

```
33 #import os #importuje moduł os, który pozwala pracować z folderami i plikami
34 #os.chdir(os.path.dirname(__file__)) #zmienia bieżący folder roboczy programu na folder, w
    którym znajduje się ten plik .py
35 import pygame #importuje bibliotekę pygame, żeby można było jej używać
36 pygame.init() #uruchamia wszystkie potrzebne moduły pygame (np. ekran, dźwięk, itd.)
37
38 SCREEN_WIDTH = 800 #szerokość ekranu w pikselach
39 SCREEN_HEIGHT = 600 #wysokość ekranu w pikselach
40
41 screen_surface = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT)) #tworzy okno gry o
    podanych rozmiarach
42 pygame.display.set_caption("Pierwsza gra") #ustawia tytuł okna gry (pokazuje się na górnym
    pasku okna)
43
44 clock = pygame.time.Clock() #Tworzysz zegar, który pozwoli Ci ustawić ile klatek na sekundę
    będzie miała Twoja gra (czyli jak szybko będzie działać)
45
46
47 def load_image(img_path: str, position):
48     image = pygame.image.load(img_path) # wczytanie obrazka z pliku
49     surface = image.convert() # konwersja do formatu pygame
50
51     transparent_color = (0,0,0) # ustawia kolor czarny (0, 0, 0) jako przezroczysty
52     surface.set_colorkey(transparent_color) # tło naszego obrazka jest czarne
53
54     rect = surface.get_rect(center = position) # tworzy prostokąt (rect) opisujący pozycję
    obrazka na ekranie i ustawia środek obrazka na podanej pozycji
55     return [image, surface, rect] # zwraca: image - oryginalny obraz; surface - gotowy do
    wyświetlenia; rect - pozycja i rozmiar
56
57 player_pos = [SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2] # tworzy wyśrodkowanego gracza na
    ekranie
58 player = load_image(r'C:\\Users\\apla\\Desktop\\Projekty Python\\Lekcja 12 - pygame -
    pierwsza aplikacja\\player.png', player_pos)
59
60
61 def print_image(img_list) -> None:
62     #[image, surface, rect]
63     image, surface, rect = img_list # rozpakowuje listę z funkcji load_image() - dostaje
    obraz, powierzchnię i prostokąt (czyli pozycję)
64     screen_surface.blit(surface, rect) # rysuje (blit) obrazek (surface) na ekranie
    (screen_surface) w miejscu opisanym przez rect
65     pass
66
67
68
69
70
71
72
73
74
75
76
```

```
77 game_status = True #zmienna, która mówi, czy gra ma dalej działać. Dopóki jest True, gra
   trwa
78
79 while game_status: #pętla gry
80     events = pygame.event.get() #pobiera wszystkie zdarzenia, np. naciśnięcie klawisza,
   ruch myszy, zamknięcie okna
81
82     for event in events: #przegląd każdego zdarzenia
83         print(event) #wypisuje zdarzenie w konsoli
84         if event.type == pygame.QUIT: #sprawdza, czy gracz kliknął "X" żeby zamknąć grę
85             game_status = False #wyłącza pętlę - kończy grę
86             pass
87
88     print_image(player) # rysuje gracza na ekranie
89
90     pygame.display.update() #Odświeża ekran gry - wszystko, co zostało narysowane, pojawia
   się na ekranie
91     clock.tick(60) #Sprawia, że gra działa maksymalnie z prędkością 60 klatek na sekundę -
   nie za szybko.
92     pass
93
94 pygame.quit() #wyłącza wszystkie moduły pygame (np. ekran, dźwięk)
95 quit() #kończy działanie programu Pythona
```