

Algorithmisches Problemlösen mit dem Legoroboter

Didaktische Hinweise

Zielgruppe

Die Einheit zum Algorithmischen Problemlösen mithilfe eines Legoroboters richtet sich an Schüler:innen der Sekundarstufe I. Bewährt haben sich die Aufgaben für Schüler:innen im ersten Lernjahr im Fach Informatik sowohl in Jahrgang 7 als auch in Jahrgang 9 am Gymnasium.

Voraussetzungen

In den Materialien wird die Verwendung von Kontrollstrukturen nicht explizit erarbeitet und thematisiert. Vorerfahrungen im algorithmisch Problemlösen mit einer grafischen Programmiersprache wie z. B. Scratch¹ sind daher hilfreich. Die Einstiegsaufgaben sind jedoch hinreichend einfach, so dass die Materialien auch ohne Vorkenntnisse genutzt werden können. Im Unterricht sollte in diesem Fall aber ausreichend Zeit eingeplant werden, um an geeigneter Stelle beispielsweise die Verwendung von Schleifen und Verzweigungen zu thematisieren. Im Abschnitt *Didaktische Hinweise zu den Arbeitsblättern und Aufgaben* wird darauf hingewiesen, welche Aufgaben sich dazu eignen.

Die meisten Aufgaben lassen sich ohne die Verwendung von Variablen lösen. Variablen können aber zum Teil helfen, um Probleme allgemeiner oder eleganter zu lösen. Die Verwendung von Variablen kann daher je nach Lerngruppe und Unterrichtsverlauf thematisiert werden.

Weiterhin sollten die Schüler:innen mit dem schuleigenen Dateisystem und bei einer online-Programmierungsumgebung auch mit dem Herunterladen von Dateien über den Webbrowser vertraut sein. Die entsprechenden Schritte müssen ansonsten im Unterricht erläutert werden.

Vorerfahrungen mit anderen Sensor-Aktor-Systemen wie z. B. dem *Calliope mini* (s. [1]) oder dem *micro:bit* (s. [3]) sind nicht notwendig, können aber hilfreich sein. Schüler:innen haben in diesem Fall die Möglichkeiten, die erlernten Konzepte auf ein neues System zu übertragen.

Lernziele

Im ersten Teil der Einheit lernen die Schüler:innen die gezielte Bewegung eines Roboters mit Fahrgestell mithilfe zweier Motoren. Im zweiten Teil kommt die Reaktion auf die Eingaben, die die Sensoren liefern hinzu. Der Schwerpunkt liegt dabei auf der Rekonstruktion bekannter Systeme aus dem Alltag.

Die Einheit lässt sich in das Modul *technische Realisierung automatisierter Prozesse* aus dem Lernfeld „Automatisierte Prozesse“ einordnen (vgl. [7], S. 22):

¹ Scratch ist ein Projekt der Scratch Foundation in Zusammenarbeit mit der Lifelong Kindergarten Group des MIT Media Lab. Es ist kostenlos unter <https://scratch.mit.edu> erhältlich.

Basis	Vertiefung	Ergänzung
Die Schülerinnen und Schüler ...		
<ul style="list-style-type: none"> - benennen Typen von Sensoren, Aktoren und Verarbeitungskomponenten von technischen Geräten und ordnen sie der Eingabe, Verarbeitung und Ausgabe zu. - lesen Sensoren aus und steuern Aktoren an. 	<ul style="list-style-type: none"> - konstruieren aus vorgegebenen Bauteilen ein Informatiksystem, z. B. einen Roboter. 	<ul style="list-style-type: none"> - vergleichen verschiedene Konstruktionen zur Lösung des gleichen Problems.
<ul style="list-style-type: none"> - implementieren einen Algorithmus zur Steuerung einer technischen Komponente. 	<ul style="list-style-type: none"> - entwickeln einen Algorithmus zur Steuerung eines einfachen Informatiksystems. 	
P1.2 P2 P3.1 P4.1 I3.1	P1.2 P2 P5.1 I2 I3.2	P1.1 P2 P3.2 P4.2 P5.1 I3.2

Tabelle 1: Ausschnitt aus dem Modul *technische Realisierung automatisierter Prozesse*, Lernfeld *Automatisierte Prozesse*, vgl. [7], S.22

Wie Tabelle 1 zeigt, können im Rahmen dieser Einheit unterschiedliche prozessbezogene und inhaltsbezogene Kompetenzen erworben werden. Bei den prozessbezogenen Kompetenzen kommt neben dem Bereich *P2 Implementieren*, dem Bereich *P5 Informatiksysteme als Werkzeuge nutzen* eine besondere Bedeutung zu. Wenn die Schüler:innen geeignete Sensoren und deren Positionierung selbständig wählen, werden Hard- und Softwarewerkzeuge nicht nur bei der Problemlösung eingesetzt (P 5.1) sondern auch unter Berücksichtigung ihrer Vor- und Nachteile ausgewählt (P 5.2). Im Bereich der inhaltsbezogenen Kompetenzen steht der Bereich *I3 Informatiksysteme*, speziell das Konstruieren und Erschließen der Funktionsweise von Informatiksystemen im Vordergrund (I 3.2 und I 3.4).

