Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 6

з курсу: «Проектування кібер-фізичних систем»

**Виконав:**студент 2-го курсу,  
групи ТВ-32  
Щур Роман Ігорович

 Посилання на GitHub репозиторій:

https://github.com/RomanSchur/DCPS\_PR-6

Київ 2025

**Завдання**

**Завдання**: Розробити модель епідемії або лісової пожежі за допомогою клітинних автоматів. Навчитися програмувати та візуалізувати динамічні процеси у складних системах.

**Програмна реалізація**

Мова програмування Python. Програма складається з восьми функцій (create\_area(), create\_line(), choose\_start\_tree(), update(), create\_visualization(), update\_visualization(), main(), update\_frame()), кожна з яких відповідає за один етап клітинного автомату.

Функція **create\_area**() створює список відповідно до отриманих характеристик, який представляє простір клітинного автомату, після створення функція перетворює список у NumPy масив.

Функція **create\_line**() відповідає за заповнення кожного рядка списку окремо, вона отримує кількість горілих і живих дерев, заповнює ними рядок і рандомно перемішує, використовуючи random.shuffle().

Функція **choose\_start\_tree**() використовується для вибору нульового дерева (дерева, з якого почнеться пожежа). Використовуючи random.randint, генерується номер рядка і стовбця, потім вибирається дерево, яке розташовується на перетині вибраного рядка і стовбця.

Функція **update**() відповідає за оновлення простору, за допомогою циклів перебирається кожен елемент (дерево) масиву, використовуючи значення, визначається, чи дерево горить, згоріло або живе. Якщо дерево горить, то час його горіння (його значення) зменшується, тим самим симулюється час горіння. Для живих дерев обираються дерева, які знаходяться поруч, перевіряється, чи є серед них дерева, які горять, і якщо такі є, за допомогою шансу підпалу визначається, чи дерево загориться.

Функція **create\_visualization**() відповідає за створення стартової фігури, яка відповідає за зображення початкового стану полотна, також використовується кольорова мапа, яка, в залежності від значення елементу полотна, позначає його тим чи іншим кольором.

Функція **update\_visualization**() відповідає за оновлення полотна та оновлення його графічного представлення.

Функція **main()** є головною функцією, тут задаються користувачем всі початкові дані полотна (розмір, початковий склад, кількість ітерацій та час горіння дерева, а також шанс підпалу дерева, якщо поруч знаходиться дерево, яке горить). Також тут, за допомогою вище згаданих функцій, створюється початкове полотно, визначається дерево, з якого починається пожежа, створюється початкова фігура. В цій функції отримується поточний стан полотна, обчислюється наступний стан за допомогою функції update\_visualization, оновлюються значення полотна та повертається оновлений об'єкт зображення для відображення наступного кадру анімації, яка створена в самому кінці main функції.

**Результати виконання**

Зображення, що містить символ, знімок екрана, Прямокутник, текст

Вміст, створений ШІ, може бути неправильним.

Рис №1 – Приклад початкового стану полотна

Зображення, що містить знімок екрана, монітор, Мультимедійне програмне забезпечення, квадрат

Вміст, створений ШІ, може бути неправильним.

Рис №2 – Приклад стану полотна після 20 кроків