Для перешкоджання виникненню таких загроз застосовують алгоритми шифрування даних. Одним із таких алгоритмів є алгоритм шифрування RSA. Це криптографічний алгоритм з відкритим ключем, який базується на складності задачі розкладу на прості множники великих цілих чисел [3].

Список використаних джерел

1. Лисенко Д. В. Метод підвищення ефективності генерації псевдовипадкових послідовностей для потокового шифрування даних: магістерська дисертація. Київ, 2022.

2. Knuth D. E. The art of computer programming. Vol. 2, Seminumerical algorithms. 3rd ed. Boston: Addison-Wesley, 1997. 762 p. URL: https://www-cs-faculty.stanford.edu/~knuth/taocp.html

3. Актуальні питання забезпечення цивільного захисту: збірник тез доповідей VIII Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих учених (м. Львів, 2021 р.) / Львівський державний університет безпеки життєдіяльності. Львів: ЛДУ БЖД, 2021.

УДК 004.422.5

Бондар А. Б., здобувачка 2 курсу
спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук,
доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ АЛГОРИТМІВ ПОШУКУ У СТВОРЕННІ СТРАТЕГІЧНИХ ІГОР РЕАЛЬНОГО ЧАСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Стратегічні ігри реального часу (RTS) є одним із найбільш вимогливих жанрів щодо алгоритмічних рішень, оскільки у них необхідно забезпечити одночасну взаємодію великої кількості автономних юнітів. Згідно з матеріалами Game Developer, ефективність RTS напряму залежить від здатності ігрового рушія швидко виконувати пошук маршруту, обробляти запити юнітів та відповідати на динамічні зміни середовища [1]. Саме тому алгоритми пошуку стали базовим інструментом для розробників цього жанру.

Однією з ключових задач у RTS є маршрутизація юнітів, яку найчастіше реалізують за допомогою алгоритму A*. Проте класичний A* зазнає значних труднощів на великих і насичених картах. Матеріали MovingAI показують, що для одночасного переміщення сотень юнітів розробники переходять до кооперативних методів, як-от Cooperative Pathfinding та Windowed HCA*, що дають змогу уникати конфліктів між агентами та покращують узгодженість групового руху [2].

Іншим важливим напрямом є Flow Field Pathfinding, описаний у доповіді GDC Vault. Цей підхід забезпечує не окремий маршрут для кожного юніта, а глобальне поле напрямів, яке зчитують усі агенти. Таке рішення було успішно

реалізоване у грі Supreme Commander і довело свою ефективність під час управління великими арміями, мінімізуючи обчислювальні витрати та підтримуючи стабільний FPS навіть у спрощеній симуляції бою [3].

Формування поведінки штучного інтелекту в RTS також базується на алгоритмах пошуку. За даними AI and Games, сучасні ігри використовують складні дерева рішень, поведінкові дерева та методи реактивного ШІ, що дають змогу NPC не просто слідувати заданим сценаріям, а адаптуватися до змін у грі [4]. Водночас поведінкові дерева (Behaviour Trees) залишаються одним з найкращих способів структурування логіки ШІ завдяки простоті редагування та модульності, що описано на GameAIPro [5].

Альтернативні теоретичні підходи у цьому напрямі охоплюють ієрархічний пошук, евристичні методи, моделі полів потенціалів, а також груповий рух на основі «стадної» поведінки (flocking). На практиці різні методи комбінуються залежно від того, чи потрібна точність маршруту, швидкість обчислення або узгоджений рух групи. Отже, RTS-ігри є унікальним середовищем, де одночасно застосовуються класичні алгоритми пошуку, кооперативні методи та поведінкові моделі ШІ.

Як авторську пропозицію можна виділити використання гібридних навігаційних карт, де Flow Field застосовується для визначення загального напрямку, а локальні маршрути уточнюються спрощеним A* або Jump Point Search. Така комбінація дає змогу зменшити кількість конфліктів між юнітами та підвищити швидкість обробки навігаційних запитів. З практичного погляду це може значно покращити продуктивність RTS на великих картах і забезпечити більш реалістичну поведінку юнітів під час масових битв.

Отже, алгоритми пошуку є центральним механізмом у створенні RTS-ігор, забезпечуючи не лише навігацію, а й інтелектуальну поведінку, координацію групових дій та адаптивність у динамічному середовищі. Їх постійний розвиток сприяє появі більш складних та реалістичних стратегічних систем, що визначає технологічний рівень сучасних ігор цього жанру.

Список використаних джерел

1. Pathfinding in real-time strategy games. *GameDeveloper*. URL: https://www.gamedeveloper.com/programming/pathfinding-in-real-time-strategy-games (дата звернення: 20.11.2025).

2. Cooperative pathfinding for multi-agent systems. *MovingAI*. URL: https://movingai.com/ (дата звернення: 20.11.2025).

3. Flow field pathfinding in games. *GDC Vault.* URL: https://www.gdcvault.com/play/1012386/Flow-Field-Pathfinding (дата звернення: 20.11.2025).

4. Real-time strategy game AI techniques. *AI and Games.* URL: https://aiandgames.com/rts-ai (дата звернення: 20.11.2025).

5. Behaviour trees and decision-making in games. *GameAIPro.* URL: https://www.gameaipro.com (дата звернення: 20.11.2025).