

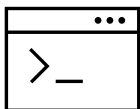
Leitfaden zur Einführung in die Robotik

Grundlagen der Programmierung zur Vorbereitung
auf die Teilnahme am RoboCupJunior Rescue



Inhaltsverzeichnis

Vorwort	2
Materialien für LehrerInnen	3
Einleitung	4
Stundenplanung - Übersicht.....	5
1. Unterrichtsstunde	8
2. Unterrichtsstunde	9
3. Unterrichtsstunde	10
4. Unterrichtsstunde	12
5. Unterrichtsstunde	13
6. Unterrichtsstunde	15
7. Unterrichtsstunde	16
8. Unterrichtsstunde	18
9. Unterrichtsstunde	19
10. Unterrichtsstunde	21
Reflektion für LehrerInnen	22
Lösungen	26
Lösung – Arbeitsblatt Grundelemente.....	27
Lösung – Arbeitsblatt Grundelemente.....	28
Lösung – Arbeitsblatt Grundelemente.....	29
Lösung – Arbeitsblatt Linienverfolger	30
Lösung – Arbeitsblatt Wandverfolger.....	31
Arbeitsblätter	32
Einstieg in die Programmierung	33
Linienverfolgung	34
Ultraschallsensor	35
Farberkennung	36
Reflektion	37
Grundelemente	38
Linienverfolger	42
Wandverfolger	44
Schlusswort	46
Anhang.....	47



Vorwort

Dieser Leitfaden unterstützt Lehrkräfte dabei, SchülerInnen praxisnah an die Themen Programmierung und Robotik heranzuführen. Durch aufeinander aufbauende Aufgaben erwerben die Lernenden Schritt für Schritt grundlegende Programmier- und Robotikkenntnisse.

Als Grundlage dient der MiBo Educational Line Following Robot Kit. Die Programmierung erfolgt über die blockbasierte Programmierumgebung Open Roberta auf der von rosslight robotics bereitgestellten Plattform, die über unsere Unternehmenswebsite erreichbar ist. Dadurch erhalten die SchülerInnen einen einfachen Einstieg in die Robotik, ohne dass Vorkenntnisse erforderlich sind.

Der Leitfaden orientiert sich am RoboCupJunior und vermittelt grundlegende Fähigkeiten für die Kategorien Rescue Line und Rescue Maze. Ziel ist es, die SchülerInnen schrittweise an die Denkweisen und Herausforderungen robotischer Wettbewerbe heranzuführen und sie auf eine mögliche Teilnahme vorzubereiten.

Die Unterrichtseinheiten 1 bis 6 bilden einen vollständigen Grundkurs und können mit dem MiBo in seiner Grundausstattung durchgeführt werden. Dabei lernen die SchülerInnen die Grundlagen der Programmierung kennen und entwickeln einen eigenen Linienfolger.

Ab der 7. Unterrichtsstunde folgen optionale Erweiterungen mit zusätzlicher Hardware. Ultraschall- und Farbsensoren sind nicht Bestandteil des MiBo-Grundkits, können jedoch problemlos ergänzt und gemeinsam mit dem Roboter eingesetzt werden.

Ergänzend enthält dieser Leitfaden Arbeitsblätter zur Vertiefung der vermittelten Inhalte. Die Aufgaben sind bewusst in kleine, überschaubare Schritte gegliedert, um schnelle Erfolgserlebnisse zu ermöglichen und die Motivation der Lernenden nachhaltig zu fördern.

Die angegebenen Zeitvorgaben dienen als Orientierung und können je nach Lerngruppe angepasst werden.

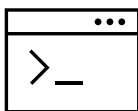
Neben fachlichen Kompetenzen fördert die Arbeit mit Robotikprojekten auch Teamarbeit, Kommunikation, Ausdauer und strukturiertes Problemlösen.

Wir wünschen Ihnen und Ihren SchülerInnen viel Freude und Erfolg bei der Einführung in die Welt der Robotik und vielleicht schon bald bei der ersten Teilnahme am RoboCupJunior.



Materialien für LehrerInnen

Zur Vorbereitung und Unterstützung der
Unterrichtseinheiten



Einführung in die Robotik – Grundlagen der Programmierung

Einleitung

Zusammenfassung

Dieser Leitfaden unterstützt Lehrkräfte dabei, SchülerInnen schrittweise an die Grundlagen der Programmierung heranzuführen. Durch praxisnahe Aufgaben erwerben die Lernenden zunächst grundlegende Programmierkenntnisse, bevor sie diese auf robotische Anwendungen und die speziellen Anforderungen des RoboCupJunior übertragen.

Material

Folgendes Material wird pro SchülerIn oder Kleingruppe benötigt:

- ein PC, Tablet oder Notebook
- Robotik-Set (*Empfehlung für den Einstieg: MiBo Line Following Robot Kit*)
- Zugänge über die Website robotics-rosslight.de zu Open Roberta

Für die optionalen Erweiterungen ab Unterrichtsstunde 7:

- zusätzliche Sensoren entsprechend der jeweiligen Unterrichtseinheit, z. B. Ultraschall- oder Farbsensor

Zeitaufwand

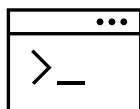
10 Unterrichtsstunden à 45 Minuten

Voraussetzungen

- Empfohlen für Klassenstufen 6 bis 10
- Keine Programmierkenntnisse erforderlich

Lernziele

- Die SchülerInnen lernen die Grundlagen der Programmierung sowie den Umgang mit Robotern kennen.
- Die SchülerInnen entwickeln ein Grundverständnis für den Aufbau eines Roboters und die dabei relevanten Komponenten.



Einführung in die Robotik – Grundlagen der Programmierung

Stundenplanung - Übersicht

Hinweis

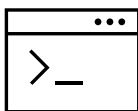
Die folgende Stundenplanung dient als Orientierung und kann bei Bedarf individuell angepasst werden. Die Dauer einer Unterrichtsstunde beträgt 45 Minuten.

Stunde	Stundeninhalt
Grundkurs: Arbeiten mit dem MiBo	
1. Stunde: Einführung	Stellen Sie den SchülerInnen das Vorhaben für die kommenden Wochen vor und geben Sie einen Einblick in den RoboCupJunior. Dadurch erhalten die SchülerInnen einen ersten Eindruck davon, welche Möglichkeiten die Roboterprogrammierung bietet. Machen Sie die SchülerInnen mit der Programmierumgebung ihres Robotersystems vertraut. Stellen Sie außerdem die verschiedenen Kategorien des RoboCupJunior vor und geben Sie einen Überblick darüber, wo die entsprechenden Codeblöcke zu finden sind.
2. Stunde: Bau der Roboter	Lassen Sie die SchülerInnen die Roboter zusammenbauen. Für den Einstieg empfiehlt es sich, den Roboter zunächst gemäß der Anleitung des MiBo Robot Kits zusammenzubauen, sodass eine funktionsfähige Grundversion entsteht. Zusätzliche Sensoren und Erweiterungen können später jederzeit ergänzt werden.
3. Stunde: Fortsetzung Roboterbau und Einstieg in die Programmierung	Der Zusammenbau der Roboter wird fortgeführt und abgeschlossen. Erfahrungsgemäß benötigen die SchülerInnen unterschiedlich viel Zeit, sodass es sinnvoll ist, bereits während der letzten Bauphasen erste Programmieraufgaben bereitzustellen. Anhand einfacher Übungen lernen die SchülerInnen die Grundlagen der Programmierung selbstständig kennen. Es empfiehlt sich, die SchülerInnen in Zweiergruppen oder kleinen Teams arbeiten zu lassen, damit sie sich gegenseitig unterstützen und gemeinsam Lösungen entwickeln können.

