

УДК 004.42; 004.94

DOI: 10.31673/2412-9070.2024.034851

О. О. ІЛЬІН, доктор техн. наук, професор, ORCID: 0000-0002-7940-4694,

В. О. ЄРМОЛЕНКО, асистент, ORCID: 0009-0007-5644-3950,

М. П. ГНІДЕНКО, канд. техн. наук, доцент, ORCID: 0009-0002-5261-3581,

Державний університет інформаційно-комунікаційних технологій, Київ

АЛГОРИТМИ ВИВЧЕННЯ СТРУКТУРИ ЛАБІРИНТІВ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЇХ ЗАСТОСУВАННЯ

У статті розглядаються лабіринти в контексті задачі їх створення та проходження для застосування в ігровій індустрії. Наводяться базові відомості про лабіринти, їх основні різновиди та структури, наводиться математична модель ідеального лабіринту. Досліджуються та вирішуються питання, пов'язані з вибором параметрів оцінки ефективності їх роботи, безпосередньо аналіз алгоритмів, проєктування тестового додатку на мові Java та аналіз отриманих результатів. Під час оцінки ефективності проаналізовано декілька алгоритмів дослідження структури за параметрами швидкості роботи та надано рекомендації по обранню найоптимальнішого з представлених.

Ключові слова: лабіринт; ідеальний лабіринт; пошук в глибину; пошук в ширину; алгоритм Тремо; Java; програма; алгоритм заповнення глухих кутів.

Вступ

Для імітації людської поведінки у комп'ютерних іграх наразі застосовується штучний інтелект, який в залежності від алгоритмів в більшій або меншій мірі імітує розумну поведінку комп'ютерного суперника. Особливістю комп'ютерного гравця є те, що на програмному рівні комп'ютерні суперники (т.з. інтелектуальні агенти) мають доступ до всіх об'єктів ігрової ситуації (простору): карти рівня, ігрової дошки і т.п. Для кращої людської імітації поведінки агента, дані, що поступають до нього, пригнічують таким чином, щоб вони співпадали з даними, які в такій же ситуації може отримувати людина за допомогою органів відчуття. Тому, якщо говорити про ігрові рівні, побудовані на основі лабіринтів, важливо застосовувати відповідні алгоритми пошуку виходу з нього (шляху в загальному випадку), а також розпізнавання їх структури. Дослідження алгоритмів пошуку шляху та структури лабіринту для виявлення «людиноподібних» підходів та оцінка ефективності таких алгоритмів дозволяє виявити кращі алгоритми, на базі яких можна розробляти ігри більш цікавого сюжету.

Таким чином, робота присвячена дослідженню алгоритмів пошуку структури лабіринту. Досягнення кінцевої мети передбачає практичну розробку програмного забезпечення на мові Java для візуалізації їх структури та отримання конкретних значень параметрів ефективності обраних алгоритмів.

Основна частина

Лабіринт — це архітектурна споруда, яка складається з великої кількості приміщень та проходів між ними (коридорів), розташованих таким чином, що потрапивши у нього, вийти назад дуже

складно або неможливо [1]. Зазвичай мають вхід та вихід (рис. 1.), один або декілька маршрутів між входом та виходом.

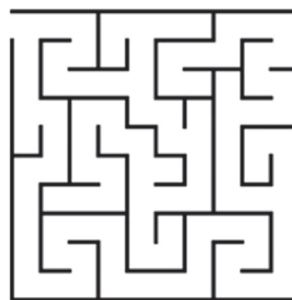


Рис. 1. Зображення лабіринту із входом та виходом та одним маршрутом між ними

Лабіринти в основному застосовувались в двох випадках: коли потрібно сховати що-небудь або кого-небудь, а також для заплутування осіб, які не знають плану лабіринту. На сьогоднішній час основна частина усіх реально споруджених лабіринтів використовується як декоративне архітектурне рішення або як тип ландшафтного дизайну. Лабіринт виступає одним із способів представлення ігрової сцени та карти ігрового рівня в задачах ігрової індустрії. При цьому певні об'єкти карти виступають сторонами (стінками) лабіринту, що дозволяє останнім бути ідеальним варіантом візуалізації лабіринту та процесу пошуку шляху на картах.

