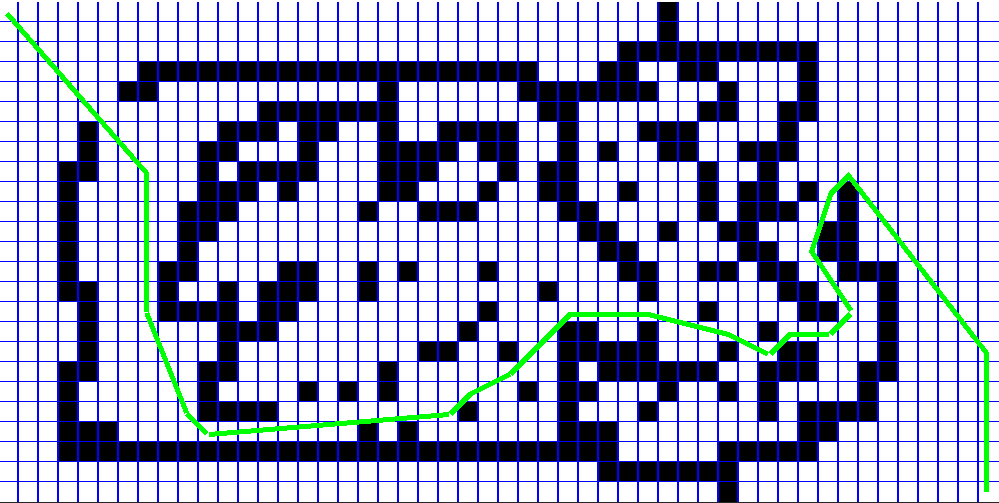
PRACA PRZEJŚCIOWA INŻYNIERSKA

PORÓWNANIE ALGORYTMÓW  
A\* ORAZ THETA\*



Damian Wysokiński, 286699

Prowadzący: dr inż. Andrzej Chmielniak

Spis treści

[Wprowadzenie 3](#_Toc19216813)

[A\* 3](#_Toc19216814)

[Theta\* 3](#_Toc19216815)

[Narzędzie programistyczne 4](#_Toc19216816)

[Schematy blokowe 5](#_Toc19216817)

[A\* 5](#_Toc19216818)

[Theta\* 6](#_Toc19216819)

[Porównanie długości wyznaczonych ścieżek 7](#_Toc19216820)

[Konfiguracja #0 7](#_Toc19216821)

[Konfiguracja #1 9](#_Toc19216822)

[Konfiguracja #2 11](#_Toc19216823)

[Konfiguracja #3 13](#_Toc19216824)

[Konfiguracja #5 15](#_Toc19216825)

[Konfiguracja #6 17](#_Toc19216826)

[Zestawienie przybliżonych wyników 19](#_Toc19216827)

[Wnioski 20](#_Toc19216828)

# Wprowadzenie

## A\*

Algorytm wyznaczający najkrótszą ścieżkę między 2 węzłami w grafie pod warunkiem, że taka ścieżka istnieje. Kształt ścieżki jest zależny od wybranej heurystyki.

Algorytm realizuje minimalizację funkcji:

, gdzie oznacza odległość po przebytej ścieżce od węzła początkowego do danego miejsca – jest zmienna, a oznacza niezmienną odległość obliczoną przez heurystykę między danym węzłem a węzłem będącego celem.

W swojej symulacji wykorzystałem 2 heurystyki: euklidesową i manhattańską.

Ścieżka powstaje w następujący sposób: każdy sprawdzony węzeł przez A\* ma informacje o poprzednim węźle, z którego algorytm dotarł do niego. I na tej podstawie można odtworzyć najlepszą ścieżkę od węzła będącego celem do węzła będącego początkiem. Warto tu zaznaczyć, że w A\* poprzednikiem węzła może być tylko węzeł leżący w sąsiedztwie danego węzła.

## Theta\*

Theta\* jest algorytmem bardzo zbliżonym do A\*. Jedyna różnica polega na tym, że poprzednikiem danego węzła może być węzeł niebędący w najbliższym sąsiedztwie danego węzła. Dzieje się tak jeśli korzystniejsza jest droga od poprzednika poprzednika do danego węzła, z pominięciem poprzednika, niż droga przez każdy z tych węzłów po kolei. Przy analizowaniu Theta\* na równomiernej siatce 2D można zauważyć jest to algorytm poruszający się po mapie pod dowolnym kątem. A\* natomiast porusza się tylko w kierunkach będącymi wielokrotnościami kąta 45°.

## Narzędzie programistyczne

Projekt napisałem w języku C++ w środowisku Visual Studio. Do pokazania zachowania się algorytmu wykorzystałem bibliotekę SFML (Simple and Fast Multimedia Library). [Link do repozytorium](https://github.com/damianski794/Przejsciowka_A_Star/)

