

РОЗРОБКА ПРОГРАМИ ДЛЯ АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ ПЕРЕТИНУ БАГАТОКУТНИКІВ З ВИКОРИСТАННЯМ ГРАФІЧНОГО ІНТЕРФЕЙСУ

¹ Вінницький національний технічний університет

Анотація:

Ця робота присвячена розробці програмного забезпечення для аналізу та візуалізації перетину багатокутників з використанням графічного інтерфейсу. Розроблена програма дозволяє користувачам візуально представляти багатокутники, завантажені з файлів з координатами, на графічному інтерфейсі. Вона автоматично визначає перетин між цими багатокутниками та виділяє вершини, які належать іншому багатокутнику, іншим кольором.

Ключові слова: програма, аналіз, візуалізація, багатокутники, графічний інтерфейс.

Abstract:

This work is devoted to the development of software for the analysis and visualization of the intersection of polygons using a graphical interface. The developed program allows users to visually represent polygons loaded from files with coordinates on a graphical interface. It automatically determines the intersection between these polygons and highlights vertices belonging to another polygon in a different color.

Keywords: software, analysis, visualization, polygons, graphical interface.

Вступ

Геометричні об'єкти, такі як багатокутники, часто використовуються у багатьох галузях, включаючи геодезію, комп'ютерну графіку, дизайн тощо. При аналізі цих об'єктів важливо виявити їх перетин для подальшої обробки та візуалізації.

Метою цієї роботи є розробка програмного забезпечення, яке дозволить ефективно аналізувати та візуалізувати перетин опуклих багатокутників. Ця програма буде здатна автоматично визначати перетин між багатокутниками та надавати зручний інтерфейс для візуального представлення результатів.

Огляд технологій

Для розробки програмного забезпечення була використана мова програмування C++, оскільки вона відома своєю зрозумілістю, швидкодією та потужними можливостями. Мова C++ дозволяє ефективно працювати з обчисленнями та оптимізацією, що особливо важливо для програм, які обробляють великі обсяги геометричних даних. Крім того, C++ є популярним вибором у галузі програмування через свою широку підтримку та активну спільноту.

Для реалізації графічного інтерфейсу була використана бібліотека GLFW. Вибір бібліотеки GLFW був обумовлений її відносною простотою підключення та зрозумілістю роботи з нею. GLFW надає зручний інтерфейс для створення вікон та обробки подій, що робить його ідеальним вибором для реалізації графічного інтерфейсу у програмах на C++.

Обидва інструменти - мова програмування C++ та бібліотека GLFW - сприяли ефективній розробці програмного забезпечення, дозволяючи швидко і зручно створити програму для аналізу та візуалізації перетину багатокутників.

Опис програми

Назва програми: Polygons

Програма Polygons призначена для аналізу та візуалізації перетину опуклих багатокутників. Вона дозволяє користувачам задавати у двох файлах координати багатокутників, та автоматично опрацьовує їх і відображає їх на графічному інтерфейсі. Програма автоматично зчитує з файлів координати, перевіряє задані фігури на опуклість та в подальшому перевіряє їх на перетини і якщо треба вираховує вершини одного багатокутника які лежать в іншому, виділяє їх іншим кольором та записує їх в окремий файл.

Програма складається з наступних файлів:

1. **main.cpp** – головний файл програми що містить точку входу.
 2. **header.h** – файл в якому відбуваються всі пудключення бібліотек та заголовкових файлів.
 3. **drawObject.h** – файл з реалізацією класу, що представляє об'єкт який відмальовується у графічному інтерфейсі.
 4. **point.h** – файл з реалізацією класу, що представляє об'єкт - точку.
 5. **vec2D.h** – файл з реалізацією класу, що представляє об'єкт - вектор.
 6. **polygon.h** – файл з реалізацією класу, що представляє об'єкт – багатокутник.
 7. **files.h** - файл з методами розробки з файлами.
 8. **graphics.h** - файл з реалізацією методів роботи графічного інтерфейсу.
 9. **mathematics.h** – файл з реалізацією всіх математичних методів обробки об'єктів.
- У програмі є три основні класи: **Point**(точка), **Vec2D**(вектор), **Polygon**(багатокутник).

Клас **DrawObject**:

Метод: **draw()**; - заготовка для методу малювання у всіх об'єктах.

