

### Spis treści

1. Wstęp
  2. Sposób instalacji aplikacji i zależności
  3. Zasady działania programu
  4. Struktura kodu i opis klas
  5. Dokumentacja metod i ich zastosowanie
  6. Sposób działania interakcji użytkownika
- 

### 1. Wstęp

Symulacja Liszaja to interaktywny program napisany w języku C++, który modeluje rozprzestrzenianie się infekcji w dwuwymiarowej siatce komórek. Program wykorzystuje bibliotekę SFML do wizualizacji i pozwala użytkownikowi analizować dynamikę zmian stanów komórek w czasie.

Główne cele aplikacji:

- Modelowanie rozprzestrzeniania się infekcji w populacji.
  - Możliwość dostosowywania parametrów symulacji, takich jak:
    - Rozmiar siatki (liczba komórek).
    - Czas trwania jednostki symulacji (w milisekundach).
    - Czas trwania odporności komórek.
  - Wizualizacja wyników w czasie rzeczywistym.
- 

### 2. Sposób instalacji aplikacji i zależności

Wymagania systemowe

- System operacyjny: Windows, Linux, MacOS.
- Kompilator obsługujący C++11 lub nowszy (np. GCC, Clang, MSVC).
- Biblioteka SFML (Simple and Fast Multimedia Library).

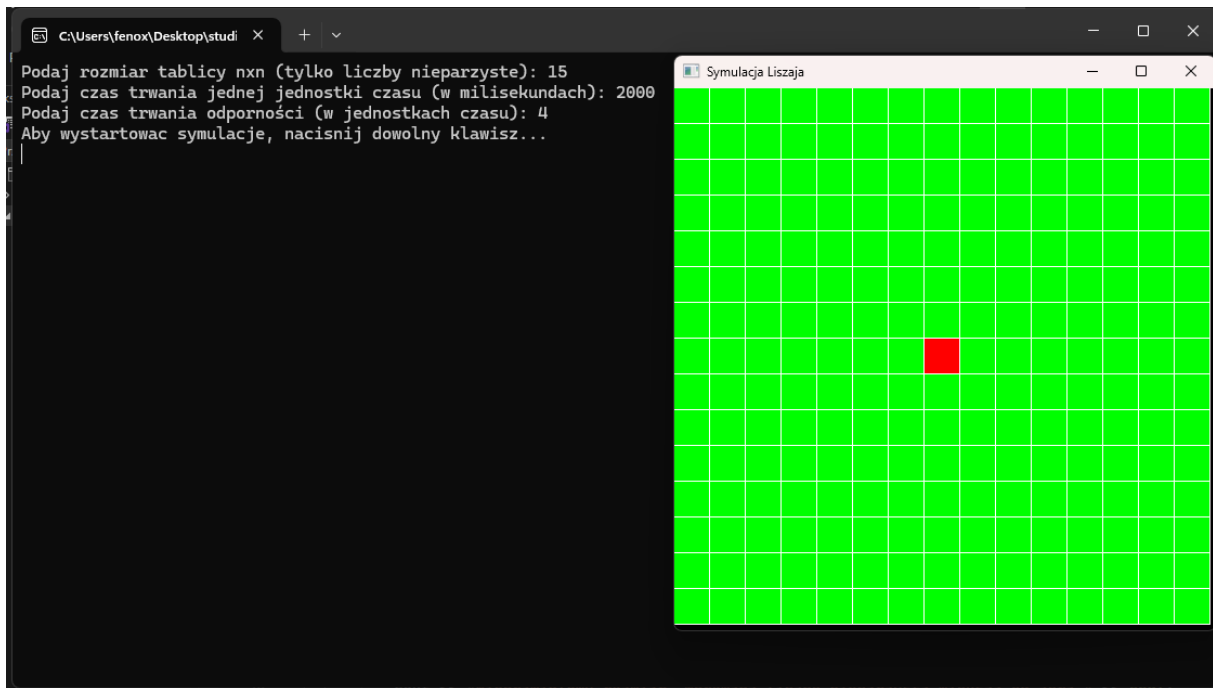
Instrukcja instalacji

1. Pobranie kodu źródłowego: Umieść pliki projektu w jednym katalogu.
2. Instalacja biblioteki SFML:
  - Na Linux (Ubuntu): należy użyć polecenia `sudo apt-get install libsFML-dev`.