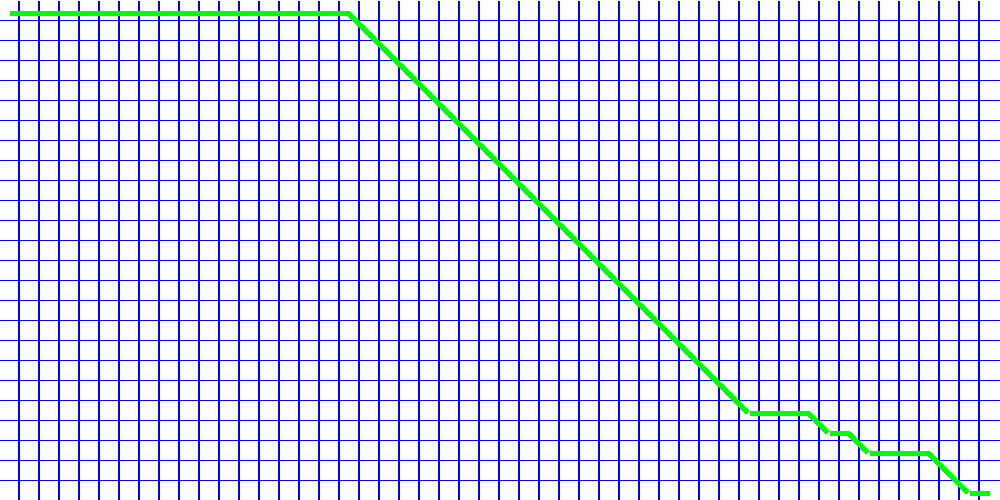
# Schematy blokowe

## A\*

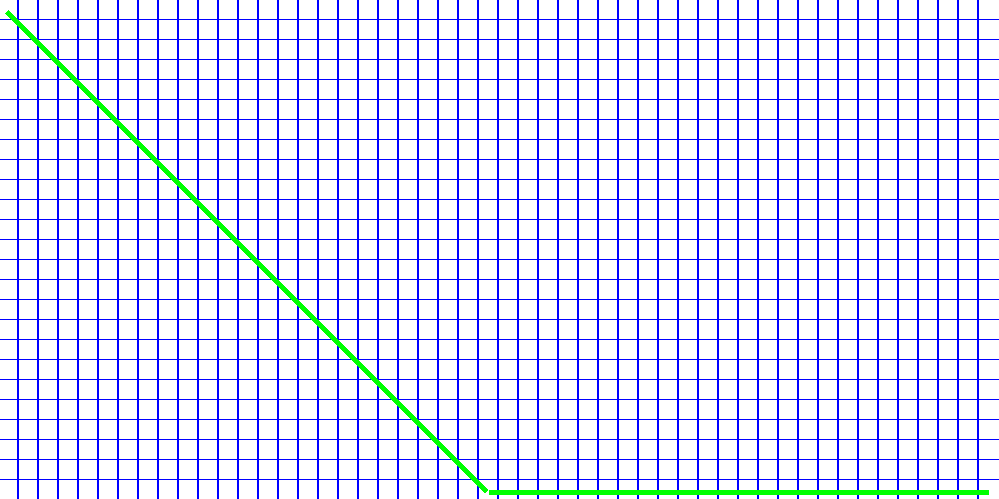
## Theta\*

# Porównanie długości wyznaczonych ścieżek

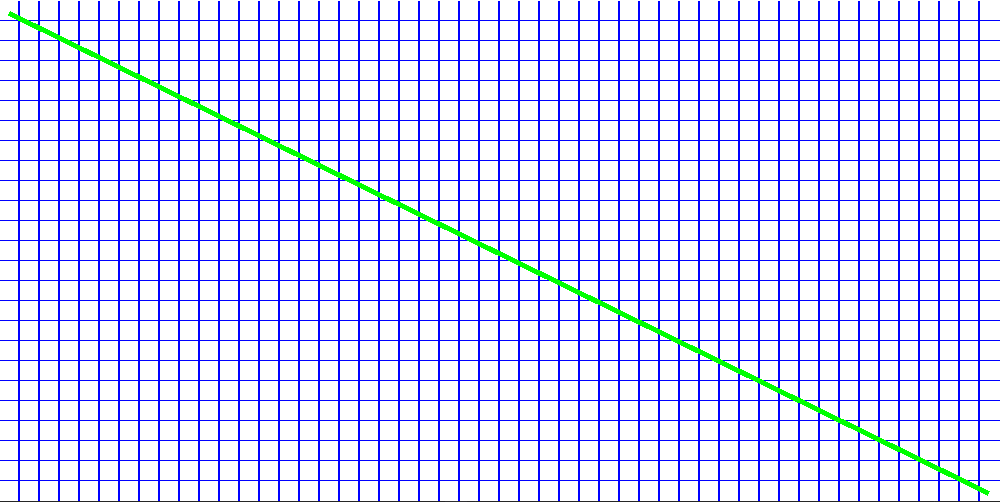
## Konfiguracja #0



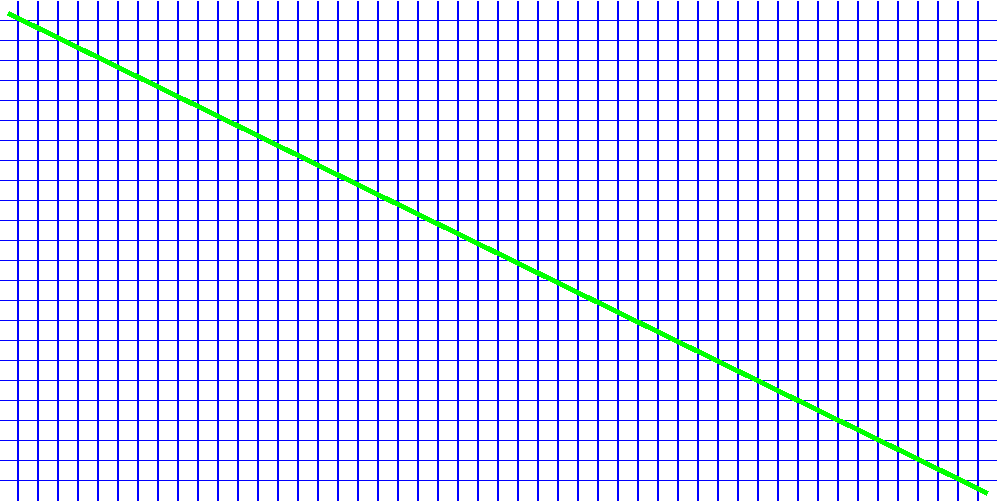
A\* h. euklidesowa, dł. ścieżki 58.9412



A\* h. manhattańska, dł. ścieżki 58.9412

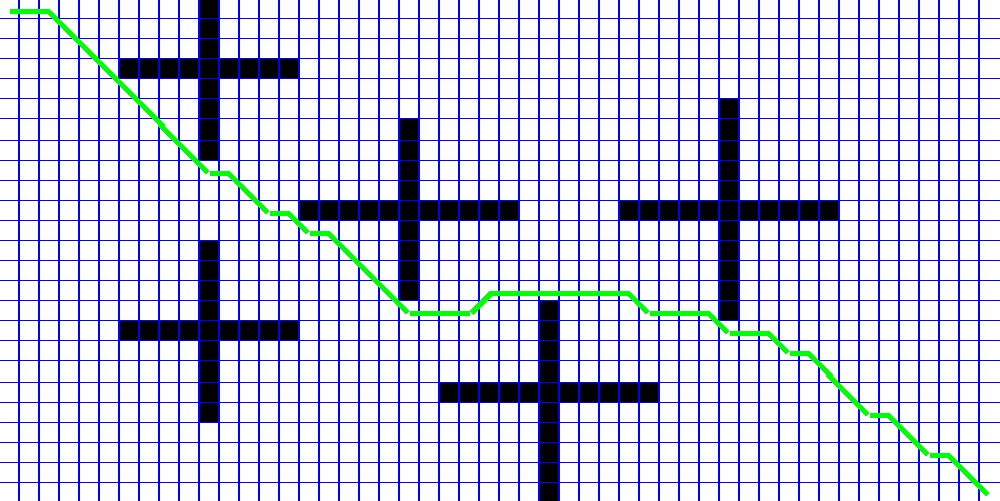


Theta\* h. euklidesowa, dł. ścieżki 54.5619

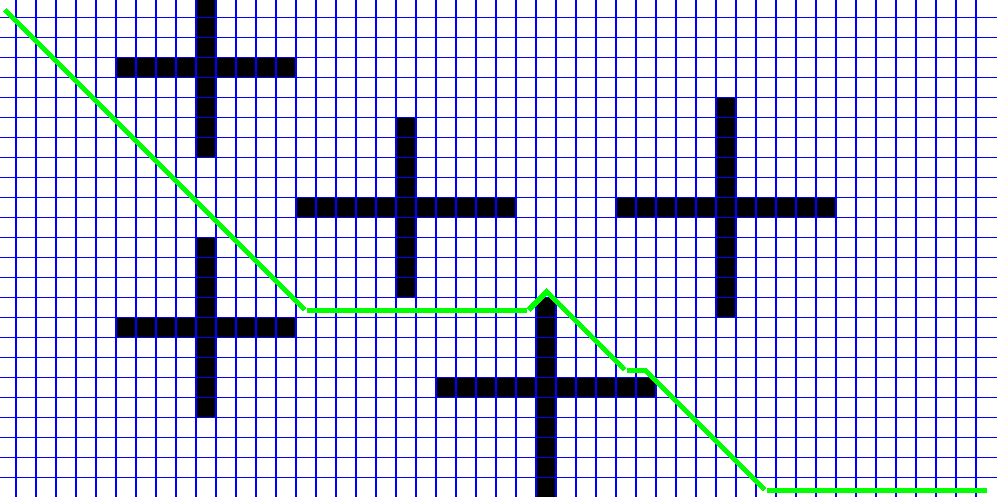


Theta\* h. manhattańska, dł. ścieżki 54.5619

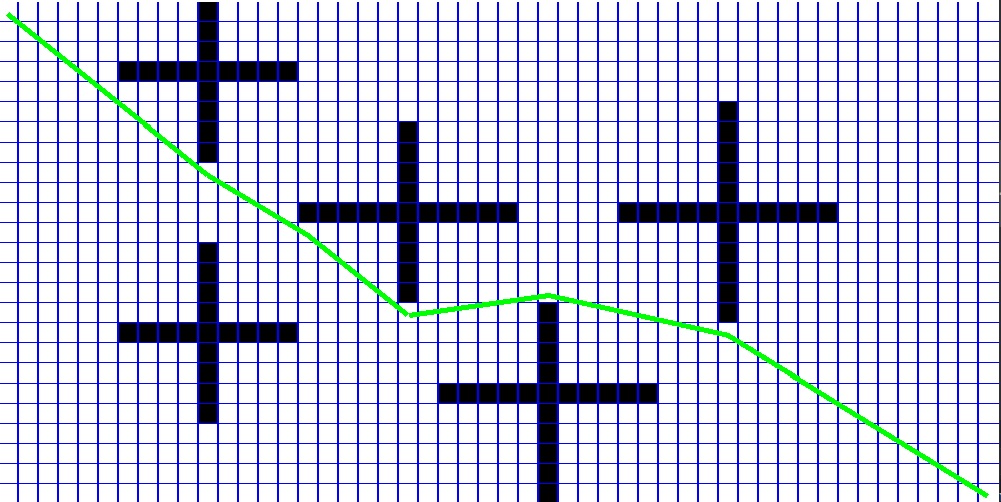
## Konfiguracja #1



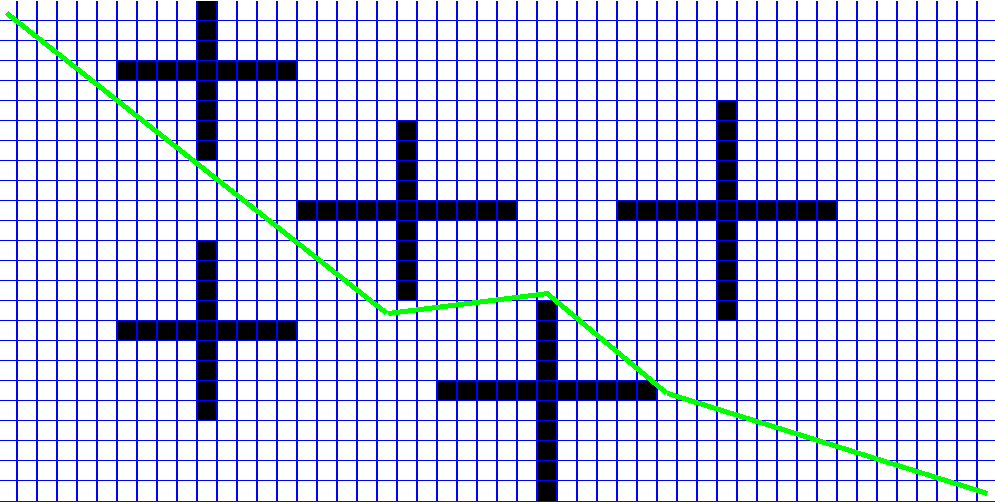
A\* h. euklidesowa, dł. ścieżki 59.7696



A\* h. manhattańska, dł. ścieżki 59.7696

