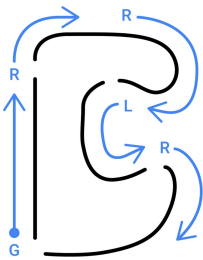


LÖSUNG

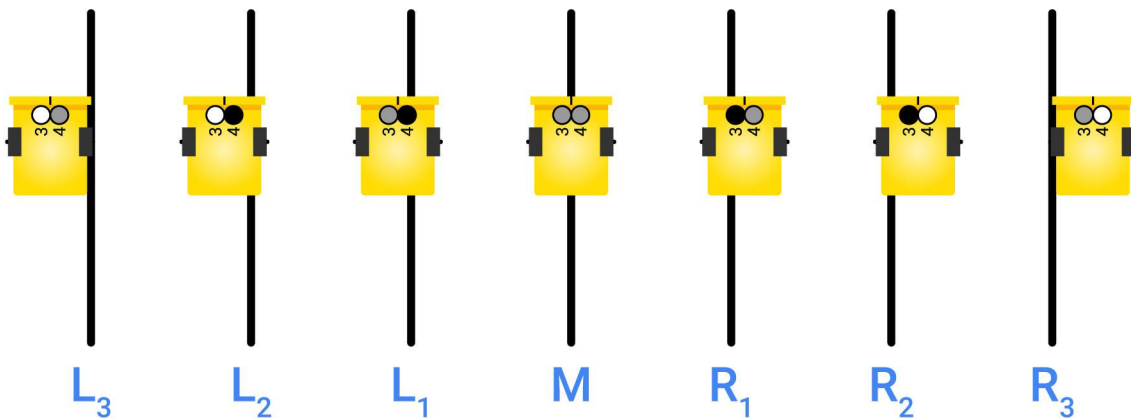
Linienfolger

Euer **Ziel** ist es, den Gepäckwagen selbständig entlang der C-förmigen Linie zu bewegen, d. h. konkret für den gewünschten **Output**: Wenn der Wagen sich auf der Linie befindet, soll er sich (vorwärts) bewegen und wenn er abgedriftet ist, soll er auf die Linie zurückkehren. Als **Input** erhält der Wagen von den beiden Sensoren die Untergrundfarbe. Welche Farben das sind, hängt vom Zustand des Wagens ab, den Ihr im Folgenden untersuchen werdet.



Die C-Kurve kann in einen Geraden-Abschnitt und in Kurven zerlegt werden (**Zerlegung**). Aufgrund der Symmetrie wird zunächst nicht weiter unterschieden zwischen Links- und Rechtskurve (**Mustererkennung**). Die erkannte Symmetrie wird später beim Algorithmischen Design genutzt.

Bisher ist der gegebene Input (die Untergrundfarben) noch unklar. Um das zu klären, untersucht Ihr die möglichen Zustände, die der Wagen auf dem geraden Streckenabschnitt annehmen kann. Durch Ausprobieren in der Open Roberta Lab Simulation erhaltet Ihr die folgenden sieben Zustände:



Theoretisch sind auch noch die Zustände weiß-weiß und schwarz-schwarz möglich, aber praktisch kommt schwarz-schwarz nicht zustande, weil die gegebene Linie nicht breit genug ist und im Zustand weiß-weiß befindet sich der Wagen nicht mehr auf der Linie, wozu es erst gar nicht kommen soll.

Ihr erkennt die Symmetrie zwischen den linken und rechten Zuständen, wodurch Ihr Euch auf eine Hälfte fokussieren könnt (**Mustererkennung**).

Außerdem ist eine Ähnlichkeit zwischen Zuständen 1 und 2 erkennbar: Der innere Sensor ist jeweils schwarz und in beiden Fällen ist der Wagen von der Linie abgedriftet (**Mustererkennung**). Die Farbe des äußeren Sensors ist dabei nicht entscheidend und kann daher ignoriert werden (**Abstraktion**). Bei Zustand 3 ist der Wagen bereits vollkommen von der Spur abgedriftet, wozu es gar nicht erst kommen soll. Der Zustand kann auch ignoriert werden (**Abstraktion**).

