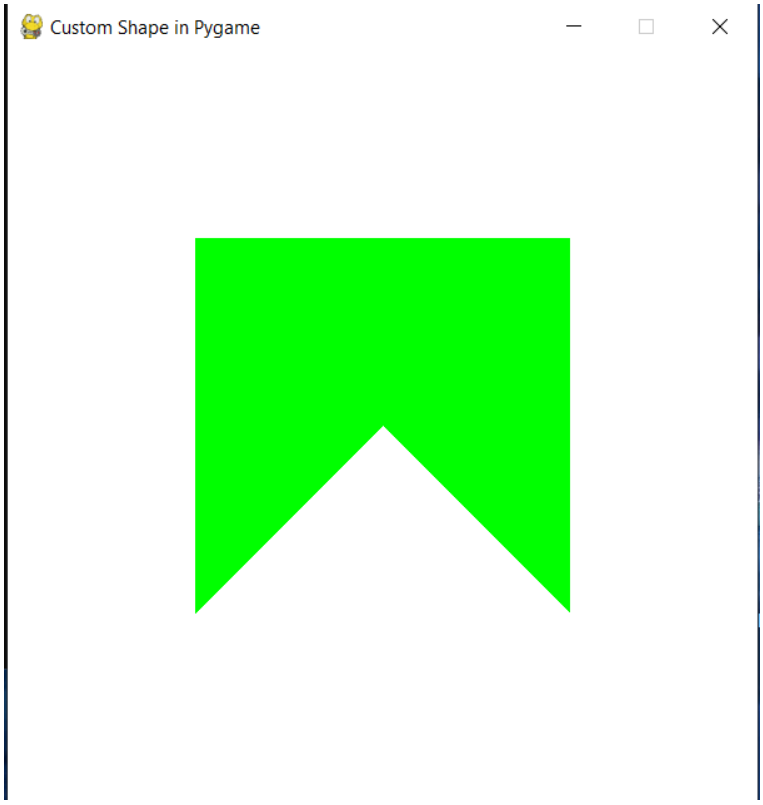
3. Wykorzystane komendy:

Do wykonania zadania musieliśmy zmodyfikować otrzymany kod dodając do niego odpowiednie metody w pygame.
Kod potrzebny do utworzenia figury nr 2:

```python
# Figury
def draw_custom_shape(surface, position):
    rect_width, rect_height = 300, 300
    tri_width, tri_height = 300, 150

    # prostokąt
    rect_x = position[0] - rect_width // 2
    rect_y = position[1] - rect_height // 2
    pygame.draw.rect(surface, GREEN, (rect_x, rect_y, rect_width, rect_height))

    # trójkąt
    triangle_points = [
        (position[0] - tri_width // 2, position[1] + rect_height // 2),
        (position[0] + tri_width // 2, position[1] + rect_height // 2),
        (position[0], position[1] + rect_height // 2 - tri_height)
    ]
    pygame.draw.polygon(surface, WHITE, triangle_points)
```

Link do Repozytorium: https://github.com/bebrabimba/Grafika-Komputerowa/blob/main/Lab3/Lab3Ex2.py

4. Wynik działania:



Wnioski:

Nauczyliśmy się pracować w środowisku Python/pygame. Dzięki użyciu stworzonych wcześniej funkcji, możemy ustawić współrzędne w taki sposób, aby program rysował nam dowolne figury, w których można zmieniać kolor, skalę czy to w jaki sposób są obrócone.