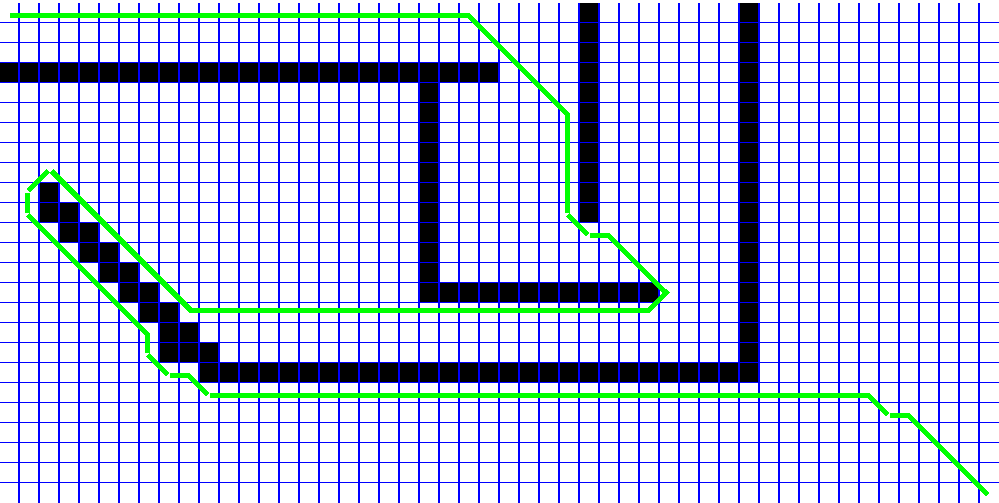


Theta\* h. euklidesowa, dł. ścieżki 56.5953

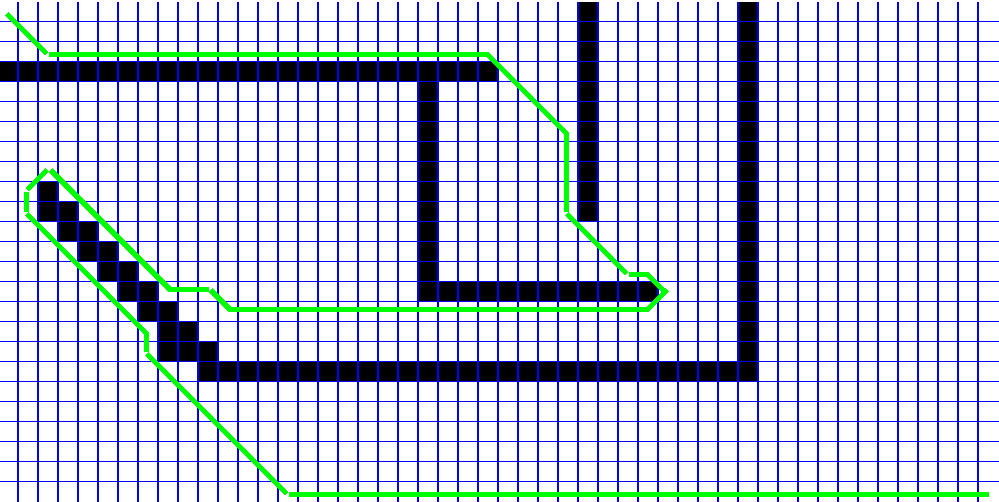


Theta\* h. manhattańska, dł. ścieżki 56.843

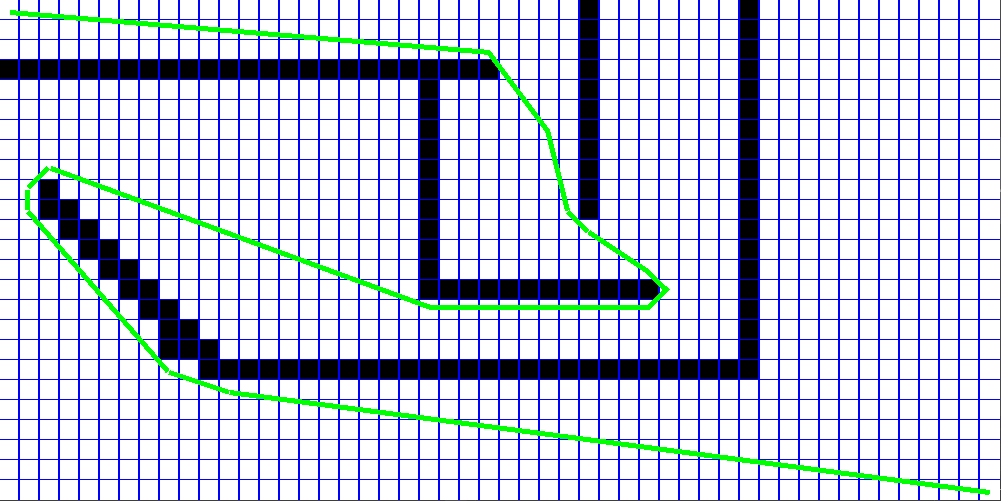
## Konfiguracja #2



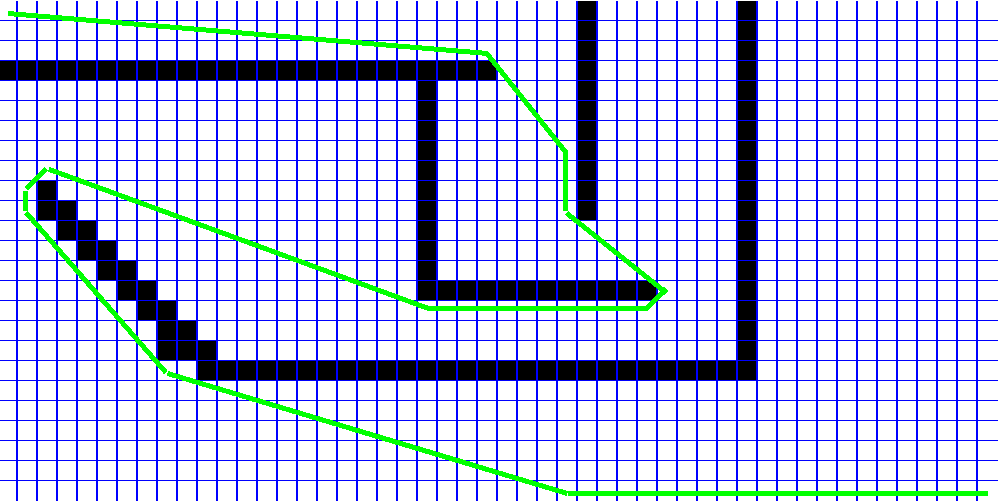
A\* h. euklidesowa, dł. ścieżki 132.841



A\* h. manhattańska, dł. ścieżki 132.255

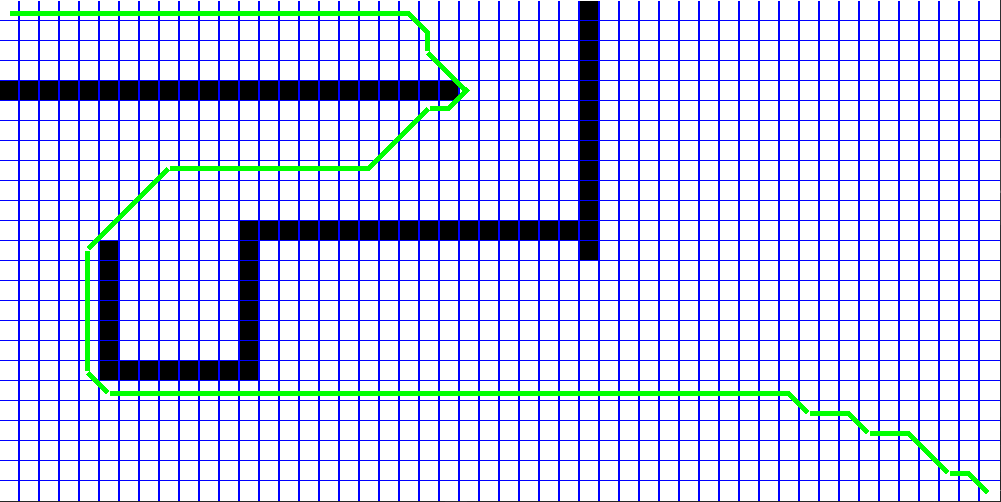


Theta\* h. euklidesowa, dł. ścieżki 97.8938

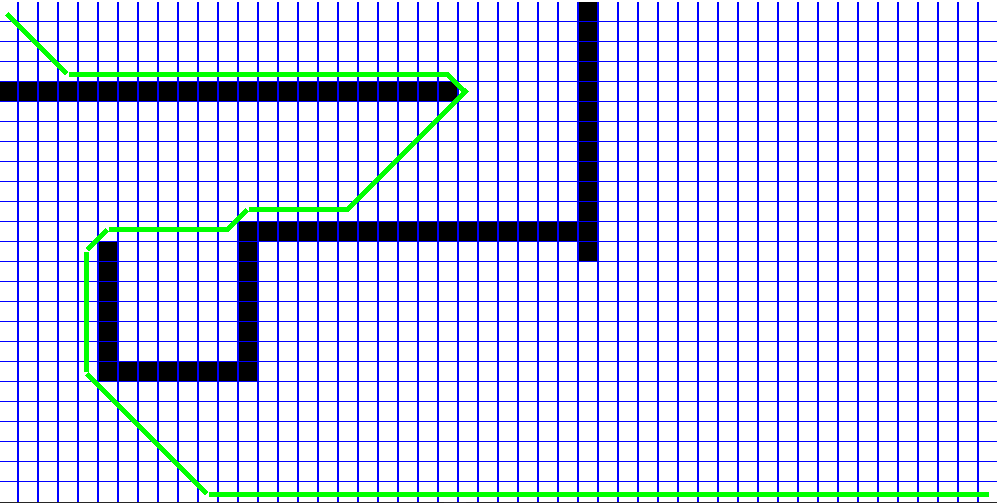


Theta\* h. manhattańska, dł. ścieżki 127.477

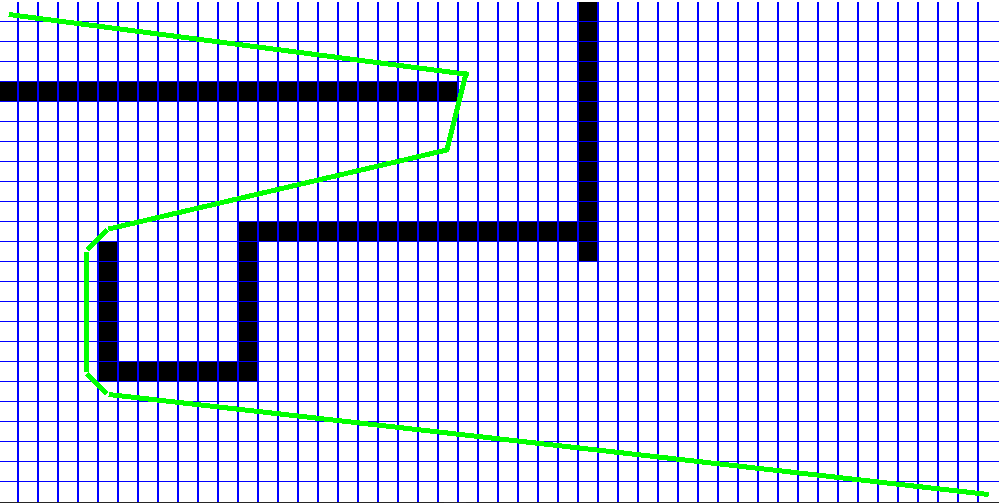
## Konfiguracja #3



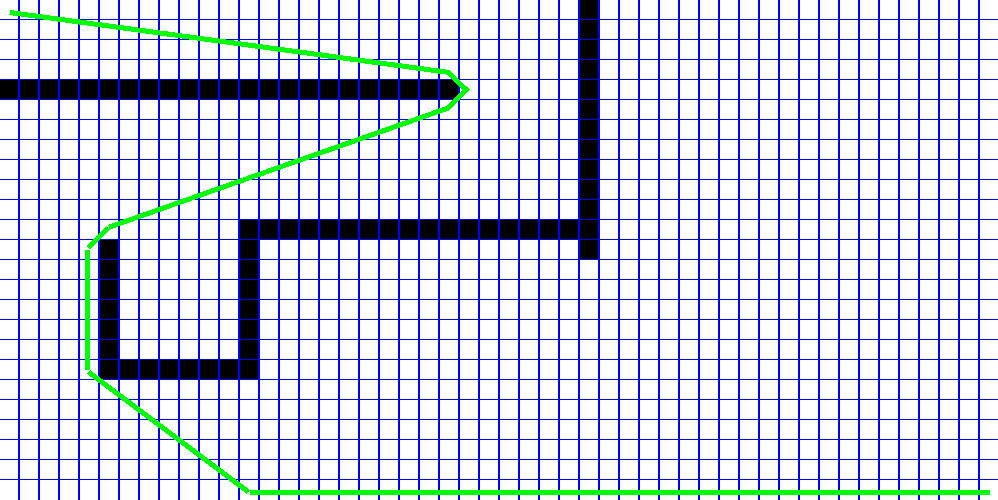
A\* h. euklidesowa, dł. ścieżki 101.042



A\* h. manhattańska, dł. ścieżki 100.456

