

Przekształcenia 2D w bibliotece **pygame**

February 27, 2024

1 Introduction

Celem jest użycie biblioteki pygame.

Co to jest Pygame?

Pygame - to stworzona przez Pete Shinnersa biblioteka przeznaczona do tworzenia gier komputerowych oraz aplikacji multimedialnych w języku Python. Pygame daje nam do dyspozycji GUI (Graficzny interfejs użytkownika, interfejs graficzny, środowisko graficzne, GUI (ang. graphical user interface)) który umożliwia na wyświetlanie grafiki, odtwarzanie dźwięków, śledzenie czasu, obsługę myszy i joysticka, obsługę CD, czy renderowanie czcionek TTF.

2 Instalacja

Instalacja jest prosta i wymaga jedynie wywołania komendy do instalacji w pythonie.

```
pip install pygame
```

3 Tworzenie okna gry

Podstawa każdej gry jest okienko w którym będzie się wszystko działo, aby utworzyć takie okno gry w pygame musimy najpierw zaimportować bibliotekę pygame. Następnie wykorzystać `pygame.init()` która służy do zainicjowania zaimportowanych modułów gry z naszej biblioteki. Wreszcie definiujemy nasze okno gry o wymiarach 500 x 500 dzięki `pygame.display.set_mode((500, 500))` i na koniec wyświetlamy je za pomocą `pygame.display.set_caption("Moja Gra")` gdzie w nawiasie możemy ustawić nazwę naszego okienka, w tym wypadku nazwaliśmy go "Moja gra".

```
import pygame
```

```
pygame.init()
```

```
# definiowanie okna gry
```

```
win = pygame.display.set_mode((500, 500))
```

```
# wyświetlenie okna gry
pygame.display.set_caption("Moja_Gra")
```

Jednak okno wyświetla nam się tylko na chwilę. Naprawimy to używając tak zwanej petli głównej gry która będzie podtrzymywać nasze okno

```
run = True
# petla główna
while run:
    # obsługa zdarzeń
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
```

Petla while spowoduje, że nasze okno nie zamyka się, ale jednak chcemy, by się zamknęło kiedy my tego chcemy. Najlepiej, żeby zamykało się wtedy kiedy klikniemy na X w prawym górnym rogu. Użyjemy do tego celu obsługi zdarzeń. Dzięki petli `for event in pygame.event.get()` odczytujemy kolejke wszystkich zdarzeń z klawiatury lub myszy wysyłanych między aplikacją pygame a użytkownikiem. Ustawiając warunek `if event.type == pygame.QUIT`: sprawdzamy czy zdarzenie `event.type` równa się `QUIT` czyli naciśnięcie na X w oknie. Jeśli warunek zostanie spełniony to petla główna przestanie działać i okno się zamknie.

Cały kod

```
import pygame
```

```
pygame.init()
# definiowanie okna gry
win = pygame.display.set_mode((500, 500))
# wyświetlanie okna gry
pygame.display.set_caption("Moja_Gra")
```

```
run = True
# petla główna
while run:
    # obsługa zdarzeń
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
```

4 Tworzenie rysunków

Tworząc gry musimy tworzyć różne obiekty jak również teksty informujące o ilości zdobywanych pkt lub opisujący menu, dlatego biblioteka pygame, oferuje prosty sposób tworzenia różnych figur oraz umożliwia wypisywanie tekstu w oknie.

Aby narysować prostokąt definiujemy jego kolor, wymiary oraz położenie w którym chcemy go narysować. Następnie w petli umieścimy metode `pygame.draw.rect()` która narysuje prostokąt w oknie, jednak nie zobaczymy go po uruchomieniu programu dopóki na końcu nie dodamy `pygame.display.update()` która odpowiada za odświeżenie ekranu naszego okna.

```
#definiujemy zmienne
KOLOR = (255, 0, 0)
x = 10
y = 30
szer = 100
wys = 100
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    #funkcja rysujaca kwadrat
    pygame.draw.rect(win, KOLOR, (x, y, szer, wys))
    #odświeżenie ekranu
    pygame.display.update()
```

5 Zadania

1. Pokazany jest obraz `shuttle.jpg` w panelu. Narysować zamiast obrazu wielokąt według wariantu (liczba n). Okno ma wymiary 600 na 600 pikseli, a wielokąt ma promień 150 pikseli. Kolejne zadanie polega na stosowaniu odpowiednich przekształceń do wielokąta (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 sa trzy podstawowe kształty: koło, kwadrat, trójkąt.
Podstawowe przekształcenia dostępne sa przez `pygame.transform`

