



Entwicklung von Computerspielen: KI Wegplanung - Weltrepräsentation

Fakultät Informatik
FWPM



KI: Wegplanung - Weltrepräsentation

Übersicht

Wegplanung mit A*

➤ **Weltrepräsentation**



KI: Wegplanung - Weltrepräsentation

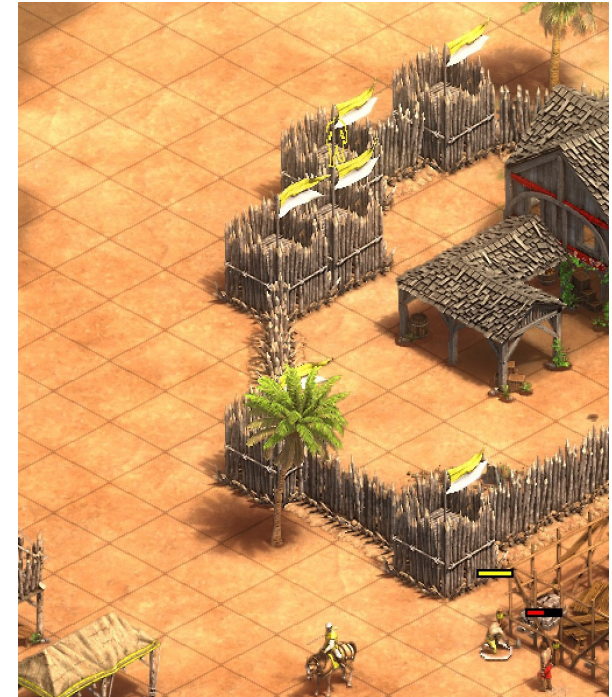
Übersicht

- Welt muss für A*-Suche als Graph repräsentiert werden
 - ein Knoten kann eine ausgedehnte Region repräsentieren
 - es muss sichergestellt sein, dass sich man innerhalb eines Knotens zwischen zwei beliebigen Punkten hindernisfrei bewegen kann
- Verbreitete Arten:
 - Regelmäßige Gitter (Grids, Tile Graph)
 - Sichtbarkeitsgraphen (Visibility Graph)
 - Wegpunkte (Waypoint Graph)
 - Navigationsnetz (Navigation Mesh)

KI: Wegplanung - Weltrepräsentation

Regelmäßige Gitter

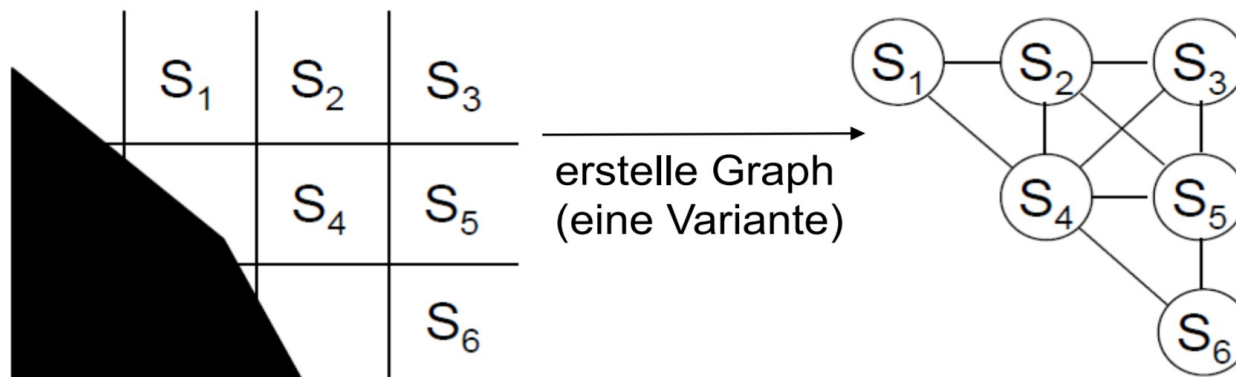
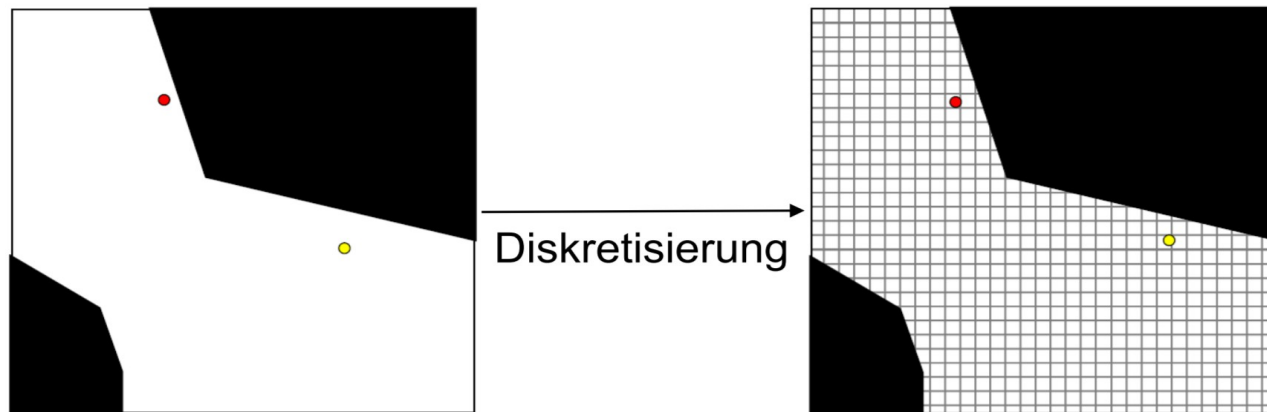
- Quadratisch oder hexagonal
3D: Würfel
- Einfache Implementierung
- Oft:
in (Echtzeit-)strategiespielen
für Freiland-Repräsentation
- Graph:
Mitte eines Felds
verbunden mit angrenzenden Feldern
- Vorteile
automatisch berechenbar
können auch zur Laufzeit erzeugt werden
- Nachteile
sehr viele Knoten → A*-Suche dauert evtl. lange
Probleme mit teilweise blockierten Feldern



