Planowanie drogi robota, algorytm A*

Karol Sydor

13 maja 2008

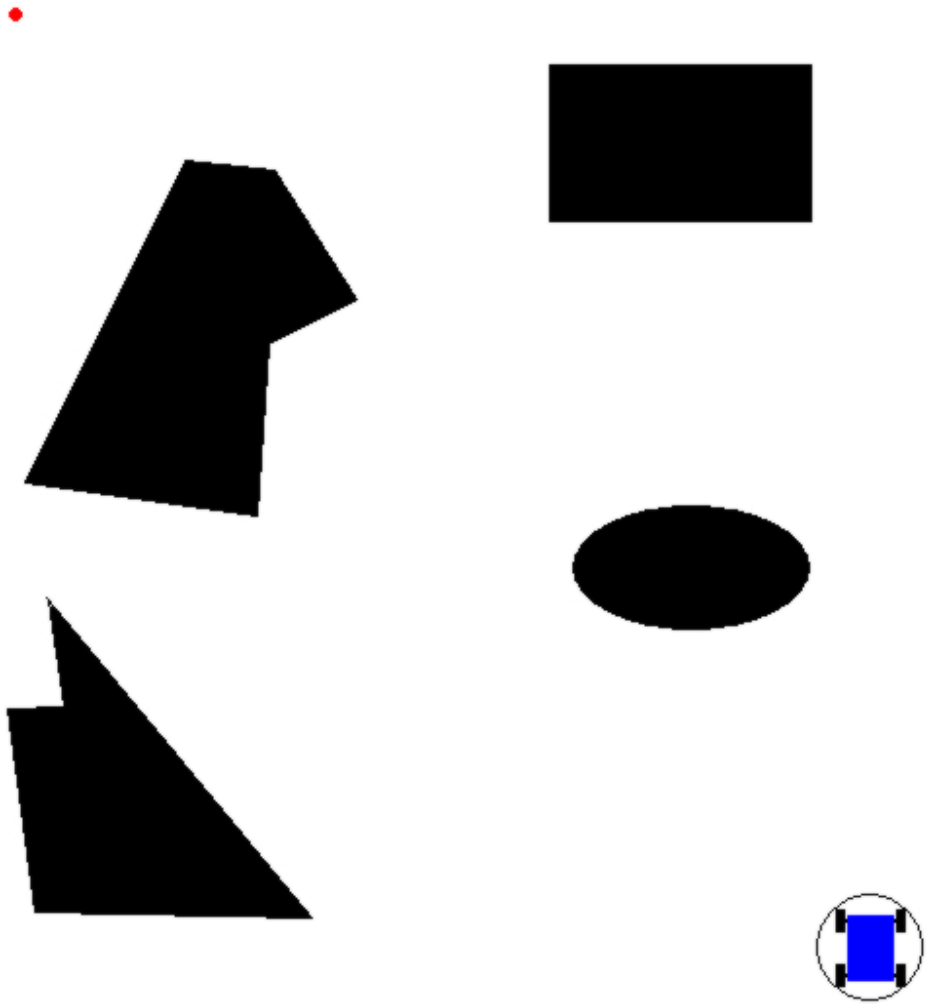
Założenia

Problem planowania trasy jest bardzo złożony i trudny. W celu uproszczenia problemu przyjmujemy założenia:

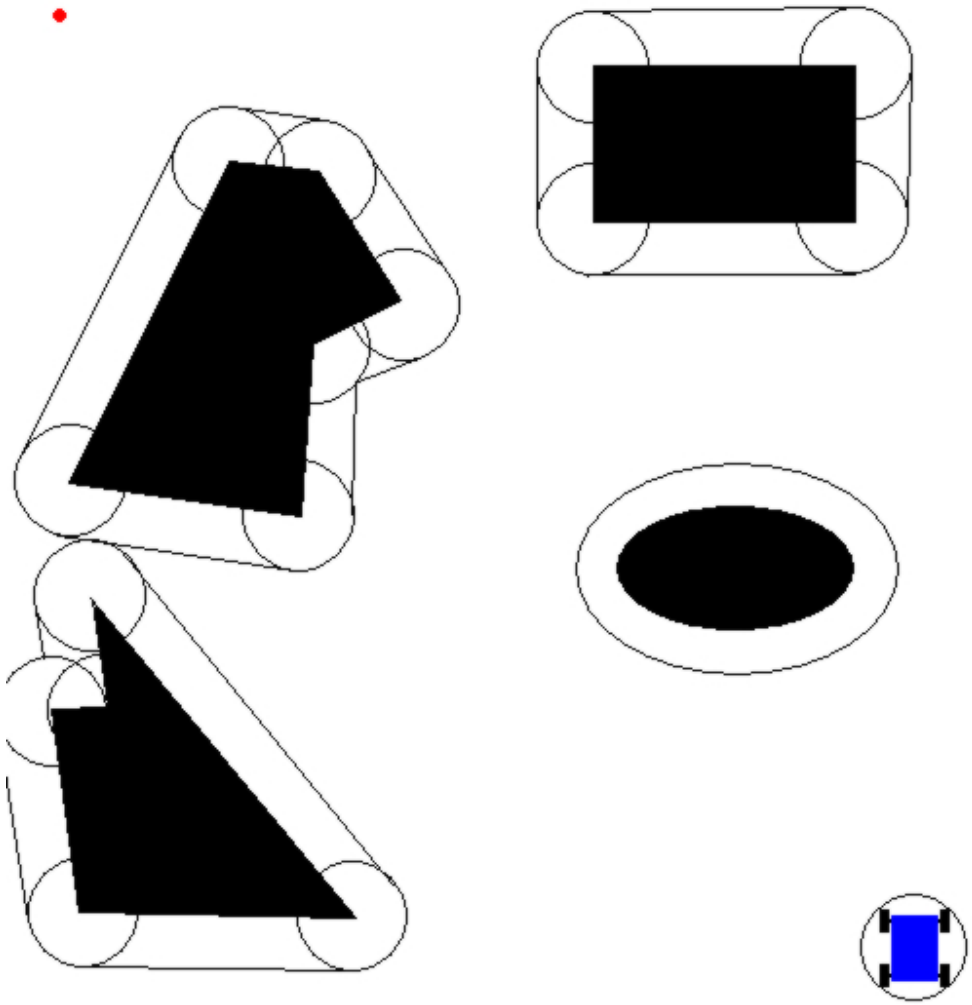
- Robot jest jedynym poruszającym się punktem
- Ograniczamy ruch do bezkontaktowego
- Robot porusza się w dwuwymiarowej płaskiej przestrzeni