6 References

1. <https://www.pygame.org/docs/>

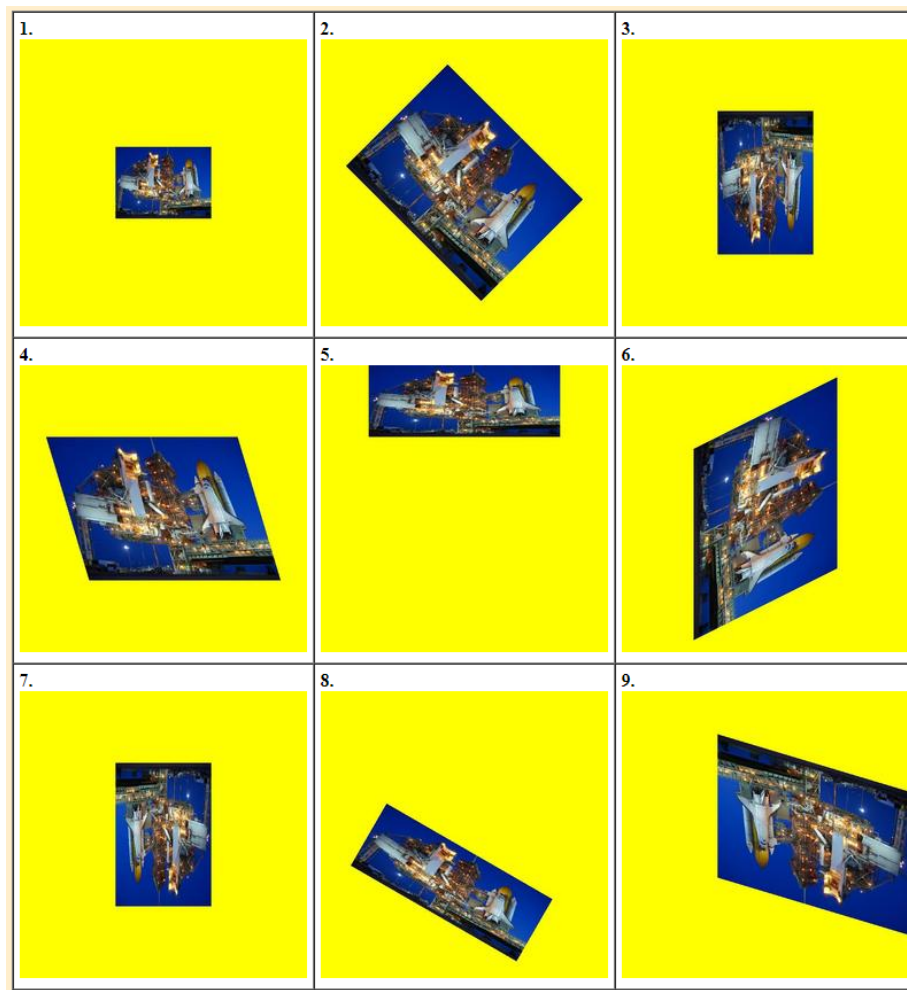


Figure 1: Zadanie 1

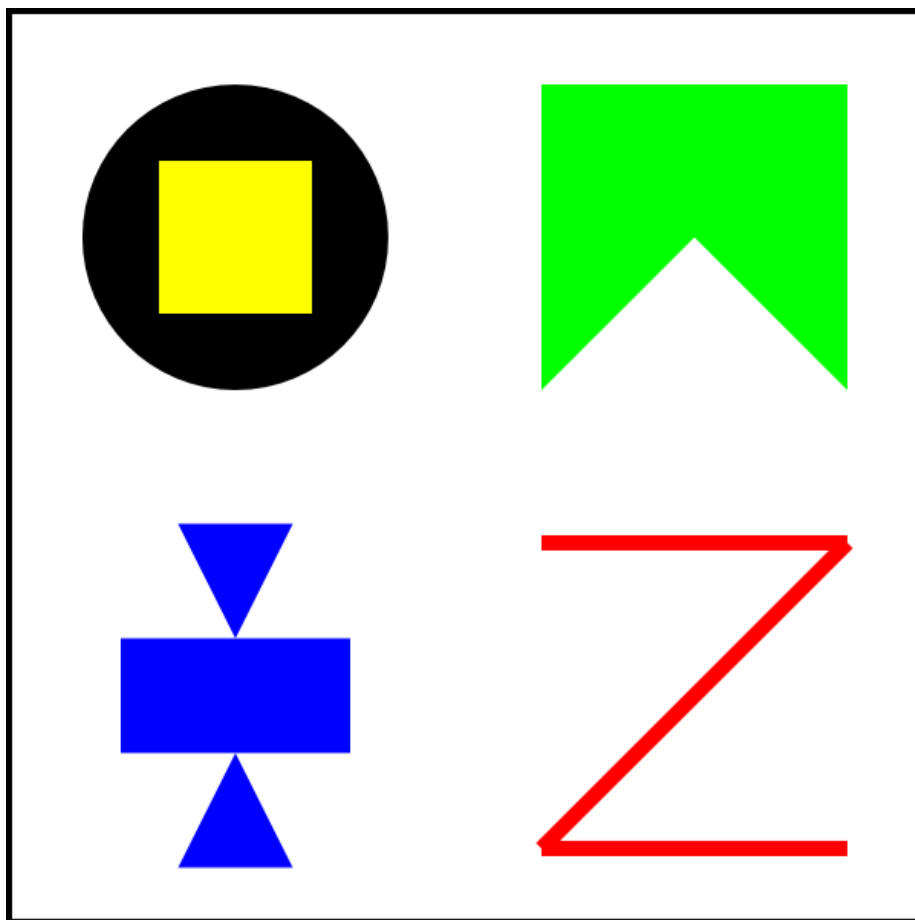


Figure 2: Zadanie 2