Werkzeuge

Die Hardware – ein Legoroboter

Die Materialien wurden für die dritte Generation der Legoroboter, das LEGO® MINDSTORMS® Education EV3 Set (s. [10]) erstellt und zuletzt auch an das neuere System LEGO® Education Spike™ Prime angepasst (s. [12])². Die Aufgaben lassen sich aber größtenteils auch mithilfe des Vorgängermodells LEGO® MINDSTORMS® NXT bearbeiten. Für die Aufgaben wird davon ausgegangen, dass ein Roboter mit Fahrgestell mit zwei Motoren für die Räder konstruiert wurde. Getestet wurden die Aufgaben für den EV3 mit dem *Robot Educator* nach einer Anleitung der Lego Group [s. [10]] Dabei handelt es sich um ein Basisfahrgestell mit verschiedenen Sensoren. Die Aufgaben können aber auch mit dem Roberta EV3, der nach einer Anleitung der Roberta Initiative konstruiert wurde, gelöst werden (s. [8]). Für den *Spike™ Prime* wurden die Aufgaben mit dem Fahrgestell nach der Anleitung der Lego Group (s. [14]) getestet.

² Im Folgenden wird für den LEGO® MINDSTORMS® Education EV3 und den LEGO® Education Spike™ Prime zur besseren Lesbarkeit nur die Bezeichnungen EV3, Spike™ Prime bzw. allgemein Legoroboter verwendet.

Für die Aufgaben werden nur die Motoren und Sensoren benötigt, die im LEGO® MINDSTORMS® Education EV3 Set enthalten sind. Je nach Aufgabe müssen die Sensoren an geeigneter Stelle platziert werden. Die Sensoren des Spike Prime sind größtenteils vergleichbar zu den Sensoren des EV3. Zu beachten ist, dass der Farbsensor nur das reflektierte Licht aber nicht das Umgebungslicht erfasst und dass der Gyrosensor in den Hub (den programmierbaren Roboter-Stein) integriert ist.

Die Programmierumgebungen

Es gibt eine Vielzahl von Programmierumgebungen zur Programmierung des Legoroboters. Es erfolgt an dieser Stelle weder eine umfassende Aufzählung noch ein vollständiger Vergleich der Vor- und Nachteile der verschiedenen Programmierumgebungen. Für die Sekundarstufe I und insbesondere für Schüler:innen mit wenig Vorerfahrungen im algorithmischen Problemlösen bietet sich eine grafische Programmiersprache an. Erstellt wurden die Materialien ursprünglich für zwei Programmierumgebungen, die nach den Erfahrungen der Autorin für den Unterricht der Sekundarstufe I geeignet sind: *Microsoft MakeCode* in der Version 4.0.11 (s. [4]) sowie *Open Roberta Lab* in der Version 3.8.1 für das System *EV3 c4ev3* (s. [2]). Beide Werkzeuge ermöglichen die Übertragung der Programme mithilfe eines USB-Kabels und arbeiten mit der vorinstallierten Lego Firmware in der Version V1.10E

Inzwischen stellt die LEGO Group sowohl für den EV3 als auch für den Spike™ Prime eine auf Scratch basierende Programmierumgebung zur Verfügung (s. [13] bzw. [12]). Da sich Scratch als Programmierumgebung im Anfangsunterricht der Sekundarstufe I sehr bewährt hat, wurden die Materialien daher entsprechend erweitert und angepasst. Hier ist der Vorteil, dass die Verbindung zwischen Roboter und Programmierumgebung sowohl per USB-Kabel als auch sehr einfach über Bluetooth hergestellt werden kann.

Die Werkzeuge weisen nach Einschätzung der Autorin unterschiedliche Vor- und Nachteile auf, so dass das Werkzeug passend für die jeweilige Lerngruppe und die örtlichen Rahmenbedingungen gewählt werden muss. An dieser Stelle werden in Tabelle 2 einige Unterschiede gegenübergestellt, die während der Erstellung der Materialien aufgefallen sind. Weitere Besonderheiten werden im Rahmen der Erläuterung der einzelnen Arbeitsblätter bzw. Aufgaben angesprochen. Diese Auflistung erhebt jedoch keinen Anspruch auf Vollständigkeit.

