

# **SPRAWOZDANIE**

Zajęcia: Grafika Komputerowa

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium: 2**

**Data: 4.02.2024**

**Temat: "Przekształcenia 2D w bibliotece pygame"**

**Wariant:**

**Zadanie 1: 5**

**Zadanie 2: 2**

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) i opcje jego modyfikowania za pomocą klawiszy od 1 do 9.

## 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 i system zmian zadany przez prowadzącego.

Kod potrzebny do utworzenia o 5-kąta:

```
def draw_angle(angle_degrees):  
    size = 100  
    center = (width // 2, height // 2)  
    angle = 360 / angle_degrees  
    points = []  
  
    for i in range(angle_degrees):  
        x = center[0] + size * math.cos(math.radians(i * angle))  
        y = center[1] + size * math.sin(math.radians(i * angle))  
        points.append((x, y))  
  
    return pygame.draw.polygon(win, BIALY, points, width=2)
```

## Kod do obsługi przycisków:

```
def handle_key_event(event):
    win.fill(CZARNY)
    draw_angle(5)
    changed = None

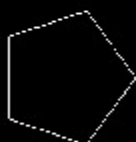
    match event.key:
        case pygame.K_1:
            changed = pygame.transform.scale(win, size: (width * 0.35, height * 0.35))
        case pygame.K_2:
            changed = pygame.transform.rotate(win, angle: 45)
        case pygame.K_3:
            changed = pygame.transform.flip(win, flip_x: 0, flip_y: 1)
        case pygame.K_4:
            changed = pygame.transform.scale_by(win, factor: (0.35, 1))
            changed = pygame.transform.rotozoom(changed, angle: 45, scale: 1)
        case pygame.K_5:
            changed = pygame.transform.scale(win, size: (width, int(height * 0.35)))
        case pygame.K_6:
            changed = pygame.transform.scale_by(win, factor: (0.35, 1))
            changed = pygame.transform.rotozoom(changed, angle: 180, scale: 1)
        case pygame.K_7:
            changed = pygame.transform.scale_by(win, factor: (0.5, 1))
            changed = pygame.transform.flip(changed, flip_x: 1, flip_y: 0)
        case pygame.K_8:
            changed = pygame.transform.scale_by(win, factor: (1, 0.4))
            changed = pygame.transform.rotate(changed, -20)
        case pygame.K_9:
            changed = pygame.transform.scale_by(win, factor: (0.35, 1))
            changed = pygame.transform.rotozoom(changed, angle: 90, scale: 1)
        case _:
            pygame.draw.rect(win, ZIELONY, rect: (200, 200, 200, 200))
            pygame.draw.polygon(win, BIALY, points: [(201, 400), (399, 400), (300, 300)])
```

```
if changed:
    win.blit(changed, dest: ((width - changed.get_width()) // 2, (height - changed.get_height()) // 2))
    pygame.display.flip()
```

1 usage

#### 4. Wynik działania po wciśnięciu 1:

 Homework



## **5. Wnioski:**

Dzięki użyciu pygame, znając odpowiednie metody możemy stworzyć wielokąty o dowolnej ilości wierzchołków a ponadto dokonywać różnorodnych przekształceń: obrócenie, przesunięcie, pochylenie.

## **Zadanie 2**

### **1. Polecenie:**

Stworzyć odpowiednią grafikę z pokazanych.

### **2. Wprowadzane dane:**

Do zadania wykorzystałem informację od prowadzącego na temat tego którą grafikę mamy zaimplementować.