Uproszczenie przestrzeni

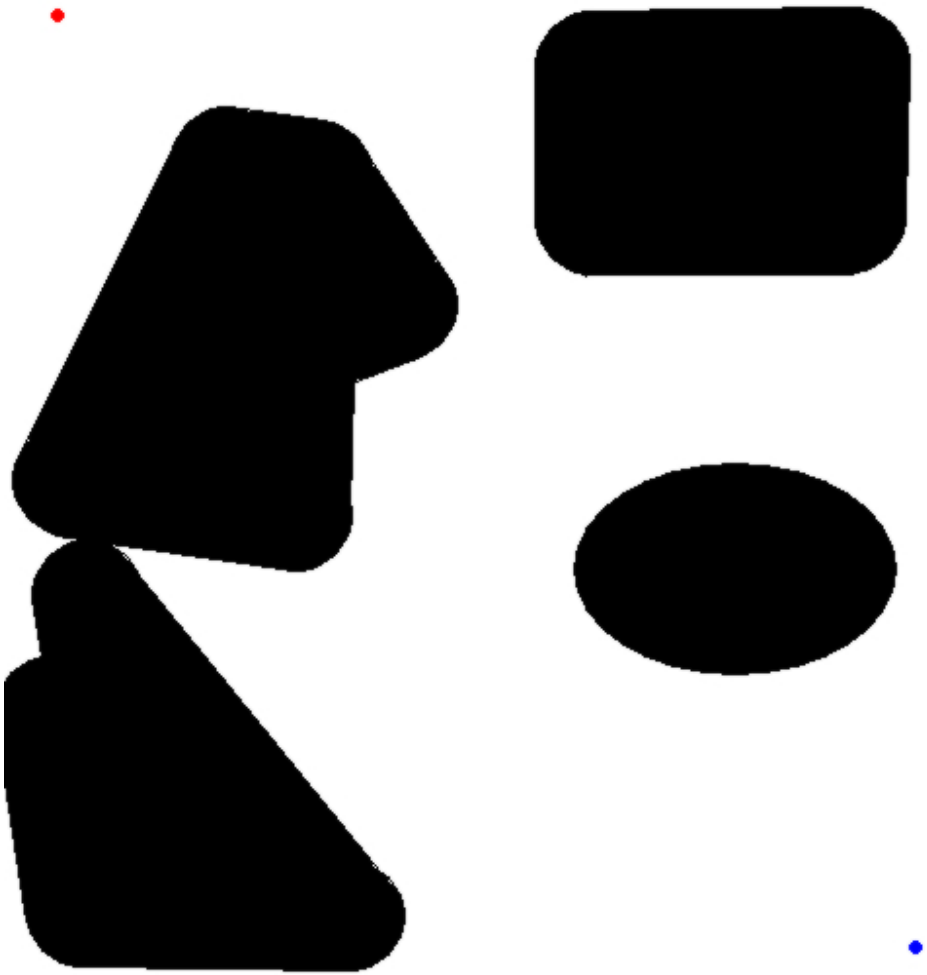
Przykład przestrzeni



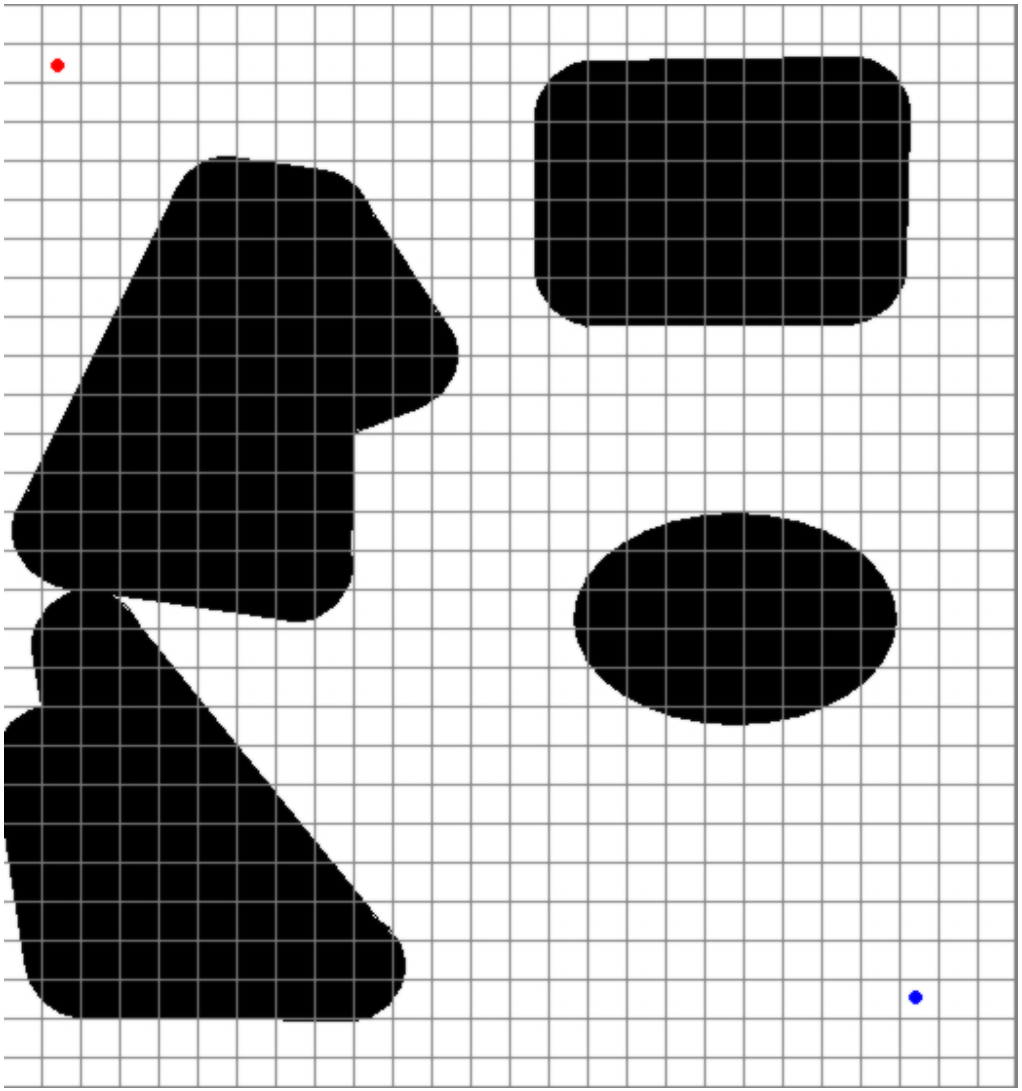
Sztuczne powiększenie przeszkód



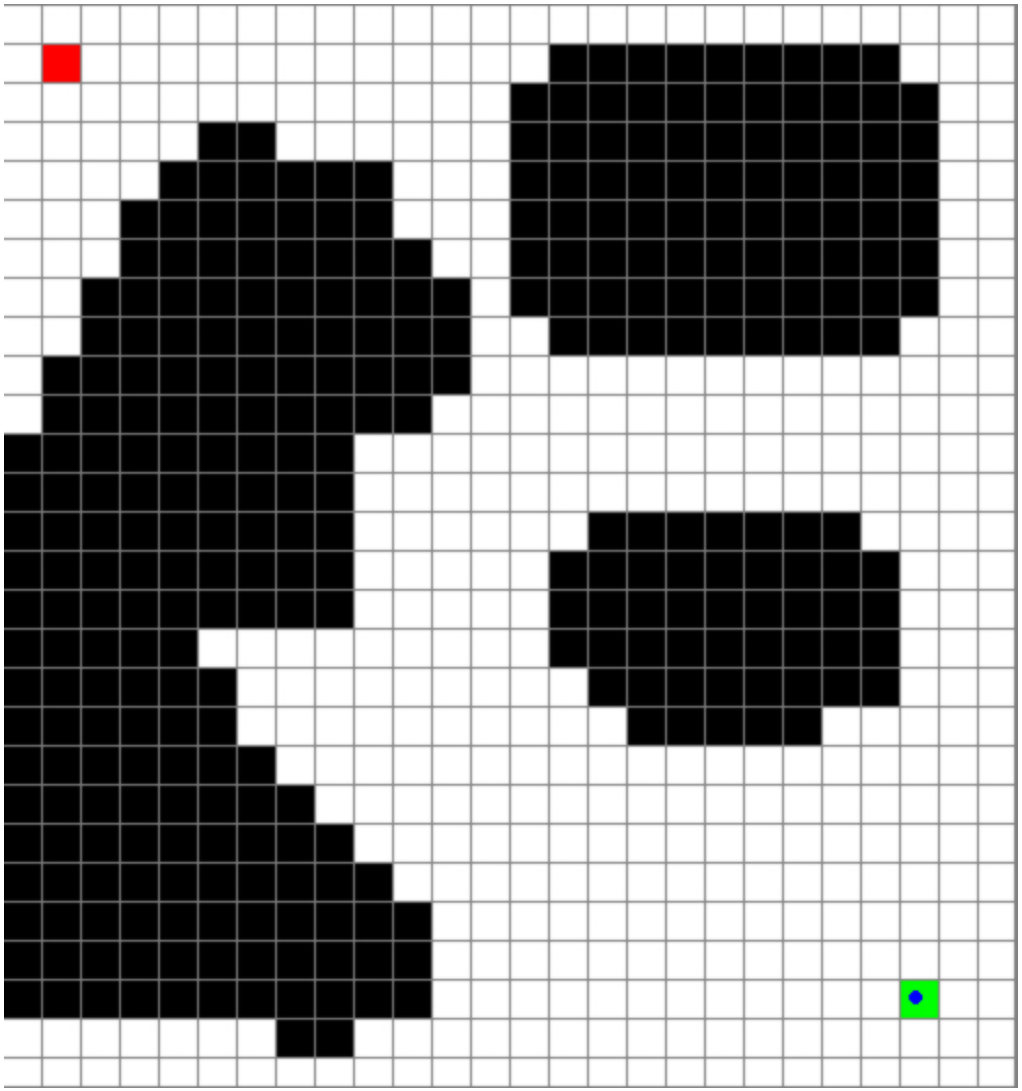
Sprowadzenie wymiarów robota do punktu



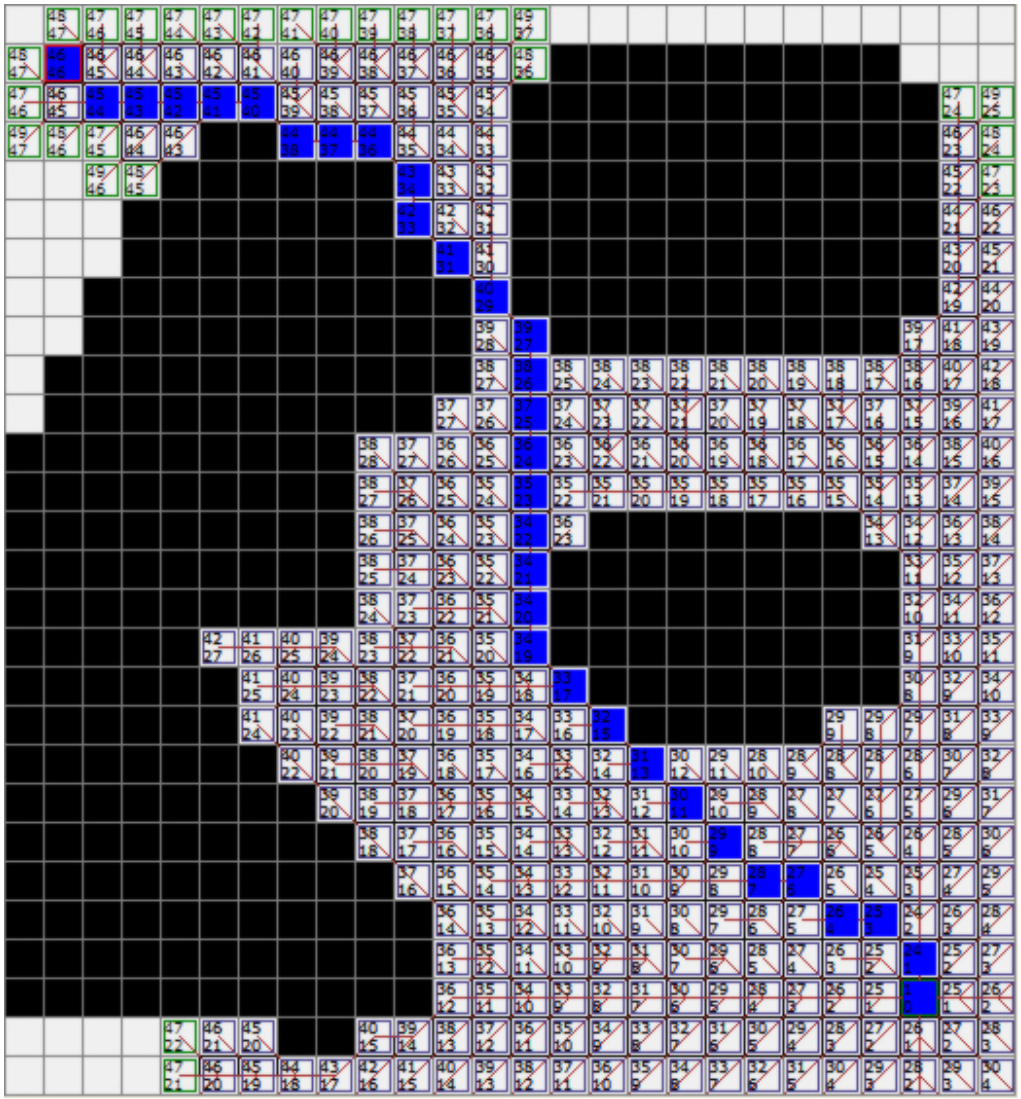