	Microsoft MakeCode	Open Roberta Lab (System EV3 c4ev3)	EV3 Classroom & Spike™ App
Verfügbarkeit online/offline	Die Programmierumgebung Microsoft MakeCode wird im Browser gestartet. Steht keine Internetverbindung zur Verfügung ist jedoch auch die Installation als App möglich, die offline verwendet werden kann (s. [5]).	Die Programmierumgebung Open Roberta Lab wird im Browser gestartet. Um die Software in einem lokalen Netzwerk ohne Internetverbindung zur verwenden, steht eine Anleitung zur Einrichtung eines lokalen Servers auf einem Raspberry Pi zur Verfügung (s. [9]).	Die App steht für verschiedene Betriebssysteme als Installationsdatei zur Verfügung.
Sprache	Die Sprache kann auf Deutsch eingestellt werden. Die Bezeichnung der Blöcke wurde in Einzelfällen jedoch noch nicht ins Deutsche übersetzt.	Alle Blöcke stehen auf Deutsch zur Verfügung. Es kann aber auch eine Vielzahl weiterer Sprachen eingestellt werden.	Alle Blöcke stehen auf Deutsch zur Verfügung. Es kann aber auch eine Vielzahl weiterer Sprachen eingestellt werden.
Wechsel zu textbasierter Programmierung	Bei Microsoft MakeCode handelt es sich um eine grafische Programmiersprache. Die Blöcke werden im Hintergrund in JavaScript übersetzt und es kann zwischen der grafischen und textbasierten Darstellung des Programms hin- und hergewechselt werden.	In der Programmierumgebung Open Roberta Lab werden die Programme in der grafischen Sprache NEPO® implementiert. Es besteht die Möglichkeit sich den generierten Quellcode in einer textbasierten Sprache anzeigen zu lassen. Im Falle der Version EV3 c4ev3 wird der Code in C/C++ generiert. Im Quellcodeeditor kann auch der textbasierte Quellcode bearbeitet werden. Die Änderungen werden jedoch nicht in die grafische Ansicht übernommen.	Die Spike™ App bietet neben den Textblöcken, die an Scratch angelehnt sind, auch Symbolblöcke oder die Programmierung in Python an. Ein Wechsel zwischen den Darstellungen innerhalb eines Projektes scheint aber nicht möglich zu sein. EV3 Classroom stellt nur die Textblöcke zur Verfügung.

	Der Wechsel in die textbasierte Programmiersprache kann für Schüler:innen, die an der textbasierten Programmierung interessiert sind, von Vorteil sein und erleichtert die Einsicht, dass grafisches und textbasiertes Programmieren gleichwertig sind.		
Einteilung der Blöcke	Die Einteilung der Blöcke unterscheidet zwischen Schleifen und Logik, wobei die Kategorie <i>Logik</i> sowohl die Verzweigungen als auch Vergleichsblöcke zur Konstruktion von Bedingungen beinhaltet. Alle Blöcke, die sich direkt auf den EV3-Stein beziehen, sind im Bereich Stein zusammengefasst, unabhängig davon, ob es sich um eine Ein- oder Ausgabe handelt.	In der Übersicht der Blöcke kann zwischen der Anfänger- und der Expertenansicht unterschieden werden. Die Expertenansicht enthält zusätzliche Blöcke, so dass die einzelnen Kategorien für die verschiedenen Blöcke weiter unterteilt sind. Verzweigungen und Schleifen sind in der Kategorie <i>Kontrolle</i> zusammengefasst. Vergleichsblöcke zur Formulierung von Bedingungen befinden sich in der Kategorie <i>Logik</i> . Alle Ausgaben sind der Kategorie <i>Aktion</i> zugeordnet.	Die Kategorien orientieren sich an Scratch. Übernommen wurden z. B. die Kategorien <i>Ereignisse</i> , <i>Steuerung</i> und <i>Operatoren</i> . Blöcke zur Auswertung der Sensoren befinden sich in der Kategorie <i>Sensoren</i> . Ergänzt wurden die Kategorien <i>Motoren</i> und <i>Bewegung</i> für Blöcke zur Motorsteuerung. In der Spike™ App können wie bei Scratch weitere Blöcke hinzugeladen werden.
Erzeugen von Variablen	Variablen werden in der Blockpalette in der Kategorie <i>Variablen</i> erzeugt.	Variablen werden im Programm im obersten <i>Start</i> -Block durch das Anklicken des Pluszeichens erzeugt. Einerseits hat man dadurch stets einen Überblick über alle Variablen, andererseits können Variablen leicht ungewollt gelöscht werden, wenn das Minuszeichen vor der Variablen aus Versehen angeklickt wird.	Variablen werden in der Kategorie <i>Variablen</i> erzeugt und stehen dann als Block zur Verfügung. In der Spike™ App kann bei bestehender Verbindung zum Roboter der aktuelle Wert während der Ausführung des Programms in der Programmierumgebung angezeigt werden.

Sensoren	In der Kategorie <i>Sensoren</i> steht für jeden Sensor ein eigener Bereich mit den entsprechenden Blöcken zur Verwendung des Sensors zur Verfügung.	In der Kategorie <i>Sensoren</i> befindet sich für jeden Sensor ein Block, der den aktuellen Wert des Sensors liefert.	In der Kategorie <i>Sensoren</i> stehen für jeden Sensor verschiedene Blöcke zur Verfügung. Es gibt Blöcke zum Abfragen des aktuellen Sensorwerts und zum Erstellen von Bedingungen. Für den EV3 stehen auch vorgefertigte Blöcke zum Warten auf bestimmte Sensorwerte zur Verfügung. Für den Spike™ Prime können diese zusammengesetzt werden. In der Kategorie <i>Ereignis</i> stehen auch Blöcke zur Verfügung, um auf Ereignisse, die von bestimmten Sensorwerten ausgelöst werden, zu reagieren.
	Unterstützt werden der Berührungssensor, der Farbsensor, der Ultraschallsensor, der Kreisel sensor, der Infrarotsensor und die IR-Fernsteuerung.	Neben dem Berührungssensor, dem Farbsensor, dem Ultraschallsensor, dem Kreisel sensor, dem Infrarotsensor und der IR-Fernsteuerung werden auch einige Sensoren des Anbieter HiTechnic sowie der Geräuschsensor des Vorgängermodells Lego Mindstorms NXT unterstützt.	Für den EV3 werden der Berührungssensor, der Farbsensor, der Ultraschallsensor, der Kreisel sensor und der Infrarot-Sensor unterstützt. Für den Spike™ Prime werden der Farbsensor, der Kraftsensor, der Ultraschallsensor und der Gyrosensor unterstützt.
Aufbau und Ablauf der Programme	Ein neues Projekt enthält bereits die Blöcke <i>beim Start</i> und <i>dauerhaft</i> . Dies minimiert die Gefahr, dass bei Programmteilen, die in einer Endlosschleife ausgeführt werden sollen,	Jedes Programm beginnt mit dem Block <i>Start</i> , der bereits in jedem neuen Programm enthalten ist.	