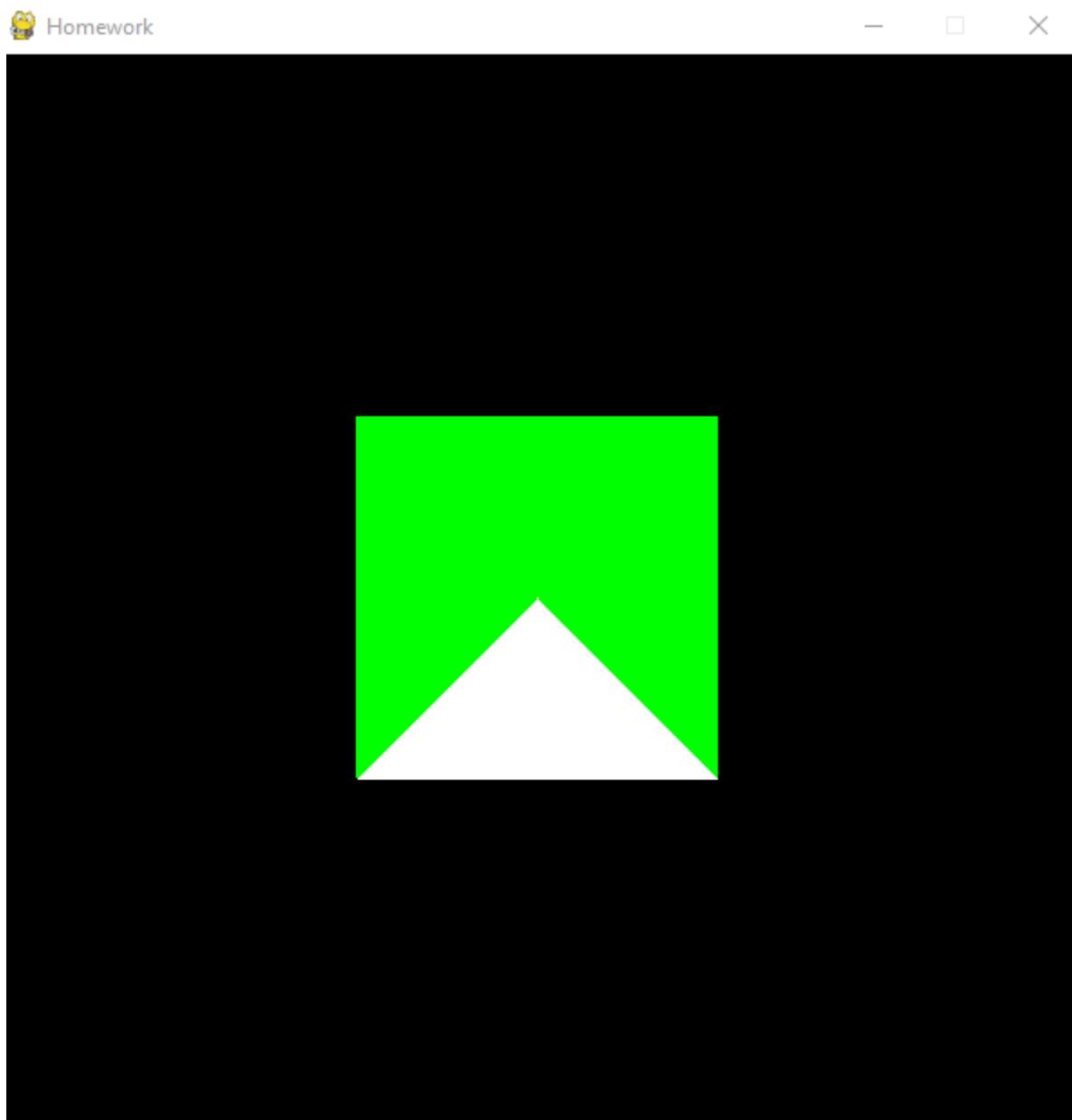
### **3. Wykorzystane komendy:**

Do wykonania zadania musieliśmy dodać grafikę, którą dodałem jako default case do 1 zadania.

Kod potrzebny do dodania w funkcji „handle\_key\_event” aby dodać nowy kolor i kształty:

```
case _:  
    pygame.draw.rect(win, ZIELONY, rect: (200, 200, 200, 200))  
    pygame.draw.polygon(win, BIALY, points: [(201, 400), (399, 400), (300, 300)])
```

#### 4. Wynik działania:



## **5.Wnioski:**

Dzięki użyciu stworzonych wcześniej funkcji, możemy zmodyfikować kod i dodać kolejny bez większych trudności przykładowo dodać nowy kształt lub kolor.