4. Stunde: Fortsetzung des Einstiegs in die Programmierung	Die SchülerInnen setzen ihre Arbeit an den ersten Programmieraufgaben fort und vertiefen die im vorherigen Unterricht erlernten Grundlagen.
5. Stunde: Einstieg in die Linienverfolgung	Zu Beginn der Unterrichtsstunde empfiehlt es sich, den SchülerInnen die für die Linienverfolgung benötigten Sensoren vorzustellen und zu erläutern, wie deren Werte ausgelesen werden können. Darüber hinaus sollten die entsprechenden Programmierbausteine sowie deren Funktion kurz erklärt werden. Im Anschluss machen sich die SchülerInnen anhand der Aufgaben mit den Grundlagen der Linienverfolgung vertraut.
6. Stunde: Fortsetzung Linienverfolgung	Die SchülerInnen arbeiten weiter an den Aufgaben zur Linienverfolgung und vertiefen ihr Verständnis für die Steuerung des Roboters mithilfe der Sensorwerte.
Erweiterungskurs: Arbeiten mit zusätzlichen Sensoren	
7. Stunde: Ultraschallsensor	Zu Beginn der Unterrichtsstunde empfiehlt es sich, den SchülerInnen den Ultraschallsensor sowie dessen Funktionsweise vorzustellen. Anschließend erarbeiten die SchülerInnen mithilfe der Aufgaben die Grundlagen der Abstandsmessung und lernen, die Sensordaten in ihren Programmen zu nutzen.
8. Stunde: Fortsetzung Ultraschallsensor	Die SchülerInnen setzen ihre Arbeit an den Aufgaben zum Ultraschallsensor fort und vertiefen die Anwendung der zuvor erlernten Konzepte.



9. Stunde: Farberkennung	Zu Beginn der Unterrichtsstunde empfiehlt es sich, den SchülerInnen den Farbsensor vorzustellen und zu erläutern, wie die Sensorwerte ausgelesen werden können. Gegebenenfalls kann es zudem hilfreich sein, die hierfür benötigten Codeblöcke kurz vorzustellen und deren Funktion zu erklären. Anschließend bearbeiten die SchülerInnen die Aufgaben zur Farberkennung und lernen, verschiedene Farbelemente auf dem Boden mithilfe des Sensors zu erkennen.
10. Stunde: Fortsetzung Farberkennung und Reflektion	Die SchülerInnen setzen ihre Arbeit an den Aufgaben zur Farberkennung fort. Die letzten 10 bis 15 Minuten der Unterrichtsstunde sollten genutzt werden, um die Ergebnisse und Erfahrungen mithilfe des Arbeitsblatts „Reflektion“ gemeinsam zu reflektieren. Hinweis: <i>Bei Bedarf können relevante Programmierbausteine nochmals besprochen werden.</i>



Einführung in die Robotik – Grundlagen der Programmierung

1. Unterrichtsstunde

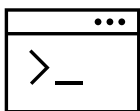
Als Einstieg in die Unterrichtseinheit empfiehlt es sich, den SchülerInnen einen Überblick über die Inhalte der kommenden Wochen zu geben und aufzuzeigen, worauf sie im Verlauf des Kurses hinarbeiten werden.

Hierfür bietet es sich an, den RoboCupJunior kurz vorzustellen und anschließend auf die Kategorien **Rescue Line** und **Rescue Maze** einzugehen. Folgende Punkte können dabei hilfreich sein:

- Der RoboCupJunior ist ein internationaler Wettbewerb für SchülerInnen.
- Roboter absolvieren autonome Rettungsmissionen in einem für Menschen schwer zugänglichen Gebiet.
- Die Roboter sollen einen vorgegebenen Parcours selbstständig bewältigen.
- In **Rescue Line** folgt der Roboter einer Linie, die zu verschütteten Personen führt.
- Der Parcours enthält verschiedene Herausforderungen wie Kurven, Kreuzungen, Lücken in der Linie, Rampen, Wippen sowie Hindernisse oder „Bumper“, beispielsweise Schaschlikspieße auf dem Boden
- In **Rescue Maze** navigiert der Roboter selbstständig durch ein Labyrinth. Zusätzliche Elemente wie schwarze, blaue oder rote Felder, Schutt und Hindernisse erhöhen den Schwierigkeitsgrad.

Zur visuellen Veranschaulichung eignen sich kurze Videos, beispielsweise von nationalen oder internationalen RoboCupJunior-Wettbewerben.

Nach der Einführung in den RoboCupJunior und die Kategorien Rescue Line und Rescue Maze stellen Sie den SchülerInnen die an Ihrer Schule verwendete Programmierungsumgebung vor. Geben Sie den SchülerInnen einen ersten Einblick in den Aufbau der Programmierungsumgebung, die verfügbaren Codeblöcke sowie deren grundlegende Funktionen.



Einführung in die Robotik – Grundlagen der Programmierung

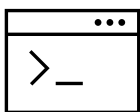
2. Unterrichtsstunde

In der zweiten Unterrichtsstunde bauen die SchülerInnen ihre Roboter zusammen. Für einen möglichst schnellen Einstieg in die Programmierung empfiehlt es sich, zunächst einen einfachen Grundaufbau zu wählen. Daher sollten die SchülerInnen den Roboter zunächst gemäß der dem Set beiliegenden Bauanleitung montieren.

Hinweis

Die standardmäßig am Roboter verbauten Sensoren sind für den Einstieg vollkommen ausreichend. Sollten die SchülerInnen zu einem späteren Zeitpunkt weitere Sensoren nutzen wollen, können diese jederzeit ergänzt werden. Für den Einstieg empfehlen wir jedoch, sich an den in der Bauanleitung beschriebenen Aufbau zu halten.

Die zweite Unterrichtsstunde eignet sich außerdem, um offene oder neu entstandene Fragen zu klären. Je nach Umfang können diese gemeinsam mit der gesamten Lerngruppe oder individuell beantwortet werden, während die übrigen SchülerInnen den Zusammenbau ihrer Roboter fortsetzen.



Einführung in die Robotik – Grundlagen der Programmierung

3. Unterrichtsstunde

In der dritten Unterrichtsstunde sollte der Zusammenbau der Roboter fortgesetzt und idealerweise abgeschlossen werden. Beachten Sie dabei, dass die benötigte Zeit je nach Vorkenntnissen der SchülerInnen variieren kann.

Sollten einige Schüler den Zusammenbau bereits abgeschlossen haben, dann empfiehlt es sich, dass diese bereits mit dem ersten Arbeitsblatt „Einstieg in die Programmierung“ beginnen. Dieses befindet sich im Kapitel „Arbeitsblätter“. Alternativ können Sie das Arbeitsblatt „Grundelemente“ nutzen. Beide Arbeitsblätter verfolgen dieselben Lernziele, setzen jedoch unterschiedliche Schwerpunkte. Musterlösungen finden Sie im Abschnitt „Lösungen“.

Der Schwerpunkt dieser Unterrichtsstunde liegt auf den Grundlagen der Roboterprogrammierung. Dazu zählen insbesondere das Geradeausfahren, das Abbiegen sowie die Verwendung von Schleifen.

