

# Архітектура програмної системи для розпізнавання та знаходження шляху в лабіринтах

Стопчак Дмитро

Кафедра інженерії програмного забезпечення  
Хмельницький НУ  
Хмельницький, Україна  
arineyt@gmail.com

Форкун Юрій

Кафедра інженерії програмного забезпечення  
Хмельницький НУ  
Хмельницький, Україна  
forkun@ridne.net

*Developed an architecture of an application for resolving tasks pertaining to path search. Determined factors affecting usage of search algorithms. Set tasks for future research.*

**Ключові слова:** лабіринт, пошук, вузол, оптимальний шлях.

## ВСТУП

Розглядається архітектура додатка для розв'язування логічних задач, пов'язаних з пошуком шляху в лабіринтах. Існує безліч алгоритмів пошуку шляху в лабіринтах, наприклад, перебір з поверненням, пошук в глибину, пошук в ширину, оснований на них хвильовий алгоритм Лі, алгоритм Дейкстри, пошуку A\* тощо. Ці алгоритми зазвичай використовуються під конкретну задачу або жорстко вказані вхідні умови [1].

## ПОСТАНОВКА ЗАДАЧІ

Завданням дослідження є формування архітектури системи, що дає можливість отримання конкретного шляху в лабіринті, надання можливості отримувати завдання з різних джерел. Використовуючи існуючі рішення (алгоритми, бібліотеки) об'єднати розрізнені компоненти в єдину систему, яка дозволить виконати набір підзадач (від отримання і перетворення даних в формальний вигляд до власне отримання рішення) з мінімальним втручанням користувача на кожному з етапів.

## ОСНОВНА ЧАСТИНА

Вхідними даними системи є зображення лабіринту та точок початку і закінчення шуканого шляху.

Вихідними даними інформаційної системи є розв'язок задачі, а саме зображення лабіринту з позначеним на ньому оптимальним шуканим шляхом.

Пошуком шляху є побудова, за допомогою комп'ютера, найкоротшого маршруту між двома точками[2]. За своєю суттю алгоритм пошуку шляху шукає на графі, починаючи з однієї (стартової) точки і досліджуючи суміжні вузли до тих пір, поки не буде досягнутий вузол призначення (кінцевий вузол). В більшості алгоритмів пошуку шляху закладена мета знайти найкоротший шлях. Деякі методи пошуку на графі, такі як пошук в ширину, можуть знайти шлях, якщо дано достатньо часу. Інші методи, які «досліджують» граф, можуть досягти точки призначення швидше.

Існує дві основні задачі пошуку шляху:

- знайти шлях між двома вузлами в графі;
- знайти оптимальний шлях.

Створювана система має на меті створити розширювану архітектуру для об'єднання алгоритмів розпізнавання і знаходження шляху в лабіринтах. Вона містить чотири основні компоненти:

- модуль, що відповідає за отримання зображення лабіринту;
- модуль, що відповідатиме за розпізнання і перетворення зображення

лабіринту в зручну програмну структуру для подальших маніпуляцій з нею;

- модуль, що відповідатиме за пошук шляху в лабіринті;
- модуль, що відповідає за формування результату.

Система передбачає заміну компонентів, з збереженням функціонування системи загалом. Тому передбачена слабка зв'язність між пакетами системи.

Це дозволить ефективно знаходити помилки в програмному забезпеченні і надає можливість їх швидкого виправлення.

На рисунку 1 представлено діаграму класів програмної системи.

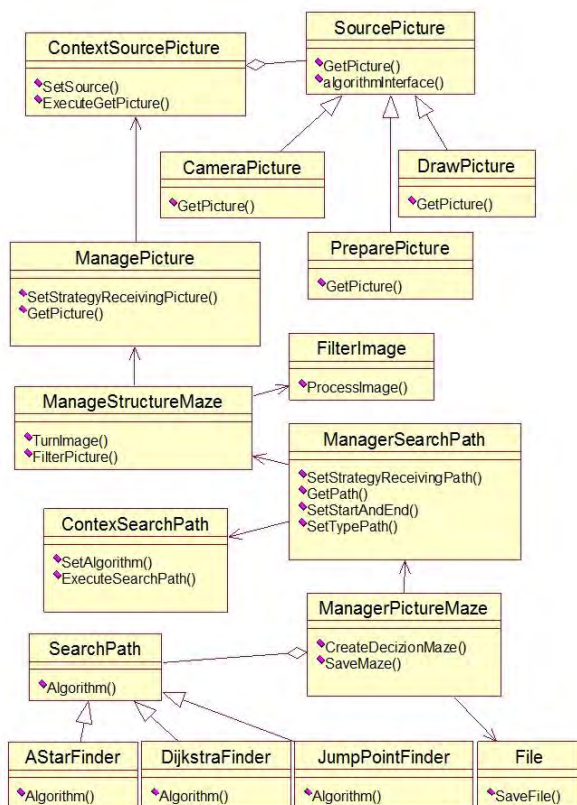


Рис. 1. Діаграма класів

Розширюваність системи надає можливість додавати в додаток вирішення пошуку альтернативними алгоритмами.

На етапі розпізнавання в разі простого лабіринту можна розглядати зображення лабіринту, як набору двох кольорів. Тобто шляхом буде один з кольорів. Ці набори точок можна перетворити в булевий масив, що позначає лабіринт. Або перетворювати в альтернативну структуру, що буде давати вигреш (в швидкості, потребі пам'яті, потребі використовуваного алгоритму). Крім того, необхідно вирішувати проблеми ліфтів, однопрохідних місць. Для таких випадків доречно передбачити експертний режим для користувача, тобто ручного вказування важких випадків.

Корисно використати поведінковий шаблон «Стратегія» в розроблюваній системі. Він дозволяє створити декілька варіантів поведінки для одного об'єкту та винести в окремий клас. Визначає сімейство алгоритмів, інкапсулює кожен з них, і робить їх взаємозамінними[3]. Відділяється процедура вибору алгоритму від його реалізації. Це дозволяє зробити вибір на підставі контексту. Наприклад, шукаючи шлях в лабіринті, можна використовувати різні алгоритми, дозволяючи класам-клієнтам взаємодіяти з алгоритмами за допомогою реалізованого інтерфейсу.

## ВИСНОВКИ

Спроекована архітектура для синтезу алгоритмів розпізнавання і знаходження шляху. Дає можливість гнучко використовувати алгоритми пошуку, змінюючи на найбільш доречний, в залежності від типу і складності шляху. Система дозволяє об'єднати задачі отримання лабіринту, формування в структури, що обробляються комп'ютером і власне пошуку шляху.

## ЛІТЕРАТУРА

- [1] B. Stout, "Smart Move: Intelligent Path-Finding" unpublished, 1997.
- [2] E. Mathew, "Direction Based Heuristic For Pathfinding In Video Games", published by Elsevier B.V, 2015.
- [3] E. Gamma, R. Helm, R.Johnson, J. Vlissides "Design Patterns Elements of Reusable Object-Oriented Software" Addison-Wesley Professional London, p. 349, August 1997.
- [4]