Nałożenie siatki



Klasyfikacja pól

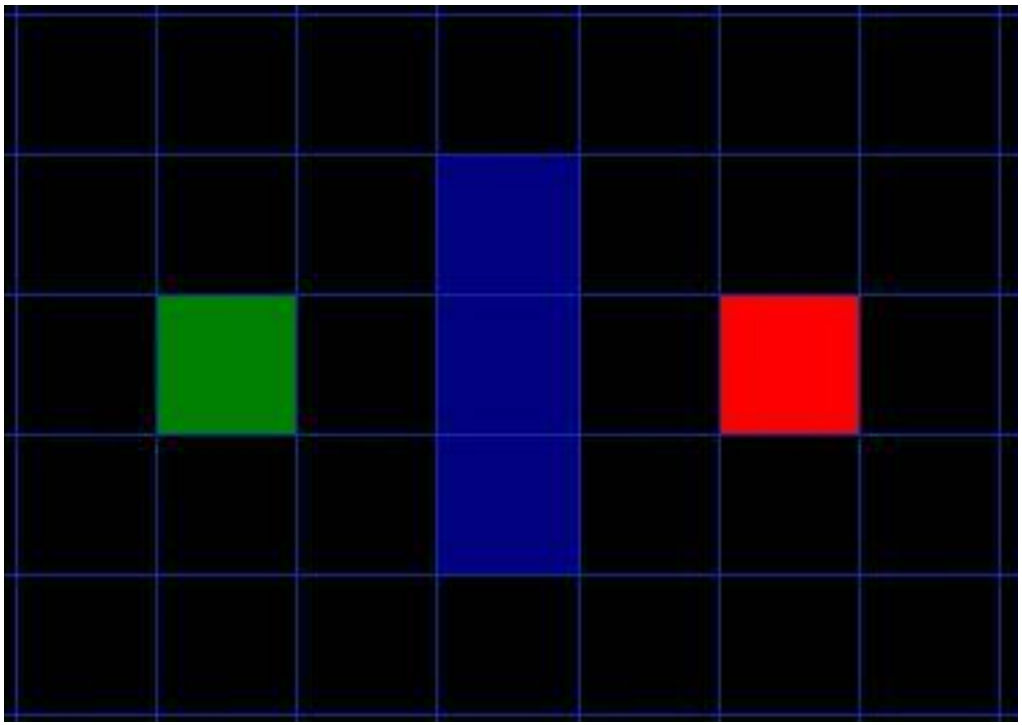


Wyznaczona ścieżka



Opis algorytmu

- Wyznaczamy punkt startowy i końcowy
- Wyznaczamy pola osiągalne i nieosiągalne



Wybór Pól

Wybór pól determinuje równanie:

$$F = G + H$$