Für die Bearbeitung der Aufgaben werden Schleifen benötigt. Da dieses Konzept von den SchülerInnen selbst erarbeitet werden soll, empfiehlt es sich, die Funktionsweise nicht vorab ausführlich zu erklären. Sollten mehrere SchülerInnen Schwierigkeiten bei der Lösungsfindung haben, kann die Arbeitsphase unterbrochen werden, um das Konzept gemeinsam zu besprechen. Gegebenenfalls kann es auch sinnvoll sein, die benötigten Codeblöcke kurz vorzustellen und zu zeigen, wo diese in der Programmierungsumgebung zu finden sind.

Nachfolgend sind einige wichtige Punkte zum Thema Schleifen aufgeführt, die als Orientierungshilfe für die Einführung im Unterricht dienen können:

- Schleifen ermöglichen die wiederholte Ausführung bestimmter Codeblöcke.
- Die Anzahl der Wiederholungen kann individuell festgelegt und von verschiedenen Bedingungen abhängig gemacht werden.
- Durch den Einsatz von Schleifen lassen sich Codezeilen einsparen und Programme übersichtlicher gestalten.
 - Eine übersichtliche Darstellung des Codes ist besonders bei längerem und komplexem Code nützlich, da so die Fehlerfindung und -behebung erheblich erleichtert wird.



Haben die SchülerInnen Schwierigkeiten zu erkennen, wann Schleifen eingesetzt werden sollten, können folgende Denkanstöße hilfreich sein:

- Welche Aufgabe erfüllt die Schleife?
- Welcher Programmteil soll wiederholt werden?
- Wie oft soll dieser Programmteil ausgeführt werden?
- An welcher Stelle im Programm muss die Schleife eingefügt werden, damit die Aufgabe erfolgreich gelöst wird?

Sollten SchülerInnen die Aufgaben bereits frühzeitig abschließen, können zusätzliche Muster, beispielsweise Dreiecke oder Sterne, abgefahren werden.

Hinweis

Auf den Arbeitsblättern befinden sich neben jeder Aufgabe Kästchen zum Abhaken oder Abstempeln. Diese ermöglichen es den SchülerInnen, ihren Lernfortschritt sichtbar nachzuverfolgen.

Gleichzeitig erhalten Lehrkräfte einen schnellen Überblick über die bearbeiteten Aufgaben und können die Ergebnisse der SchülerInnen bei Bedarf gemeinsam besprechen.

Hinweis zur Verbindung des Roboters mit Open Roberta

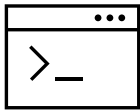
Damit eine Verbindung zwischen dem Roboter und der Open-Roberta-Programmierungsumgebung hergestellt werden kann, muss auf dem verwendeten Computer der Open Roberta Connector installiert sein.

Für die Einrichtung gehen Sie wie folgt vor:

1. Starten Sie den Open Roberta Connector.
2. Verbinden Sie den Roboter mit dem Computer.
3. Aktivieren Sie die Option „anderen Server verwenden“.
4. Tragen Sie anschließend folgende Adresse ein:

openroberta.services.external.rosslight.de

Nur wenn diese Adresse korrekt hinterlegt ist, kann eine Verbindung zum Roboter hergestellt werden. Sollte keine Verbindung zustande kommen, prüfen Sie daher zunächst die Einstellungen des Open Roberta Connectors.



Einführung in die Robotik – Grundlagen der Programmierung

4. Unterrichtsstunde

Spätestens in der vierten Unterrichtsstunde sollten alle SchülerInnen mit den Grundlagen der Programmierung begonnen haben. Beachten Sie dabei die Hinweise aus der vorherigen Unterrichtsstunde zu den Arbeitsblättern.

Um die SchülerInnen bei der Bearbeitung der Aufgaben zu unterstützen, können gezielte Fragen hilfreich sein. Diese fördern das eigenständige Problemlösen und lenken die Aufmerksamkeit auf die wesentlichen Schritte der Aufgabenbearbeitung.

Folgende Fragen können dabei unterstützen:

- Welche der Codeblöcke werden zur Lösung der Aufgabe benötigt?
- Wie lässt sich die Aufgabe Schritt für Schritt mit eigenen Worten beschreiben?
- Welche Aktionen muss der Roboter ausführen, um das gewünschte Ergebnis zu erreichen?
- In welchen Kategorien der Programmierumgebung könnten sich die benötigten Codeblöcke finden?
- Welche Codeblöcke aus diesen Kategorien werden tatsächlich benötigt?

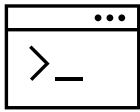
Hinweis

Auf den Arbeitsblättern befinden sich neben jeder Aufgabe Kästchen zum Abhaken oder Abstempeln. Diese ermöglichen es den SchülerInnen, ihren Lernfortschritt sichtbar nachzuverfolgen.

Gleichzeitig erhalten Lehrkräfte einen schnellen Überblick über die bearbeiteten Aufgaben und können die Ergebnisse der SchülerInnen bei Bedarf gemeinsam besprechen.

Hinweis

Gegebenenfalls kann es hilfreich sein, die für die jeweilige Aufgabe benötigten Programmierbausteine nochmals gemeinsam zu besprechen.



Einführung in die Robotik – Grundlagen der Programmierung

5. Unterrichtsstunde

In dieser Unterrichtsstunde erfolgt der Einstieg in die Linienverfolgung. Dabei lernen die SchülerInnen, Sensordaten auszulesen und mithilfe von Variablen und Verzweigungen auf unterschiedliche Situationen zu reagieren.

Zu Beginn empfiehlt es sich, gemeinsam das grundlegende Prinzip der Linienverfolgung zu besprechen. Dabei kann die Funktionsweise vereinfacht über das „Wenn ... dann ...“-Prinzip erklärt werden:

Wenn ein Sensor die schwarze Linie erkennt, dann soll der Roboter seine Fahrtrichtung entsprechend anpassen.

Die Sensoren erfassen dabei kontinuierlich den Untergrund. Abhängig davon, ob eine schwarze Linie erkannt wird oder nicht, muss der Roboter unterschiedlich reagieren. Damit der Roboter der Linie folgen kann, wird diese Abfrage fortlaufend wiederholt.

Arbeiten mit Variablen:

Für die Programmierung der Linienverfolgung werden zunächst Variablen verwendet. Diese speichern Informationen darüber, ob die schwarze Linie von einem Sensor erkannt wurde. Beispielsweise kann eine Variable für den linken und eine weitere für den rechten Sensor angelegt werden.

Vereinfacht bedeutet dies:

- Linker Sensor erkennt Schwarz → Variable „schwarz links“ wird auf wahr gesetzt.
- Rechter Sensor erkennt Schwarz → Variable „schwarz rechts“ wird auf wahr gesetzt.
- Erkennt der jeweilige Sensor keine schwarze Linie, erhält die entsprechende Variable den Wert falsch.

So kann das Programm jederzeit auf die gespeicherten Informationen zugreifen und anhand dieser entscheiden, wie der Roboter reagieren soll.

Im nächsten Schritt werden Verzweigungen genutzt. Mithilfe von „Wenn ... dann ...“-Bedingungen prüft der Roboter die zuvor gespeicherten Informationen und passt seine Bewegung entsprechend an.



Das Prinzip lässt sich beispielsweise so erklären:

Wenn links eine schwarze Linie erkannt wird, dann korrigiert der Roboter seine Fahrtrichtung in die entsprechende Richtung.

Wenn rechts eine schwarze Linie erkannt wird, dann korrigiert er in die andere Richtung.

Wenn keine Korrektur notwendig ist, dann fährt der Roboter geradeaus weiter.

Diese Abfragen befinden sich innerhalb einer Schleife und werden fortlaufend wiederholt. Der Roboter überprüft somit während der Fahrt immer wieder die Sensorwerte und kann seine Fahrtrichtung kontinuierlich an den Verlauf der Linie anpassen.