Клас **Point**:

Змінні: **x, y** – координати точки.

Методи:

1. **Point(float x, float y)**; - конструктор.
2. **Point(float coords[2])**; - конструктор.
3. **draw()**; - метод для виведення точки на графічний інтерфейс.

Клас **Vec2D**:

Змінні:

1. **x, y** – координати вектора.
2. **vertexA, vertexB** – точки початок та кінець вектора.
3. **length** – довжина вектора.

Методи:

1. **calcVecLength()**; - для обчислення довжини вектора.
2. **calcVecCoords()**; - для обчислення координат вектора.
3. **Vec2D(Point vert1, Point vert2)**; - конструктор
4. **Vec2D(float vert1[2], float vert2[2])**; - конструктор
5. **getVertA()**; - метод для отримання початку вектора.
6. **getVertB()**; - метод для отримання кінця вектора.
7. **X()**; - метод для отримання координати **x** вектора.
8. **Y()**; - метод для отримання координати **y** вектора.
9. **Length()**; - метод для отримання довжини вектора.
4. **draw()**; - метод для виведення вектора на графічний інтерфейс.

Клас **Polygon**:

Змінні:

1. **vertexes** – масив вершин багатокутника.
2. **edges** – масив граней багатокутника.

Методи:

1. **createEdges()**; - метод створення масиву граней.
2. **Polygon()**; - конструктор.
3. **getVertexesAll()**; - метод для отримання масиву вершин багатокутника.
4. **getEdgesAll()**; - метод для отримання масиву граней багатокутника.
5. **getVertex(int id)**; - метод для отримання однієї вершини за індексом.
6. **getEdge(int id)**; - метод отримання однієї грані за індексом.

7. **draw()**; - метод для виведення багатокутника на графічний інтерфейс.

Пояснення основного функціоналу програми

В результаті після компіляції програма складається з 4 файлів: **Polygons.exe**, **pol1.txt**, **pol2.txt**, **result.txt**. У файлах **pol1.txt** та **pol2.txt**, записуються координати вершин багатокутників, потім при запуску основної програми **Polygons.exe**, програма зчитує координати з файлів використовуючи функцію **ReadPolsVerts()**, перевіряє їх на задоволення умов, та записує їх в глобальні масиви точок **pol1Verts**, **pol2Verts**. Далі по цим масивам створюються два багатокутники **pol1**, **pol2** використовуючи конструктор, та перевіряються на опуклість, у разі не задоволення хоча б одної умови, програма зупиняється.

Після валідації запускається алгоритм перевірки двох багатокутників на перетин **checkPolygons(pol1, pol2)**. Алгоритм перевірки перетину двох багатокутників полягає у наступному: спочатку для кожної вершини одного багатокутника перевіряється, чи знаходиться вона всередині іншого багатокутника. Це виконується за допомогою алгоритму підрахунку перетинів, де кожна вершина перевіряється на перетин з ребрами іншого багатокутника. Якщо хоча б одна вершина з одного багатокутника знаходиться всередині іншого, це свідчить про їх перетин. Парна кількість перетинів означає, що точка знаходиться ззовні багатокутника. Якщо перетин виявлено, виводиться повідомлення "the polygons are crossing" а також зберігаються координати вершин що лежать в іншому багатокутнику в масив **resPoints**. Цей алгоритм дозволяє ефективно визначати перетин багатокутників за допомогою перевірки кожної вершини одного багатокутника відносно іншого.

Після цього всі об'єкти що повинні бути виведені на екран, зберігаються в масив «мальованих елементів», який використовується для динамічного виведення об'єктів.

На кінець всіх обчислень, результуючі точки записуються в файл **result.txt**.

Останньою дією програми є запуск методу відмальовування який використовує бібліотеку **GLFW** для створення вікна графічного інтерфейсу та відображення всіх елементів масиву **drawObjs**.

