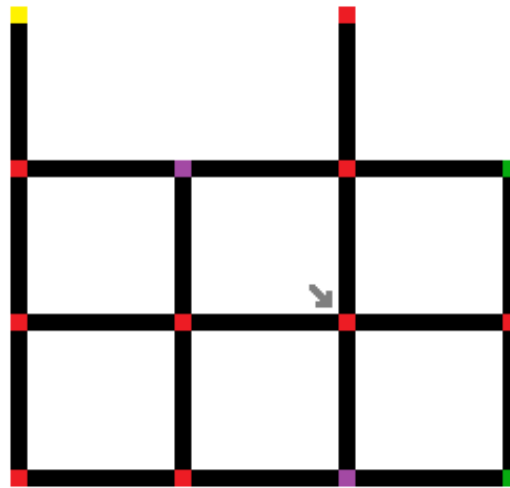


Vorüberlegungen

1 Aufgabenstellung

Sokoban ist ein Spiel bei dem mit einer Spielfigur Kisten auf vorgegebene Ziele geschoben werden müssen. Kisten können nur geschoben und nicht gezogen werden. Außerdem können nicht mehrere Kisten gleichzeitig bewegt werden.

In dieser Aufgabe sind Roboter zu konstruieren, die ein Sokoban-Feld zunächst einlesen und dann lösen. Zu den Regeln eines normalen Sokoban-Spiels wird hierbei zusätzlich zu dem Schieber ein Zieher eingesetzt. Außerdem wird ein Kartierer benötigt und ein Bauteil, dass die Lösung des Spiels berechnet.



Das Spielfeld besteht hier aus schwarzen Linien. Die Kreuzungen sind durch verschiedene Farben gekennzeichnet. Grün steht für die Ziele der Kisten, Violett für die Starträume, Gelb für den Startraum des Schiebers und ein Pfeil markiert den Startraum des Ziehers und des Kartierers. Rot wird für normale Kreuzungen verwendet.

2 Zeitplan

Beginn: 10. September 2010

Abgabe: 27. September 2010

Konstruktion der Roboter: ca. 1 Tag

Testen von Sensoren und anderen Komponenten: ca. 2 Tage

Implementierung der zusammenfassenden Navigation: ca. 2 Tage

Implementierung des Kartierers: ca. 3-4 Tage

Implementierung des Solvers: ca. 4-5 Tage

Implementierung des Ziehers: ca. 3 Tage

Implementierung des Schiebers: ca. 3 Tage

Testen und Optimierung: ca. 3-4 Tage

3 Aufgabenverteilung

Teammitglieder:

Andreas Bigontina, Michael Bigontina, Christoph Bruns, Maximilian Burger, Sebastian Hagen, Wiebke Köpp, Anastasia Panteloglou, Till Rohrmann

Organisation:

Till Rohrmann, Wiebke Köpp

Kartierer:

Maximilian Burger, Sebastian Hagen

Solver:

Till Rohrmann

Zieher:

Christoph Bruns, Michael Bigontina

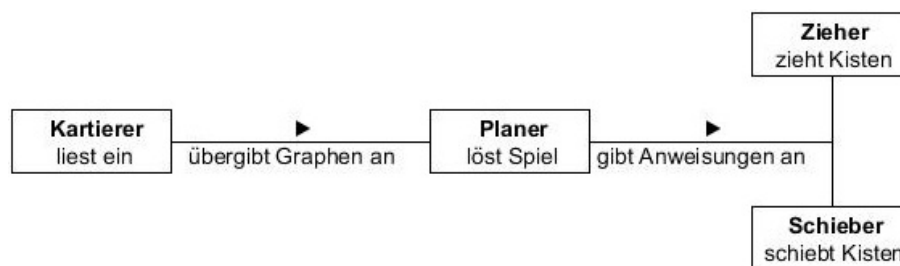
Schieber

Andreas Bigontina, Anastasia Panteloglou

Fehlerbehandlung:

Wiebke Köpp

4 Lösungsansatz

**Navigation:**

Alle drei Roboter verwenden eine gemeinsame Navigation. Die Roboter werden mit dieser über einfache Befehle wie *'Bewege dich nach Norden'* oder *'Bewege dich zum nächsten Raum'* gesteuert. Außerdem können auch Koordinaten angegeben werden, zu denen sich ein Roboter bewegen soll.

Die Navigation wird mithilfe eines Pilot realisiert, der über verschiedene Behaviors sicherstellt, dass die Roboter der Linie folgen, von ihr nicht abweichen und dass die farbigen Räume erkannt werden.

Aufbau der Roboter:

Alle drei Roboter besitzen drei Sensoren. Zwei Lichtsensoren werden zum Erkennen der Linie verwendet, sie sind am Roboter vorne knapp über dem Boden montiert. Zwischen ihnen ist ein Abstand, der in etwa der Breite der Linie entspricht, so dass die Sensoren im Fahren rechts und links der Linie messen und erkennen, wenn der Roboter sich in die Linie dreht und somit korrigiert werden muss. Zwischen den Lichtsensoren, etwas nach hinten versetzt, wird ein Farbsensor montiert. Die Roboter haben außerdem



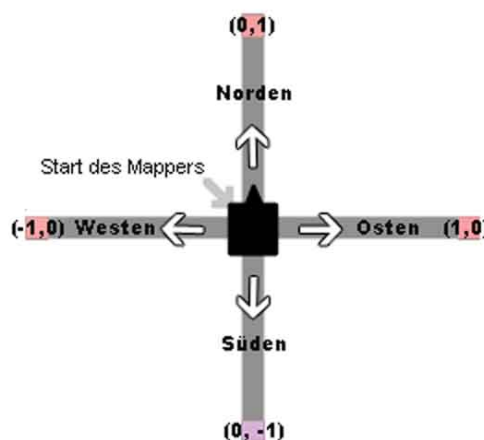
die Verwendung eines Bricks, zwei Motoren mit größeren Rädern zur Bewegung und zwei kleinere Stützräder gemeinsam. Hierbei wird versucht die Roboter möglichst kompakt zu halten, damit beispielsweise bei Zieher und Schieber eine Drehung auch dann möglich ist, wenn Zieher und Schieber sich auf benachbarten Feldern befinden.