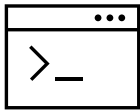
Nach der gemeinsamen Einführung bearbeiten die SchülerInnen die Aufgaben zur Linienverfolgung. Dabei sollen die grundlegenden Konzepte zunächst möglichst selbstständig angewendet werden.

Sollten mehrere SchülerInnen Schwierigkeiten bei der Umsetzung haben, empfiehlt es sich, das Zusammenspiel aus Sensoren, Variablen, Verzweigungen und Schleifen noch einmal gemeinsam zu besprechen.

Hinweis

Die SchülerInnen müssen an dieser Stelle nicht sofort verstehen, wie das gesamte Programm aufgebaut wird. Entscheidend ist zunächst, dass sie die Logik hinter der Linienverfolgung verstehen:

Der Sensor liefert eine Information → die Information wird gespeichert → das Programm trifft eine Entscheidung → der Roboter reagiert.



Einführung in die Robotik – Grundlagen der Programmierung

6. Unterrichtsstunde

In dieser Unterrichtsstunde setzen die SchülerInnen ihre Arbeit an der Linienverfolgung fort. Der Schwerpunkt liegt darauf, das entwickelte Programm zu testen, Fehler zu erkennen und die Linienverfolgung schrittweise zu verbessern.

Zu Beginn kann es hilfreich sein, das Grundprinzip aus der vorherigen Unterrichtsstunde kurz zu wiederholen:

Sensoren erfassen → Information in Variablen speichern → Bedingungen prüfen → Fahrtrichtung anpassen.

Anschließend testen die SchülerInnen ihre Programme auf einer einfachen schwarzen Linie. Dabei sollen sie beobachten, wie der Roboter auf unterschiedliche Linienverläufe reagiert und an welchen Stellen Schwierigkeiten auftreten.

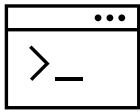
Zur Unterstützung können folgende Fragen gestellt werden:

- Erkennt der Roboter die schwarze Linie zuverlässig?
- Reagiert der Roboter richtig, wenn links oder rechts eine Linie erkannt wird?
- Welche Wenn-Dann-Bedingung wird in dieser Situation ausgeführt?
- Was passiert, wenn keiner oder mehrere Sensoren gleichzeitig die Linie erkennen?
- Wie kann das Programm angepasst werden, damit der Roboter zuverlässiger der Linie folgt?

Im weiteren Verlauf können die Parcours schrittweise schwieriger gestaltet werden. Hierfür eignen sich beispielsweise engere Kurven, Schlangenlinien oder andere Linienverläufe. So können die SchülerInnen ihr bestehendes Programm testen und an neue Situationen anpassen.

Hinweis

Fehler und unerwartetes Verhalten des Roboters sind ein wichtiger Teil des Lernprozesses. Ermutigen Sie die SchülerInnen dazu, zunächst zu beobachten, warum der Roboter falsch reagiert, und anschließend gezielt die entsprechende Bedingung im Programm zu überprüfen.



Einführung in die Robotik – Grundlagen der Programmierung

7. Unterrichtsstunde

Hinweis: Benötigte Erweiterung

Ab dieser Unterrichtsstunde wird zusätzliche Hardware benötigt.
Der Ultraschallsensor ist nicht Bestandteil des MiBo-Grundkits und muss separat ergänzt werden. Die folgenden Unterrichtseinheiten sind daher optional.

In dieser Unterrichtsstunde lernen die SchülerInnen den Ultraschallsensor kennen und erfahren, wie der Roboter mithilfe des Sensors Abstände zu Hindernissen erkennen und darauf reagieren kann.

Zu Beginn empfiehlt es sich, die Funktionsweise des Ultraschallsensors kurz zu erläutern. Der Sensor sendet Ultraschallwellen aus, die von einem Hindernis zurückgeworfen werden. Anhand der Zeit bis zur Rückkehr des Signals kann der Abstand zum Hindernis bestimmt werden.

Für die SchülerInnen kann das Grundprinzip vereinfacht dargestellt werden:

Abstand messen → Bedingung prüfen → Roboter reagieren lassen

Dabei kann an das bereits aus der Linienverfolgung bekannte „Wenn ... dann ...“-Prinzip angeknüpft werden. Der Roboter misst fortlaufend den Abstand vor sich und entscheidet anhand eines festgelegten Grenzwertes, wie er reagieren soll.

Beispielsweise:

Wenn der gemessene Abstand kleiner als 15 cm ist, dann befindet sich ein Hindernis vor dem Roboter.

Die SchülerInnen können anschließend überlegen, welche Reaktion in diesem Fall sinnvoll ist. Der Roboter könnte beispielsweise anhalten, zurückfahren oder seine Fahrtrichtung ändern.

Für das Verständnis der Programmierung kann die Abfolge gemeinsam mit den SchülerInnen betrachtet werden:

1. Ultraschallsensor auslesen

Der Roboter ermittelt den Abstand zum nächsten Hindernis.



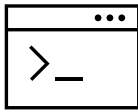
2. Abstand überprüfen
Mithilfe einer Bedingung wird geprüft, ob der gemessene Abstand einen festgelegten Grenzwert unterschreitet.
3. Entscheidung treffen
Ist ein Hindernis zu nah, wird eine entsprechende Aktion ausgeführt.
4. Messung wiederholen
Innerhalb einer Schleife wird der Abstand fortlaufend überprüft.

So erkennen die SchülerInnen, dass die bereits erlernten Konzepte Schleifen und Verzweigungen auch für andere Sensoren und Aufgaben eingesetzt werden können.

Anschließend bearbeiten die SchülerInnen die Aufgaben des Arbeitsblatts „Ultraschallsensor“ und entwickeln schrittweise eine einfache Hinderniserkennung. Das bestehende Arbeitsblatt sieht dabei zunächst das Ausweichen vor Hindernissen und anschließend die Verknüpfung mit dem Linienfolger vor.

Hinweis

Lassen Sie die SchülerInnen zunächst unterschiedliche Abstände messen und beobachten, welche Werte der Ultraschallsensor liefert. So entwickeln sie ein besseres Verständnis dafür, welcher Grenzwert für die Hinderniserkennung sinnvoll ist.



Einführung in die Robotik – Grundlagen der Programmierung

8. Unterrichtsstunde

In dieser Unterrichtsstunde setzen die SchülerInnen ihre Arbeit mit dem Ultraschallsensor fort. Der Schwerpunkt liegt darauf, die Hinderniserkennung zu testen und mit der bereits programmierten Linienverfolgung zu verbinden.

Zunächst sollten die SchülerInnen überprüfen, ob der Roboter Hindernisse zuverlässig erkennt und darauf wie vorgesehen reagiert. Dabei können unterschiedliche Abstände und Positionen der Hindernisse getestet werden.

Anschließend werden die beiden bisher erarbeiteten Programmteile miteinander kombiniert: Der Roboter soll weiterhin der Linie folgen, gleichzeitig jedoch regelmäßig prüfen, ob sich ein Hindernis vor ihm befindet. Wird ein Hindernis erkannt, soll der Roboter entsprechend reagieren und anschließend die Linienverfolgung fortsetzen.

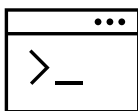
Dabei können folgende Fragen helfen:

- Erkennt der Roboter das Hindernis früh genug?
- Welche Wenn-Dann-Bedingung wird bei einem Hindernis ausgeführt?
- Was soll der Roboter tun, nachdem das Hindernis umfahren wurde?
- Wie findet der Roboter anschließend zurück zur Linie?
- Was muss im Programm zuerst überprüft werden: die Linie oder ein Hindernis?