	die entsprechende Schleife vergessen wird.		
	Durch die ereignisgesteuerte Reaktion auf Sensorwerte sowie den Einsatz mehrerer <i>dauerhaft</i> -Blöcke können parallele Programmabläufe implementiert werden. Soll das Programm an einer bestimmten Stelle beendet werden, muss daher explizit der Block <i>Programm beenden</i> aus dem Bereich <i>Stein</i> verwendet werden.	Das Programm wird sequenziell abgearbeitet. Danach ist das Programm beendet.	Die Ereignisgesteuerte Reaktion auf Sensorwerte und das Warten auf bestimmte Sensorwerte sind hier noch flexibler möglich als bei MakeCode. Dadurch werden verschiedene Programmieransätze unterstützt. Mithilfe des stoppe-Blocks kann das Programm beendet werden.
Simulator	Der Simulator zeigt nur die Ausgaben auf dem Stein und die Bewegung der Motoren an. Die Bewegungen des Roboters lassen sich daran nicht direkt erkennen.	Es steht ein umfangreicher Simulator zur Verfügung. Hier können auch Umgebungen für den Roboter, in denen sich dieser bewegt, simuliert werden. Algorithmen können dadurch umfangreich getestet werden, bevor sie auf den Roboter geladen werden. Auch wenn die Simulation nicht die Erfahrung mit dem echten Roboter ersetzt, kann die Simulation eine gute Ergänzung sein, insbesondere wenn nur wenige Roboter zur Verfügung stehen. Die Simulation enthält auch eine Anzeige aller aktuellen Sensorwerte des simulierten Roboters.	Ein Simulator steht nicht zur Verfügung. Die aktuellen Sensorwerte und der Status der Motoren werden bei bestehender Verbindung aber in der Programmierumgebung angezeigt. Für den Spike™ Prime können auch die aktuellen Werte der Variablen angezeigt werden.

Quellcode und ausführbare Datei	Das ausführbare Programm und der Quellcode befinden sich in derselben Datei, die entweder auf dem Roboter oder zur späteren Bearbeitung auf dem Rechner gespeichert werden kann.	Auf dem Roboter wird nur der ausführbare Code gespeichert. Möchte man das Programm zur späteren Bearbeitung auf dem Rechner speichern, muss es hingegen als <i>xml</i> -Datei exportiert werden.	Die letzte Version eines Projektes wird automatisch gespeichert und kann über die Startseite der Programmierungsumgebung erneut geöffnet werden. Der aktuelle Stand eines Projektes kann aber auch explizit gespeichert werden. Für den Spike™ Prime können die Programme im Speicher des Roboters über die Programmierungsumgebung aufgelistet und ggf. gelöscht werden. Zur weiteren Bearbeitung muss auch hier die Projektdatei auf dem Rechner gespeichert werden.
--	--	--	---

Tabelle 2: Gegenüberstellung der Programmierungsumgebungen Microsoft MakeCode, Open Roberta Lab und EV3 Classroom bzw. SPIKE™ App

Didaktische Hinweise zu den Arbeitsblättern und Aufgaben

Der Einstieg - Erste Fahrten

Das Besondere an den Legorobotern im Vergleich zu anderen Sensor-Aktor-Systemen, wie dem Calliope mini (s. [1]) oder dem micro:bit (s. [3]), sind die an Motoren befestigten Räder als Aktoren, mit denen sich der Roboter bewegen lässt. Der Einstieg erfolgt daher anhand erster Fahrten mit dem Roboter.

Alle Programmierumgebungen stellen Blöcke zum Ansteuern der Motoren zur Verfügung. Sie unterscheiden sich jedoch darin, ob sie zusätzlich Blöcke anbieten, die als Blackbox die Motoren so ansteuern, dass bestimmte Bewegungen ausgeführt werden, oder ob die Blöcke zum Einstellen der Drehrichtung und Geschwindigkeit der Motoren von den Schüler:innen so gewählt und kombiniert werden müssen, dass bestimmte Bewegungen erzeugt werden. Je nach Angebot der Blöcke ist das erste Arbeitsblatt daher unterschiedlich aufgebaut. Ziel ist dabei einerseits, dass die Schüler:innen schnell erste Erfolge erzielen, aber auch dass sie reflektieren, wie sich ein Befehl wie „fahre 100 cm geradeaus“ im Inneren zusammensetzt. Dazu gehört beispielsweise, dass die Motoren sich in eine bestimmte Richtung drehen und abhängig vom Umfang der Räder eine passende Anzahl Umdrehungen durchführen. Es folgen einige spezielle Hinweise zu den verschiedenen Programmierumgebungen.