Класифікація лабіринтів. Існують наступні типи лабіринтів [2]:

- **двовимірні** — саме такий тип лабіринтів зустрічається частіше всього в задачах по розробці ігор (рис. 1);

- **багатовимірні** — тип лабіринту, в якому розмірність перевищує за три, в іграх це означає, що ігрові сцени мають портали у іншу розмірність, час та простір;

- *ортогональні* — звичайний прямокутний гамма-лабіринт;
- *дельта* — утворений зі з'єднаних між собою трикутників, кожна комірка якого може мати по три коридори (рис. 2);

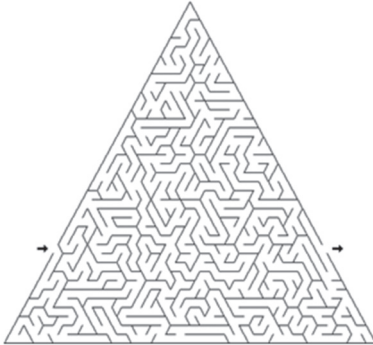


Рис. 2. Дельта-лабіринт [2]

- *сигма* — утворений із шестикутників, може мати по шість проходів;
- *тета* — лабіринт, що утворений у вигляді кола, одна сторона входу якого починається з зовнішньої частини кільця, а друга — зсередини, може мати чотири і більше коридори;
- *crack* — аморфний лабіринт без постійного заміщення має стіни та коридори під різним кутом;
- *фрактальні* — лабіринт в лабіринті, кожен сектор даного лабіринту є окремим лабіринтом, може бути нескінченно рекурсивним і повторювати свою будову нескінченне число разів, утворюючи ідеальний фрактал;
- *ідеальні* — такий тип лабіринту описується як лабіринт, що не має проходів-петель, замкнених ланцюгів чи областей, яких неможливо досягнути, від точки до точки існує один і тільки один шлях, фактично є деревом з великою кількістю вершин (рис. 3);

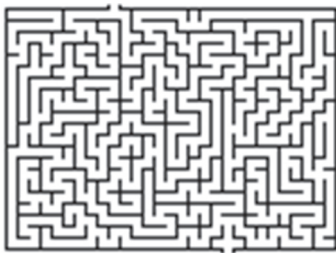


Рис. 3. Ідеальний лабіринт [2]

- *плетені* — такий тип лабіринту не має глухих кутів, замість них у лабіринті використовуються коридори, які замикаються та повертаються до початку або конкретної точки, і для проходження такого лабіринту може потребуватися більше часу, аніж при ідеальному лабіринті;
- *одномаршрутні* — найпростіший тип лабіринту, в якому він фактично є одним коридором, який не має розвилок і постійно повертається. Щоб пройти лабіринт цього типу потрібно просто йти вздовж одної сторони коридору. Якщо порівнюва-

ти з іншими лабіринтами при умовах, коли шлях відомий, то забирає набагато більше часу, оскільки пройти потрібно весь лабіринт.

Представлення лабіринту у математиці та у кодї програми. Математичну модель лабіринту можна зобразити за допомогою графу, каркасного дерева та матриці суміжності. На рис. 4 наведено ідеальний лабіринт, який складається з 24 комірок, з'єднаних випадковим чином, а на рис. 5 наведено його представлення у вигляді графу.

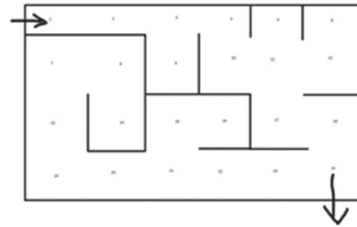


Рис. 4. Зображення лабіринту ідеального типу з 24 кімнат

Щоб представити лабіринт у вигляді графу потрібно домовитись, що проходи між комірками є ребрами графу, а самі комірки є вершинами графу. Стіни зображаються, як відсутність ребер між вершинами.



Рис. 5. Граф лабіринту ідеального типу, представленого на рис. 4

Математично представляється за допомогою виразу

$$G(V, E),$$

де G — граф; V — кількість вершин (комірок) у лабіринті; E — кількість ребер (коридорів) у лабіринті, при чому кількість ребер E дорівнює $V - 1$.

Можна помітити, що даний граф утворює собою остонове або каркасне дерево, тобто конструкцію, що не має циклів у своїй структурі і при цьому має з'єднання до кожної вершини. Вигляд дерева розглянутого графу представлено на рис. 6.

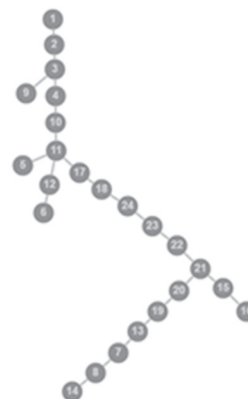


Рис. 6. Каркасне дерево лабіринту ідеального типу

Під час дослідження лабіринт піддавався вивченню кожним з алгоритмів, який повністю вивчав його структуру. Робота кожного алгоритму займала певний час. У таблиці наведено результати дослідження швидкодії розглянутих алгоритмів — пошуку в глибину, пошуку в ширину, алгоритму Тремо та пошуку глухих кутів. Дослідження проводились на лабіринтах різних розмірів: 25×25, 50×50, 75×75 та 100×100 комірок.

