GameOfLife

Version

Table of Contents

Contents:

GameOfLife documentation

Projekt “Game of Life” to gra symulacyjna oparta na automacie komórkowym zaproponowanym przez Johna Conwaya. Aplikacja została napisana w języku Python z wykorzystaniem biblioteki Tkinter do tworzenia graficznego interfejsu użytkownika.

Funkcje projektu:

- Interfejs graficzny z siatką komórek, które można aktywować kliknięciem myszki.
- Obsługa podstawowych zasad gry (np. narodziny, śmierć i przetrwanie komórek).
- Możliwość definiowania własnych reguł gry.
- Zapisywanie i wczytywanie szablonów z katalogu *resources/templates* .
- Okno startowe umożliwiające wybór ustawień gry.
- Osobne okno do podglądu i wyboru zapisanych szablonów.
- Możliwość pauzowania i wznawiania symulacji.
- Czysty podział logiki gry i interfejsu użytkownika.

Struktura katalogów:

- *logic/* – logika gry (np. zarządzanie siatką i regułami).
- *gui/* – interfejs graficzny użytkownika.
- *resources/templates/* – zapisane szablony wzorów początkowych.
- *main.py* – punkt wejścia do aplikacji.

Wymagania:

- Python 3.9 lub nowszy
- Biblioteka standardowa (Tkinter)

Instrukcja uruchomienia:

Aby uruchomić aplikację, wykonaj w terminalu:

```
python main.py
```

—

GameOfLife package

Subpackages

GameOfLife.gui namespace

Submodules

GameOfLife.gui.main_window module

class GameOfLife.gui.main_window. MainWindow (*master* , *width* , *height* , *rules* , *initial_grid = None*) [source]

Bases: `object`

Główne okno gry – wyświetla planszę, kontrolki i obsługuje grę.

_build_interface (*width* , *height*) [source]

Buduje cały interfejs użytkownika.

_create_buttons_section ( ) [source]

Tworzy sekcję przycisków sterujących grą.

_create_exit_section ( ) [source]

Tworzy sekcję z szablonami i powrotem do startu.

_create_rules_section ( ) [source]

Tworzy sekcję zasad gry.

_create_template_section ( ) [source]
Tworzy sekcję z szablonami i powrotem do startu.

apply_rules ( ) [source]
Zastosowuje zasady z panelu.

back_to_start ( ) [source]
Przechodzi do okna startowego.

centred_window ( ) [source]
on_mouse_down (*event*) [source]
Określa tryb przeciągania na podstawie kliknięcia.

on_mouse_drag (*event*) [source]
Przeciąganie myszą w celu aktywacji/dezaktywacji komórek.

randomize ( ) [source]
Losuje układ komórek.

randomize_rules ( ) [source]
Losuje zasady gry.

reset ( ) [source]
Czyści planszę.

run_loop ( ) [source]
Pętla animacji.

save_template ( ) [source]
Zapisuje aktualny szablon.

step ( ) [source]
Wykonuje jeden krok symulacji.

toggle_cell (*y* , *x*) [source]
Przełącza stan komórki.

toggle_run ( ) [source]
Startuje lub zatrzymuje animację.

update_canvas ( ) [source]
Aktualizuje kolory komórek na podstawie stanu planszy.

update_delay (*value*) [source]
Aktualizuje opóźnienie animacji na podstawie suwaka.

update_rules_entry ( ) [source]
Aktualizuje pola wejściowe zasad.

GameOfLife.gui.start_window module

class GameOfLife.gui.start_window. StartWindow (*master*) [source]

Bases: `object`

Okno startowe do konfiguracji początkowych parametrów gry.

_build_buttons ( ) [source]

Tworzy przyciski sterujące aplikacją.

_build_rules ( ) [source]

Tworzy interfejs do konfiguracji zasad gry.

