

---

# **Gra w życie S30517**

***Wydanie none***

**Sebastian Wilusz**

**11 cze 2025**



---

## Contents:

---



Add your content using reStructuredText syntax. See the [reStructuredText](#) documentation for details.



# ROZDZIAŁ 1

---

## config module

---

config.py

Zawiera globalne ustawienia konfiguracyjne dla aplikacji „Gra w życie Conwaya”. Stałe te są używane w różnych modułach gry do kontrolowania wyglądu, rozmiarów i wydajności.





**class** game.GameOfLife

Klasy bazowe: object

Klasa reprezentująca logikę gry w życie Conwaya. Odpowiada za aktualizację siatki, rysowanie komórek i obsługę ich stanu.

**clear()**

Czyści siatkę, ustawiając wszystkie komórki jako martwe.

**count\_neighbors(x, y)**

Zlicza żywych sąsiadów komórki (z uwzględnieniem otaczającego torusa).

**Parametry**

- **x** – Pozycja X komórki
- **y** – Pozycja Y komórki

**Zwraca**

Liczba żywych sąsiadów

**draw(screen)**

Rysuje żywe komórki oraz siatkę na ekranie.

**Parametry**

**screen** – Obiekt ekranu Pygame

**toggle\_cell(x, y)**

Zmienia stan komórki (żywa/martwa) na podanej pozycji, jeśli gra jest zatrzymana.

**Parametry**

- **x** – Pozycja X komórki
- **y** – Pozycja Y komórki

**update()**

Aktualizuje siatkę zgodnie z zasadami gry w życie: - Komórka żywa z 2 lub 3 sąsiadami przeżywa. - Martwa komórka z dokładnie 3 sąsiadami ożywa.

```
class gui.Button(text, pos, size, font_size=30)
```

Klasy bazowe: object

Klasa reprezentująca przycisk interfejsu użytkownika.

**Parametry**

- **text** (str) – Tekst wyświetlany na przycisku.
- **pos** (tuple[int, int]) – Pozycja lewego górnego rogu przycisku w formacie (x, y).
- **size** (tuple[int, int]) – Rozmiar przycisku w formacie (szerokość, wysokość).
- **font\_size** (int) – Rozmiar czcionki tekstu przycisku, domyślnie 30.

```
draw(screen)
```

Rysuje przycisk na podanym ekranie.

**Parametry**

**screen** (pygame.Surface) – Obiekt ekranu (powierzchnia Pygame), na którym ma być narysowany przycisk.

```
is_clicked(mouse_pos)
```

Sprawdza, czy mysz kliknęła przycisk.

**Parametry**

**mouse\_pos** (tuple[int, int]) – Pozycja kliknięcia myszy.

**Zwraca**

True jeśli kliknięto na przycisk, False w przeciwnym wypadku.

**Typ zwracany**

bool

```
is_hovered(mouse_pos)
```

Sprawdza, czy mysz znajduje się nad przyciskiem.

**Parametry**

**mouse\_pos** (*tuple[int, int]*) – Aktualna pozycja myszy w formacie (x, y).

`gui.pause_menu(screen)`

Wyświetla menu pauzy z opcjami: wznowienia gry, ustawień, powrotu do menu i wyjścia.

**Parametry**

**screen** (*pygame.Surface*) – Ekran, na którym renderowane jest menu.

**Zwraca**

Jedna z wartości: «resume», «settings», «menu», «quit».

**Typ zwracany**

str

`gui.settings_menu(screen)`

Wyświetla menu ustawień, w którym można zmieniać FPS i kolor komórek.

**Parametry**

**screen** (*pygame.Surface*) – Ekran, na którym renderowane jest menu.

**Zwraca**

«saved» jeśli zapisano ustawienia, «cancel» jeśli anulowano.

**Typ zwracany**

str

`gui.start_screen(screen)`

Wyświetla ekran startowy gry z opcją rozpoczęcia lub zakończenia gry.

**Parametry**

**screen** (*pygame.Surface*) – Ekran, na którym renderowany jest interfejs.

**Zwraca**

«start» lub «quit» w zależności od wyboru użytkownika.

**Typ zwracany**

str

`main.create_screen()`

Tworzy i zwraca powierzchnię Pygame o rozmiarze określonym w konfiguracji.

**Zwraca**

Obiekt `pygame.Surface` reprezentujący okno gry.

`main.main()`

Główna funkcja programu zarządzająca ekranem startowym oraz uruchamianiem gry. Po wybraniu opcji „start” uruchamia pętlę gry.

`main.run_game(game)`

Uruchamia główną pętlę gry, obsługując zdarzenia, rysowanie i pauzę.

**Parametry**

**game** – Instancja klasy `GameOfLife` zarządzająca logiką gry.



### c

config, ??

### g

game, ??

gui, ??

### m

main, ??