

## Nr. 1

**Zustand:**

(int anzahlOrksLinks, int anzahlElbenLinks, boolean pferdIstLinks)

**Startzustand:**

Es gibt einen Startzustand (alle Orks, Elben und das Pferd sind links):

$S = (3, 3, 1)$

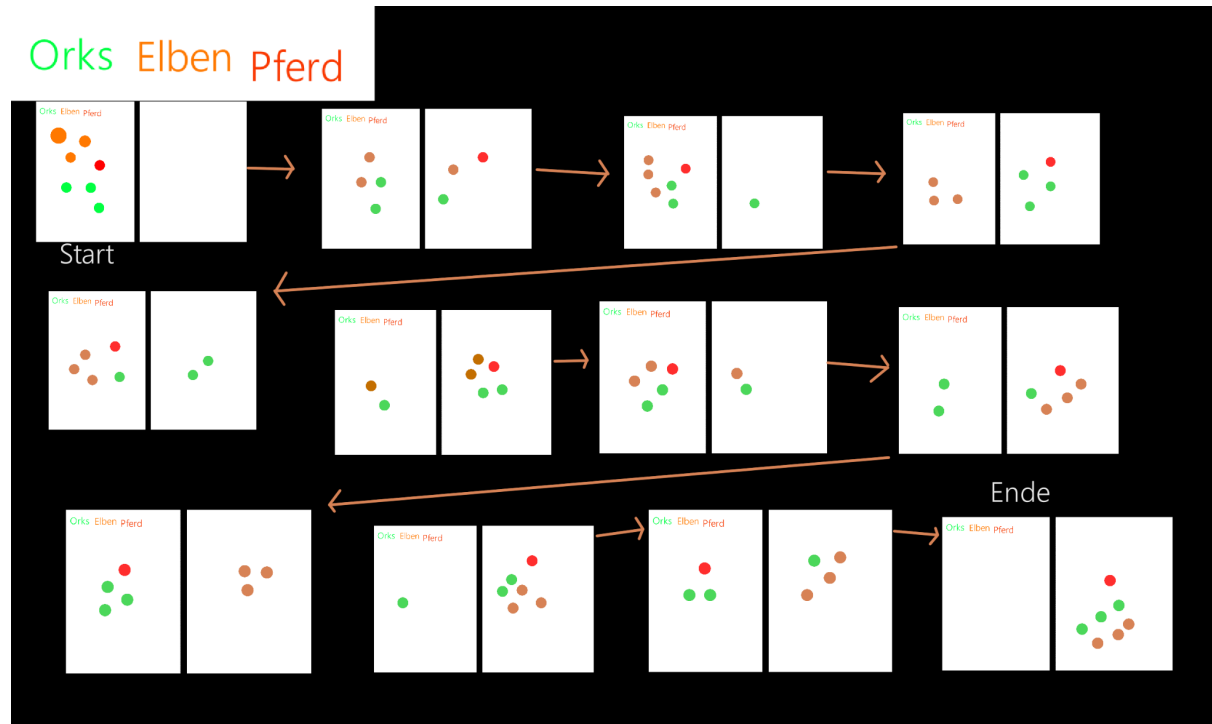
**Aktionen**

- 1 Elb und das Pferd bewegt sich nach rechts
- 1 Ork und das Pferd bewegt sich nach rechts
- 1 Elb und das Pferd bewegt sich nach links
- 1 Ork und das Pferd bewegt sich nach links
- 2 Elben und das Pferd bewegen sich nach rechts
- 2 Orks und das Pferd bewegen sich nach rechts
- 2 Elben und das Pferd bewegen sich nach links
- 2 Orks und das Pferd bewegen sich nach links
- 1 Ork, 1 Elb und das Pferd bewegen sich nach links
- 1 Ork, 1 Elb und das Pferd bewegt sich nach links

**Zielzustand**

$E = (0, 0, 1)$

Visualisierung mit Problemgraph:



## Nr. 2, Teil 1 (von Würzburg nach München)

Tiefensuche Handsimulation (Graph search variante, ohne Wiederholungen)

(Es werden zufällige Unterknoten ausgewählt)

Schritt 1:

Würzburg

Schritt 2:

Frankfurt

Schritt 3:

Mannheim

Schritt 4:

Karlsruhe

Schritt 5:

Augsburg

Schritt 6:

München

Ziel erreicht: Würzburg -> Frankfurt -> Mannheim -> Karlsruhe -> Augsburg -> München

## Breitensuche Handsimulation (Graph search)

Schritt 1:

Würzburg

Schritt 2:

Erfurt, Frankfurt, Nürnberg

Schritt 3:

Mannheim, Kassel, München, Stuttgart

Ziel erreicht: Würzburg -> Nürnberg -> München

## A\*-Suche Handsimulation (ohne Zyklen)

$$g(\text{Würzburg}) = 0$$

$$f(\text{Würzburg}) = 0 + 170 = 170$$

wir starten bei Würzburg mit 170km

Schritt 1:

Frankfurt:

$$h(\text{Frankfurt}) = 100\text{km}$$

$$g(\text{Frankfurt}) = 217\text{km}$$

$$f(\text{Frankfurt}) = 317\text{km}$$

$$h(\text{Erfurt}) = 400\text{km}$$

$$g(\text{Erfurt}) = 186\text{km}$$

$$f(\text{Erfurt}) = 586\text{km}$$

$$h(\text{Nürnberg}) = 537\text{km}$$

$$g(\text{Nürnberg}) = 103\text{km}$$

$$f(\text{Nürnberg}) = 640\text{km}$$

-> Frankfurt auswählen

Schritt 2:

Mannheim:

$$h(\text{Mannheim}) = 200\text{km}$$

$$g(\text{Mannheim}) = 85\text{km}$$

$$f(\text{Mannheim}) = 285$$

Kassel:

$$h(\text{Kassel}) = 460\text{km}$$

$$g(\text{Kassel}) = 173\text{km}$$

$$f(\text{Kassel}) = 633$$

-> Mannheim auswählen, aktueller weg:

Schritt 3:

-> immer weiter Unterknoten auswählen, da es von Frankfurt bis München mit nur einem Unterknoten weitergeht

-> Auf den Unterknoten "München" treffen

-> Ziel erreicht, Distanz zurückgelegt: 716km

-> Mir fällt auf, dass die Heuristik für Nürnberg 537km beträgt, was unrealistisch hoch ist und zu einem nicht optimalen Pfad führt, der hier als Ergebnis rauskommt.

## Vergleich

Bei der Tiefensuche werden am meisten Unterknoten durchsucht, was dazu führt, dass man die Hauptschleife am meisten durchläuft. Bei dem A\* Algorithmus wurde die Hauptschleife auch oft durchlaufen, aber das liegt nur daran, dass in diesem Beispiel ein Weg mit vielen Unterknoten genommen wurde. Bei der Breitensuche wurde die Hauptschleife weniger oft durchlaufen, da sie weniger tief geht und das oberflächlichste Ergebnis sucht.

2.

ich bin der Meinung, dass die Heuristik falsch ist.  $h(\text{Nürnberg})$  ist unrealistisch hoch. Das führt zu einem nicht optimalen Pfad. Ich finde, sie sind daher nicht korrekt und korrigiere die Restabschätzung von Nürnberg auf 70. Außerdem darf  $h(n)$  nicht größer sein als die echte Reststrecke.

## Korrektur:

$$g(\text{Würzburg}) = 0$$

$$f(\text{Würzburg}) = 0 + 170 = 170$$

wir starten bei Würzburg mit 170km

Schritt 1:

Frankfurt:

$$h(\text{Frankfurt}) = 100\text{km}$$

$$g(\text{Frankfurt}) = 217\text{km}$$

$$f(\text{Frankfurt}) = 317\text{km}$$

$$h(\text{Erfurt}) = 400\text{km}$$

$$g(\text{Erfurt}) = 186\text{km}$$

$$f(\text{Erfurt}) = 586\text{km}$$

$$h(\text{Nürnberg}) = 70\text{km}$$

$g(\text{Nürnberg}) = 103\text{km}$

$f(\text{Nürnberg}) = 173\text{km}$

-> Nürnberg auswählen

Schritt 2:

Der Unterknoten von Nürnberg ist München, das Ziel ist erreicht.

### Nr. 3

Eine dominierende Heuristik sind besser passende (genauere) Werte für den A\* Algorithmus. Das perfekte Beispiel hierzu ist die unrealistisch hohe Restdistanz von Nürnberg nach München, die ich zu von 537km zu 70km korrigiert habe. Die Heuristik von 70km dominiert, da sie eine realistischere, passendere Distanz angibt und einen besseren Pfad nach München liefert.

### Nr. 4

Der A\* Algorithmus ist optimal, wenn die Heuristik korrekt ist. Wie man bei dem Beispiel mit den Städten gesehen hat, ist der schnellste Weg nach München der über Nürnberg. Nachdem ich die Heuristik korrigiert habe, hat der A\* Algorithmus genau diesen Weg genommen. Davor hatte er einen sehr langen Weg nach München genommen, der über Mannheim, Karlsruhe und Augsburg ging, da die Heuristik für Nürnberg nicht stimmte. Mit einer gut geschätzten Heuristik kann man, wie in diesem Beispiel, sehr schnell einen optimalen Weg finden. Die Tiefensuche findet nicht den optimalsten Weg, sondern sucht einfach irgendeinen und braucht hier zum Beispiel viel mehr Durchläufe der Hauptschleife. Für den A\* Algorithmus muss man zwar die Heuristik berechnen, hat dafür aber am Ende ein optimaleres Ergebnis als die Tiefen- oder Breitensuche, da es nicht zufällige Unterknoten auswählt.