(AoE nur teilweise
Tilebasiert)

KI: Wegplanung - Weltrepräsentation

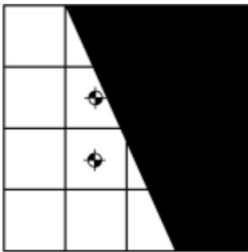
Gitter Generierung



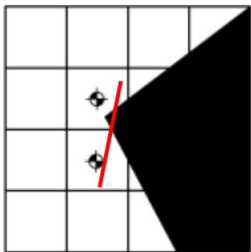
KI: Wegplanung - Weltrepräsentation

Teilweise blockierte Felder

➤ Können zu ungültigen Graphen führen



gültig: von jedem beliebigen Punkt in einem Feld gelangt man direkt zu einem beliebigen Punkt im angrenzenden Feld (Sichtverbindung)



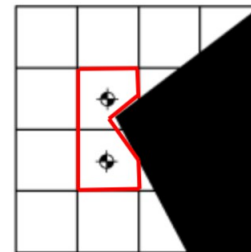
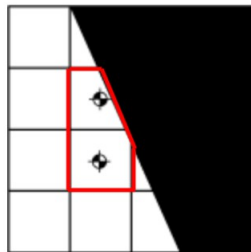
ungültig: es gibt Punkte in einem Feld, von denen aus man nicht zu einem anderen Punkt im angrenzenden Feld kommt (keine Sichtverbindung)

KI: Wegplanung - Weltrepräsentation

Teilweise blockierte Felder

- Teilweise blockierte Felder gelten als komplett blockiert
kann Gebiete komplett unbegebar / undurchdringlich machen
A* findet evtl. keinen Pfad
- Sehr feines Gitter
Wegplanung wird sehr teuer (Zeit und Speicher)
- Adaptive Diskretisierung
z.B. feineres Gitter nur an Rändern
oder feineres Gitter nur an Stellen, wo es notwendig ist

=> Polygon bestehend aus angrenzenden Feldern muss konvex sein



KI: Wegplanung - Weltrepräsentation

Blockiger Pfad

- Nicht nur bei Repräsentation durch Gitter
diese ist aber sehr anfällig dafür
- Pfadglättung → hier nicht behandelt
→ siehe z.B. [I. Millington, J. Funge: Artificial Intelligence for Games, Morgan Kaufmann, 2. Auflage, 2009 , Kap. 4.4.7]

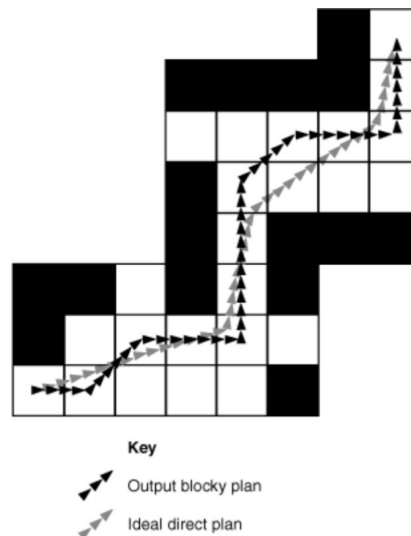


Abbildung aus: I. Millington, J. Funge:
Artificial Intelligence for Games, Morgan
Kaufmann, 2. Auflage, 2009



