- **Великі мовні моделі (LLM)** генерують діалогові тести, які дозволяють здійснювати формувальне оцінювання та забезпечують миттєвий фідбек [5].
- **Контекстний бандит (принцип грального автомату) LinUCB** здійснює адаптивне переналаштування ваг у цільовій функції, реагуючи на успішність проходження проміжних етапів [6].

Таким чином, алгоритм поєднує **детерміноване ядро (ILP/LNS)** із **адаптивними AI-компонентами**, що дозволяє у режимі реального часу перебудовувати освітній маршрут користувача.

Висновки. У роботі запропоновано системно-аналітичний підхід до побудови індивідуальних освітніх траєкторій, який включає:

1. Математичну модель на основі DAG та ILP-формалізації.
2. Алгоритм оптимізації з використанням як точних, так і евристичних методів.
3. Інтеграцію сучасних інструментів штучного інтелекту (CatBoost, LLM, контекстні бандити).

Отримана система забезпечує персоналізацію навчання, мінімізує когнітивні та часові витрати й водночас зберігає прозорість прийняття рішень. Це створює передумови для масштабування адаптивних освітніх систем нового покоління, які відповідають вимогам ринку праці та принципам системного аналізу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

[1] J. Reich and J. A. Ruipérez-Valiente, The MOOC pivot, Science, vol. 363, no. 6423, pp. 130–131, 2019. doi: 10.1126/science.aav7958.

[2] Y. Wang and H. Guan, Internet cognitive fatigue and online learning effectiveness, Frontiers in Public Health, vol. 10, p. 927454, 2022. doi: 10.3389/fpubh.2022.927454.

[3] D. Pisinger and S. Ropke, Large neighborhood search, in Handbook of Metaheuristics, 2nd ed., Springer, 2019, pp. 99–127.

[4] A. V. Dorogush, V. Ershov, and A. Gulin, CatBoost: gradient boosting with categorical features support, in Advances in Neural Information Processing Systems, vol. 31, 2018. [Online]. Available: https://arxiv.org/abs/1810.11363

[5] A. Norberg, D. Vorobyeva, and E. Chan, Effects of adaptive feedback using large language models in online learning, Journal of Computer Assisted Learning, 2025. doi: 10.1007/s40593-025-00457-x

[6] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, Contextual bandits with linear payoff functions, in Proc. 14th Int. Conf. Artificial Intelligence and Statistics, 2011, pp. 208–214.

UDC 004.93'1:336.74

SOFTWARE AND ALGORITHMICAL IMPLEMENTATION OF THE METHOD OF CONSTRUCTING THE VORONOG DIAGRAM

SKORIN YURII (yuriy.skorin@hneu.net), ZHU ZHUNBYAO (1385618657@139.com)
Kharkiv National Economic University named after Semen Kuznets

The existing areas of application of the Voronoi diagram were considered, it was determined that these diagrams are used in quite a few industries, such as cartography, architecture, biochemistry, etc. The task of implementing this method was formulated. Based on the results of the study of the subject environment, a list of tasks was determined that were solved to achieve the goal. A detailed analysis of the subject area was carried out, during which the requirements for input and output data were formed and described. Based on the analysis, an object-oriented model was built. Based on the data obtained in the analysis process, appropriate mathematical methods for constructing Voronoi diagrams were formulated. Based on the results of object-oriented design, a toolkit was selected for the software implementation of the automated system. Requirements for hardware and software were formulated. A user manual was developed for the software product that was developed as part of the research.

Statement of the problem.

The development of modern technologies leads to increasing automation of processes performed by humans in various spheres of life, including education, medicine, industry, agriculture, etc. These tasks are either impossible to solve using simple methods, or their solution will require a significant amount of time and resources. There are special methods and algorithms for solving such tasks.

The Voronoi diagram is an important tool in solving so many problems. In general, a Voronoi diagram is a partition of a plane in which each region of the partition forms a set of points that are closer to one of the elements of the set than to any other element of the set.

Discretization of the region using Voronoi cells provides a fast search for natural neighbors for a given point. Searching for natural neighbors based on information about neighboring nodes obtained from the Voronoi diagram significantly reduces the time required to construct interpolated functions.

Voronoi diagrams are a fairly powerful tool for data visualization and analysis, widely used in various fields of science and many areas of human activity.

Voronoi diagrams are extremely relevant in modern scientific research and practical applications due to their versatility and effectiveness in various fields.