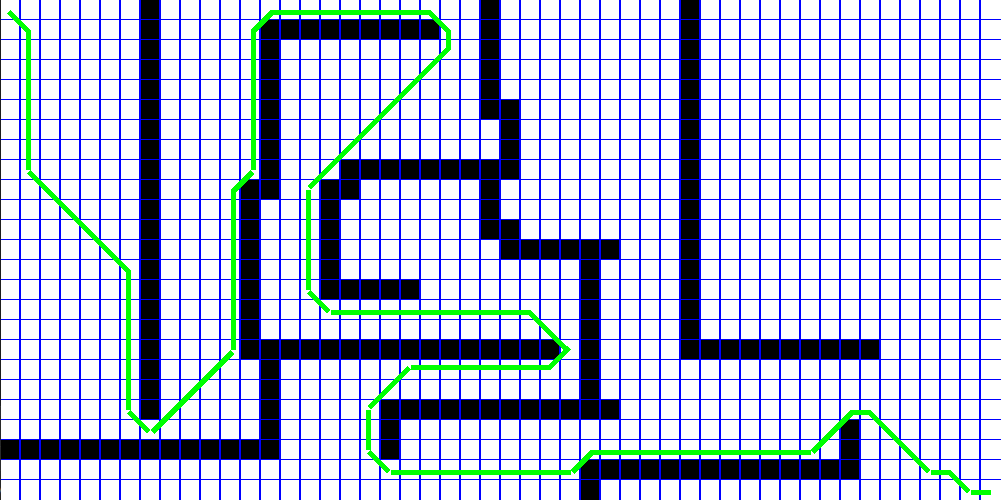


Theta\* h. euklidesowa, dł. ścieżki 97.8938

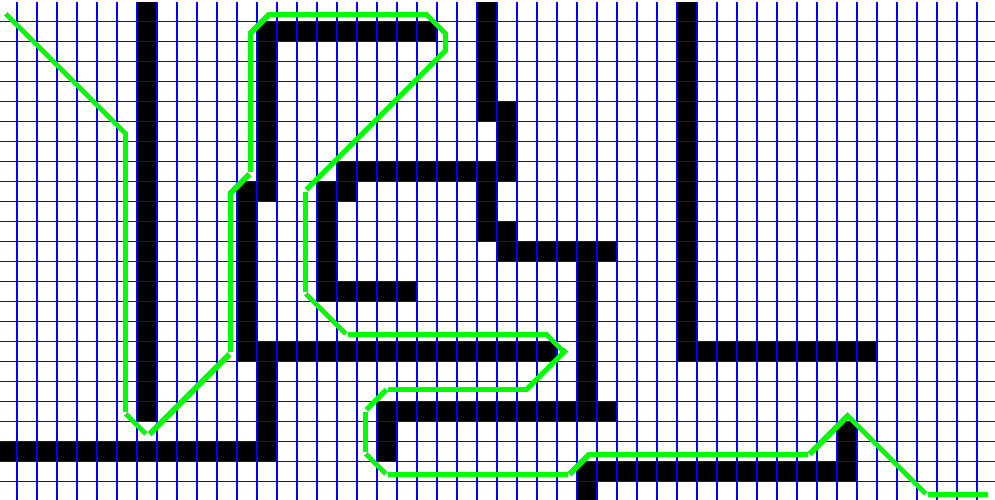


Theta\* h. manhattańska, dł. ścieżki 97.474

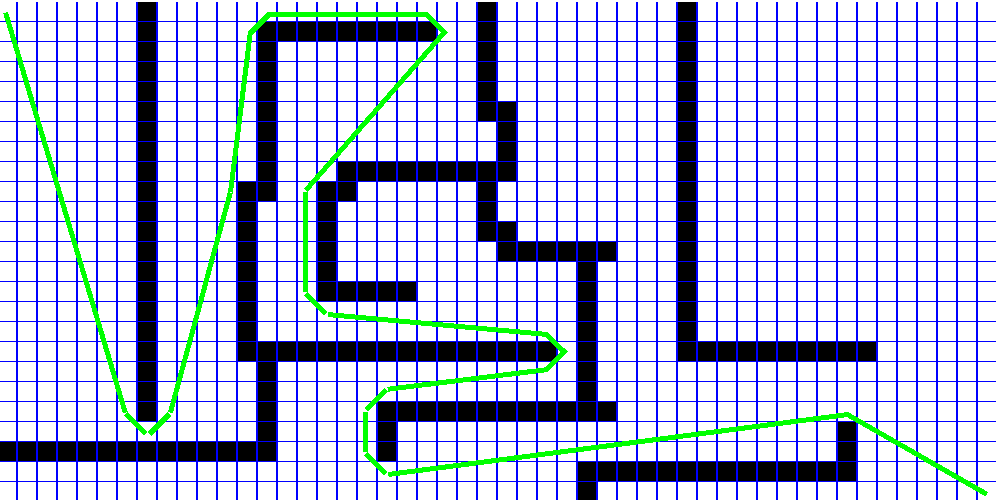
## Konfiguracja #5



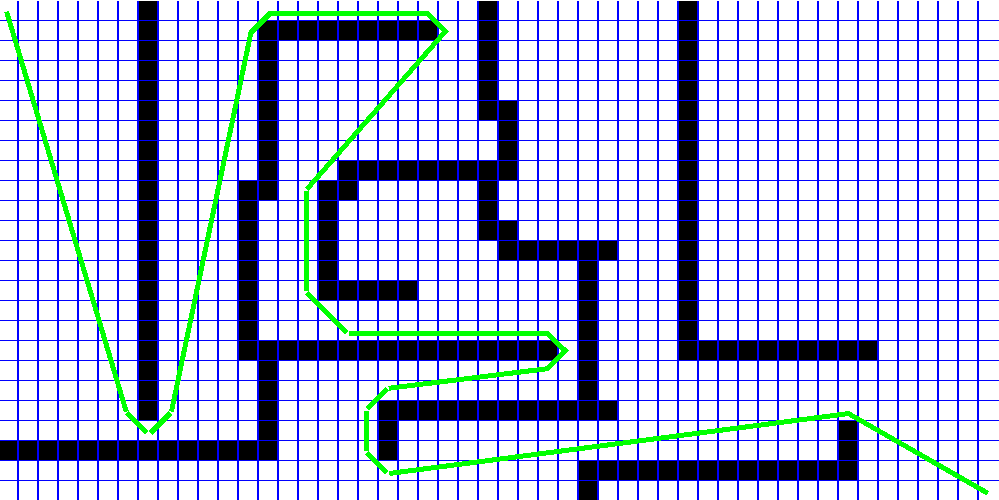
A\* h. euklidesowa, dł. ścieżki 134.498



A\* h. manhattańska, dł. ścieżki 134.498

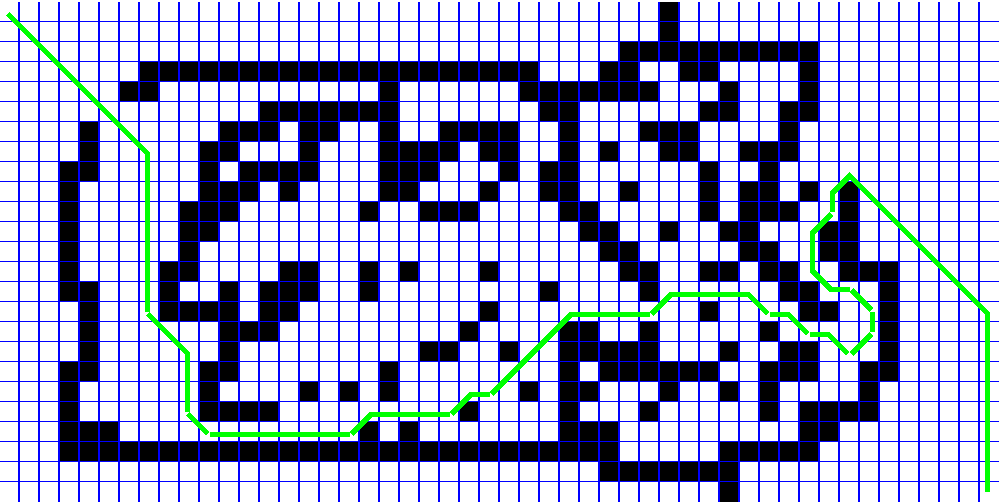


Theta\* h. euklidesowa, dł. ścieżki 129.067

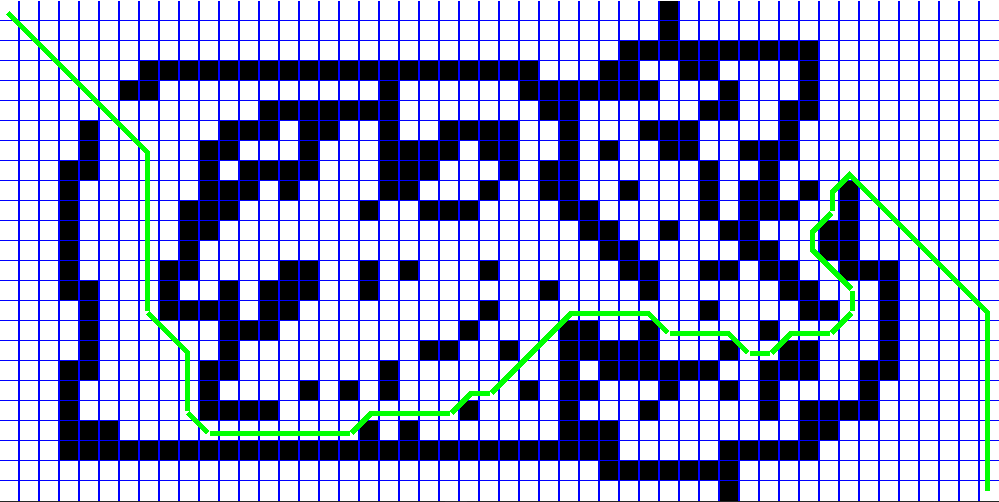


Theta\* h. manhattańska, dł. ścieżki 129.389

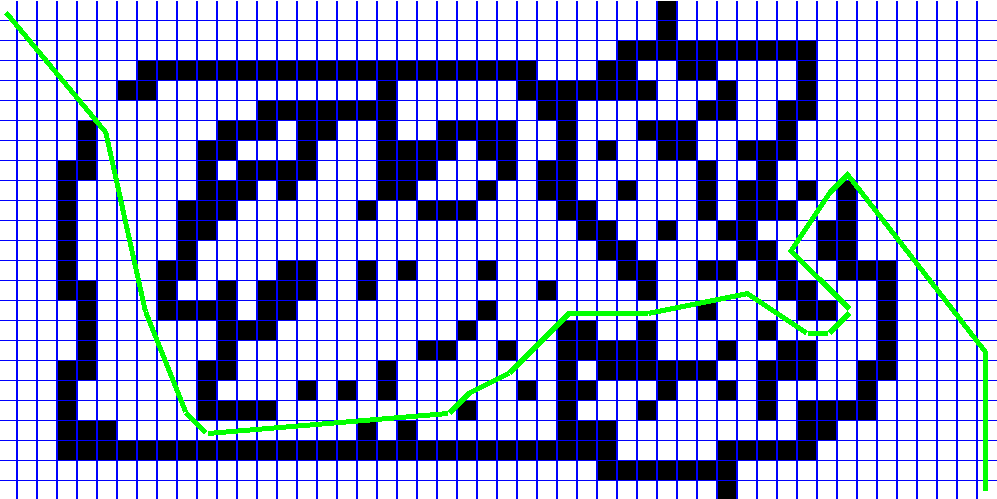
## Konfiguracja #6



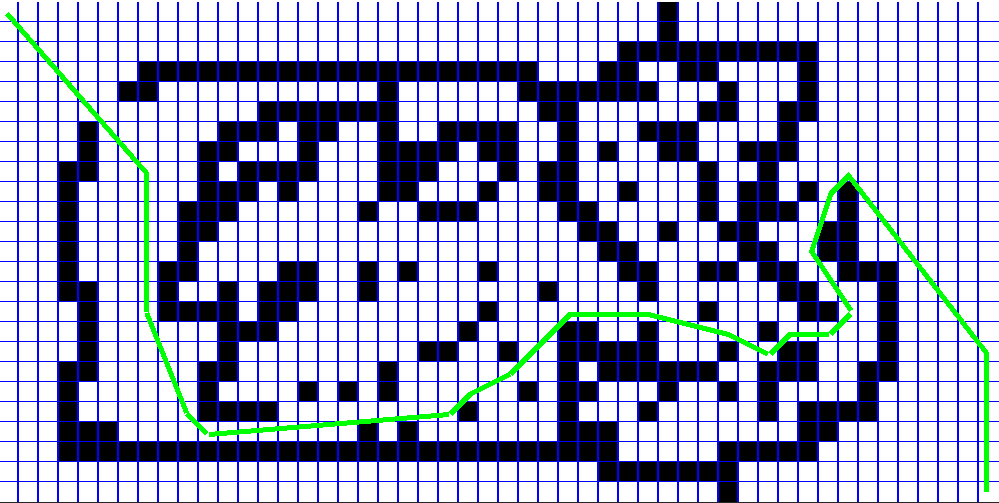
A\* h. euklidesowa, dł. ścieżki 92.2549



A\* h. manhattańska, dł. ścieżki 88.8406