KI: Wegplanung - Weltrepräsentation

Gitter Bewertung

➤ Vorteil

erzeugen eines Graphen ist sehr einfach

Größe unabhängig von geometrischer Komplexität des Levels

➤ Nachteile

Größe des Graphen abhängig von Größe des Levels

Handhabung teilweise blockierter Felder

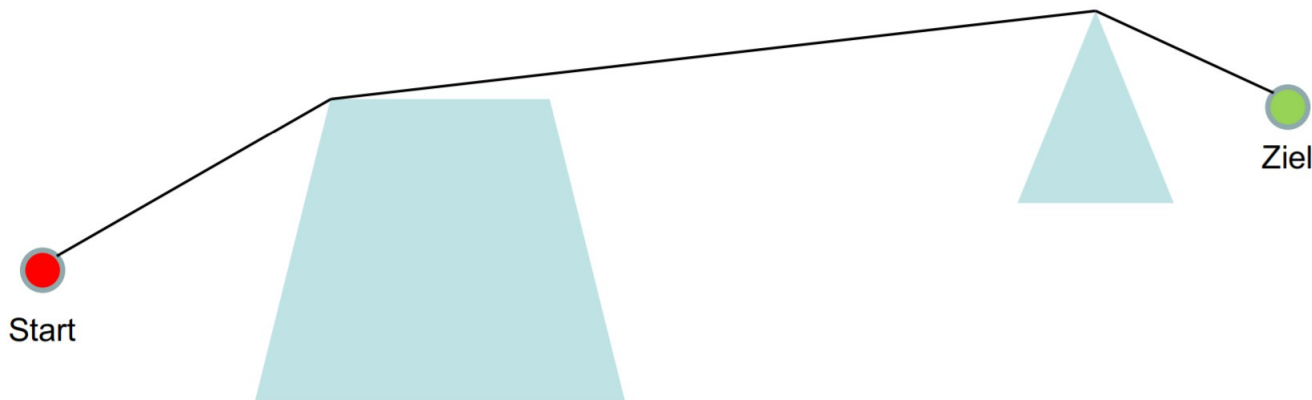
ausgeprägte blockige Pfade

➤ Meist in Strategiespielen verwendet

KI: Wegplanung - Weltrepräsentation

Sichtbarkeitsgraph

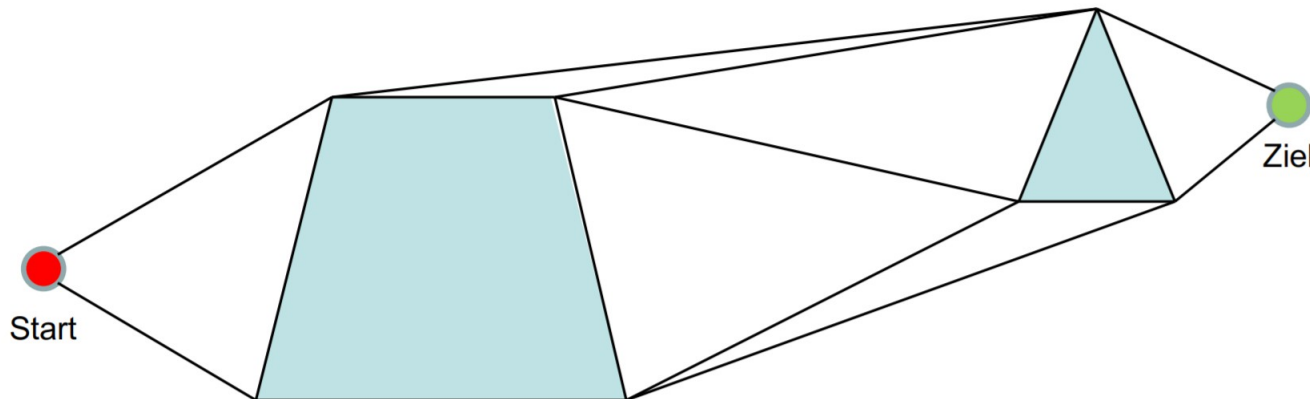
- T. Lozano-Pérez, M. Wesley, 1979
- Idee: kürzester Pfad besteht aus Geradenstücken, die an konvexen Vertices von Polygonen beginnen/enden



KI: Wegplanung - Weltrepräsentation

Sichtbarkeitsgraph

- Erstelle Graph:
Verbinde alle Vertices sowie Start und Ziel mit Geradenstücken
so, dass diese keine Hindernisse durchschneiden
- Aufwand: $O(n^2)$ n : #Vertices