Schnellere SchülerInnen können schwierigere Hindernispositionen testen und ihr Programm entsprechend optimieren.

Hinweis

Fehler können gezielt zur Weiterentwicklung des Programms genutzt werden. Ermutigen Sie die SchülerInnen, zunächst zu beobachten, an welcher Stelle der Programmablauf nicht wie vorgesehen funktioniert, bevor Änderungen am Programm vorgenommen werden.



Einführung in die Robotik – Grundlagen der Programmierung

9. Unterrichtsstunde

Hinweis: Benötigte Erweiterung

Für diese Unterrichtsstunde wird ein zusätzlicher Farbsensor benötigt. Dieser ist nicht Bestandteil des MiBo-Grundkits. Die Unterrichtseinheit kann daher optional durchgeführt werden, sofern die entsprechende Hardware vorhanden ist.

In dieser Unterrichtsstunde lernen die SchülerInnen den Farbsensor als weitere Möglichkeit kennen, Informationen aus der Umgebung des Roboters zu erfassen und im Programm zu verarbeiten.

Zu Beginn empfiehlt es sich, die grundlegende Funktionsweise des Farbsensors gemeinsam zu betrachten. Der Sensor erfasst den Untergrund und liefert entsprechende Messwerte, anhand derer unterschiedliche Farben voneinander unterschieden werden können. Wichtig ist hierbei, dass die SchülerInnen zunächst beobachten, wie sich die ausgegebenen Werte bei verschiedenen Farben verändern.

Lassen Sie die SchülerInnen den Sensor zunächst auf verschiedenen Untergründen testen. Besonders sinnvoll ist der Vergleich zwischen weiß, schwarz und den für die Aufgaben benötigten Farben. Die SchülerInnen sollen beobachten, welche Unterschiede sich in den Messwerten erkennen lassen und wie diese für die Programmierung genutzt werden können.

Anschließend bearbeiten die SchülerInnen das Arbeitsblatt „Farberkennung“. Dabei übertragen sie die bereits bekannten Konzepte der Variablen und Wenn-Dann-Bedingungen auf den Farbsensor. Erkennt der Roboter beispielsweise eine bestimmte Farbe, kann eine zuvor festgelegte Aktion ausgeführt werden.

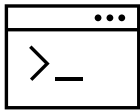
Zur Unterstützung können folgende Fragen hilfreich sein:

- Welche Werte liefert der Sensor bei unterschiedlichen Farben?
- Wie kann der Roboter die Farben voneinander unterscheiden?
- Welche Bedingung muss überprüft werden?
- Wie soll der Roboter auf die erkannte Farbe reagieren?



Hinweis

Es empfiehlt sich, die verschiedenen Farben zunächst einzeln zu testen, bevor mehrere Farberkennungen in einem Programm miteinander kombiniert werden. So können die SchülerInnen besser nachvollziehen, welche Messwerte zu welcher Reaktion führen.



Einführung in die Robotik – Grundlagen der Programmierung

10. Unterrichtsstunde

In der 10. Unterrichtsstunde schließen die SchülerInnen die noch offenen Aufgaben zur Farberkennung ab und testen ihre Programme in der Praxis. Dabei sollen sie überprüfen, ob der Roboter verschiedene Farben zuverlässig erkennt und die jeweils programmierte Reaktion korrekt ausführt.

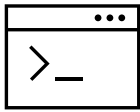
Treten dabei Fehler auf, sollen die SchülerInnen möglichst selbstständig überprüfen, an welcher Stelle im Programm die Ursache liegt. Hierfür können insbesondere die Sensorwerte sowie die verwendeten Variablen und Wenn-Dann-Bedingungen betrachtet werden.

Für die letzten 10 bis 15 Minuten der Unterrichtsstunde empfiehlt es sich, die bisherige Arbeit gemeinsam zu reflektieren. Hierfür kann das Arbeitsblatt „Reflexion“ genutzt werden. Die SchülerInnen setzen sich damit auseinander, welche Aufgaben sie bereits erfolgreich lösen können, wo noch Schwierigkeiten bestehen und welche nächsten Schritte für die Weiterentwicklung ihres Roboters sinnvoll wären.

Das Arbeitsblatt kann anschließend eingesammelt werden, um einen Überblick über den Lernfortschritt und die Selbsteinschätzung der SchülerInnen zu erhalten. Alternativ kann es bei den SchülerInnen verbleiben und als Orientierung für weitere Robotikprojekte genutzt werden.

Hinweis

Die Reflexion sollte sich nicht nur auf die Fähigkeiten des Roboters beziehen. Ermutigen Sie die SchülerInnen auch dazu, ihren eigenen Lernfortschritt zu betrachten und festzuhalten, welche Programmierkonzepte sie inzwischen selbstständig anwenden können.



Einführung in die Robotik – Grundlagen der Programmierung

Reflektion für LehrerInnen

Hinweis

Diese Checkliste unterstützt Sie dabei, den Lernfortschritt der SchülerInnen regelmäßig zu reflektieren und mögliche Schwierigkeiten frühzeitig zu erkennen. Es empfiehlt sich, die entsprechenden Punkte nach jeder oder jeder zweiten Unterrichtsstunde zu überprüfen.

1.- 2. Unterrichtsstunde

Haben alle SchülerInnen das Ziel der Unterrichtseinheit verstanden?	☐
Finden sich die SchülerInnen in der verwendeten Programmierumgebung zurecht?	☐
Sind alle SchülerInnen bzw. Teams mit dem Aufbau des MiBo fertig?	☐



3. - 4. Unterrichtsstunde

Haben alle SchülerInnen mit den vorgesehenen Programmieraufgaben begonnen?	☐
Können die SchülerInnen den MiBo gezielt fahren und lenken?	☐
Haben die SchülerInnen das Prinzip einer Schleife verstanden und können es erklären?	☐
Können die SchülerInnen geeignete Programmierblöcke selbstständig auswählen und miteinander verbinden?	☐
Können die SchülerInnen einfache Fehler in ihrem Programm erkennen und beheben?	☐

5. – 6. Unterrichtsstunde

Haben die SchülerInnen verstanden, wie die Liniensensoren des MiBo funktionieren?	☐
Verstehen die SchülerInnen, welche Informationen die Sensoren liefern und wie diese im Programm genutzt werden?	☐
Können die SchülerInnen Variablen anlegen und für die Linienverfolgung verwenden?	☐
Haben die SchülerInnen das Wenn-Dann-Prinzip bzw. Verzweigungen verstanden und können es erklären?	☐
Können die SchülerInnen Sensoren, Variablen, Verzweigungen und Schleifen zu einem Linienfolger verbinden?	☐
Folgt der MiBo einer einfachen schwarzen Linie zuverlässig?	☐
Grundkurs abgeschlossen: Sind die SchülerInnen in der Lage, ihren MiBo selbstständig zu programmieren und einen einfachen Linienfolger umzusetzen?	☐



OPTIONALE ERWEITERUNGEN

7. – 8. Unterrichtsstunde

Verstehen die SchülerInnen die grundlegende Funktionsweise des Ultraschallsensors?	☐
Können die SchülerInnen Abstandswerte im Programm auslesen und verwenden?	☐
Können sie mithilfe einer Wenn-Dann-Bedingung auf ein Hindernis reagieren?	☐
Erkennt und umfährt der Roboter Hindernisse zuverlässig?	☐
Können die SchülerInnen Hinderniserkennung und Linienverfolgung miteinander verbinden?	☐

9. – 10. Unterrichtsstunde

Verstehen die SchülerInnen die grundlegende Funktionsweise des Farbsensors?	☐
Können die SchülerInnen unterschiedliche Sensorwerte erkennen und auswerten?	☐
Können sie Farben mithilfe von Bedingungen unterscheiden?	☐
Führt der Roboter abhängig von der erkannten Farbe die gewünschte Aktion aus?	☐
Können die SchülerInnen bereits bekannte Programmierkonzepte selbstständig auf neue Sensoren übertragen?	☐



Optional bei eingesamelter Reflektion der SchülerInnen

Stimmt der Eindruck der SchülerInnen mit der Realität überein? Kann der Roboter alle Aufgaben erledigen, die er laut der Reflektion erledigen kann?