- Na Windows: należy pobrać paczkę SFML ze strony <https://www.sfml-dev.org> i skonfigurować środowisko IDE.
- 3. Kompilacja programu: Przykładowa komenda dla systemów Unix/Linux:  
`g++ main.cpp Komórka.cpp Siatka.cpp Symulacja.cpp Renderowanie.cpp -o SymulacjaLiszaja -lsfml-graphics -lsfml-window -lsfml-system`
- 4. Uruchomienie programu: W systemie Linux/MacOS użyj komendy `./SymulacjaLiszaja`.

### 3. Zasady działania programu





Po uruchomieniu programu widać następujące okna:



Należy podać parametry, o które prosi konsola. Następnie, aby wystartować należy kliknąć dowolny klawisz (należy pamiętać, aby okno symulacji było in Focus).

Model rozprzestrzeniania się infekcji

- Każda komórka może znajdować się w jednym z czterech stanów:

- Zdrowa (zielona). 
- Zarażona (czerwona). 
- Odporna (niebieska). 
- Martwa (czarna). 

- Zarażona komórka ma 50% szans na zainfekowanie sąsiednich komórek (góra, dół, lewo, prawo).
- Po 6 jednostkach czasu komórka przechodzi w stan Odporna.
- Komórka Odporna wraca do stanu Zdrowa po określonym czasie.
- 5% szans, że zarażona komórka umrze zamiast stać się odporną.

Program renderuje aktualny stan i wyświetla statystyki.

Symulacja kończy się automatycznie, gdy nie ma już zmian w siatce przez określoną liczbę kroków.

## 4. Struktura kodu i opis klas

### 4.1. Klasa Komórka (Komórka.h, Komórka.cpp)

- Opis: Klasa reprezentująca pojedynczą komórkę.
- Zadania:
  - Przechowuje stan komórki i licznik czasu jej aktualnego stanu.
  - Obsługuje przechodzenie pomiędzy stanami.

### 4.2. Klasa Siatka (Siatka.h, Siatka.cpp)

- Opis: Klasa przechowująca dwuwymiarową siatkę komórek.
- Zadania:
  - Przechowywanie wszystkich komórek.
  - Obsługuje zarażanie sąsiadów.
  - Aktualizuje stan każdej komórki.

### 4.3. Klasa Symulacja (Symulacja.h, Symulacja.cpp)

- Opis: Klasa zarządzająca przebiegiem symulacji.
- Zadania:
  - Inicjalizuje początkowy stan siatki.
  - Zarządza kolejnymi krokami symulacji.
  - Zapisuje statystyki dotyczące stanu komórek.

### 4.4. Klasa Renderowanie (Renderowanie.h, Renderowanie.cpp)

- Opis: Klasa odpowiadająca za wyświetlanie siatki komórek w czasie rzeczywistym.
- Zadania:
  - Każdy stan komórki ma przypisany kolor.

- Rysowanie prostokątów reprezentujących komórki.

#### 4.5. Główna funkcja programu (main.cpp)

- Opis: Główna funkcja programu.
  - Zadania:
    - Obsługuje interakcję użytkownika (np. wybór parametrów symulacji).
    - Uruchamia pętlę symulacji i renderowania.
    - Wyświetla statystyki w terminalu.
- 

### 5. Dokumentacja metod i ich zastosowanie

#### 5.1. Klasa Komórka

- **pobierzStan()** – Zwraca aktualny stan komórki.
- **ustawStan(Stan nowyStan)** – Zmienia stan komórki na nowy.
- **aktualizuj()** – Aktualizuje licznik i zmienia stan komórki.
- **ustawCzasOdporności(int)** – Ustawia globalny czas trwania odporności.

#### 5.2. Klasa Siatka

- **pobierzRozmiar()** – Zwraca rozmiar siatki.
- **pobierzKomórkę(x, y)** – Zwraca referencję do komórki na pozycji (x, y).
- **zarazSąsiadów()** – Iteruje po siatce i zaraża sąsiadujące komórki.
- **aktualizuj()** – Aktualizuje stan każdej komórki w siatce.