Інструкція з використання програми

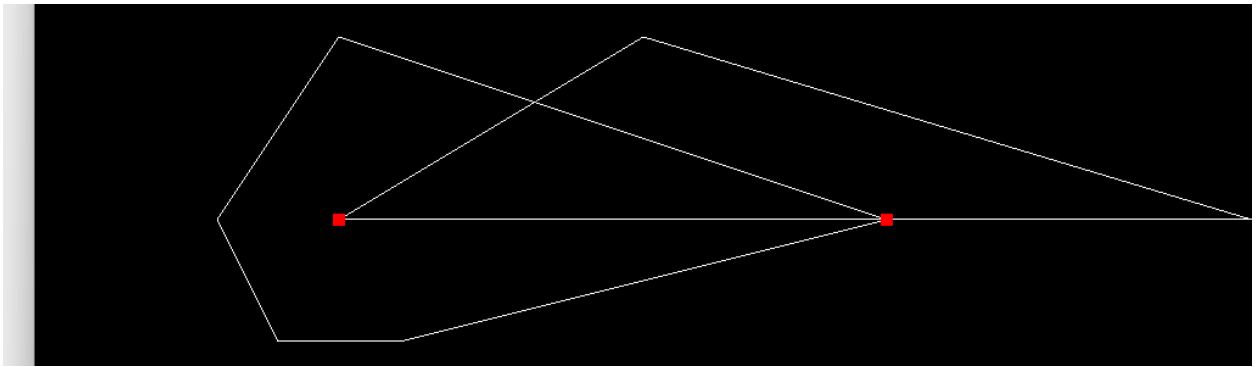
1. Скачати архів з програмою за посиланням: <https://drive.google.com/file/d/1LEUk2KqRAWYJmk6MGKCjbg2CGvKFtuZH/view>
2. Розархівувати все що знаходиться в архіві.
3. Записати координати вершин у файли **pol1.txt** та **pol2.txt**. Вершини повинні бути записані у порядку по, або проти часової стрілки! Координати повинні бути строго у форматі: «**x=0.01; y=0.01**» та у межах від -1 до 1 включно [-1;1]! Кожна вершина повинна бути записана в окремому рядку. В іншому випадку програма може працювати не коректно, або не працювати зовсім.
4. Зберегти текстові файли.
5. Запустити **Polygons.exe**.
6. Після запуску, якщо все було виконано по інструкціям, якщо багатокутники перетинаються буде виведено повідомлення, а також запущений графічний інтерфейс з зображеннями фігур.
7. Після завершення програми можна відкривати файл **result.exe** з результатами програми.

Приклад використання програми

Pol1.txt
x=-0.5; y=0
x=0; y=0.3
x=1; y=0

Pol2.txt
x=-0.7; y=0
x=-0.5; y=0.3
x=0.4; y=0
x=-0.4; y=-0.2
x=-0.6; y=-0.2

the polygons are crossing



result.txt

x=-0.500000; y=0.000000

x=0.400000; y=0.000000

Висновки

Розроблена програма для аналізу та візуалізації перетину багатокутників є потужним інструментом з точними алгоритмами та зручним інтерфейсом, що сприяє ефективній роботі з геометричними даними. Вона може знайти застосування як у дослідженнях, так і у практичних сферах, від геодезії до комп'ютерної графіки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Методичні вказівки до виконання лабораторних робіт з дисципліни "Програмування". Частина 2. Об'єктно-орієнтоване програмування мовою C++ для студентів спеціальностей 121 – «Інженерія програмного забезпечення», 123 – «Комп'ютерна інженерія» денної та заочної форм навчання / Уклад. Д. В. Кисюк, О. В. Дудник. – Вінниця : ВНТУ, 2024 – 73 с.
2. Методичні вказівки до виконання курсової роботи з дисципліни "Програмування" для студентів спеціальності 123 - "Комп'ютерна інженерія" освітніх програм "Комп'ютерна інженерія" та "Системне програмування" всіх форм навчання / Уклад. О. І. Черняк, Л. А. Савицька – Вінниця: ВНТУ, 2022 – 47с.

Марценюк Роман Володимирович – студент групи ІСП-23Б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: mister.roman240706@gmail.com

Науковий керівник: **Клепа Ірина Анатоліївна** – PhD, старший викладач кафедри вищої математики Вінницький національний технічний університет, м. Вінниця

Roman Martseniuk – Department of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email : mister.roman240706@gmail.com

Supervisor: **Iryna Klepa** – PhD, Senior Lecturer, Assistant of the Department of Higher Mathematics, Vinnytsia National Technical University, Vinnytsia