

IMPLEMENTING THE A* ALGORITHM IN ETERNAL VIGIL⁸

Serkan Sadulov – Student

Faculty of Natural Sciences and Education,
University of Ruse “Angel Kanchev”
Tel.: +359878925635
E-mail: serkansadulov@gmail.com

Mustafa Mustafov – Student

Faculty of Natural Sciences and Education,
University of Ruse “Angel Kanchev”
Tel.: +359890569310
E-mail: mustafasunayev607@gmail.com

Pr. Assist. Prof. Kamelia Shoilekova, PhD

Department of Informatics and Information technologies,
University of Ruse “Angel Kanchev”
Tel.: +359887859224
E-mail: kshoilekova@uni-ruse.bg

Abstract: This paper delves into the exploration of advanced methods for enhancing pathfinding mechanisms in the context of game development, with a particular focus on optimizing the movement of monsters and mobs within the game world. Emphasizing the application of the A* algorithm, this research aims to assess its effectiveness and elucidate its potential advantages in enhancing the gaming experience. The study primarily employs the A* algorithm to develop a pathfinding solution that intelligently navigates monsters and mobs through complex game environments. The investigation extends to both theoretical analysis and practical implementation, leveraging the algorithm's capabilities to find optimal routes while considering different types of factors. Utilizing computational tools and gaming frameworks, the research employs the A* algorithm to model and evaluate pathfinding scenarios. This includes the creation of a simplified virtual environment to test the algorithm's performance and the integration of A* within the context of an actual game development project. The paper seeks to highlight the advantages and commercial viability of employing the A* algorithm for enhancing pathfinding in the realm of game development. Through empirical analysis and simulation, the study aims to provide valuable insights into the potential benefits and practical implications of this approach.

Keywords: A* algorithm, Dijkstra's algorithm, game, shortest paths.

ВЪВЕДЕНИЕ

A* е алгоритъм за намиране на най-кратък път в граф, комбинирайки G-стойност (дължина на пътя от начало) и H-стойност (предполагаемо разстояние до целта). Използва се в игри за интелигентно движение на герои (NPC), подобрявайки изцяло игровото изживяване. Тъй като е подходящ за разнообразни терени и сценарии, A* е ключов инструмент в дизайна на игрите и оптимизацията на игралния опит [5].

В разработката на играта Eternal Vigil, една от най-важните функционалности е как чудовищата достигат до главния герой, кулите или кристала с цел да ги атакуват. Това е постигнато благодарение на имплементирането на A* (А-стар) алгоритъма, който е подобен на алгоритъма на Дейкстра, само че при обхождането на графа и намирането на най-кратък път използва евристична функция, която дава приоритет на възлите.

⁸ Докладът е представен на конференция на Русенския университет на 27 октомври 2023 г. в секция „Математика, информатика и физика“ с оригинално заглавие на български език: ИЗПОЛЗВАНЕ НА A* АЛГОРИТЪМА В ETERNAL VIGIL.

ИЗЛОЖЕНИЕ

Същност на A* алгоритъма

A* алгоритъмът е базиран на комбинация от два основни компонента - G-стойност (дължина на пътя от началната точка) и H-стойност (предполагаемо разстояние до целта) и е разширение на алгоритъма на Дейкстра за намиране на най-кратък път. Тази комбинация прави A* ефективен алгоритъм за търсене на най-кратък път в графи и графоподобни структури. Тези две стойности функционират по следния начин:

