

УДК 004.514:004.93

ОПТИМІЗАЦІЯ НАВІГАЦІЇ NPC У UNREAL ENGINE 5 ЗА ДОПОМОГОЮ ВИКОРИСТАННЯ PATHFINDING АЛГОРИТМІВ

Карасьов М.А., Новіков Ю.С.

e-mail: mykhailo.karasov@nure.ua, yuriy.novikov@nure.ua

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

The purpose of this work is to explore various pathfinding algorithms and their application in game development using the Unreal Engine 5. These algorithms are widely used in different applications to ensure realistic movement of NPCs (non-playable characters). Without such algorithms, game characters would move in a straight line, getting stuck in various objects. Therefore, different types of pathfinding algorithms, such as A*, Dijkstra, Theta*, and others, are widely applied. The optimal use of these algorithms allows for efficient utilization of the end user's computing resources, which is a crucial aspect in the development of any software.

У сучасних ігрових застосунках, особливо в іграх з відкритим світом або великими інтерактивними рівнями, важливим аспектом є навігація неігрових персонажів (NPC)[1]. Ефективна система навігації дозволяє створювати реалістичний ігровий процес, де NPC можуть орієнтуватися в просторі, уникати перешкод і взаємодіяти з оточенням природним чином.

У рушії Unreal Engine 5 використовується вбудована система навігації, яка забезпечує швидкий та ефективний пошук маршрутів для NPC. Основним алгоритмом, що застосовується для пошуку шляху, є алгоритм A* (A-star). Цей алгоритм поєднує підхід пошуку в ширину з евристичною оцінкою, що дозволяє знаходити оптимальний шлях з точки зору витрат на переміщення.

Однак, крім A*, існують й інші методи та алгоритми, які можуть значно покращити як продуктивність програмної системи, так і поведінку NPC. Вибір алгоритму залежить від багатьох факторів, включаючи розмір і складність ігрового світу, кількість NPC, а також необхідний рівень реалістичності їхньої поведінки.

Розглянемо наступні, найбільш широко застосовувані алгоритми, для пошуку шляху:

– алгоритм A*(A – Star) – використовується у Unreal Engine 5 за замовченням. Використовує вартість шляху від початкової точки до поточної;

– алгоритм Дейкстри[2] – алгоритм шукає найкоротший шлях від початкової точки, до усіх можливих вершин, незалежно від кінцевої точки;

– алгоритм Theta* – модифікована версія A*, дозволяє NPC пересуватися більш природньо.

Проведемо дослідження щодо продуктивності при використанні кожного з алгоритмів.

Під час дослідження будуть порівнянні три вище перерахованих алгоритми (див. табл. 1), з різної кількості NPC, за наступними метриками:

- час пошуку шляху (мс) – скільки часу зайняло у NPC прорахувати шлях;
- завантаження CPU (%).

Тестування буде проводитися на рівні Unreal Engine 5, де для NPC будуть розташовані перешкоди (див. рис. 1). На рисунку 1 червоною стрілкою виділено початок руху NPC, зеленою – кінець. NPC має прорахувати маршрут з початку до кінця, та дістатися кінця маршруту. На рівні присутні перешкоди для NPC, тому навігаційним алгоритмам необхідно буде прорахувати складний маршрут, а не просто пряму лінію.