Gdzie:

- G - koszt ruchu z punktu startu do aktualnej pozycji
- H - szacunkowy koszt ruchu przejścia z aktualnej pozycji do celu - funkcja heurystyczna

Lista otwartych

Lista otwartych to lista węzłów oczekujących na sprawdzenie

Lista zamkniętych

Lista zamkniętych to lista węzłów które zostały już sprawdzone

węzeł - rodzic

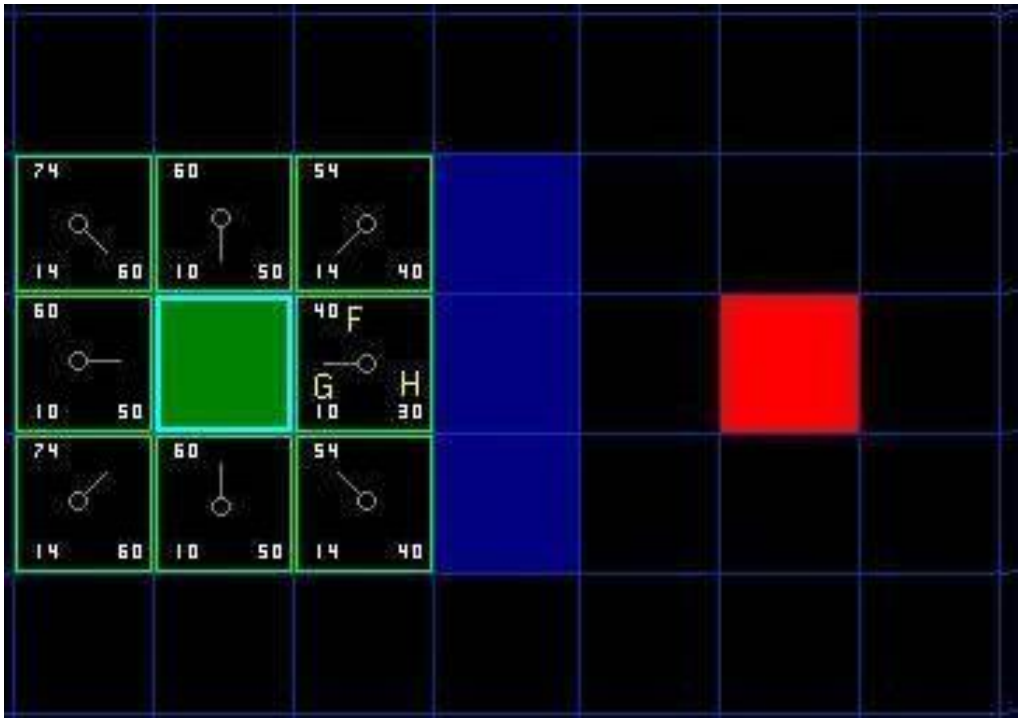
Węzeł rodzic to taki węzeł z którego przeszliśmy do aktualnego węzła. Informacja kto jest rodzicem danego węzła jest ważna do określenia ścieżki po zakończeniu sprawdzania