Kartierer:

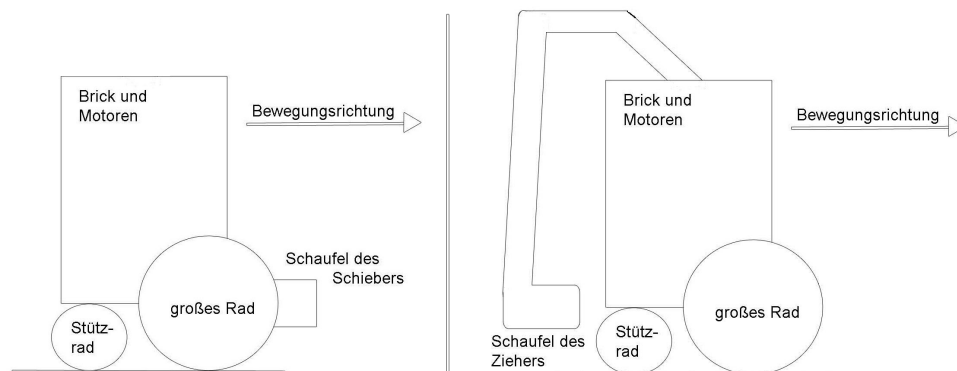
Der Kartierer beginnt an seinem Startpunkt das Feld einzulesen. Er speichert das Feld in einem Graphen, in dessen Knoten Informationen über den Typ des jeweiligen Knotens (definiert durch die Farbe) und dessen Nachbarn (im Norden, Osten, Süden und Westen) gespeichert sind. Der Startpunkt ist hierbei als Punkt $(0,0)$ definiert. Die Ausrichtung zu Beginn wird als Norden definiert. Bewegt sich der Roboter nun nach Norden wird die y -Koordinate inkrementiert, bei Osten wird die x -Koordinate inkrementiert und bei Süden und Westen werden x - und y -Koordinate entsprechend dekrementiert.

Der Kartierer führt nun eine Tiefensuche auf dem Feld aus und analysiert jede Kreuzung, die er zuvor noch nicht analysiert hat, indem er sich auf ihr dreht und die Richtungen speichert, die dieser Knoten besitzt. Die auf dem Feld erkannte Farbe wird gleichzeitig gespeichert.

Nach dem kartieren schickt der Roboter den gesamten Graphen per Bluetooth an den Computer, der dann die Berechnung der Lösung durchführt.

**Solver:**

Da die Leistung eines Bricks voraussichtlich nicht für die Berechnung der Lösung ausreicht, wird die Berechnung auf einem Computer durchgeführt. Der Computer empfängt zunächst die Daten vom Mapper und schreibt sie in ein Format, mit dem der Solver arbeiten kann. Der Solver berechnet dann mögliche Zustände und errechnet mit Hilfe eines A*-ähnlichen Algorithmus welche Zustandsübergänge insgesamt zur Lösung führen. Eine Kombination von Kistenpositionen entspricht hierbei einem Zustand. Eine Kistenbewegung entspricht einem Zustandsübergang. Bei der Prüfung, ob zwei Zustände gleich sind, wird außerdem auch noch die Roboterposition mit einbezogen.



Aufbau von Schieber und Zieher

Schieber:

Zusätzlich zu dem beschriebenen Aufbau aller Roboter besitzt der Schieber eine Art unbewegliche Schaufel, um eine Kiste sicher schieben zu können. Zum Bewegen auf dem Feld ohne Kisten benutzt er die normale Navigation. Das Aufnehmen der Kiste und Ablegen der Kiste werden gesondert implementiert. Sind zwischen Aufnehmen und Ablegen mehrere Felder zurückzulegen, wird auch hier die normale Navigation verwendet.

Der Schieber erhält über Bluetooth einzelne Befehle, führt diese dann aus und sendet dann eine Bestätigung zurück an den Computer, um anzuzeigen, dass er seine Aktion abgeschlossen hat und für neue Befehle bereit ist.

Zieher:

Der Zieher hat statt der beim Schieber angebrachten Schaufel einen Art Arm mit einer Schaufel am Ende, der mit einem Motor verbunden ist, um gehoben und gesenkt werden zu können. Sonst unterscheidet er sich von Schieber nur durch das Ablege- und Aufnahmeverhalten bei Kisten. Der Arm ist so am Roboter angebracht, dass der Zieher beim Ziehen vorwärts fahren kann, damit auch hier beim Fahren die normale Navigation benutzt werden kann. Der Zieher zieht die Kiste also hinter sich her. Auch der Zieher ist über Bluetooth mit dem Computer verbunden und erhält von ihm Befehle.