Міністерство освіти і науки України

Національний університет імені Івана Франка

Факультет прикладної математики та інформатики

Кафедра програмування

КУРСОВА РОБОТА

на тему:

**«МЕТОДИ І АЛГОРИТМИ ПЕРЕСУВАННЯ ОБ’ЄКТІВ НА ПЛОЩИНІ»**

**Виконав:**

студент групи ПМІ-31 Довгань М.А.

**Науковий керівник:**

доцент кафедри програмування, кандидат фізико-математичних наук Черняхівський В.В.

Львів – 2020

ЗМІСТ

[ВСТУП 3](#_Toc41003058)

[Об’єктом дослідження 3](#_Toc41003059)

[Предметом дослідження 3](#_Toc41003060)

[Метою дослідження 4](#_Toc41003061)

[КЛАСИФІКАЦІЯ ЛАБІРИНТІВ 4](#_Toc41003062)

[Розмірність 4](#_Toc41003063)

[Топологія 5](#_Toc41003064)

[Теселяція 6](#_Toc41003065)

[Маршрутизація 6](#_Toc41003066)

[Текстура 7](#_Toc41003067)

[Фокус 8](#_Toc41003068)

[АЛГОРИТМИ СТВОРЕННЯ ЛАБІРИНТІВ 9](#_Toc41003069)

# ВСТУП

Пересування об’єктів на площині це доволі поширена задача з якою можна зіштовхнутися в найрізноманітніших сферах діяльності людини. Знаходження оптимальних алгоритмів для пересування по різноманітним поверхням при визначених умовах актуально для промисловості, проектування великих інфраструктурних об’єктів, задач будівництва, тощо. Якщо розглядати конкретно область інформаційних технологій, то ця задача виникає в таких сферах як комп’ютерна графіка, розробка інтерактивних ігор, симуляція різноманітних процесів.

Завдяки тому, що ця проблема виникає в багатьох сферах діяльності людини, було винайдено велику кількість різноманітних методів і алгоритмів для її вирішення. Проте все ще існують такі види задач пошуку шляху для яких важко знайти найкраще вирішення за необхідний проміжок часу, наприклад геометрична задача комівояжера.

Отже, можна впевнено сказати, що дана тема не втрачає своєї актуальності і в теперішній час та потребує подальших досліджень.

В задачі пошуку методів та алгоритмів пересування об’єктів на площині, сама площина може бути представлена у багатьох формах за допомогою різноманітних структур, наприклад у вигляді графу, математичної площини або лабіринту. Для цієї роботи було вирішено обрати в якості моделі площини лабіринт, через наочність та легкість сприйняття цієї структури людиною.

Об’єктом дослідження цієї роботи є алгоритми генерації та пошуку шляхів для двовимірних лабіринтів.

Предметом дослідження цієї роботи є реалізація алгоритмів генерації та пошуку шляхів для двовимірних лабіринтів і реалізація візуального відображення роботи алгоритмів пошуку шляхів за допомогою мови програмування C#.

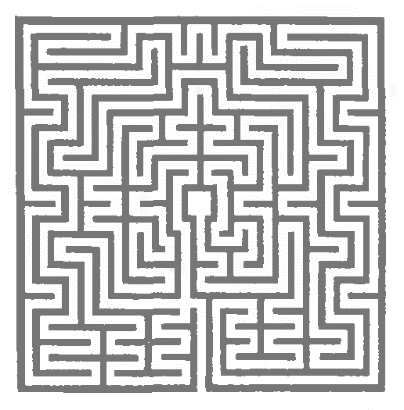
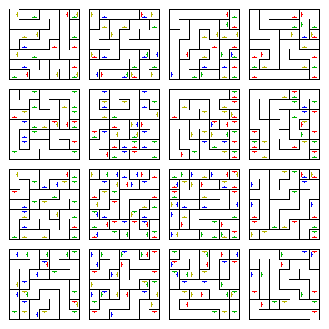
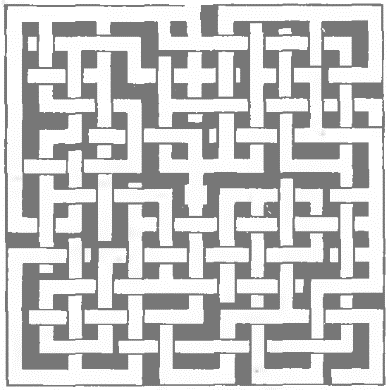
Метою дослідження цієї роботи є створення програми яка б дозволяла користувачу:

* Розглянути декілька алгоритмів генерації лабіринтів
* Розглянути декілька алгоритмів пошуку виходу з лабіринтів
* Порівняти різні алгоритми пошуку виходу з лабіринту

# КЛАСИФІКАЦІЯ ЛАБІРИНТІВ

Лабіринти можна розділити по сімох різних класифікаціях: розмірності, гіперрозмірності, топології, теселяції, маршрутизації, текстурі та пріоритету. Лабіринт може використовувати по одному елементу з кожного класу в будь якому поєднанні.