Oto kilka przykładowych funkcji heurystycznych:

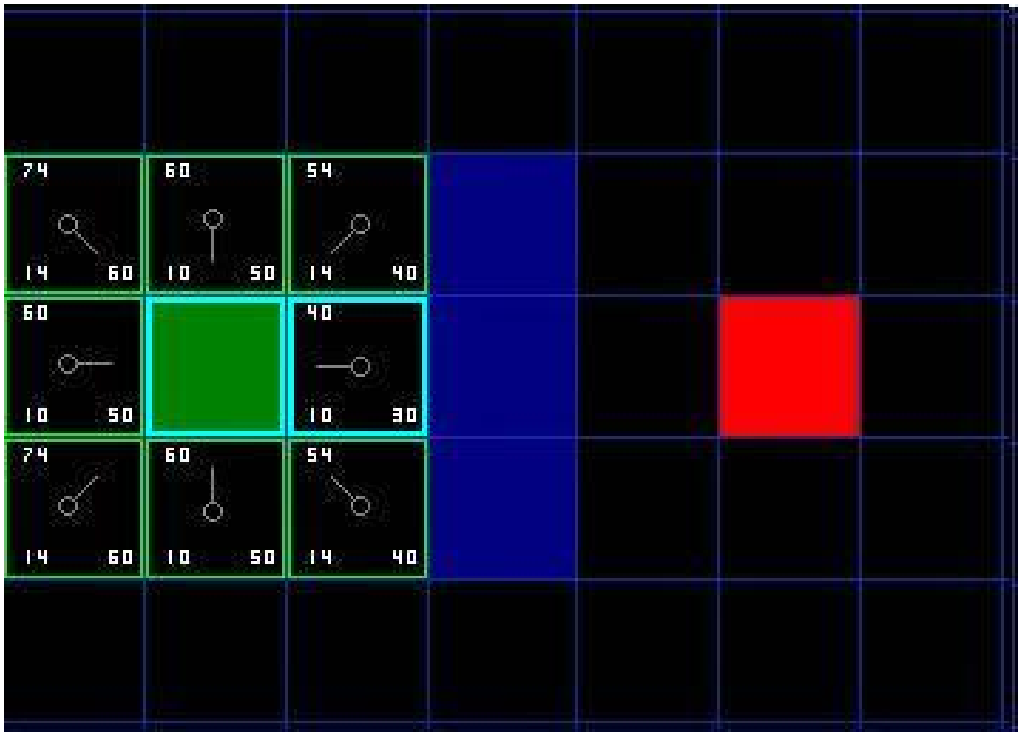
- Manhattan: $H = D * |n.x - goal.x| + |n.y - goal.y|$
- Diagonal: $H = D * max(|n.x - goal.x|, |n.y - goal.y|)$
- Euklides: $H = D * \sqrt{(n.x - goal.x)^2 + (n.y - goal.y)^2}$
- Euklides bez pierwiastka: $H = D * ((n.x - goal.x)^2 + (n.y - goal.y)^2)$

W opisywanym przykładzie stosujemy funkcję heurystyczną Manhattan, oraz przyjmujemy współczynnik D = 10, natomiast krok w pionie lub poziomie (do wyliczania G) kosztuje 10, po skosie 14.

