

Aufgabe:

Pfadfinde-Algorithmus, welcher in einer dynamischen
Gridumgebung in Anbetracht der Multiagenten
Aufgabenstellung gute Leistungen erzielt.

Bekannte Algorithmen:

DFS - BFS - Dijkstra - A* - D* - D*Lite - Theta* - ARA* -
AD* - REA* - MPGAA* - D*ExtraLite - REA* - EPEA* -
A* mit JPS Optimierung ... und viele Mehr

Pathfinding für Multiagenten Systeme

Versucht alle Pfade, für alle Agenten vorzuberechnen, um einen optimalen Fluss ohne Blockaden zu erreichen.

Rechenintensiv, und da wir auf min. 50% der Agenten keinen Einfluss haben nicht effektiv.

Beispiel:

<https://youtu.be/AGDk2RozpMQ?t=35>

Pathfinding in dynamischen Umgebungen

z.B. D*ExtraLite

Grundsätzlich komplexer und Rechenintensiver als statische Ansätze. Da die Umgebung sich selten verändert, nicht wirklich notwendig.

Fallbacklösung.

A* Algorithmus

While there are nodes in the open list:

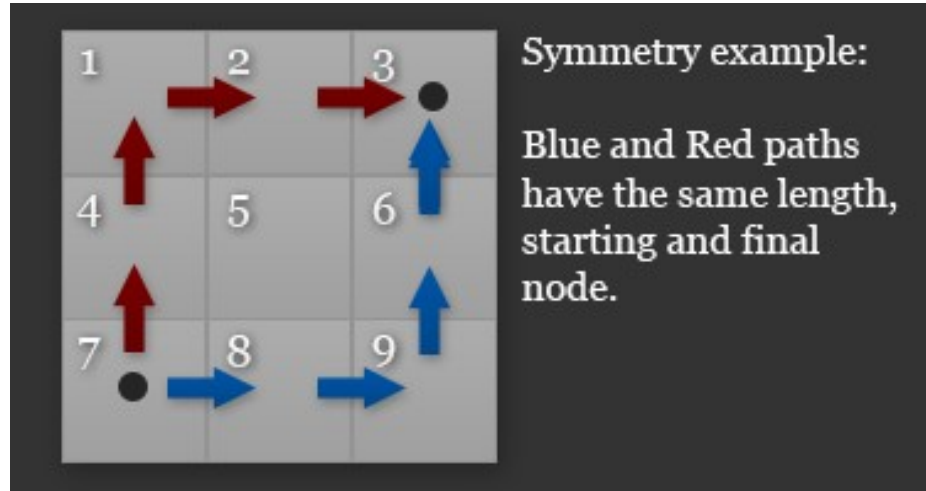
1. Pick the node from the open list having the smallest F score. Put it on the closed list.
(you don't want to consider it again).
2. For each neighbor (adjacent cell) which isn't in the closed list:
 1. Set its parent to current node.
 2. Calculate G score (distance from starting node to this neighbor) and add it to the open list
 3. Calculate F score by adding heuristics (distance to target) to the G value.

A^*

Video:

<https://www.youtube.com/watch?v=-L-WgKMFuhE>

Optimierung durch Pfadsymmetrien



A* mit Jump Point Search Optimierung

Vorteile:

Requires no precomputation. Speed up factor of 3-10X.

Always as fast as A*.

Nachteil:

limitation of uniform grid costs.

Nachfolger:

JPS+ benötigt Vorberechnung, ist aber noch schneller.

Evtl. Option später für mehrere Agenten.