Theta\* h. euklidesowa, dł. ścieżki 86.5466



Theta\* h. manhattańska, dł. ścieżki 86.1354

# Zestawienie przybliżonych wyników

(dokładne wyniki w [*/SPRAWOZDANIE/dane\_podsumowanie.xlsx*](https://github.com/damianski794/Przejsciowka_A_Star/blob/master/SPRAWOZDANIE/dane_podsumowanie.xlsx))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mapa | A\* | | | | | | Theta\* | | | | | | |
| h. euklidesowa | | | h. manhattańska | | | h. euklidesowa | | | h. manhattańska | | | |
| #0 | 58,94 | 236,3 | 235,8 | 58,94 | 17,7 | 17,2 | 54,56 | 235,6 | 233,8 | 54,56 | 19,8 | 19,5 |
| #1 | 59,77 | 166,1 | 165,3 | 59,77 | 41,5 | 41,0 | 56,60 | 147,3 | 146,3 | 56,84 | 41,4 | 40,7 |
| #2 | 132,84 | 109,2 | 108,6 | 132,26 | 62,3 | 59,3 | 126,84 | 116,3 | 114,9 | 127,48 | 68,1 | 66,7 |
| #3 | 101,04 | 199,2 | 108,6 | 100,46 | 36,1 | 34,9 | 97,90 | 185,5 | 183,7 | 97,47 | 40,3 | 39,4 |
| #5 | 134,50 | 49,2 | 47,4 | 134,50 | 45,7 | 43,3 | 129,07 | 48,4 | 47,7 | 129,39 | 48,7 | 48,1 |
| #6 | 92,25 | 101,0 | 98,2 | 88,84 | 87,3 | 86,1 | 86,55 | 108,0 | 105,4 | 86,14 | 92,8 | 91,6 |
|  | dł. ścieżki | czas (ms) | czas  (ms  prio.) | dł. ścieżki | czas (ms) | czas (ms  prio.) | dł. ścieżki | czas (ms) | czas (ms  prio.) | dł. ścieżki | czas (ms) | czas (ms  prio.) |

* Dla A\* wykorzystanie heurystyki manhattańskiej zamiast euklidesowej przyśpiesza algorytm około 1.7 razy
* Dla Theta\* wykorzystanie heurystyki manhattańskiej zamiast euklidesowej przyśpiesza algorytm około 1.4 razy