Mit der Programmierumgebung MakeCode

Bei der Programmierumgebung *Microsoft MakeCode* gibt es verschiedene Blöcke zum Ansteuern der Motoren. Da beim Fahren meist zwei Motoren parallel laufen müssen, kommt dem *Kettenfahrzeug-Motoren*-Block hier eine besondere Bedeutung zu, da dieser zwei Motoren gleichzeitig ansteuert und koordiniert. In den ersten Aufgaben probieren die Schüler:innen verschiedene Parameter für den *Kettenfahrzeug-Motoren*-Block aus, um ein Gefühl für den Zusammenhang zwischen Rotation der Motoren und Bewegung des Roboters zu bekommen. Auf dieser Grundlage können dann auch komplexere Fahrten programmiert werden.

Mit der Programmierumgebung EV3Classroom

Die Programmierumgebung *EV3 Classroom* stellt in der Kategorie *Bewegung* einen ähnlichen Block zur Verfügung, um zwei Motoren parallel anzusteuern. Der für *MakeCode* beschriebene Ansatz des ersten Arbeitsblattes wurde daher übernommen. Die abschließenden Aufgaben laden zum Erkunden der weiteren Blöcke aus der Kategorie *Bewegung* ein.

Mit der Programmierumgebung Open Roberta Lab

Die Programmierumgebung *Open Roberta Lab* stellt zusätzlich zur direkten Ansteuerung der Motoren Blöcke für das Fahren zur Verfügung. Diese Blöcke können die Schüler:innen intuitiv verwenden, ohne die zugrundeliegenden Motorbewegungen zu durchschauen. Der im Arbeitsblatt verfolgte Ansatz hat jedoch zum Ziel, dass die Schüler:innen sich zunächst durch ein experimentelles Vorgehen den Zusammenhang zwischen Motorbewegung und Bewegung des Roboters erschließen. Dieses Verständnis hilft zum einen beim gezielten Lösen komplexerer Probleme und das Vorgehen ist zum anderen exemplarisch für das Erkunden der Funktionsweise eines technischen Systems. Deshalb wird in den Materialien für die Programmierumgebung *Open Roberta Lab* das experimentelle Implementieren von Roboterfahrten mit den Blöcken aus der Kategorie *Bewegen* der Verwendung der Blöcke aus der Kategorie *Fahren* vorangestellt. Die Blöcke aus der Kategorie *Fahren* kommen also erst zum Einsatz, wenn diese Blöcke auch von den Schüler:innen selbst erstellt werden könnten. Für

komplexere Fahrten können diese dann als Kurzschreibweise für Bewegungen, die sich aus mehreren Motor-Blöcken zusammensetzen, aufgefasst werden. Da das selbständige Implementieren von Bewegungen mithilfe der Motor-Blöcke anspruchsvoller ist als das Verwenden der Blöcke aus der Kategorie *Fahren*, stehen die Motor-Blöcke nur in der zweiten Ansicht für Fortgeschrittene zur Verfügung.

Für Lerngruppen, in denen das Zusammensetzen von Bewegungen aus den Motor-Blöcken für einige Schüler:innen zu komplex ist, kann eine andere Reihenfolge beim Erkunden der Blöcke aus den Kategorien *Bewegung* und *Fahren* durchaus sinnvoll sein. Bei der Arbeit mit der Programmierumgebung *Open Roberta Lab* können die Abschnitte *Mein erstes Programm* und *Drehungen* aus dem Arbeitsblatt *Erste Fahrten* daher je nach Lerngruppe auch hintenangestellt, nur optional für leistungsstärkere Schüler:innen zur Verfügung gestellt oder ganz weggelassen werden.

Mit der Programmierumgebung *Spike™ App*

Die *Spike™ App* bietet in der Kategorie *Bewegung* verschiedene Blöcke an, die das Fahrgestell des Roboters gezielte Bewegungen ausführen lässt. In der Kategorie *Motoren*, stehen auch Blöcke zur Verfügung, mit denen sich mehrere Motoren direkt starten lassen. Allerdings erfolgt hier beim Starten mehrerer Motoren keine Koordinierung z. B. zu einer vorwärts Bewegung. Diese Blöcke scheinen daher eher für andere Bewegungen als Fahrbewegungen gedacht zu sein. Das Arbeitsblatt konzentriert sich daher auf den Einsatz der Blöcke aus der Kategorie *Bewegung*. Es enthält aber auch entsprechende Aufgaben, die zum Erkunden und Reflektieren des Zusammenhangs von Motorbewegung, Roboterbewegung und den verwendeten Blöcken einladen.