KI: Wegplanung - Weltrepräsentation

Sichtbarkeitsgraph

- Problem: Komplexität des Graphen steigt mit Komplexität des Levels

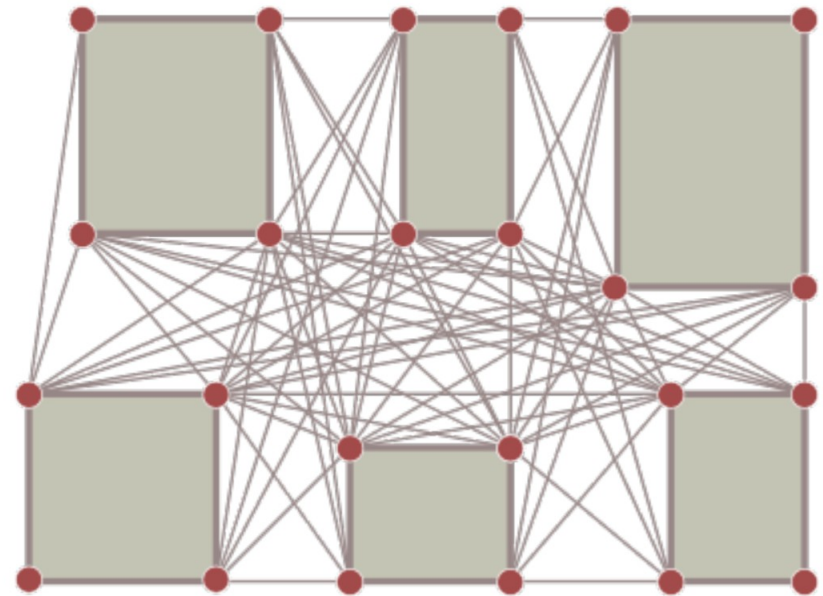


Abbildung aus:
A. Patel: Amit's Game Programming Information, 2009,
<http://www-cs-students.stanford.edu/~amitp/gameprog.html>

KI: Wegplanung - Weltrepräsentation

Sichtbarkeitsgraph

- Problem: Annahme, dass NPCs Punkte sind und keine Ausdehnung haben
- Pfade verlaufen extrem eng am Rand (bündig)
- Lösung: Verschiebung der Punkte weg von Vertices
Entfernung abhängig von Ausdehnung des NPC

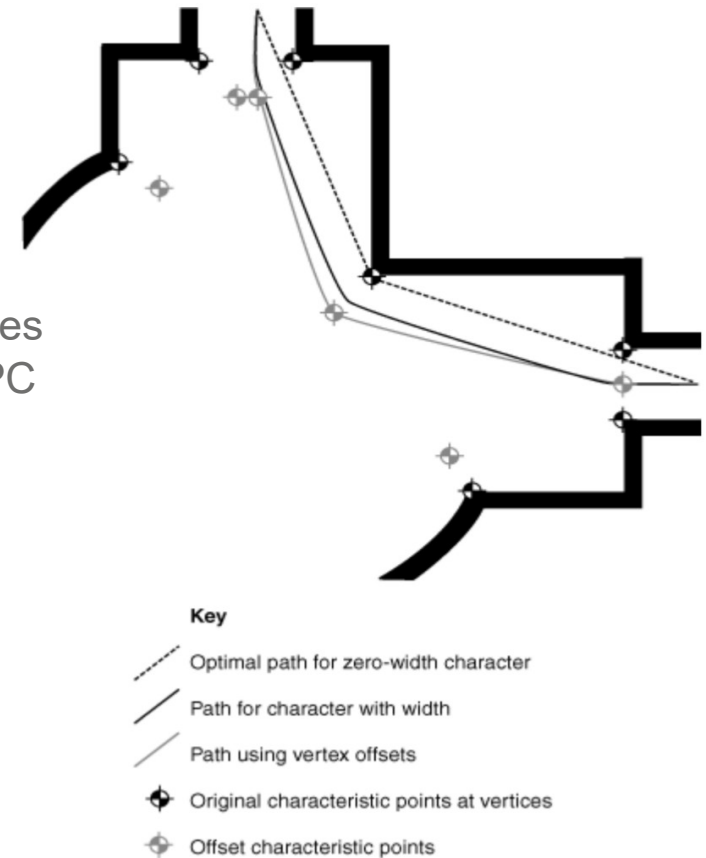


Abbildung aus: I. Millington, J. Funge:
Artificial Intelligence for Games, Morgan
Kaufmann, 2. Auflage, 2009



KI: Wegplanung - Weltrepräsentation

Sichtbarkeitsgraph - Bewertung

➤ Vorteil

Größe unabhängig von der Größe des Levels
Einfach zu handhaben

➤ Nachteile

Größe abhängig von geometrischer Komplexität des Levels

- Dann kann der Graph sehr groß werden

Handhabung von realistischen Pfaden

- Die nicht zu eng am Rand entlanggehen

NPCs bewegen sich wie auf Schienen

Pfadglättung schwierig

- Vertices definieren nicht freie Flächen (wie bei Gitter), sondern Ecken von Polygonen

