

```
1 import os
2 os.chdir(os.path.dirname(__file__))
3 #CEL: zbieranie bonusów z ekranu
4
5 import pygame
6 from random import choice, randint
7 pygame.init()
8
9 SCREEN_WIDTH = 800
10 SCREEN_HEIGHT = 600
11
12 screen_surface = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
13 pygame.display.set_caption("Pierwsza gra")
14
15 #~~~~~
16 #  FUNCKJE DO OBRAZÓW ↓
17 #~~~~~
18 def load_image(img_path: str, position):
19     image = pygame.image.load(img_path)
20     surface = image.convert()
21
22     transparent_color = (0,0,0)
23     surface.set_colorkey(transparent_color)
24
25     rect = surface.get_rect(center = position)
26
27     return [image, surface, rect]
28
29 def print_image(img_list) -> None:
30     image, surface, rect = img_list
31     screen_surface.blit(surface, rect)
32     pass
33
34 #~~~~~
35 #  PORUSZANIE GRACZA ↓
36 #~~~~~
37 def set_position_image(img_list, position):
38     image, surface, rect = img_list
39     rect = surface.get_rect(center = position)
40     return [image, surface, rect]
41
42 def calculate_player_movement(keys):
43     speed = 10
44     delta_x = 0
45     delta_y = 0
46
47     if keys[pygame.K_LSHIFT]:
48         speed *= 2
49     if keys[pygame.K_w]:
50         delta_y -= speed
51     if keys[pygame.K_s]:
52         delta_y += speed
53     if keys[pygame.K_d]:
54         delta_x += speed
55     if keys[pygame.K_a]:
56         delta_x -= speed
57     return [delta_x, delta_y]
58
59 def limit_position(position):
60     x, y = position
61     x = max(0, min(x, SCREEN_WIDTH))
62     y = max(0, min(y, SCREEN_HEIGHT))
63     return [x, y]
64
65
66
67
68
```

```

69 #~~~~~
70 #  OBIEKTY BONUSOWE ↓
71 #~~~~~
72 # ♦ Utworzenie listy grafik bonusowych
73 # ♦ Utworzenie obiektów bonusowych - obiekty gotowe do wstawienia [image, surface, rect]
74
75 # ♦ FUNKCJA DO PRZYGOTOWANIA OBIEKTÓW:
76 # ♦ dodanie na samej górze: "from random import choice, randint"
77 def generate_bonus_object():
78     # Losuje nazwę obrazka bonusowego z listy grafik
79     image_name = choice(bonus_images)
80
81     # Losuje współrzędne
82     x = randint(0, SCREEN_WIDTH)
83     y = randint(0, SCREEN_HEIGHT)
84     # Tworzy listę z wylosowaną pozycją
85     position = [x, y]
86     # Tworzy gotowy obiekt z [image, surface, rect]
87     new_obj = load_image(image_name, position)
88
89     # Dodaje nowo utworzony obiekt bonusowy do listy 'bonus_objects', która przechowuje wszystkie
90     # aktywne bonusy w grze (czyli te, które mają być wyświetlone i mogą zostać zebrane).
91     bonus_objects.append(new_obj)
92     pass
93
94 # ♦ FUNKCJA WYŚWIETLAJĄCA OBIEKTY BONUSOWE
95 def print_bonus_objects():
96     for obj in bonus_objects:
97         print_image(obj)
98         pass
99     pass
100
101 #-----TESTY-----
102 # ♦ stworzenie zmiennej FRAMES_PER_SECOND = 60
103 # ♦ podmiana liczby klatek w clock.tick(FRAMES_PER_SECOND)
104 # ♦ przygotowanie zmiennej 'frames_cnt = 0', która zlicza wyświetlane klatki
105 # ♦ zliczamy klatki po 'pygame.display.update()', 'frames_cnt += 1'
106 # ♦ zmiana formy generowania bonusów na tą z 'if..'
107
108 #-----TESTY-----
109
110 # ♦ USUWANIE OBIEKTÓW BONUSOWYCH PO DOTKNIĘCIU
111 def check_collisions():
112     # Pobierz prostokąt (rect) gracza z listy 'player'; element [2] to właśnie rect.
113     rect_player = player[2]
114
115     # Iteruj po indeksach od końca listy 'bonus_objects' do początku.
116     # To ważne, ponieważ usuwanie elementów od końca nie psuje indeksowania listy.
117     for i in range(len(bonus_objects)):
118         index = len(bonus_objects) - i - 1 # Przeliczanie indeksu na odwrotną kolejność
119
120         # Pobierz obiekt bonusowy o wyliczonym indeksie.
121         obj = bonus_objects[index]
122
123         # Pobierz prostokąt (rect) tego obiektu – używany do detekcji kolizji.
124         rect = obj[2]
125
126         # Sprawdź, czy prostokąt gracza koliduje z prostokątem bonusu.
127         # Funkcja 'colliderect()' zwraca True, jeśli prostokąty się nachodzą.
128         if rect.colliderect(rect_player):
129             # Jeśli doszło do kolizji, usuń dany obiekt bonusowy z listy.
130             # Dzięki temu nie będzie już rysowany ani wykrywany.
131             bonus_objects.pop(index)
132             pass
133         pass
134     pass
135
136 # ♦ przed 'print_bonus_objects()' dodać 'check_collisions()'
137

```

```
138 #~~~~~
139 #   PĘTLA GRY ↓
140 #~~~~~
141 player_pos = [SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2]
142 player = load_image(r'C:\Users\apla\Desktop\Projekty Python\Lekcja 12 - pygame - pierwsza
143 aplikacja\grafiki\player.png', player_pos)
144 background_color = [9, 42, 121]
145
146 FRAMES_PER_SECOND = 60
147 frames_cnt = 0
148
149 clock = pygame.time.Clock()
150
151 #Lista grafik obiektów bonusowych
152 bonus_images = [
153     r'grafiki\bonus_1.png',
154     r'grafiki\bonus_2.png',
155     r'grafiki\bonus_3.png'
156 ]
157
158 bonus_objects = [] #gotowy obiekt to 3-elementowa lista z funkcji load_image
159
160 game_status = True
161
162 while game_status:
163     events = pygame.event.get()
164     for event in events:
165         #print(event)
166         if event.type == pygame.QUIT:
167             game_status = False
168             pass
169
170     #-----PODSTAWA-----
171     pressed_keys = pygame.key.get_pressed()
172
173     delta_x, delta_y = calculate_player_movement(pressed_keys)
174     player_pos[0] += delta_x
175     player_pos[1] += delta_y
176
177     player_pos = limit_position(player_pos)
178     player = set_position_image(player, player_pos)
179
180     screen_surface.fill(background_color)
181     #-----
182     print_image(player)
183
184     #-----ROZSZERZENIE-----
185     # generate_bonus_object()
186     # print_bonus_objects()
187     #TESTY
188     if frames_cnt % (FRAMES_PER_SECOND * 1) == 0: #mnożnik mówi co ile sekund generować
189         generate_bonus_object()
190         pass
191     check_collisions()
192     print_bonus_objects()
193     #-----
194
195     pygame.display.update()
196     frames_cnt += 1
197
198     clock.tick(FRAMES_PER_SECOND)
199     pass
200
201 pygame.quit()
202 quit()
```