Allgemeine Hinweise zu den Aufgaben

Die Aufgabe *Geometrische Figuren* bietet sich an, um die Verwendung von Schleifen zu thematisieren, wenn die Schüler:innen noch keine Vorerfahrungen im algorithmischen Problemlösen haben. Beim Quadrat beispielsweise müssen wiederholt die gleiche Strecke gefahren und eine 90°-Drehung ausgeführt werden.

Die Aufgabe *Slalomfahren* kann als Überleitung zur Verwendung von Sensoren dienen. Anders als in der Simulation sind die Bewegungen des Roboters nicht sehr exakt. Wird eine Slalomfahrt ohne Sensoren implementiert, so muss zum einen darauf geachtet werden, dass die Hindernisse immer an genau der gleichen Stelle stehen. Zum anderen wird die Steuerung mit zunehmender Anzahl an Hindernissen schwieriger, da sich die Ungenauigkeiten der Roboterbewegungen aufsummieren. Daraus kann der Wunsch resultieren, dass der Roboter die Hindernisse selbständig erkennt und im Bedarfsfall ein Ausweichmanöver durchführt.

Weitere Ausgaben des Roboters wie Bildschirmanzeigen und Töne werden nicht explizit eingeführt. Erfahrungsgemäß erkunden die Schüler:innen diese Ausgaben selbständig als Erweiterungen ihrer Problemlösungen.

Die Sensoren

Der grundsätzliche Umgang mit den Sensoren wird am Beispiel des Farbsensors erläutert. Dazu gehören das Erkunden und Interpretieren der Sensorwerte. Neben der Anzeige der Sensorwerte auf dem Display des Roboters werden bei *EV3Classroom* und der *Spike™ APP* die Werte auch in der Programmierumgebung angezeigt.

Im Umgang mit Sensoren stellen die Programmierumgebungen z. T. unterschiedliche algorithmische Ansätze zur Verfügung. Alle Programmierumgebungen verfügen über Blöcke, welche den aktuellen

Wert eines Sensors liefern. Diese können verwendet werden, um Bedingungen zu formulieren. Die Bedingungen können in einen *warte*-Block oder in eine Verzweigung eingesetzt werden. In *Microsoft MakeCode* und *EV3 Classroom* gibt es zu den einzelnen Sensoren passende *warte*-Blöcke. *Microsoft MakeCode* stellt außerdem einige Blöcke zur Kalibrierung zur Verfügung. Hier kann beispielsweise festgelegt werden, ab welchem Helligkeitswert die Werte als dunkel bzw. hell interpretiert werden sollen.

Microsoft MakeCode, *EV3 Classroom* und die *Spike™ App* bieten zusätzlich die Möglichkeit, ereignisorientiert auf einen Sensorwert zu reagieren. Zu jedem Sensor gibt es einen *wenn*-Block, mit dem darauf reagiert werden kann, dass eine bestimmte Bedingung erfüllt ist. Dadurch können verschiedene Programmstränge erstellt werden. Wenn das entsprechende Ereignis eintritt (der Sensor den eingestellten Wert meldet), wird dieser Programmteil ausgeführt. Durch die Ereignissteuerung kann auf die Endlosschleife, in der eine Bedingung explizit immer wieder überprüft wird, verzichtet werden. Dieser Ansatz ist daher zunächst intuitiver und weniger fehleranfällig und eignet sich besonders für Schüler:innen ohne Vorerfahrungen im algorithmischen Problemlösen. Bei komplexeren Algorithmen kann eine Vielzahl paralleler Programmstränge das Programm jedoch unübersichtlich und schwerer nachvollziehbar machen.

Für *Microsoft MakeCode*, *EV3 Classroom* und die *Spike™ App* werden alle drei Programmieransätze vorgestellt, so dass jede:r herausfinden kann, zu welchem Ansatz er oder sie den besten Zugang findet. Bei komplexeren Aufgabenstellungen kann ein Ansatz auch geeigneter sein als ein anderer. Je nach Lerngruppe kann es aber sinnvoll sein, sich zunächst auf einen Ansatz zu konzentrieren und die Einführung weitere Ansätze hintenanzustellen oder wegzulassen.

Für *Open Roberta Lab* wird sowohl die Verwendung des *warte*-Blocks als auch von Verzweigungen in Kombination mit einer Endlosschleife vorgestellt. Je nach Problemstellung müssen die Schüler:innen dann entscheiden, welche Variante besser geeignet ist. Da das Programm immer mit dem *Start*-Block beginnt, vergessen Schüler:innen hier häufiger die Endlosschleife für die Abfrage der Sensoren. Die Notwendigkeit sollte daher wiederholt verdeutlicht werden. Der tatsächliche Programmablauf wird dadurch aber transparenter als bei der ereignisgesteuerten Programmierung.

Für die Erprobung der Konzepte am Beispiel des Farbsensors und des Ultraschallsensors bietet es sich an, den Kontext des Reinigungsroboters (s. nächster Abschnitt) zu nutzen. Da sowohl das Erkennen eines Abgrunds z. B. an einer Treppe als auch das Ausweichen bei Hindernissen typische Anforderungen an einen Reinigungsroboter sind.