- Bei Abweichung von geradlinigen Verbindungen ist freie Fläche nicht garantiert

➤ Sehr weite Verbreitung

KI: Wegplanung - Weltrepräsentation

Wegpunkte

- Idee: Lege Punkte fest, entlang derer die Bewegung erfolgen kann
- Auf ersten Blick ähnlich zu Sichtbarkeitsgraph
 - Vertices des Graphen werden von Polygonvertices weg verschoben in Richtung des freien Raums
 - Aber nicht vollständig (soweit direkt sichtbar) vernetzt

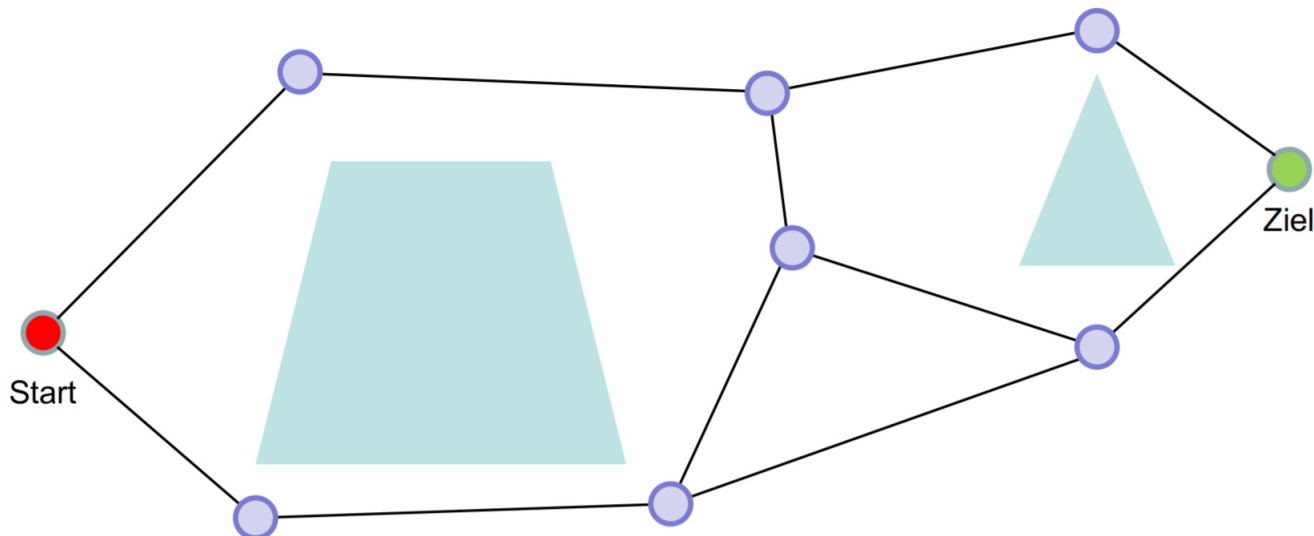


Abbildung aus: Vorlesung EVC
2020, Prof. Dr. J. Schmidt

KI: Wegplanung - Weltrepräsentation

Wegpunkte - Probleme

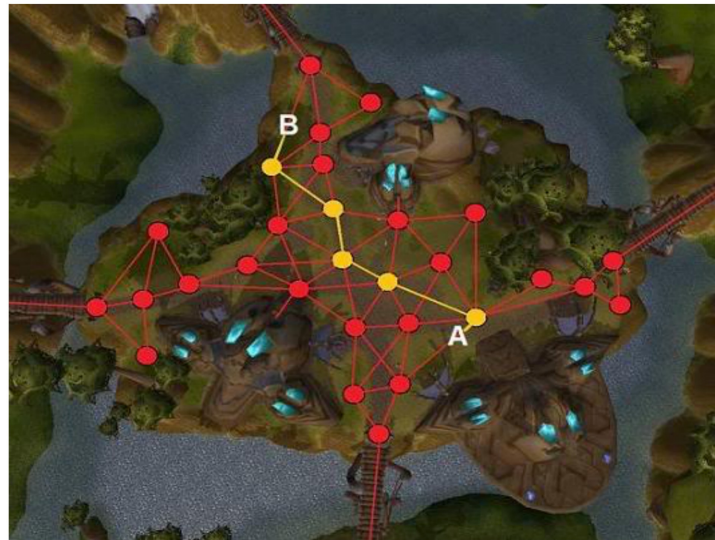
- Große offene Gebiete benötigen sehr viele Wegpunkte
- Nur so erreicht man eine vollständige Abdeckung



