**Комунальний заклад освіти**

**«Дніпровський науковий ліцей інформаційних технологій»**

**ВИПУСКНА РОБОТА**

**Тема:** Програмна реалізація алгоритмів пошуку шляхів на графах у тривимірному просторі.

**Виконав:**

Ліцеїст Класу 11-Б-2 Напольських Денис Сергійович

**Керівник:**

Ентін Й. А.

**ЗМІСТ**

[1. ВСТУП 3](#_Toc184580258)

[2. ОСНОВНА ЧАСТИНА 4](#_Toc184580259)

[2.1 Теоретична частина з предмету 4](#_Toc184580260)

[2.2 Теоретична частина з інформатики 4](#_Toc184580261)

[2.3 Опис роботи 4](#_Toc184580262)

[3. ВИСНОВКИ 4](#_Toc184580263)

[4. ДЖЕРЕЛА ІНФОРМАЦІЇ 4](#_Toc184580264)

# ВСТУП

Тема:  
Програмна реалізація алгоритмів пошуку шляхів на графах у тривимірному просторі.

Мета:  
Створити проект з програмною реалізацією алгоритму, який розв'язує дану задачу.

Актуальність теми:

Визначення шляхів між двома заданими точками є важливою задачею у багатьох наукових і технічних галузях, особливо в матеріалознавстві, фізиці твердого тіла та інженерії композитних матеріалів. У кристалах і композитах структура матеріалу може значно впливати на поведінку частинок, електронів або хвиль, що проходять крізь нього, і, відповідно, на фізичні та механічні властивості самого матеріалу. Розуміння шляхів між точками дозволяє дослідникам і інженерам моделювати процеси переносу тепла, електричної провідності, поширення звуку чи світла. У випадку композитів, які складаються з різних матеріалів, визначення шляху допомагає оптимізувати властивості, такі як міцність, жорсткість та стійкість до зношування. Крім того, визначення таких шляхів є необхідним у проектуванні складних комп'ютерних симуляцій та інструментів машинного навчання, що застосовуються у створенні нових матеріалів і пристроїв із наперед заданими властивостями.

# ОСНОВНА ЧАСТИНА

## Теоретична частина

Пошук шляху на тривимірній кубічній сітці – задача комп'ютерних наук, що має безліч застосувань, від планування руху роботів до аналізу біологічних молекул.

Суть задачі полягає у знаходженні найкоротшого або найефективнішого шляху між двома заданими точками в тривимірному просторі, який представлений у вигляді регулярної сітки.

На відміну від двовимірних графів, тривимірні сітки мають додатковий вимір, що ускладнює пошук шляху та/або збільшує витрати часу на його знаходження.

**Алгоритми пошуку шляху**

Існує багато алгоритмів для пошуку шляху на тривимірних сітках, кожен з яких має свої переваги та недоліки. Найбільш поширені з них:

* **Алгоритм Дейкстри:** Класичний алгоритм, який знаходить найкоротший шлях від однієї вершини до всіх інших. Він ефективний для графів без від'ємних циклів.
* *A*\* Евристичний алгоритм, який використовує додаткову інформацію про відстань до мети, щоб швидше знаходити найкоротший шлях.
* *D*\* Динамічний алгоритм, який дозволяє ефективно перераховувати шлях при зміні оточення.
* **Jump Point Search:** Оптимізований варіант алгоритму A\*, який дозволяє пропускати великі області сітки, що значно прискорює пошук.
* **Пошук у ширину (BFS)**: Починає пошук з кореневого вузла і досліджує всіх його сусідів на одному рівні, перш ніж перейти до наступного рівня. Цей процес продовжується, поки не буде знайдено цільовий вузол або не будуть досліджені всі досяжні вузли.
* **Пошук у глибину (DFS):** починає пошук з кореневого вузла і рухається по одній гілці графа якомога глибше, перш ніж повернутися назад і досліджувати інші гілки.

**Фактори, що впливають на вибір алгоритму:**

**Розмір сітки:** Для великих сіток більш ефективними будуть алгоритми, які використовують евристики або дозволяють пропускати частину сітки.

**Вимоги до точності:** Якщо потрібно знайти не обов'язково найкоротший, а просто прийнятний шлях, можна використовувати спрощені алгоритми.

Пошук шляху на тривимірних сітках має широкий спектр застосувань у робототехніці для планування руху роботів в тривимірному просторі з урахуванням перешкод, у комп’ютерних іграх для штучний інтелекту персонажів що мають рухатися в ігровому світі та в логістиці для оптимізації маршрутів доставки.

**Серед можливих напрямків:**

* **Розробка нових алгоритмів:** Створення більш ефективних алгоритмів для великих сіток і складних середовищ.
* **Паралельні обчислення:** Розробка паралельних алгоритмів для прискорення пошуку шляху на багатоядерних процесорах і графічних процесорах.

## Теоретична частина за інформатики

**Теоретична частина з інформатики**

Програма розроблена на мові програмування C#, що дозволяє легко працювати з об'єктно-орієнтованим підходом та інтегрувати сторонні бібліотеки. У розробці використано дві ключові бібліотеки:

**OpenTK** – це кросплатформна бібліотека для роботи з OpenGL, що забезпечує низькорівневий доступ до графічного апаратного забезпечення. Вона дозволяє створювати ефективну графіку завдяки використанню API OpenGL, а також включає підтримку роботи з вікнами, клавіатурою та мишею.

**ImGui.NET** – це обгортка для бібліотеки ImGui (Immediate Mode GUI) на C#. Вона забезпечує швидке створення графічного інтерфейсу користувача (GUI), де кожен кадр виконується безпосередньо, що спрощує інтерактивність і дозволяє динамічно змінювати інтерфейс під час роботи програми.

Елементи управління, їх властивості та методи:

У програмі було реалізовано такі основні елементи управління:

Button:

Властивості: текст, розмір.

Дає можливість змінювати значення типу int

InputInt:

Властивості: крок, поточне значення.

Дає можливість змінювати значення типу int введенням його з клавіатури або кнопками ⮝|⮟

InputDouble:

Властивості: крок, поточне значення.

Дає можливість змінювати значення типу double введенням його з клавіатури або кнопками ⮝|⮟

## Опис роботи

## Програмно-апаратні вимоги

* Роздільна здатність монітору: 1600 х 900
* Процесор: intel i5 4200 ( на слабших процесорах не перевіряв )
* Графічний процесор: з підтримкою OpenGL 3
* Оперативна пам’ять: 1 GB

## Комплектація програми

У комплектацію входять лише:

* .exe файл;
* cimgui.dll
* glfw3.dll

## Використані програмні засоби

Visual studio 2022 для написання коду;

ShaderToy для тестування шейдерів;

# ВИСНОВКИ

# ДЖЕРЕЛА ІНФОРМАЦІЇ

* [Determining electrical percolation threshold of randomly distributed conductor materials in polymer composites via pathfinding algorithms](https://www.sciencedirect.com/science/article/abs/pii/S0266353822001464)
* [ВІКІПЕДІЯ: Теорія протікання](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F_%D0%BF%D1%80%D0%BE%D1%82%D1%96%D0%BA%D0%B0%D0%BD%D0%BD%D1%8F)
* [ВІКІПЕДІЯ: Пошук шляху](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA_%D1%88%D0%BB%D1%8F%D1%85%D1%83)