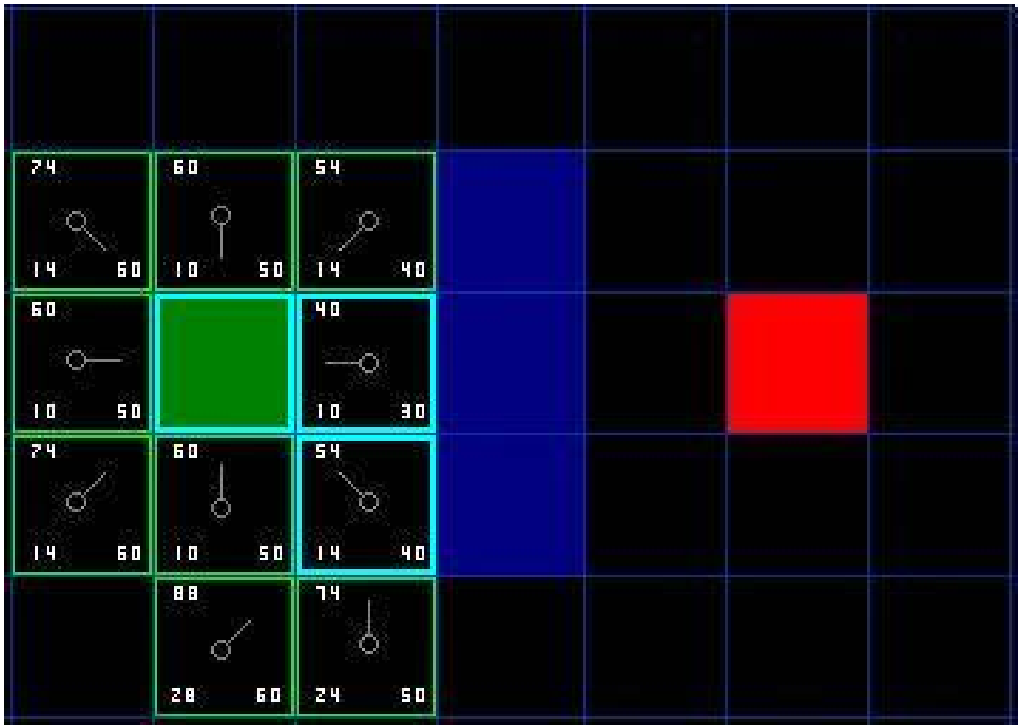
- Zaczynamy w punkcie startowym, dodajemy go do Listy otwartych
- Szukamy dostępnych przyległych pól
- Pole startowe usuwamy do Listy zamkniętych



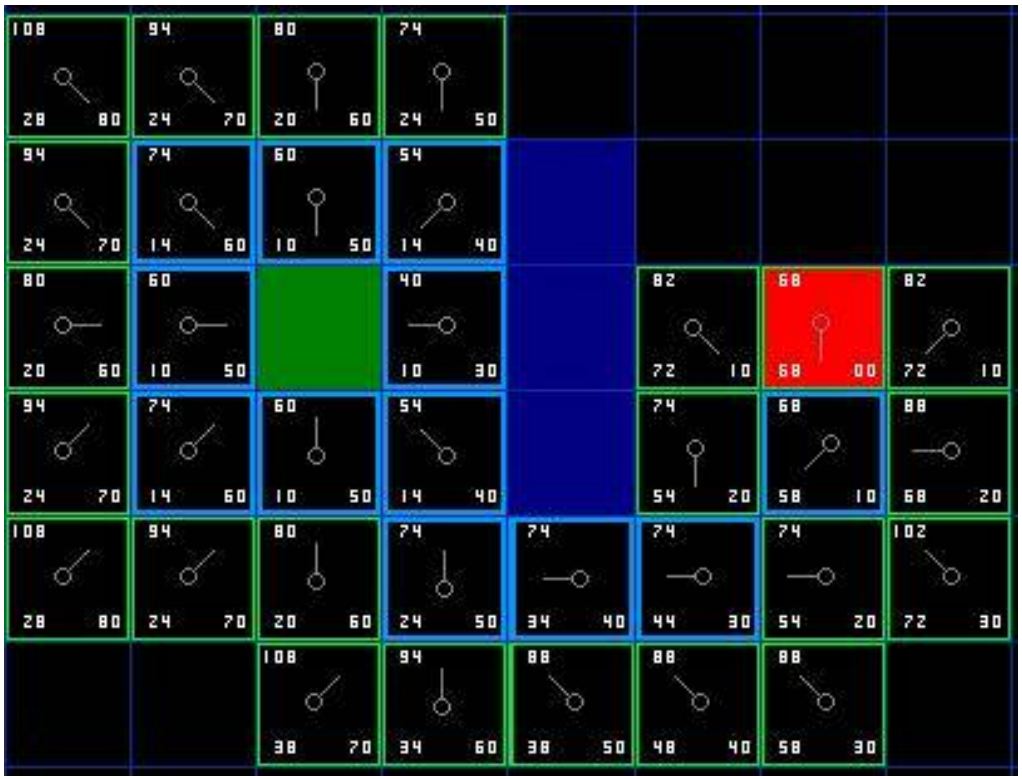
- Wybieramy spośród wszystkich dostępnych punktów ten o najmniejszym współczynniku F. Rodzica dodajemy do Listy zamkniętych.