* Dla A\* heurystyka manhattańska wyznacza krótsze ścieżki o średnio 0.8%
* W przypadku Theta\* wybór heurystyki ma pomijalny wpływ (0.045%) na długość ścieżki
* Theta\* wyznacza znacznie krótsze ścieżki od A\*: średnio 5,4% dla heurtystyki euklidesowej i 4.5% dla manhattańskiej
* Ustawienie najwyższego priorytetu na aplikację i zabicie nieistotnych procesów przyśpiesza algorytm średnio o 1,76%

# Wnioski

1. Theta\* wyznacza wyraźnie krótsze ścieżki w porównaniu do A\*
2. Dla A\* minimalnie krótsze ścieżki daje wykorzystywanie heurystyki manhattańskiej
3. Dla Theta\* wybór heurystyki nie ma większego wpływu na długość ścieżki
4. Dla obu algorytmów heurystyka manhattańska wyraźnie skraca czas obliczeń
5. Ustawienie najwyższego priorytetu dla aplikacji oraz zabijanie niepotrzebnych procesów daje niewielki wpływ na szybkość obliczeń
6. Do zadań, gdzie ważna jest szybkość obliczeń, a nie dokładność, wybrałym A\* z heurystyką manhattańską
7. Do zadań, gdzie istotna jest precyzja i szybkość, wybrałbym Theta\* z heurystyką manhattańską

Kod źródłowy:

* <https://github.com/damianski794/Przejsciowka_A_Star>

Bibliografia:

* <https://arxiv.org/ftp/arxiv/papers/1401/1401.3843.pdf>
* [https://pl.wikipedia.org/wiki/Algorytm\_A\*](https://pl.wikipedia.org/wiki/Algorytm_A*)
* [https://en.wikipedia.org/wiki/Theta\*](https://en.wikipedia.org/wiki/Theta*)
* <https://gamedev.stackexchange.com/questions/75158/line-of-sight-on-a-2d-grid>