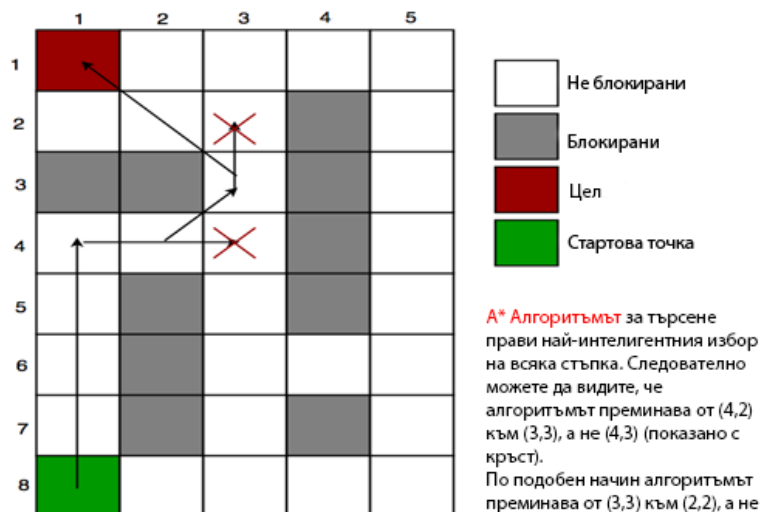
1. G-стойност (път от началото) - актуалната дължина на пътя от началната точка (началния връх) до текущия връх, който се разглежда. Тази стойност се измерва във върхове, ребра или други метрики, в зависимост от приложението. Идеята е да се определи колко дълъг е изминатият път.

2. H-стойност (предполагаемо разстояние до целта) - оценка за разстоянието от текущия връх до целта, като се използва някаква евристика (предположение). Тази стойност се определя на базата на предположения и колко близо е текущият връх до целта. Идеята е да се определи колко близо (или далече) е текущото местоположение от целта.

При A* алгоритъма се изчислява общата стойност за всяка възможна опция (съседен връх), като сумира G-стойността и H-стойността (1). Този общ резултат се нарича F-стойност и се използва за определяне на приоритета на възможните опции. Разглежданият алгоритъм обикновено избира опцията с най-ниска F-стойност, тъй като това вероятно е най-добрият кандидат за следващата стъпка [1], [2], [3].

$$F(x) = G(x) + H(x) \quad (1)$$

По този начин A* алгоритъмът комбинира информацията за текущия път (G-стойност) с информация за това колко близо (или далече) е текущият връх до целта (H-стойност) и търси най-краткия път, като приоритизира опциите въз основа на този комбиниран критерий. Този алгоритъм е широко използван в различни области, включително географско информационно системи, видеоигри, роботика и други.



Фиг. 1. Пример на работа на алгоритъма

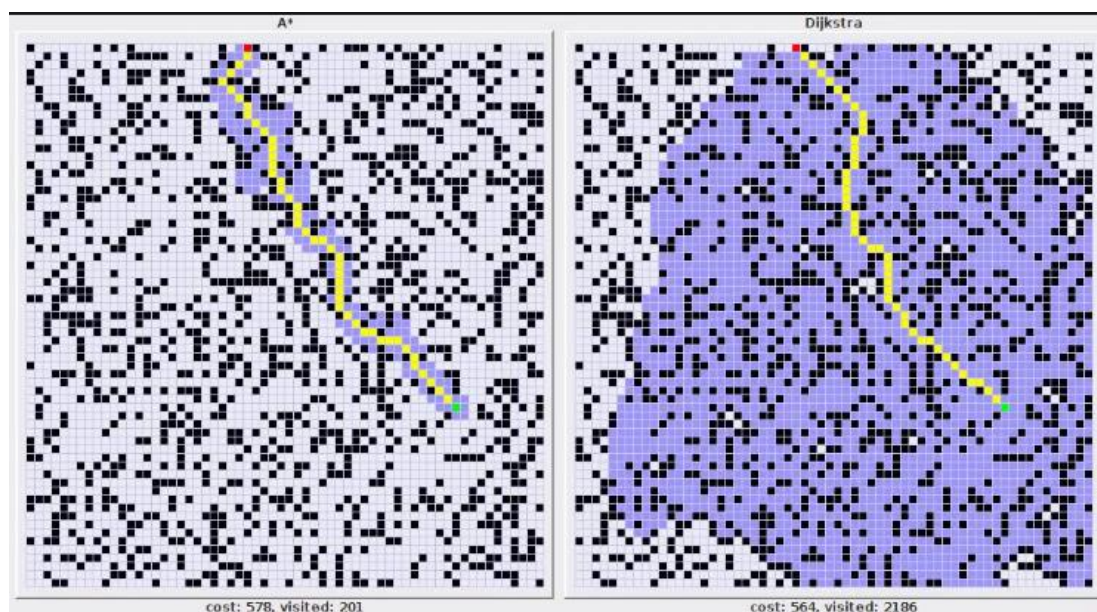
Предимствата на A* алгритъма пред алгоритъма Дейкстра

Алгоритъмът A* се предпочита пред алгоритъма на Дейкстра при разработката на игри поради неговата ефективност във време и памет. Той е по-бърз, подходящ за различни игрови терени, лесно се адаптира към динамични среди и е предназначен за намиране на най-кратки пътища (Фиг. 2).