_build_rules_board ( ) [source]

Tworzy interfejs do konfiguracji planszy gry.

open_main_window_with_template (*template*) [source]

Zamknij StartWindow i otwórz MainWindow z wybranym szablonem.

open_templates ( ) [source]

Otwiera okno wyboru szablonu i ustawia dane w polach.

parse_rules ( ) [source]

Parsuje tekst z pola zasad do obiektu Rules.

set_random_rules ( ) [source]

Ustawia losowe zasady gry w polach tekstowych.

start_game ( ) [source]

Uruchamia główne okno gry z podanymi parametrami.

GameOfLife.gui.templates_window module

class GameOfLife.gui.templates_window. TemplateSelectorWindow (*master* , *on_template_selected*) [source]

Bases: `object`

_build_interface ( ) [source]

Tworzy GUI.

_clear_preview ( ) [source]

_update_preview (*template*) [source]

Aktualizuje podgląd zasad i planszy.

confirm_selection ( ) [source]
Zatwierdza wybór szablonu i otwiera główne okno gry z szablonem.

delete_template ( ) [source]
Usuwa wybrany szablon po potwierdzeniu.

on_select (*event*) [source]
Obsługuje wybór szablonu z listy.

GameOfLife.logic namespace

Submodules

GameOfLife.logic.board module

class GameOfLife.logic.board. Board (*rows* , *cols* , *initial_grid*) [source]
Bases: `object`

Klasa reprezentująca planszę gry i jej stan. Przechowuje komórki i wiek komórek.

get_neighbors (*y* , *x*) [source]
Zwraca liczbę żywych sąsiadów komórki. :param y: wiersz :param x: kolumna :return: liczba sąsiadów

randomize (*prob = 0.2*) [source]
Losowo wypełnia planszę komórkami. :param prob: prawdopodobieństwo żywej komórki

reset ( ) [source]
Resetuje planszę do stanu początkowego.

step (*rules*) [source]
Wykonuje jeden krok symulacji. :param rules: zasady gry

toggle_cell (*y* , *x*) [source]
Zamienia stan komórki na przeciwny. :param y: wiersz :param x: kolumna

GameOfLife.logic.game_state module

class GameOfLife.logic.game_state. GameState (*rows* , *cols* , *rules* , *initial_grid*) [source]
Bases: `object`

Klasa łącząca planszę gry i zasady. Przechowuje bieżący stan gry.

apply_rules (*rules*) [source]
Zastępuje zasady gry nowymi.

randomize (*prob = 0.2*) [source]
Losowo ustawia komórki na planszy.

reset ( ) [source]
Resetuje planszę gry.

step ( ) [source]
Wykonuje jeden krok symulacji.

toggle_cell (*y* , *x*) [source]
Przełącza stan komórki.

GameOfLife.logic.rules module

class GameOfLife.logic.rules. Rules (*birth* , *survive* , *age*) [source]
Bases: `object`

Klasa reprezentująca zasady gry w życie. Określa warunki przetrwania i narodzin komórek.

static generate_random ( ) [source]
Generuje losowe zasady gry. :return: instancja Rules z losowymi wartościami

should_live (*alive* , *neighbors*) [source]
Określa, czy komórka powinna być żywa w następnym kroku. :param alive: czy komórka jest aktualnie żywa :param neighbors: liczba żywych sąsiadów :return: True jeśli komórka ma być żywa, False w przeciwnym razie

GameOfLife.logic.template module

class GameOfLife.logic.template. Template (*name* , *grid* , *rules*) [source]
Bases: `object`

Klasa reprezentująca szablon (template) gry. Zawiera rozmiar, stan planszy i zasady.

static from_dict (*data*) [source]
Tworzy instancję szablonu z danych słownikowych.

static load_templates (*directory = './resources/templates'*) [source]

Wczytuje wszystkie szablony z plików JSON w folderze. Zwraca listę krotek: (nazwa_szablonu, Template)

static save_template (*template* , *directory = './resources/templates/'*) [source]

Zapisuje szablon do foldera ‘templates’. Jeśli plik już istnieje, zgłasza wyjątek.

to_dict ( ) [source]

Zwraca dane szablonu jako słownik.

Submodules

GameOfLife.main module

GameOfLife.main. main ( ) [source]

Plik main uruchamia cały program.