☐

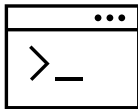
Wurden die nächsten Schritte korrekt identifiziert? Sind diese realistisch?

☐



Lösungen

Zu den Arbeitsblättern „Grundelemente“,
„Linienverfolger“ und „Wandverfolger“



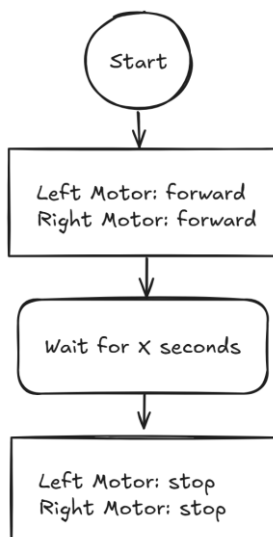
Einführung in die Robotik – Grundlagen der Programmierung

Lösung – Arbeitsblatt Grundelemente

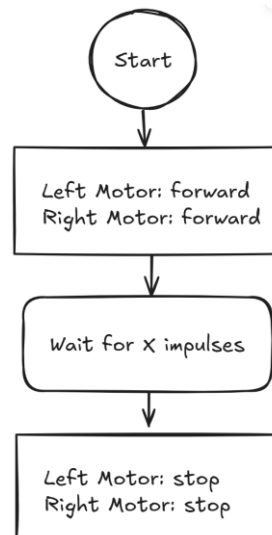
Aufgabe 1: Geradeaus fahren

Lösungsvorschläge

Flussdiagramm – Vorschlag 1



Flussdiagramm – Vorschlag 2

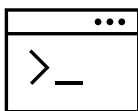


Pseudocode

LeftMotor forward
RightMotor forward

Wait for X seconds/impulses

LeftMotor stop
Right Motor stop



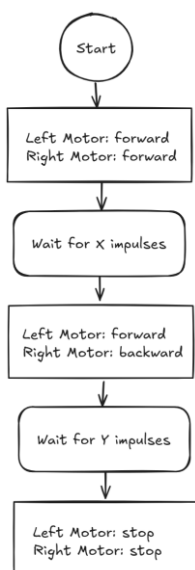
Einführung in die Robotik – Grundlagen der Programmierung

Lösung – Arbeitsblatt Grundelemente

Aufgabe 2: Abbiegen

Lösungsvorschläge

Flussdiagramm



Pseudocode

```

LeftMotor forward
RightMotor forward

Wait for X seconds/impulses

LeftMotor forward
RightMotor backward

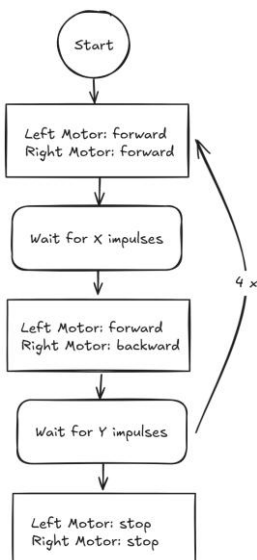
Wait for Y seconds/impulses

LeftMotor stop
Right Motor stop
  
```

Aufgabe 3: Viereck fahren

Lösungsvorschläge

Flussdiagramm



Pseudocode

```

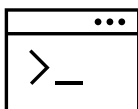
for 4x
  LeftMotor forward
  RightMotor forward

  Wait for X seconds/impulses

  LeftMotor forward
  RightMotor backward

  Wait for Y seconds/impulses

LeftMotor stop
Right Motor stop
  
```



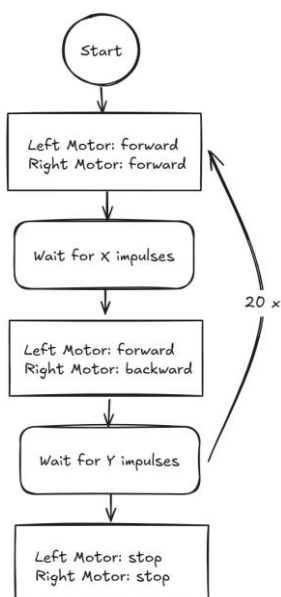
Einführung in die Robotik – Grundlagen der Programmierung

Lösung – Arbeitsblatt Grundelemente

Aufgabe 4: Schleifen

Lösungsvorschläge

Flussdiagramm



Pseudocode

```

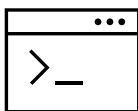
for 20x
  LeftMotor forward
  RightMotor forward

  Wait for X seconds/impulses

  LeftMotor forward
  RightMotor backward

  Wait for Y seconds/impulses

LeftMotor stop
Right Motor stop
  
```



Einführung in die Robotik – Grundlagen der Programmierung

Lösung – Arbeitsblatt Linienverfolger

Aufgabe 1: Linie verfolgen

Lösungsvorschläge

Vorschlag 1

```
# Two Color Sensors, put together
while True:
    if LineSensorLeft is White:
        if LineSensorRight is White:
            Drive Forward
        else:
            Drive Right curve
    else:
        if LineSensorRight is White:
            Drive Left curve
        else:
            Drive forward
```

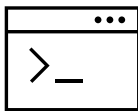
Vorschlag 2

```
# One Color Sensor
while True:
    if LineSensor is White:
        Drive Right curve
    else:
        Drive Left curve
```

Aufgabe 2: Abbiegen mit Verzweigungen

Lösungsvorschläge

```
# Two Color Sensors, put apart, turn right
while True:
    if LineSensorLeft is White:
        if LineSensorRight is White:
            Drive Forward
        else:
            Drive Right curve
    else:
        if LineSensorRight is White:
            Drive Left curve
        else:
            Drive X cm forward
            Turn 90°
```



Einführung in die Robotik – Grundlagen der Programmierung

Lösung – Arbeitsblatt Wandverfolger

Aufgabe 1: Wand verfolgen mit Schräge

Lösungsvorschläge

```
while True:
    if UltrasonicSensor < Threshold:
        Drive Right curve
    else:
        Drive Left curve
```

Aufgabe 2: Wand verfolgen mit rechtem Winkel

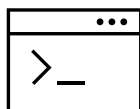
Lösungsvorschläge

```
while True:
    if UltrasonicSensorFront < Threshold:
        Turn 90° to the right
    else if UltrasonicSensorSide < Threshold:
        Drive Right curve
    else:
        Drive Left curve
```



Arbeitsblätter

Zur Bearbeitung durch die SchülerInnen



Einführung in die Robotik – Grundlagen der Programmierung

Einstieg in die Programmierung

Zusammenfassung

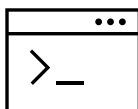
Das Ziel dieser Aufgaben ist, die Grundlagen der Programmierung zu lernen. Am Ende dieser Aufgaben, wird euer Roboter in der Lage sein, im Quadrat zu fahren und ihr werdet wissen, wie man Schleifen benutzt, um euer Programm zu verkürzen.