1. Подходящ за игри с различни терени: A^* лесно се адаптира към игри с разнообразни терени и препятствия. В случай, че се използва подходяща евристична функция, може да се твърди, че алгоритъмът намира пътища, които избягват препятствия или се приспособяват към специфичните изисквания на играта.

2. Интерактивност и динамичност: в игрите често е необходимо да се пресмятат пътища в реално време, като се вземат предвид движенията на играчите и други динамични събития. A^* може лесно да се актуализира при всяко изменение в средата, без да е необходимо да се пресмята целия граф, както при алгоритъма на Дейкстра.

3. Подходящ за намиране на най-кратки пътища: A^* се използва предимно за намиране на най-кратки пътища, което често е ключов аспект в игрите. Този алгоритъм предоставя възможност за бързо и ефективно намиране на пътища, които са оптимални спрямо целта на играта [4].



Фиг. 2. Изобразено обхождането чрез A^* и Дейкстра

Фиг. 2 показва намиране на път през лабиринт. Отляво е " A^* " (А-звезда), а отдясно е "Алгоритъмът на Дейкстра". И двата алгоритъма са проектирани, така че да намират най-краткия път между две точки в граф, който в този случай е представен чрез решетка, където черните квадрати са препятствия, а белите квадрати са възможни пътища.

Ключовите разлики между двата алгоритъма:

1. Хевристична функция (A^*): A^* използва хевристика, за да оцени разходите от текущата точка до крайната точка, което му помага да приоритизира кои пътища да изследва. Той комбинира реалния разход от началото до текущата точка с оценката на разходите от текущата точка до края. Това е причината A^* да е посетил по-малко квадрати (201) в сравнение с Дейкстра (2186) преди да намери най-краткия път.

2. Изследване на пътя (Дейкстра): Алгоритъмът на Дейкстра не използва хевристика; това е неориентирано търсене, което просто изследва всички възможни пътища и избира този с най-ниските натрупани разходи. Това води до по-широка област на търсене, което е визуализирано със сините квадрати, показващи всички възли, които алгоритъмът е проверил.

3. Производителност: Както се вижда от броя на посетените възли, A^* е обикновено по-ефективен за намиране на път в решетка с единична дестинация, защото не изследва толкова много ненужни пътища, колкото алгоритъма на Дейкстра.

4. Обща стойност: И двата алгоритъма са намерили път със сравнително подобна обща стойност (A^* с 578, Дейкстра с 564), но поради хевристичната функция, A^* го е направил по-ефективно.