Lerngruppen ohne Vorerfahrungen im algorithmischen Problemlösen werden Bedingungen und Verzweigungen bei der Auswertung der Sensoren zunächst intuitiv verwenden. Aufbauend auf den Erfahrungen der Schüler:innen kann die Verwendung von Verzweigungen als algorithmische Kontrollstruktur im Unterricht thematisiert werden. Insbesondere das Zusammenspiel von Schleife und Verzweigung bei der Auswertung der Sensoren wäre hier transparent zu machen.

Zu den Aufgaben

An den Einstieg in die Arbeit mit Sensoren können sich die weiteren Aufgaben in beliebiger Reihenfolge anschließen. Die Problemstellungen sind so gehalten, dass sie mit beiden Programmierungsumgebungen bearbeitet werden können. Grundidee der meisten Aufgaben ist die Rekonstruktion von Geräten, die den Schüler:innen aus ihrer Lebenswelt bekannt sind. Die Problemstellung wird dazu ggf. vereinfacht, das grundsätzliche Problem bleibt aber erhalten. Die

Aufgabe beginnt in der Regel mit einer sehr starken Vereinfachung, um einen leichten Einstieg zu ermöglichen. Je nach Leistungsstärke können die Schüler:innen dann die Anregungen in den weiteren Aufgabenteilen nutzen, um die Komplexität ihrer Problemlösung zu steigern.

Es werden in den Aufgaben möglichst wenig Vorgaben bezüglich der Sensoren gemacht. Die Schüler:innen sollen selbst überlegen, welche Sensoren sinnvoll eingesetzt werden können und wo diese am Roboter platziert werden sollten. Die Lösung ist hier nicht immer eindeutig.

Unterschiedliche Ansätze und Implementierungen können in der Gruppe diskutiert werden.

Es folgen einige aufgabenspezifische Hinweise.

Bientanz

Der erste Teil der Aufgabe, die Simulation der Bienenbewegung beim Schwänzeltanz, kann ohne die Verwendung von Sensoren gelöst werden. Erst zum Auffinden der Futterquelle wird der Farbsensor benötigt. Diese Aufgabe kann daher zur Überleitung vom Fahren zur Reaktion auf Sensoren dienen.

Eine Verfeinerung der Simulation von Bientänzen kann auch Anlass zur Kommunikation der Roboter untereinander geben.

Einparkhilfe

Der zweite Teil dieser Aufgabe eignet sich ebenfalls zur Überleitung von reinen Fahrbewegungen zur Reaktion auf Sensoren.

Weitere Materialien zum Thema *autonomes Fahren* sind z. B. in der Dokumentation zu der Programmierumgebung *Microsoft MakeCode* unter dem Punkt Unterrichtsmaterial – Informatik zu finden (s. [6]).

Reinigungsroboter

Auch wenn der Legoroboter weder saugen noch wischen kann, kann die Bewegung eines Reinigungsroboters simuliert werden. Dabei sind verschiedene Aspekte zu berücksichtigen. Der Roboter muss sich zufällig oder systematisch durch den Raum bewegen, Hindernissen ausweichen und darf nicht die Treppe hinunterfallen. Hindernisse können sowohl mit dem Ultraschall- als auch mit dem Farbsensor erkannt werden. Diese Aufgabe bietet daher viel Potenzial für unterschiedliche Lösungsansätze, die von den Schüler:innen diskutiert und verglichen werden können.

Wird mit dem Berührungssensor gearbeitet, bietet sich das Robotermodell Roberta EV3 nach der Anleitung aus dem Open Roberta Projekt an (s. [2]). Dieses beinhaltet eine Vorrichtung, welche das Auslösen der Berührungssensoren optimiert.

Als Einstieg bietet sich ein kurzes Video von der Fahrt eines Wisch- oder Staubsaugerroboters durch einen Wohnraum an, um die genaue Problemstellung von den Schüler:innen selbst herausarbeiten zu lassen.

Diese Aufgabe eignet sich auch als Kontext für die Aufgaben des zweiten Arbeitsblattes, die in das Arbeiten mit Sensoren einführen.

Orientierung an Bodenmarkierungen

Markierungen auf dem Boden sind eine gute Möglichkeit, Robotern Grenzen oder einen Weg vorzugeben. Das Verfolgen einer schwarzen Linie ist daher ein klassisches Problem bei der Roboterprogrammierung, das vielfältige Anwendungen hat. Auch in Roboterwettbewerben ist das Verfolgen einer Linie häufig Teil der Aufgabe.

Einfacher ist das Verfolgen einer Linie, wenn zwei Farbsensoren zur Verfügung stehen, die links und rechts am Roboter montiert werden, so dass das Verlassen der Linie an beiden Seiten registriert werden kann. Das Problem kann aber auch mithilfe eines einzelnen Farbsensors gelöst werden.

Zum Testen sollten Bahnen mit unterschiedlichen schwarzen Linien zur Verfügung stehen.

Der Roboter im Straßenverkehr

Am Beispiel der Ampel und der Verkehrsschilder soll gezeigt werden, dass Signale, die für uns Menschen gut und leicht zu erkennen sind, nicht unbedingt für einen Roboter geeignet sind. Andere Positionierungen und Codierungen können hier sinnvoll sein. In Bezug auf den Legoroboter bietet sich die Positionierung auf dem Boden an. Die Abbildung der Ampel legt nahe, sich auf die Farben Rot und Grün zu beschränken, diese Entscheidung kann aber den Schüler:innen überlassen werden. Für die eingeschränkte Menge an Verkehrszeichen bietet sich eine Codierung durch einfarbige Markierungen an.