KI: Wegplanung - Weltrepräsentation

Wegpunkte - Probleme

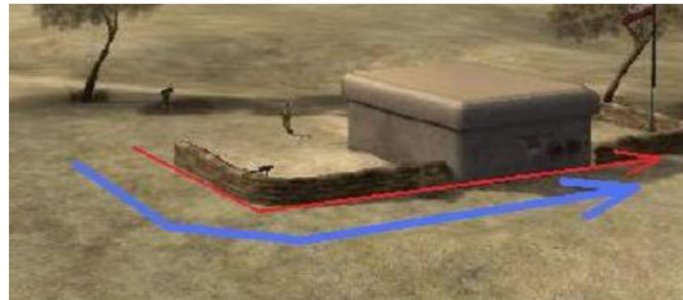
- NPCs bewegen sich nur entlang der definierten Verbindungen
- Führt zu Zickzack-Bewegungen
- NPC bewegt sich wie auf Schienen



KI: Wegplanung - Weltrepräsentation

Wegpunkte - Probleme

- Position der Wegpunkte abhängig von Ausdehnung des NPC
- Verschiedene Wegpunktnetze für unterschiedliche NPC nötig um optimale
- Pfade zu erreichen
 - rot: Soldat
 - blau: Panzer





KI: Wegplanung - Weltrepräsentation

Wegpunkte - Bewertung

➤ Vorteil

Einfach zu definieren und zu handhaben

➤ Nachteile

Größe abhängig von Größe und geometrischer Komplexität des Levels
dann kann der Graph sehr groß werden

Handhabung von realistischen Pfaden abhängig von Größe des NPCs
dürfen nicht zu eng am Rand entlanggehen

NPCs bewegen sich wie auf Schienen

Pfadglättung schwierig

bei Abweichung von geradlinigen Verbindungen ist offene Fläche nicht
garantiert

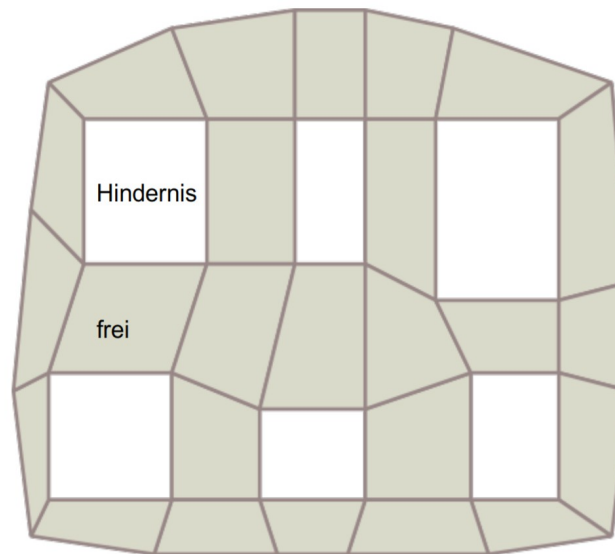
Oft mehrere Wegpunktnetze nötig

➤ Sehr weite Verbreitung

KI: Wegplanung - Weltrepräsentation

Navigationsnetz

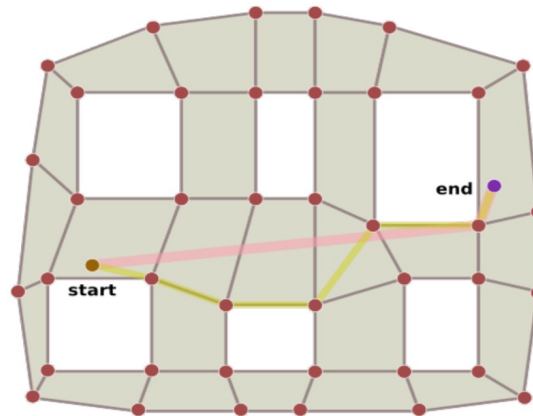
- Navigation Mesh → NavMesh
- Idee: Darstellung von **freien** Flächen als Polygone
- Polygone müssen **konvex** sein
garantiert freie Bewegung innerhalb eines Polygons



KI: Wegplanung - Weltrepräsentation

Navigationsnetz - Vertexbewegung

- direkte Verwendung des Graphen für A*-Suche
- Bewegung entlang der Vertices des Netzes



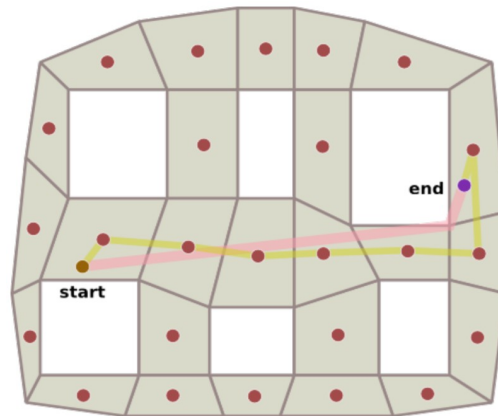
pink: idealer Pfad
gelb: berechneter Pfad

- Vorteile:
 - kürzester Pfad geht direkt an Hindernissen vorbei
 - direkte Verwendung des Netzes
- Nachteile
 - Zickzack-Bewegung
 - Bewegung entlang von Wänden (wall hugging)