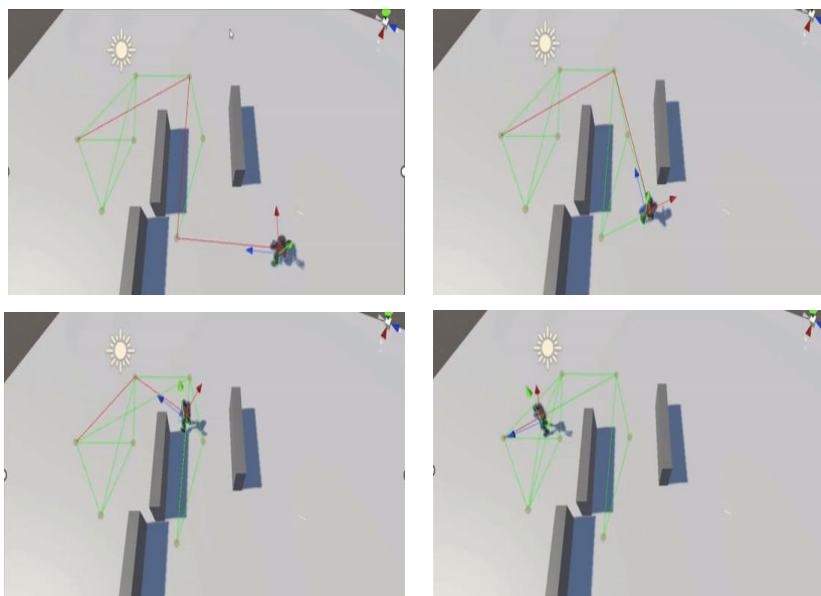
Жълтият път във всяка от картинките представлява най-краткия път, намерен от съответния алгоритъм от началната точка до дестинацията. Лилавите квадрати на страната на A^* показват възлите, които алгоритъмът A^* е преценил, че си струва да изследва въз основа на хевристичната функция. Сините квадрати на страната на Дейкстра показват всички възли, които са били проверени по време на търсенето.

Имплементиране на A^* алгоритъма в играта Eternal Vigil

Благодарение на алгоритъма A^* , който успешно е приложен в играта Eternal Vigil, се открива възможност за бързо и надеждно намиране на оптимални пътища в игровата среда на разработеното приложение. Този алгоритъм изгражда основния стълб на функционалността в играта, като насочва всичките чудовища (NPCs) по кратки и оптимални маршрути до техните поставени цели.

Този алгоритъм е избран заради ефективното му използване за намиране на най-кратък път в тегловен граф с множество препятствия. Той е подходящ за съчетаване със сложните изисквания, заложили в същността на играта, предоставяйки едно добро възможно решение за движение и навигация в игровия свят (Фиг. 3).

Използван е в играта за управление на предвижването на чудовищата, намиране на път до кристала, играчите и различните мишени, които са им зададени.



Фиг. 3 – Начин на използване на A^* в играта

ЗАКЛЮЧЕНИЕ

Алгоритъмът A^* представлява мощен инструмент за намиране на най-кратки пътища в игрите, като комбинира G-стойност (дължина на пътя от началото) и H-стойност (предполагаемо разстояние до целта). Подходящ е за различни сценарии и среди и се използва широко в разработката на игрите и оптимизацията на игралния опит. Съществената разлика между A^* алгоритъма и алгоритъма на Дейкстра е, че A^* предлага по-голяма ефективност, адаптивност към динамични среди и способността да намира най-кратки пътища, без да е необходимо да се пресметне трасето по целия граф, което го прави предпочитан избор при разработката на игри.

БЛАГОДАРНОСТ

Това изследване е подкрепено от проект 23-ФПНО-02 „Изследване на ефективни механизми за управление на знания, прилагани в софтуерното инженерство при създаване на проекти с гъвкави методологии“, финансиран от Фонд „Научни изследвания“ на Русенския университет.

REFERENCES

Magzhan, Kairanbay, and Hajar Mat Jani. "A review and evaluations of shortest path algorithms." *Int. J. Sci. Technol. Res* 2.6 (2013): 99-104.

Darmann, A., Pfersch, U., Schauer, J. (2014). The Shortest Path Game: Complexity and Algorithms. In: Diaz, J., Lanese, I., Sangiorgi, D. (eds) Theoretical Computer Science. TCS 2014. Lecture Notes in Computer Science, vol 8705. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-44602-7_4.

Patel, A. *Introduction to A**. Retrieved April 29, 2016, from <http://theory.stanford.edu/~amitp/GameProgramming/concave1.png>.

Patel, A. *Introduction to A**. Retrieved April 29, 2016, from <http://theory.stanford.edu/~amitp/GameProgramming/concave2.png>.

Java t point, Informed Search Algorithms, Copyright 2011-2021 www.javatpoint.com. All rights reserved. Developed by JavaTpoint, <https://www.javatpoint.com/ai-informed-search-algorithms>.

A* Search. *Brilliant.org*. Retrieved 15:57, October 5, 2023, from <https://brilliant.org/wiki/a-star-search/>.