Sobald ihr eine der Aufgaben abgeschlossen habt, könnt ihr euch die jeweilige Aufgabe bei eurer Lehrerin oder eurem Lehrer abzeichnen lassen.

Aufgabe 1: Geradeaus fahren	Lasst euren Roboter vorwärtsfahren.	☐
Aufgabe 2: Abbiegen	Programmiert den Roboter so, dass er nach ca. 1 Meter rechts abbiegt.	☐
Aufgabe 3: Viereck fahren	Jetzt soll der Roboter ein 1x1 Meter großes Quadrat abfahren.	☐
Aufgabe 4: Schleifen benutzen	Ändert nun euer Programm so ab, dass euer Roboter das Quadrat 5-mal abfährt. Tipp: Es gibt eine einfache Möglichkeit, wie ihr die Aufgabe lösen könnt, ohne die Blöcke zu kopieren. Schaut euch dazu die Blöcke an, die euch noch so zur Verfügung stehen.	☐



Viel Erfolg!



Einführung in die Robotik – Grundlagen der Programmierung

Linienverfolgung

Wiederholung - Schleifen

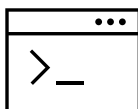
In Mit einer Schleife können Programmteile mehrfach ausgeführt werden. Für die Linienverfolgung benötigen wir eine Endlosschleife, damit der Roboter seine Sensoren während der Fahrt immer wieder überprüft.

Zusammenfassung

Das Ziel der heutigen Aufgaben ist es, die Verwendung von Verzweigungen zu lernen und zu verstehen, wie der Roboter mithilfe von Sensoren einer Linie folgen kann.

Sobald ihr eine der Aufgaben abgeschlossen habt, könnt ihr euch die jeweilige Aufgabe bei eurer Lehrerin oder eurem Lehrer abzeichnen lassen.

Aufgabe 1: Sensoren testen	Testet die Liniensensoren auf schwarzem und weißem Untergrund. Beobachtet, wie die Sensoren reagieren.	☐
Aufgabe 2: Variablen anlegen	Erstellt zwei Variablen, in denen gespeichert wird, ob links bzw. rechts eine schwarze Linie erkannt wird.	☐
Aufgabe 3: Wenn... dann...	Überlegt, wie der Roboter auf die Informationen der Sensoren reagieren muss. Nutzt dafür eine Wenn-Dann-Verzweigung.	☐
Aufgabe 4: Linie verfolgen	Verknüpft Sensoren, Variablen und Verzweigungen so, dass euer Roboter selbstständig der schwarzen Linie folgt.	☐



Einführung in die Robotik – Grundlagen der Programmierung

Ultraschallsensor

Wiederholung - Verzweigungen

Mit Verzweigungen kann der Roboter auf unterschiedliche Situationen reagieren. Dafür wird eine Bedingung überprüft und abhängig vom Ergebnis eine bestimmte Aktion ausgeführt.

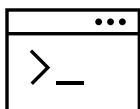
Beispiel: Wenn der Roboter ein Hindernis erkennt, dann soll er seine Fahrtrichtung

Zusammenfassung

Ziel der heutigen Aufgaben ist es, den Ultraschallsensor kennenzulernen und damit Hindernisse zu erkennen. Dabei nutzt ihr bereits bekannte Wenn-Dann-Bedingungen, um den Roboter auf unterschiedliche Abstände reagieren zu lassen.

Sobald ihr eine Aufgabe abgeschlossen habt, könnt ihr sie bei eurer Lehrerin oder eurem Lehrer abzeichnen lassen.

Aufgabe 1: Abstand messen	Testet den Ultraschallsensor mit verschiedenen Hindernissen und beobachtet die gemessenen Abstände.	☐
Aufgabe 2: Verknüpfung der Programmteile	Legt einen Abstand fest, ab dem der Roboter ein Hindernis erkennen soll. Programmiert eine passende Wenn-Dann-Bedingung.	☐
Aufgabe 3: Hinderniss umfahren	Programmiert euren Roboter so, dass er auf ein erkanntes Hindernis reagiert und diesem ausweicht.	☐
Aufgabe 4: Mit Linienverfolgung verbinden	Verbindet die Hinderniserkennung mit eurem Linienfolger. Nach dem Umfahren soll der Roboter die Linie wiederfinden und ihr weiter folgen.	☐



Einführung in die Robotik – Grundlagen der Programmierung

Farberkennung

Zusammenfassung

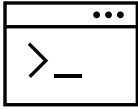
Das Ziel der heutigen Aufgaben ist es, verschiedene Farben mithilfe eines Farbsensors zu identifizieren.

Sobald ihr mit einer der Aufgaben fertig seid, könnt ihr euch die jeweilige Aufgabe bei eurer Lehrerin oder eurem Lehrer abzeichnen lassen.

Aufgabe 1: Grüne Markierungen erkennen	Nutzt den Farbsensor, um die grünen Vierecke auf den Kacheln zu erkennen. Der Roboter soll nach der Erkennung in die entsprechende Richtung abbiegen.	☐
Aufgabe 2: Rote Kreuze erkennen	Benutzt den Farbsensor, um die roten Kreuze auf den Kacheln erkennen zu können. Der Roboter soll nach der Erkennung der Kreuze fünf Sekunden stehen bleiben.	☐
Aufgabe 3: Erkennung der schwarzen Kacheln	Programmiert euren Roboter so, dass er die schwarzen Kacheln erkennt und nach der Erkennung rückwärts von der Platte runterfährt und sich um 180° dreht, bevor der Roboter weiterfährt. Tipp: <i>Der Kontrast der schwarzen Kacheln ist sehr groß, da die anderen Kacheln weiß sind. Überlegt euch wie ihr euch das zu Nutze machen könnt.</i>	☐
Aufgabe 4: Erkennung der blauen Kacheln	Nun soll euer Roboter die blauen Kacheln erkennen. Nach der Erkennung soll euer Roboter in der Mitte der blauen Kachel für 5 Sekunden stehen bleiben.	☐



Viel Erfolg!



Einführung in die Robotik – Grundlagen der Programmierung

Reflektion

Zusammenfassung

Ziel dieser Reflexion ist es, auf die vergangenen Unterrichtsstunden zurückzublicken. Überlegt, was ihr bereits gelernt habt, was euer Roboter schon kann und wo noch Herausforderungen bestehen.

1. Was hat dir an den Aufgaben am meisten Spaß gemacht? Warum?

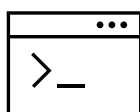
2. Was hast du in den letzten Unterrichtsstunden gelernt?

3. Was kann dein / euer Roboter bereits gut?

4. Was funktioniert bei deinem / eurem Roboter noch nicht so gut?

5. Welche Herausforderung konntest du erfolgreich lösen? Wie bist du dabei vorgegangen?

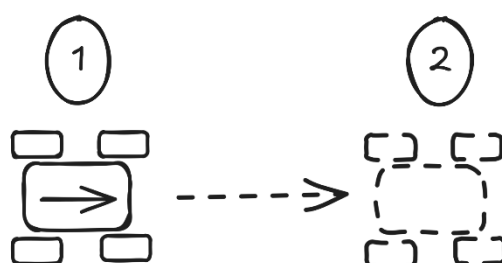
6. Was möchtest du als Nächstes verbessern oder ausprobieren?