KI: Wegplanung - Weltrepräsentation

Navigationsnetz - Polygonbewegung

- Verwende **Polygonmittelpunkte** als Vertices des Suchgraphen



- Vorteile:

- Bewegung ungefähr in der Mitte
- Geringere Zickzackbewegung

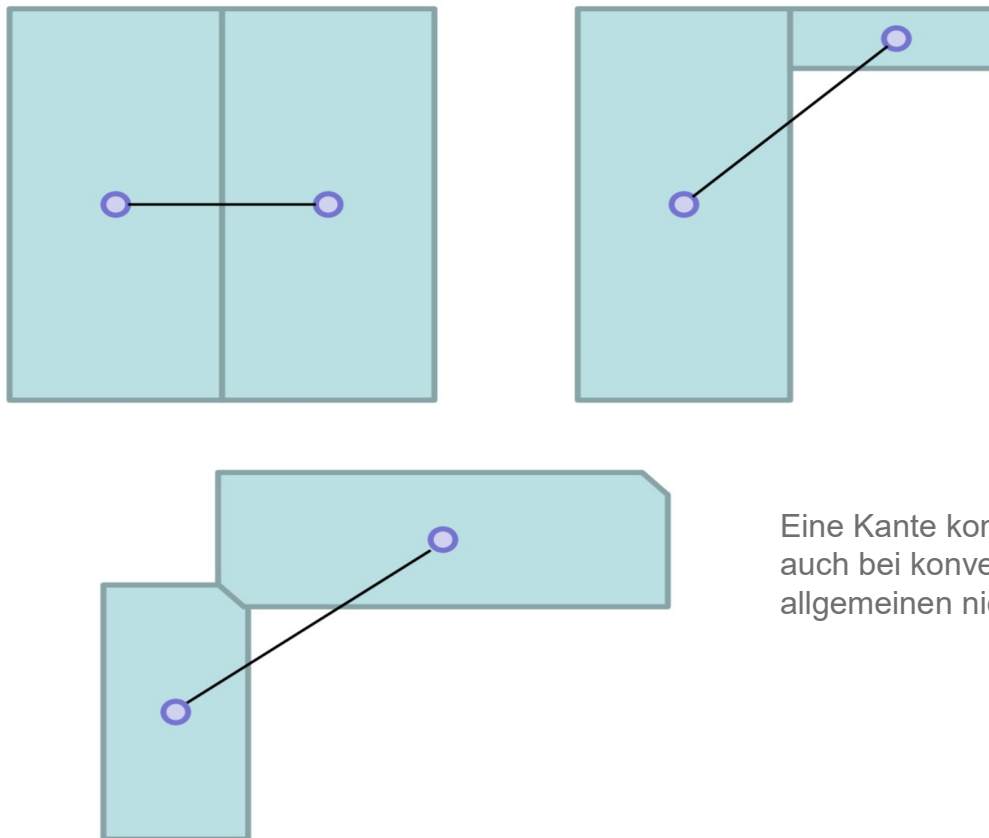
- Nachteile

- Im ersten/letzten Polygon muss man immer zum Mittelpunkt gehen (2 konvexe Polygone im allgemeinen konkav)
- Direkter (ungeblockter) Weg zwischen zwei Mittelpunkten ist ebenfalls nicht immer garantiert
- Suchgraph \neq NavMesh (Polygonfläche \rightarrow Vertex, verbinde angrenzende Polygone)

pink: idealer Pfad
gelb: berechneter Pfad

KI: Wegplanung - Weltrepräsentation

Navigationsnetz - Polygonbewegung



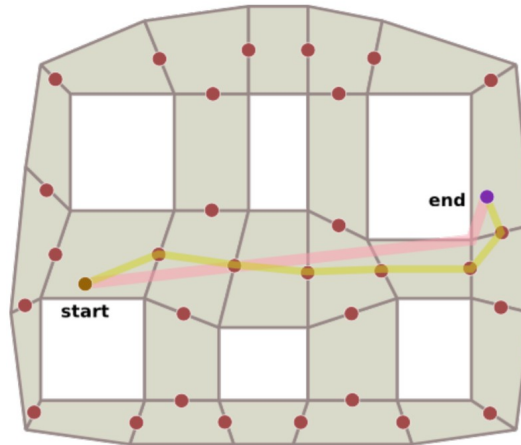
Eine Kante komplett zu teilen genügt
auch bei konvexen Polygonen im
allgemeinen nicht