Розмірність вказує на те скільки вимірів у просторі займає лабіринт. Існують такі розмірності лабіринтів:

* **2D** – більшість лабіринтів має саму таку розмірність. Лабіринт такої розмірності можна зобразити на площині, або намалювати на листку паперу.
* **3D** – тривимірний лабіринт. Лабіринт в якому можна рухатись не тільки в чотирьох напрямках на площині, а ще вверх і вниз. Частіше за все такі лабіринти зображають у вигляді декількох 2D лабіринтів, які є рівнями пов’язаними між собою спусками та підйомами.
* **Вищі** **розмірності**. Наприклад 3D лабіринт з додатковими порталами для переміщення у часі. Лабіринти вищих розмірностей складно зобразити у легкому для людського сприйняття вигляді, тому на практиці вони менш поширені.
* **Переплетення** – це 2D лабіринт але в якому коридори можуть перекривати один одного. Лабіринт з такою розмірністю легко зобразити на двовимірній поверхні. В реальному світі такий лабіринт можна побудувати використовуючи мости для перекриття одним коридором іншого.

Топологія – вказує на те яким чином лабіринт розташований у просторі.

* **Звичайний** – це стандартний лабіринт розташований у Евклідовому просторі.
* **Планарний –** це будь який лабіринт розташований у просторі незвичним чином. Зазвичай це означає що краї лабіринту з’єднані особливим чином. Прикладом може бути будь який лабіринт розташований на тривимірній фігурі. Наприклад лабіринт на поверхні призми, або на поверхні стрічки Мебіуса.

Теселяція – вказує на те якої форми кожна окрема клітинка в лабіринті.

* **Прямокутний** – стандартний вид лабіринту з комірками які мають прямі кути.
* **Дельта –** лабіринт що складається із з’єднаних один з одним трикутників.
* **Сигма –** лабіринт що складається з шестикутників з’єднаних між собою. Таким чином у кожної клітинки є шість стін і максимальна кількість коридорів на розвилці дорівнює шести.
* **Тета –** лабіринт який складається з концентричних кіл і в якому початок або кінець може знаходитись на зовнішньому колі і в центрі.
* **Епсилон –** лабіринт що складається з восьмикутників.
* **Дзета –** лабіринт що складається із прямокутних комірок але має проходи також і по діагоналі.
* **Омега –** так називають будь який лабіринт який має не прямокутну теселяцію.
* **Crack –** це лабіринт який не має конкретної формі теселяції. В такому лабіринті стіни проводяться пі випадковими кутами.
* **Фрактальний –** це лабіринт створений з менших лабіринтів, де кожний наступний лабіринт теж створений з таких самих лабіринтів.

Маршрутизація **–** вказує на те які типи проходів має лабіринт в межах геометрії яка обумовлена **розмірністю, топологією та теселяцією.** Основний параметр при генерації лабіринту.

* **Ідеальний –** це лабіринт який немає циклів і недоступних областей. В такому лабіринті між будь якими двома точками існує один і тільки один шлях. Лабіринти такого типу мають єдиний розв’язок.
* **Плетений –** це лабіринт який немає тупиків. В такому лабіринті проходи переплітаються та зіштовхуються один з одним. Добре продуманий алгоритм цього типу може бути набагато складнішим за **ідеальний** лабіринт такого ж розміру.
* **Одномаршрутний –** такий лабіринт не має жодної розвилки або перехрестя. Він представляє собою довгий коридор з багатьма поворотами. Для лабіринту такого типу існує лише один спосіб обходу. Задача пошуку виходу для такого лабіринту доволі тривіальна. Достатньо рухатися вперед для досягнення виходу.
* **Розріджений –** це лабіринт в якому стіни присутні не у всіх комірках, тобто існують доволі великі відкриті області. Також в лабіринтах такого типу є недосяжні області.
* **Частково плетений –** цей тип лабіринту має одночасно і цикли і тупики.

Текстура – вказує на стиль проходів в лабіринті при будь якій маршрутизації та геометрії.