A* mit Jump Point Search Optimierung

Source Code z.B.: <https://github.com/kevinsheehan/jps>

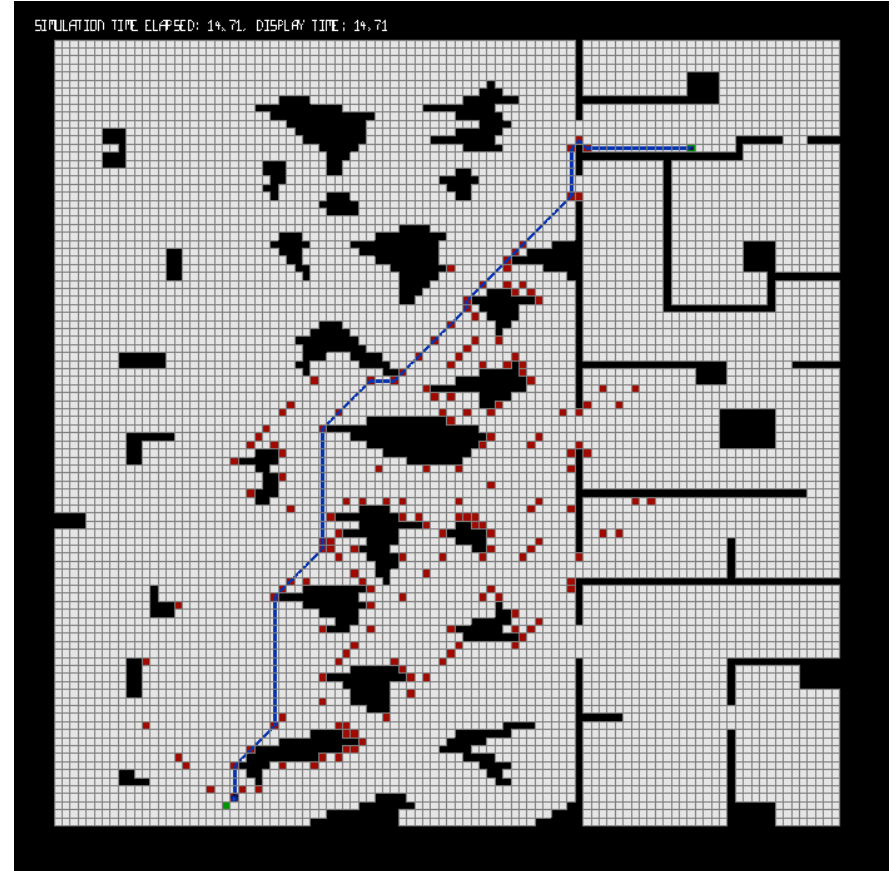
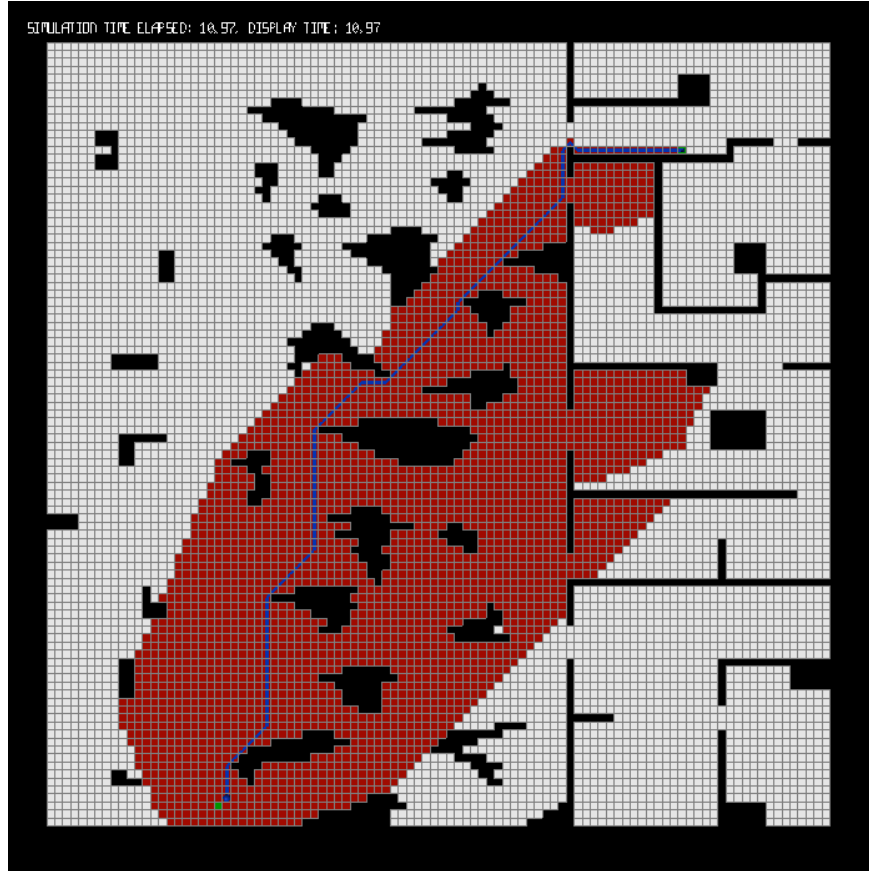
Beispiele:

<https://www.youtube.com/watch?v=ROG4Ud08lLY>

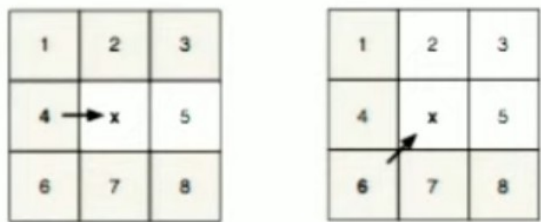
Erklärung:

https://www.youtube.com/watch?v=__ZLnTwYNPk

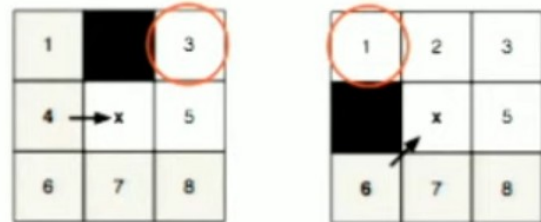
A* vs. A* with JPS



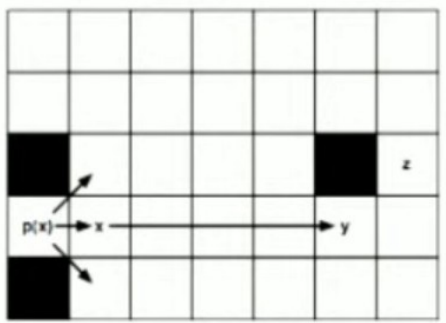
Durch JPS angewandte Heuristik



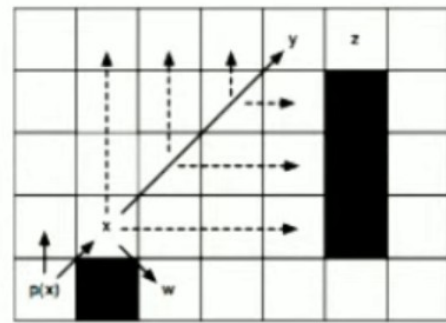
Pruning Rules



Stopping Rules



Example: Straight Jump



Example: Diagonal Jump

Ansatz: Global Path Planning with Local Replanning

Globale Planung auf der Gruppenebene. Globale Sicht verfügbar.

Falls während der Ausführung Hindernisse auftreten, globale Ausführung pausieren, einen Ansatz für lokale Problemlösung auswählen. Dabei verlässt sich der Agent auf die lokale Sicht (z.B. 5 Felder).

Anschließend fortsetzen der globalen Ausführung.

Falls lokale Problemlösung fehlschlägt > Globale Neuplanung.