Insbesondere die erste Aufgabe zur Reaktion auf eine Ampel kann gut mit dem Folgen einer schwarzen Linie kombiniert werden.

Solarpanel

Der Farbsensor misst die direkte Reflektion nur genau, wenn er sich unmittelbar über der reflektierenden Oberfläche befindet. Da sich die Lichtquelle, nach der sich der Roboter ausrichten soll, weiter weg befindet, muss hier die Helligkeit des Umgebungslichts verwendet werden.

In einer einfachen Variante kann vorher der Helligkeitswert ermittelt werden, der erreicht wird, wenn der Farbsensor auf die Lampe ausgerichtet ist. Der Roboter dreht sich dann solange bis dieser Helligkeitswert erreicht wurde.

In einer komplexeren Lösung ermittelt der Roboter selbständig den hellsten Bereich in seiner Umgebung unabhängig von der absoluten Helligkeit. Dazu wird jedoch eine Variable benötigt, in der sich der Roboter den bislang hellsten gemessenen Wert und die zugehörige Position merken kann. Zur Bestimmung der aktuellen Position des Roboters benötigen wir den Kreisel sensor. Dieser wird zu Beginn auf 0 gesetzt, so dass er dann den Drehwinkel zur Ausgangsposition angibt.

Da für diese Aufgabe die Helligkeit des Umgebungslichts erfasst werden muss, ist sie nicht für die zum Testen verwendete Version des Spike™ Prime geeignet.

Ausblick

Alle Problemstellungen, die bisher betrachtet wurden, lassen sich mithilfe eines Standardrobotermodells mit Fahrgestell lösen. Komplexere Problemstellungen können auch einen Umbau bzw. eine Konstruktion des Roboters beinhalten. Dazu bietet sich eine Exkursion an, in der Schüler:innen bewusst nach Informatiksystem in ihrer Umwelt suchen können, die sie mithilfe eines Legoroboters (ggf. vereinfacht) nachbauen.

Literaturverzeichnis

- [1] Calliope GmbH (2029). *Calliope*. <https://calliope.cc/> [Datum des Zugriffs: 06.02.2020]
- [2] Das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme. *Open Roberta Lab. EV3 c4ev3. Release 3.8.1* <https://lab.open-roberta.org/> [Datum des Zugriffs: 06.02.2020]
- [3] Micro:bit Educational Foundation. *micro:bit* <https://microbit.org/de/> [Datum des Zugriffs: 06.02.2020]
- [4] Microsoft (2019). *Microsoft MakeCode. Version 4.0.11* <https://makecode.mindstorms.com/> [Datum des Zugriffs: 06.02.2020]
- [5] Microsoft (2018). *MakeCode offline verwenden*. <https://makecode.mindstorms.com/offline> [Datum des Zugriffs: 06.02.2020]
- [6] Microsoft (2018). *Unterrichtsmaterial – Informatik. Programmieraufgaben*. <https://makecode.mindstorms.com/coding> [Datum des Zugriffs: 06.02.2020]
- [7] Niedersächsisches Kultusministerium (2014). *Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5 – 10. Informatik*. Hannover: Unidruck
- [8] Roberta Initiative (2017). *Bauanleitung Roberta EV3*. https://www.roberta-home.de/fileadmin/user_upload/Bauanleitung_EV3_05_12_2017.pdf [Datum des Zugriffs: 06.02.2020]
- [9] Roberta Initiative. *Open Roberta Lab Server*. (<https://www.roberta-home.de/lab/lokale-installation/>) [Datum des Zugriffs: 07.02.2020]
- [10] The LEGO Group (2015). *Bauanleitung für den Robot Educator*. <https://education.lego.com/de-de/support/mindstorms-ev3/building-instructions#robot> [Datum des Zugriffs: 06.02.2020]
- [11] The LEGO Group (2019). *Lego Mindstorms Education EV3* <https://education.lego.com/de-de/product/mindstorms-ev3> [Datum des Zugriffs: 06.02.2020]
- [12] The LEGO Group (2021). *LEGO® Education SPIKE™ App v. 2.0.4*. <https://education.lego.com/de-de/downloads/spike-app/software> [Datum des Zugriffs: 21.03.2022]
- [13] The LEGO Group (2021). *Software für MINDSTORMS v. 1.5.3*. <https://education.lego.com/de-de/downloads/mindstorms-ev3/software> [Datum des Zugriffs: 21.03.2022]
- [14] The LEGO Group (2021). *SPIKE™ Prime-Bauanleitungen*. <https://education.lego.com/de-de/product-resources/spike-prime/downloads/bauanleitungen> [Datum des Zugriffs: 21.03.2022]

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#).

Erprobt wurden die Materialien mit dem Roboter LEGO® MINDSTORMS® Education EV3.

Die vorliegenden Materialien werden nicht von der LEGO Gruppe gesponsert, genehmigt oder unterstützt.