* **Зміщення –** лабіринт із зміщеними проходами це такий лабіринт в якому довжина проходів в якомусь одному напрямку загалом більша ніж довжина проходів у інших напрямках. Наприклад, лабіринт із горизонтальним зміщенням має набагато більше довгих горизонтальних проходів ніж вертикальних.
* **Прольоти –** цей показник відповідає за те на скільки довгими будуть неперервні коридори в лабіринті. Тобто в лабіринті з високим значенням прольотів буде багато довгих проходів без розвилок. В протилежному варіанті, з низьким показником прольотів, лабіринт буде виглядати більш випадковим і не мати довгих коридорів.
* **Елітність –** цей показник визначає довжину розв’язку лабіринту відносно його розміру. Більш елітні лабіринти мають короткий прямий розв’язок, в той час як менш елітні мають довгий розв’язок з великою кількістю поворотів. Проте добре спроектований елітний лабіринт набагато складніше розв’язати ніж менш елітний.
* **Симетричність –** симетричний лабіринт має симетричні проходи. Наприклад симетричні відносно центра лабіринту, або відносно вертикальної чи горизонтальної осі. Лабіринт може бути повністю або частково симетричним.
* **Однорідність –** однорідний алгоритм генерує всі можливі лабіринти з однаковою вірогідністю. Лабіринт має однорідну структуру, якщо він, якщо він виглядає як типовий лабіринт згенерований однорідним алгоритмом.
* **Текучість –** в лабіринті з високим показником текучості невелика кількість тупиків, але вони мають велику довжину. В лабіринті з низьким показником текучості велика кількість коротких тупиків.

Фокус – цей клас демонструє те, що алгоритми створення лабіринтів можуть бути поділені на два головні типи: ті що додають стіни та ті що вирізають проходи.

* **Додавання стін –** алгоритми що починають з порожньої площини та додають стіни. В реальному житті більшість лабіринтів створених штучно створені саме за допомогою додавання стін.
* **Вирізання проходів –** алгоритми що починають з заповненої області та вирізають проходи. В реальному житті прикладом лабіринту створеного за таким алгоритмом може бути підземна шахта.
* **Шаблон –** лабіринт може бути згенеровано за допомогою одночасно двох підходів. Так реалізовані деякі комп’ютерні алгоритми для створення лабіринтів. Шаблон застосовується для створення лабіринту на основі якогось графічного зображення. Шляхом додавання стін і вирізання проходів зображення перетворюється на лабіринт і зберігає структуру яка задана на зображенні.

# АЛГОРИТМИ СТВОРЕННЯ ЛАБІРИНТІВ

Існує багато методів генерації лабіринтів. Для кожного типу маршрутизації лабіринту існує свій алгоритм генерації.

* **Ідеальний –** створення стандартного ідеального лабіринту зазвичай потребує його «вирощування» із забезпеченням відсутності циклів і недосяжних областей. Розпочинають із зовнішньої стіни і випадковим чином додають фрагмент стіни який доторкається до зовнішньої, а його інший кінець знаходиться в нерозміченій частині лабіринту. Цей алгоритм є прикладом алгоритму додавання стін.
* **Плетений –** для створення лабіринту без тупиків треба додавати до лабіринту сегменти стін випадковим чином, але щоб при додаванні нового сегменту не створювався глухий кут. Можна розділити весь алгоритм на чотири етапи:

1. Перейти до зовнішньої стіни
2. Обійти лабіринт додаючи окремі сегменти стіни які дотичні з кожною вершиною стіни
3. Обійти всі сегменти стін у випадковому порядку додаючи стіну там де це не створить глухого кута

* **Одномаршрутний –** для створення лабіринту без розгалужень тільки з одним проходом треба спочатку згенерувати ідеальний лабіринт, потім закрити вихід і додати стіни які б розділяли кожний прохід на дві частини. При цьому всі глухі кути перетворяться на U- подібні повороти, а вихід буде розташовано поруч із входом.
* **Розріджений –** для початку треба обрати радіус розрідженості лабіринту. Тобто на якій приблизно відстані будуть знаходитись стіни лабіринту одна від іншої. Після цього можна починати випадковим чином вирізати лабіринт із суцільної області. На кожному кроці треба перевіряти чи нема по ходу обраного напрямку в радіусі розрідженості комірок які вже є частиною лабіринту.
* **3D –** для генерації тривимірних лабіринтів і лабіринтів більших розмірностей зазвичай використовують алгоритм вирізання проходів. Можна використовувати алгоритм створення двомірного ідеального лабіринту, але на кожному кроці обирати одну з шести сусідніх комірок, а не чотирьох, як в двомірному випадку.
* **Crack –** для генерації можна використовувати метод генерації ідеального лабіринту з додаванням стін. Проте в цьому алгоритмі оперують не комірками лабіринту а пікселями. Спочатку обирається перший випадковий піксель і замальовується. Це перший фрагмент стіни. На кожному наступному кроці ми обираємо випадковий вже замальований піксель і одну випадкову точку. Малюємо лінію з пікселя до цієї точки. Якщо на шляху лінії зустрічається інший замальований піксель, то перестаємо малювати лінію за довільну, визначену заздалегідь відстань. Повторюємо цей крок, поки можемо намалювати наступну лінію. Отриманий лабіринт буде мати декілька основних великих стін та багато менших стін, які відходять одна від одної у випадкових напрямках.