Abbildung aus: Vorlesung EVC
2020, Prof. Dr. J. Schmidt

KI: Wegplanung - Weltrepräsentation

Navigationsnetz - Kantenbewegung

- Verwende **Kantenmittelpunkte** als Vertices des Suchgraphen



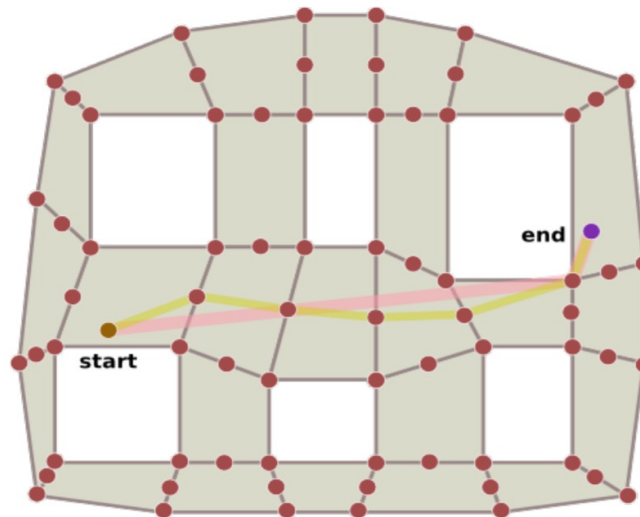
pink: idealer Pfad
gelb: berechneter Pfad

- Vorteile:
 - Bewegung ungefähr in der Mitte
 - Noch geringere Zickzackbewegung
 - Es können auch mehr als ein Vertex pro Kante eingefügt werden → glattere Bewegung
- Nachteile
 - Suchgraph \neq NavMesh
 - (Kante \rightarrow Vertex, Vertex \rightarrow Kante, voll vernetzt innerhalb eines Polygons)

KI: Wegplanung - Weltrepräsentation

Navigationsnetz – hybride Bewegung

- Kombination aus **Vertex**- und **Kanten**bewegung

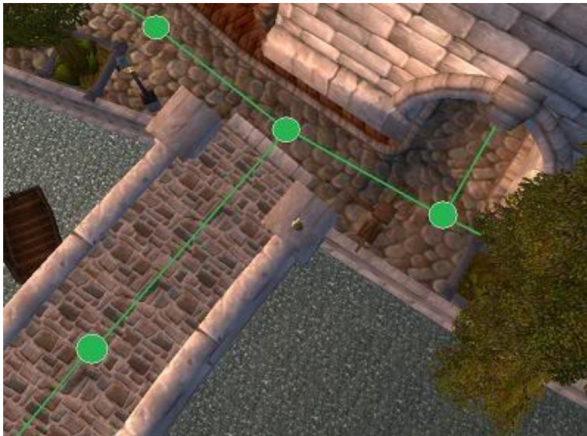


pink: idealer Pfad
gelb: berechneter Pfad

- Vorteile:
 - Bewegung ungefähr in der Mitte
 - außer an Hindernissen, dort direkt eng vorbei
- Nachteile
 - Suchgraph \neq NavMesh
 - Graph größer

KI: Wegplanung - Weltrepräsentation

Wegpunkte - Navigationsnetz



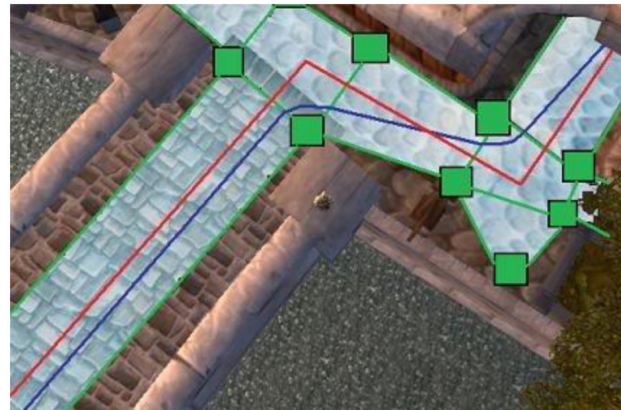
KI: Wegplanung - Weltrepräsentation

Wegpunkte - Navigationsnetz

➤ Pfadglättung

bei Wegpunkten nicht sicher → bleibe auf Kanten (rot)

bei NavMesh: sicher solange geglätteter Pfad innerhalb des Polygons
(= freie Fläche) befindet (blau)





KI: Wegplanung - Weltrepräsentation

Navigationsnetz - Bewertung

- Vorteile
 - automatisch generierbar
 - Speicherbedarf bei großen freien Flächen gering
 - Pfadglättung einfach möglich
- Nachteile
 - Vorsicht bei direkten Wegen von beliebigen Punkten innerhalb eines Polygons zu einem angrenzenden – funktioniert nicht immer
(„reibungslös“)!
- Sollte heute das Mittel der Wahl sein
 - Empfehlung: hybride Bewegung

** Kann selbst bei Fehlfunktion „funktionieren“; Klassisches „Character läuft gegen Wand und slidet langsam Richtung Eck.“ Problem... Sollte trotzdem vermieden werden*