They can be applied to any data that can be represented as points in space. This makes them a universal tool for analyzing data of various natures.

Voronoi diagrams allow you to visualize complex data, making it easier to understand and analyze. Voronoi diagrams can be used to identify implicit structures and patterns in data, which is important for making informed decisions.

Voronoi diagrams are used to solve optimization problems, for example, to determine the optimal location of objects in space.

Voronoi diagrams are a convenient tool that is used in a fairly wide range of scientific disciplines and practical problems.

Voronoi diagrams:

- help determine the optimal location of stores to minimize competition and maximize profits;
- used to plan the location of public facilities such as schools, hospitals, etc.;
- help determine the optimal location of base stations to ensure maximum coverage when solving the issue of optimizing wireless networks.

Versatility, intuitive visualization, and the ability to reveal hidden patterns make Voronoi diagrams an indispensable tool for data analysis and informed decision-making.

Based on the above, the goal of the study was formulated, namely, the software implementation of a system for searching for the closest elements using the Voronov diagram construction method.

The object of the study is an automated system for searching for the nearest elements using the Voronov diagram construction method.

The subject of research is software development technologies and methods, methods for constructing a Voronov diagram.

Presentation of the main material.

Nowadays, there is a wide variety of algorithms specifically for solving computational geometry problems. However, the scope of application of such algorithms is usually very narrow.

For example, known algorithms for constructing a Voronoi diagram can be effectively applied only to points or segments.

However, when solving practical problems, one usually has to deal with more complex geometric objects that can be defined as points and segments, as well as curves of arbitrary shape.

Constructing a Voronoi diagram for such geometric objects is a non-trivial task, and therefore the study of Voronoi diagrams is reduced to the study of their approximations [1].

A Voronoi diagram is a mathematical object that is not strictly described in this way. The Voronoi diagram of a given finite set of points M of a plane (space) is a partition in which each region (Voronoi cell) of the given partition forms a set of points that is closer to one of the elements of the set M than to any other element of this set [2].

The goal of the software implementation of the Voronoi diagram method is to provide a fast search for natural neighbors for a given point.

Searching for natural neighbors obtained from the Voronoi diagram significantly reduces the time required to construct functions.

To achieve the goal, the following was done:

- subject area analysis;
- select the main input and output objects;
- analyze the system limitations;
- choose the necessary algorithm;
- specify external storage;
- display the notation on the sheet.

Conclusions.

During the research, the existing areas of application of the Voronoi diagram were considered and it was found that today these diagrams are used in many fields, such as cartography, biochemistry and architecture.

The method of constructing the Voronoi diagram has applications in many fields, therefore, the task of implementing this method was formulated.

The goal and objectives of the diploma project have been formulated, the object and subject of the research have been determined.

Based on the results of the study of the subject environment, a list of tasks was determined that were solved to achieve the goal.

A detailed analysis of the subject area was conducted, during which the requirements for input and output data were formed and described.

Based on the analysis, an object-oriented model was built. Based on the data obtained in the analysis process, appropriate mathematical methods for constructing Voronoi diagrams were formulated.

The following algorithms were built to implement the methods of system classes:

- area calculation algorithm;
- algorithm for independent point location;
- algorithm for loading data into a file;
- Point editing algorithm.

Based on the results of object-oriented design, a toolkit was selected for the software implementation of the automated system.

A formalized programming language, namely C#, was chosen as the programming language in the Visual Studio Code development environment.

Based on the object-oriented model, a software product was created, which is implemented in accordance with the principles of object-oriented programming and consists of ten classes.

Hardware and software requirements have been formulated. A user manual has been developed for the software product that was developed as part of this research.

The manual describes the user's capabilities when working with the software product, reveals the purpose of windows and commands, and provides forms that correspond to the screen.

REFERENCES

1. Mark de Berg. Computational Geometry: Algorithms and Applications / de Berg Mark, Cheong Otfried, van Kreveld Marc, Overmars Mark // Springer, 2008. - 398 p.
2. Satyan L. Devadoss. Discrete and Computational Geometry / L. Devadoss Satyan, O'Rourke Joseph // Princeton University Press, 2011. - 272 p.
3. Kenneth E. Kendall. Systems Analysis and Design, Global Edition / E. Kendall Kenneth, E Kendall Julie // Pearson Education, 2019. - 578 p.
4. Hector Uriel Perez Rojas, Angel Teheran Garcia. Hands-On Visual Studio 2022: A developer's guide to new features and best practices with .NET 8 and VS 2022 for maximum productivity / Uriel Perez Rojas Hector, Angel Teheran Garcia Miguel // Packt Publishing, 2024. - 336 p.
5. Joseph Albahari. C# 9.0 in a Nutshell: The Definitive Reference / Albahari Joseph // O'Reilly Media, 2021. - 1058 p.
6. Beecham S. Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis / S. Beecham, T. Hall, A. Rainer, et al. // Empirical Software Engineering 2003 8(1): 7-42. [Electronic resource]. – Access mode : https://doi.org/10.1023/A:1021764731148.