Durch Zerlegung, Mustererkennung und Abstraktion hat sich das komplexe Problem bereits deutlich vereinfacht für den Fall, dass sich der Wagen auf der Geraden befindet. Ihr könnt für diesen Fall jetzt konkrete Zustände/Probleme beschreiben:

Zustand: Zentral auf Geraden (Zustand M)

Input: (linker Sensor = grau UND rechter Sensor = grau)

Output: fahre vorwärts

Zustand: rechts abgedriftet von Geraden (R1 oder R2)

Input: (linker Sensor = Schwarz)

Output: fahre nach links (bis Zustand = M)

Zustand: links abgedriftet von Geraden (L1 oder L2)

Input: (rechter Sensor = Schwarz)

Output: fahre nach rechts (bis Zustand = M)

Da das Lösen von Problemen ein **iterativer Prozess** ist, **testet** mal, ob der bisherige Lösungsansatz das tut, was er soll. Entwickelt dazu **Pseudocode** und setzt diesen anschließend im Open Roberta Lab um.

Pseudocode 0

```
wiederhole unendlich oft
  wenn (zentral auf Gerade)
    dann fahre vorwärts
  sonst wenn (rechts abgedriftet)
    dann fahre nach links
  sonst wenn (links abgedriftet)
    dann fahre nach rechts
```

Pseudocode 1

wiederhole unendlich oft

 wenn (Sensor_links = grau UND Sensor_rechts = grau)

 dann fahre vorwärts

 sonst wenn (Sensor_links = schwarz)

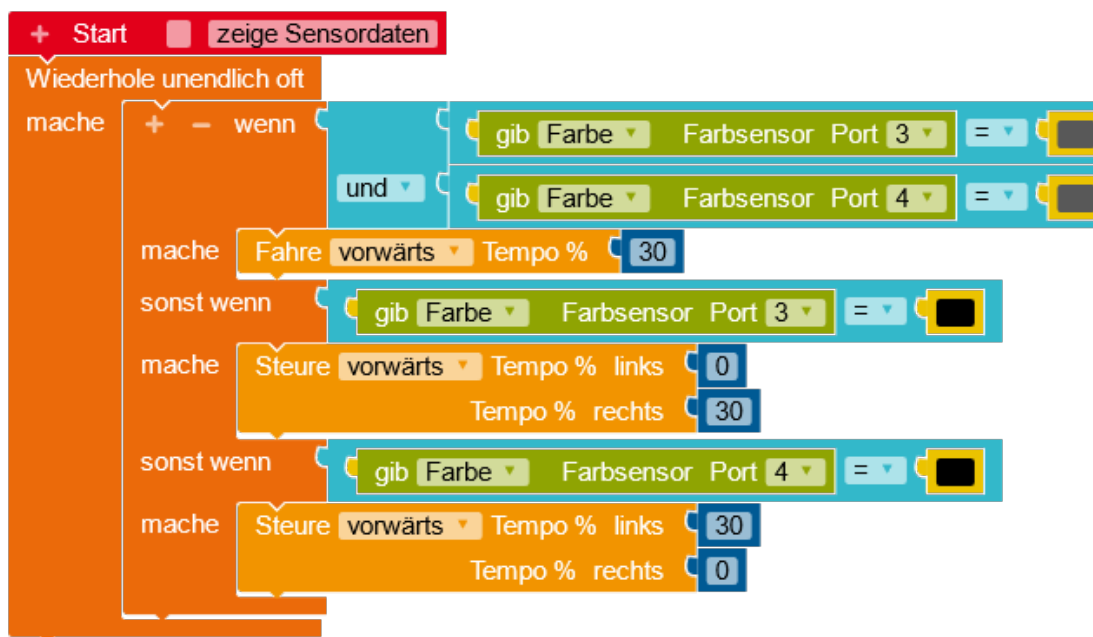
 dann fahre nach links

 sonst wenn (Sensor_rechts = schwarz)

 dann fahre nach rechts

NEPO Code

Den fertigen NEPO Code sieht ihr hier unten.



Beim Testen fällt auf, dass der Code nicht nur auf dem geraden Streckenabschnitt funktioniert, sondern auch in den Kurven. Das lässt sich dadurch erklären, dass die oben identifizierten Zustände ebenfalls in der Kurve vorliegen.

Glückwunsch!

Großartig! Das Lösen dieses Problems war echt nicht leicht! Dennoch konntet Ihr mittels Computational Thinkings eine kurze und simple Lösung entwickeln.

Den Erfolg müsst Ihr feiern!



Vielen Dank!
Wir freuen uns
über Euer Feedback.