Часові характеристики роботи досліджуваних алгоритмів

Алгоритм	Розмір лабіринту			
	25×25	50×50	75×75	100×100
Пошук в глибину, мс	0,00762	0,03474	0,13634	0,25740
Пошук в ширину, мс	0,01781	0,10024	0,25001	0,45312
Пошук глухих кутів, мс	0,78980	9,39720	35,49480	88,46320

З наведених результатів можна зробити висновок про те, що згідно часових характеристик найкращим є алгоритм пошуку в глибину, а найгіршим — пошук глухих кутів. Слід зазначити, що алгоритм пошуку глухих кутів призначений для пошуку усіх доступних шляхів між входом та виходом лабіринту. І його також можна застосувати для пошуку структури лабіринту, якщо входи та виходи вважати «закритими». Також він має принципову особливість у порівнянні із іншими: він створений для використання комп'ютерною програмою, яка має можливість бачити й обробляти весь лабіринт одночасно [6]. Алгоритми пошуку в глибину та ширину є менш універсальними і призначені для знаходження одного можливого варіанту проходження лабіринту та дослідження його структури. Характер дослідження структури лабіринту всіх згаданих алгоритмів розрізняється, що легко помітити під час візуалізації цього процесу

у розробленому програмному забезпеченні. З цього погляду, розроблена програма може виступати як засіб дослідження роботи алгоритмів під час навчання студентів мові програмування Java.

Висновки

В роботі проведено дослідження роботи ряду алгоритмів вивчення структури лабіринтів. Чисельні виміри дозволили виявити кращий алгоритм для використання при написанні та оптимізації роботи інтелектуальних агентів у комп'ютерних іграх, де ігровий простір містить лабіринт в тому чи іншому виді.

Список використаної літератури

1. Yerry Soepriyanto, Akhmad Fakhruddina, Sirkabudena, Eka Pramono Adia. *Development of Maze Game on Interactive Whiteboard for Fine Motor Learning // Advances in Social Science, Education and Humanities Research. 2017. Vol. 118. P. 651.*
2. *Maze Classification [Електронний ресурс]. URL: <https://www.astrolog.org/labyrnth/algrithm.htm>*
3. *Introduction to Algorithms, Second Edition / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein // MIT Press and McGraw-Hill, 2001. P. 594–611.*
4. *Jamis Buck. Mazes for Programmes. Code your own twisty little passages. The Pragmatic Bookshelf.*
5. *A Comprehensive and Comparative Study of Maze-Solving Techniques by Implementing Graph Theory / Adil M. J. Sadik, Maruf A. Dhali, Hasib M. A. B. Farid [et al.] // Department of Electrical and Electronic Engineering. 2010. P. 52–53.*
6. *Maze-solving algorithm [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Maze-solving_algorithm*

O. Ilyin, V. Yermolenko, M. Hnydenko

ALGORITHMS FOR DETECTING THE STRUCTURE OF LABYRINTHS AND STUDYING THE EFFICIENCY OF THEIR APPLICATION

The article considers labyrinths in the context of the task of their creation and completion for use in the gaming IT industry. Artificial intelligence is currently used to simulate human behavior in computer games, which, depending on the algorithms, more or less imitates the intelligent behavior of a computer opponent. The peculiarity of a computer player is that at the software level, computer opponents have access to all objects of the game situation: the level map, the game board, etc. For a better human imitation of the behavior of the agent, the data coming to it is suppressed in such a way that it coincides with the data that a person in the same situation can receive using the senses. Therefore, it is important to apply appropriate algorithms for finding a way out of it, as well as recognizing their structure. The study of path-finding algorithms and the structure of the labyrinth for the detection of "human-like" approaches, and the evaluation of the effectiveness of such algorithms, allows us to identify better algorithms on the basis of which it is possible to develop games with a more interesting plot. Thus, this work is devoted to the research of algorithms for finding the structure of a labyrinth. Achieving the final goal involves the practical development of software in the Java language for visualizing their structure and obtaining specific values of the performance parameters of the selected algorithms. The work provides basic information about labyrinths, their main varieties and structures, and provides a mathematical model of an ideal labyrinth. Questions related to the selection of parameters for evaluating the effectiveness of their work, the direct analysis of algorithms, the design of a test application in Java and the analysis of the obtained results are investigated and resolved. During the performance evaluation, several structure research algorithms were analyzed in terms of speed of operation, and recommendations were given to choose the most optimal of the ones presented.

Keywords: maze; ideal maze; depth-first search; breadth-first search; Tremo's algorithm; Java; program; dead-end filling algorithm.