7. Bjarnason E. Requirements Are Slipping Through the Gaps — A Case Study on Causes & Effects of Communication Gaps in Large-Scale Software Development / E. Bjarnason, K. Wnuk, B. Regnell // IEEE Computer Society 2011. [Electronic resource]. – Access mode : https://doi.org/10.1109/RE.2011.6051639.
8. Bardsiri V. K. A Flexible Method to Estimate the Software Development Effort Based on the Classification of Projects and Localization of Comparisons / V. K. Bardsiri, D. N. A. Jawawi, S. Z. M. Hashim, et al. // Empirical Software Engineering 2014 19(4): 857-884. [Electronic resource]. – Access mode : https://doi.org/10.1007/s10664-013-9241-4.

УДК: 004.9

ЗАСТОСУВАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ ПАРАДИГМИ В РОЗРОБЦІ СТРАТЕГІЧНОЇ ГРИ ЖАНРУ TOWER DEFENSE НА БАЗІ UNITY

СКРИПЛЬОНОК С.Д., МОЛЧАНОВА М.О., МАЗУРЕЦЬ О.В.
(ssskriplyonok@gmail.com, momolchanova@gmail.com, exe.chong@gmail.com)
Хмельницький національний університет

Розглянуто застосування об'єктно-орієнтованої парадигми програмування під час розробки стратегічної гри жанру Tower Defense на базі рушія Unity. ООП дозволила забезпечити модульність, масштабованість і зручність повторного використання коду, що є важливим для складних ігрових проєктів. Для опису логіки роботи гри було створено діаграми станів, варіантів використання та класів, які визначають основні системи та їхні взаємодії. Виокремлено п'ять ключових класів, серед яких центральне місце займає BattleController, що реалізує основні механіки бою.

Вступ

Об'єктно-орієнтована парадигма програмування забезпечує модульність, масштабованість і повторне використання коду [1], що є важливим під час розробки складних ігрових систем. Завдяки концепціям інкапсуляції, наслідування та поліморфізму ООП дозволяє моделювати ігрові об'єкти як незалежні сутності з власною поведінкою, спрощуючи підтримку та розвиток проєкту.

Розробка стратегічної гри жанру Tower Defense у середовищі Unity потребує чіткої архітектури та взаємодії численних об'єктів. Використання ООП разом із компонентною системою Unity дає змогу структуровано організувати ігрові елементи, спростити масштабування та внесення змін, що підвищує ефективність створення ігрових проєктів [2].

Проєктування та програмна реалізація гри жанру Tower Defense на базі Unity

Для забезпечення чіткого розуміння структури та логіки роботи гри було проведено проєктування її основних систем і взаємодій. З цією метою створено діаграми станів (рисунок 1), варіантів використання (рисунок 2), які відображають поведінку ігрових об'єктів, сценарії взаємодії користувача з системою.

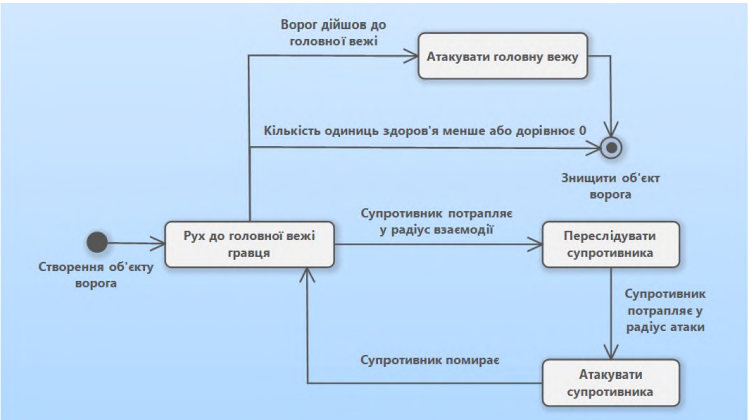

Рисунок 1 – Діаграма станів поведінки ворога