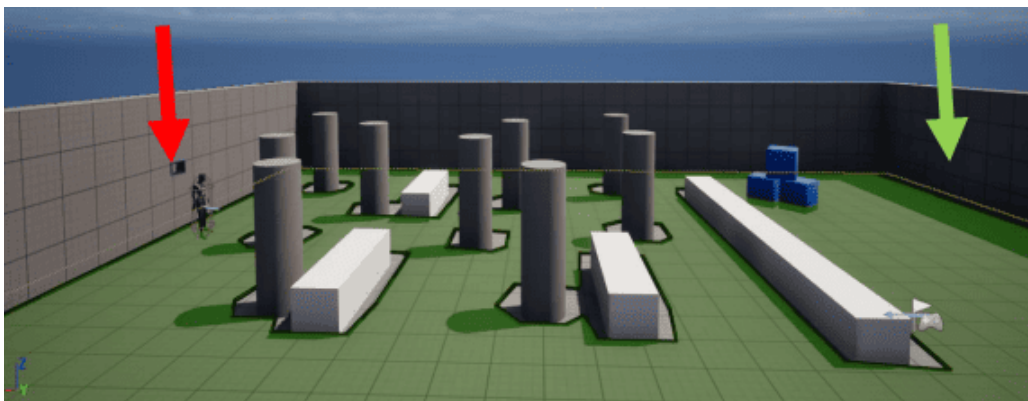


Рисунок 1 – Рівень для NPC, з перешкодами

У якості алгоритму A* використовується функція з коробки MoveTo (A* є вбудованим у Unreal Engine 5 алгоритмом). Інші два алгоритми були реалізовані за допомогою C++ класів.

Таблиця 1 – Продуктивність алгоритмів пошуку шляху

Кількість NPC	A*(Час, мс)	A*(CPU, %)	Дейкстра (Час, мс)	Дейкстра (CPU, %)	Theta*(Час, мс)	Theta*(CPU, %)
1	2 – 4	5 – 8%	6 – 8	10 – 12%	3 – 5	7 – 10%
10	10 – 12	12 – 18%	42 – 51	26 – 32%	16 – 21	18 – 25%
50	45 – 60	35 – 50%	253 – 397	81 – 100%	83 – 122	52 – 70%
100	120 – 160	65 – 85%	800 – 987	100%	220 – 287	75 – 93%

З отриманих результатів можна зробити висновок, що найоптимальнішим алгоритмом є вбудований алгоритм A*, який оптимально використовувати у більшості випадків.

Алгоритм Theta* працює повільніше у порівнянні з алгоритмом A*, але його головна перевага – це більш природна траєкторія руху NPC. Завдяки вдосконаленому механізму вибору маршруту, NPC можуть пересуватися менш прямолінійно, що створює більш реалістичне сприйняття їхньої поведінки. Проте цей алгоритм має вищі обчислювальні витрати, що стає критичним при великій кількості NPC у грі. Тому його доцільно використовувати в тих випадках, коли точність руху є важливішою за швидкість обчислень, наприклад, у сюжетно орієнтованих іграх, де кожен NPC має унікальні анімації та взаємодії з оточенням.

Алгоритм Дейкстри показав найгірші результати, а при великій кількості NPC привів до неможливості подальшої роботи застосунку.

Отже, майже в усіх випадках найкращим вибором залишається алгоритм A*. Він є ефективним, швидко прораховує шлях і не вимагає значних системних ресурсів. Його головна перевага – баланс між продуктивністю та точністю. Крім того, важливо зазначити, що алгоритм A* вже вбудований у Unreal Engine 5 за замовчуванням, що значно спрощує його використання для розробників. Інші алгоритми, такі як Theta* або Дейкстри, вимагають окремої реалізації через написання додаткових класів на C++ або використання зовнішніх бібліотек.

Проте, у тих випадках, коли важливі не лише швидкість, а й плавність руху NPC, можна розглянути використання алгоритму Theta*. Він дозволяє отримати більш реалістичний результат, що може бути корисним у проектах з детальною анімацією персонажів або в стратегічних іграх, де поведінка NPC має виглядати максимально природно. Але його застосування варто обмежувати сценаріями з невеликою кількістю NPC, щоб уникнути зайвого навантаження на систему.

Список використаних джерел:

1. O. Mazurova, O. Samantsov, O. Topchii and M. Shirokopetleva, A Study of Optimization Models for Creation of Artificial Intelligence for the Computer Game in the Tower Defense Genre, 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 491-496, doi: 10.1109/PICST51311.2020.9468057.

2. Логвін І. Ю. Дослідження та використання підходів створення ігрового ШІ, для реалізації поведінки монстрів в ігрових додатках / І. Ю. Логвін // Радіоелектроніка та молодь у ХХІ столітті : матеріали 27-го Міжнар. молодіж. форуму, 10–12 травня 2023 р. – Харків : ХНУРЕ, 2023. – Т. 6, ч.1. – С. 275-276.