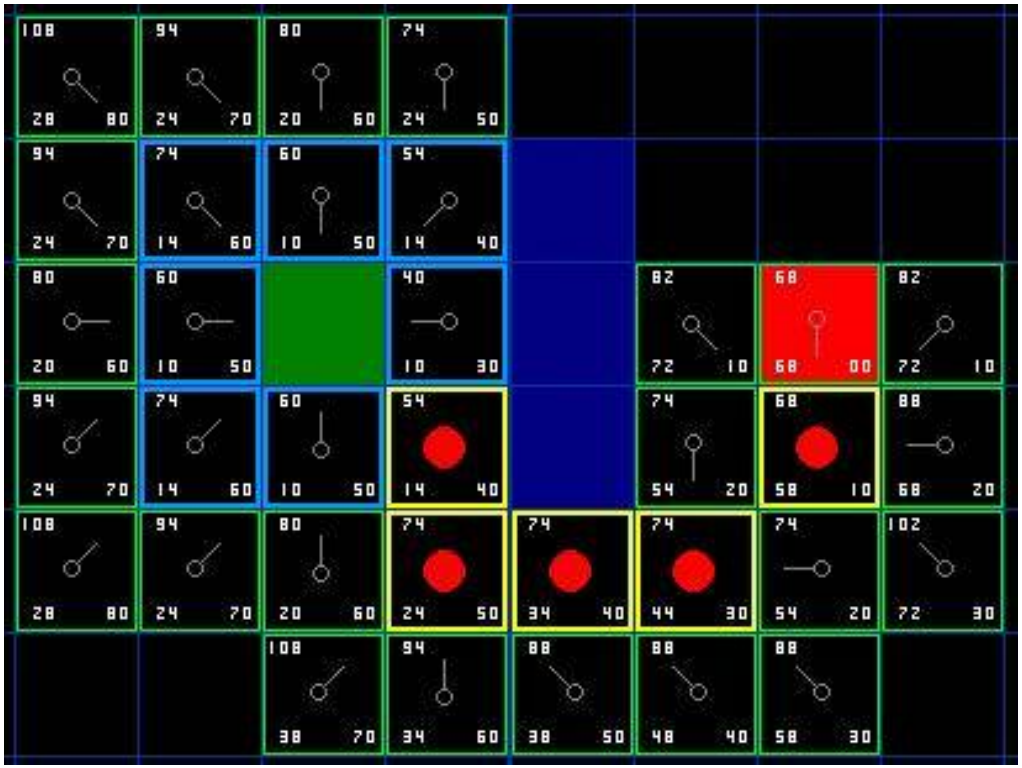
- Kontynuujemy poszukiwania, postępując wg. algorytmu.
- Jeżeli węzeł w otoczeniu rodzica jest już na Liście otwartych, to sprawdzamy czy z aktualnego węzła nie wygenerujemy lepszej (niższy koszt) ścieżki dla niego.



- Algorytm kończy działanie w momencie kiedy punkt końcowy zostanie dodany do Listy zamkniętych, lub kiedy punkt końcowy zostanie osiągnięty.



- Ścieżka zostaje wyznaczona poprzez ruch od punktu końcowego, po kolejnych rodzicach, aż do punktu początkowego.



IDA*

IDA* - Interative deepening A* - iteracyjne pogłębianie. Jest modyfikacją polegającą na tym że dołożony jest dodatkowy próg na funkcję f(n):

- Jeśli $f(n) > T$ cofaj się; T jest tu zmiennym progiem
- Jeśli nie znaleziono rozwiązania zwiększ T i powtarzaj.
- Poprzednie limity T nie są zapamiętywane

Taka modyfikacja powtarza niektóre ścieżki, ale pochłania mniej pamięci.

SMA*

SMA* - Simplified Memory A* - jest modyfikacją która wykorzystuje całą dostępną dla niej pamięć.

- SMA* nie powtarza ścieżek, pamięta wszystkie rozwinięte węzły, do póki jest dla nich miejsce w pamięci
- Jeżeli nie ma miejsca dla nowego węzła, usuwany jest pierwszy węzeł z listy.
- Dla usuniętych węzłów zapamiętywana jest najlepsza ścieżka.

Taka modyfikacja dzięki ograniczeniu pamięci doskonale nadaje się do systemów o niewielkiej jej ilości.

Real-Time A*

Real Time A* - jest modyfikacją której priorytetem jest czas wyliczenia ścieżki

- Algorytm cofa się tylko w przypadku, kiedy koszt cofnięcia i wyliczenia problemu z innego węzła jest niższy niż wyliczenie ścieżki z miejsca aktualnego
- Modyfikacja która umożliwia takie działanie polega na tym że funkcja G jest liczona nie od miejsca startu, ale jako wszystkie przejścia wykonane przez algorytm
- Dodatkowo wprowadzona jest zmienna α która zawiera minimum z wyliczanych f(n) - dzięki temu nie ma konieczności przeszukiwania drzewa, a węzeł o wyższym f(n) nie będzie nigdy rozwijany.

Taka modyfikacja nie zawsze daje optymalną ścieżkę, ale daje ją w optymalnym czasie.

Źródła informacji:

- Odnajdywanie ścieżki typu A* dla początkujących http://www.policyalmanac.org/games/aStarTutorial_pl.htm
- kurs: Artificial Intelligence - Joelle Pineau http://www.cs.mcgill.ca/ jpineau/comp424/
- Sztuczna inteligencja - Włodzisław Duch
- Wikipedia

Dziękuje za uwagę.