Einführung in die Robotik – Grundlagen der Programmierung

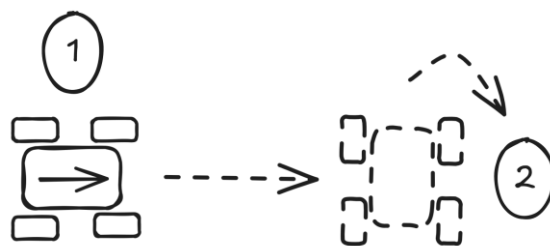
Grundelemente

Aufgabe 1: Geradeaus fahren



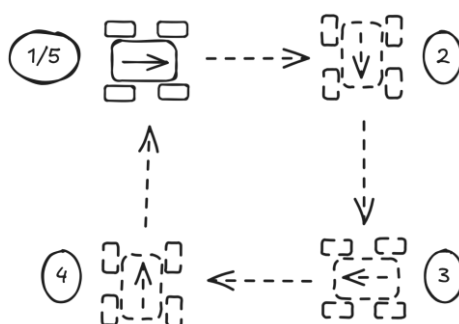
Lasst euren Roboter geradeaus fahren. Visualisiert euren Code als Flussdiagramm oder Pseudocode:

Aufgabe 2: Abbiegen



Programmiert den Roboter so, dass er nach ca. 1 Meter rechts abbiegt. Visualisiert euren Code als Flussdiagramm oder Pseudocode.

Aufgabe 3: Viereck fahren



Programmiert den Roboter nun so, dass der Roboter ein Quadrat abfährt. Visualisiert euren Code als Flussdiagramm oder Pseudocode.



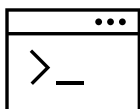
Aufgabe 4: Schleifen

Ändert das Programm nun so ab, dass euer Roboter das Quadrat 5-mal abfährt.
Visualisiert euren Code als Flussdiagramm oder Pseudocode.

Tipp: Es gibt eine einfache Möglichkeit, wie ihr die Aufgabe lösen könnt, ohne die Blöcke zu kopieren. Schaut euch dazu die Blöcke an, die euch außerdem zur Verfügung stehen.



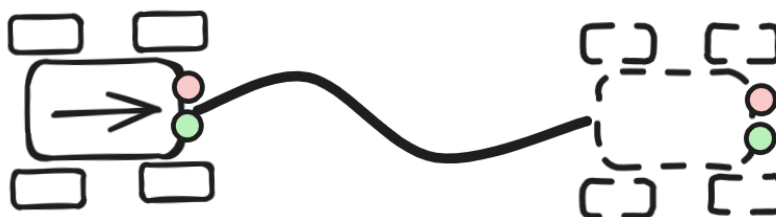
Viel Erfolg!



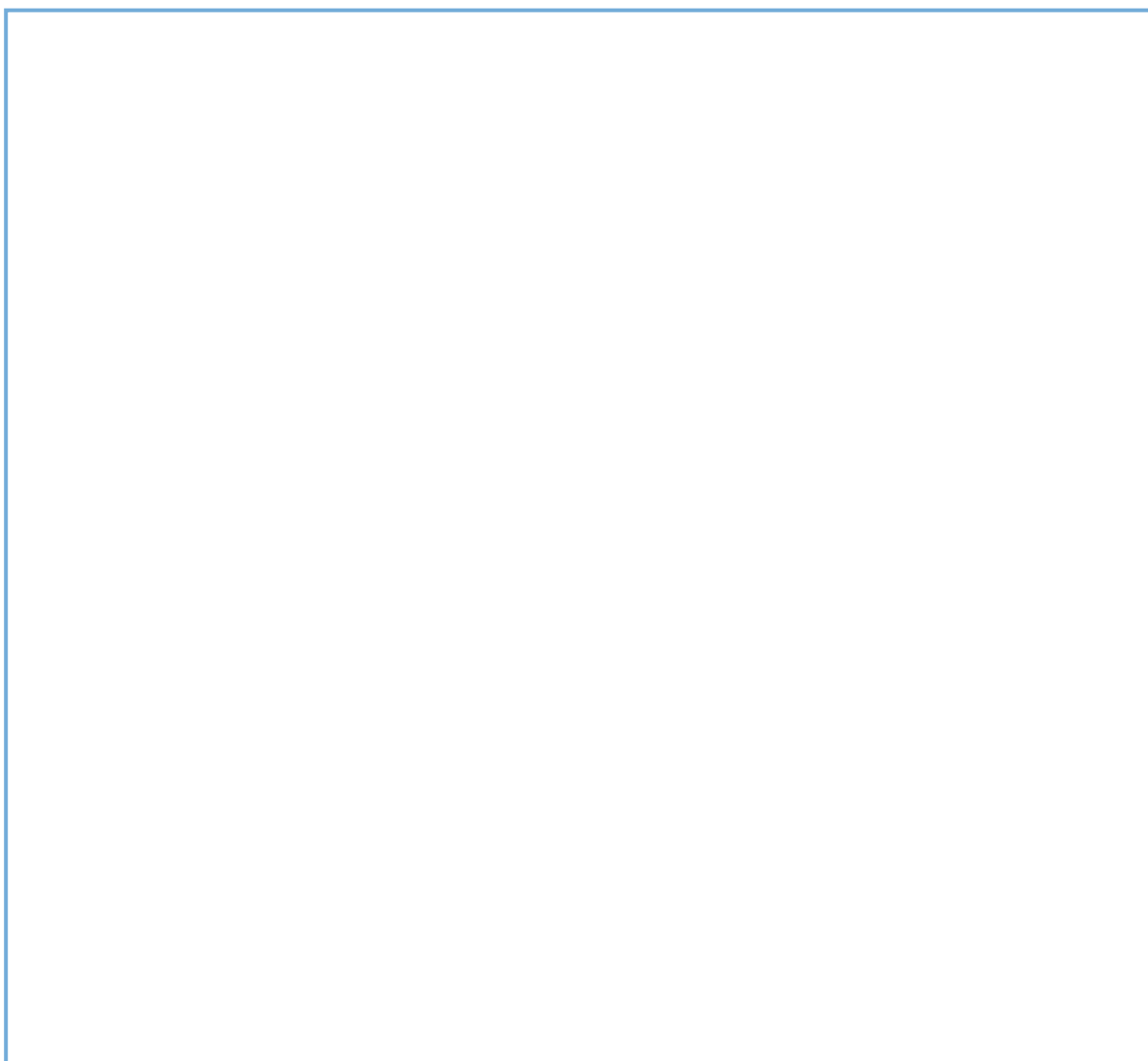
Einführung in die Robotik – Grundlagen der Programmierung

Linienverfolger

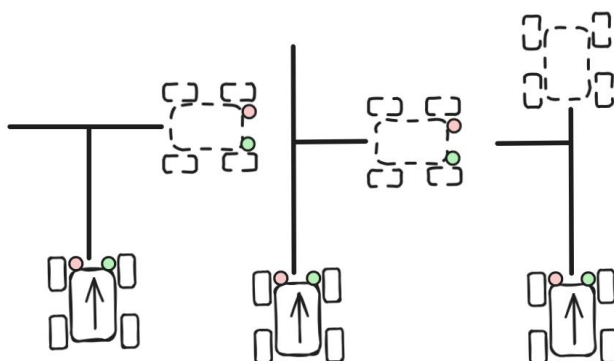
Aufgabe 1: Linie verfolgen



Programmiert euren Roboter so, dass er die Linie verfolgt. Nutzt hierfür den Sensor, welcher euch am Anfang der Unterrichtsstunde vorgestellt wurde.

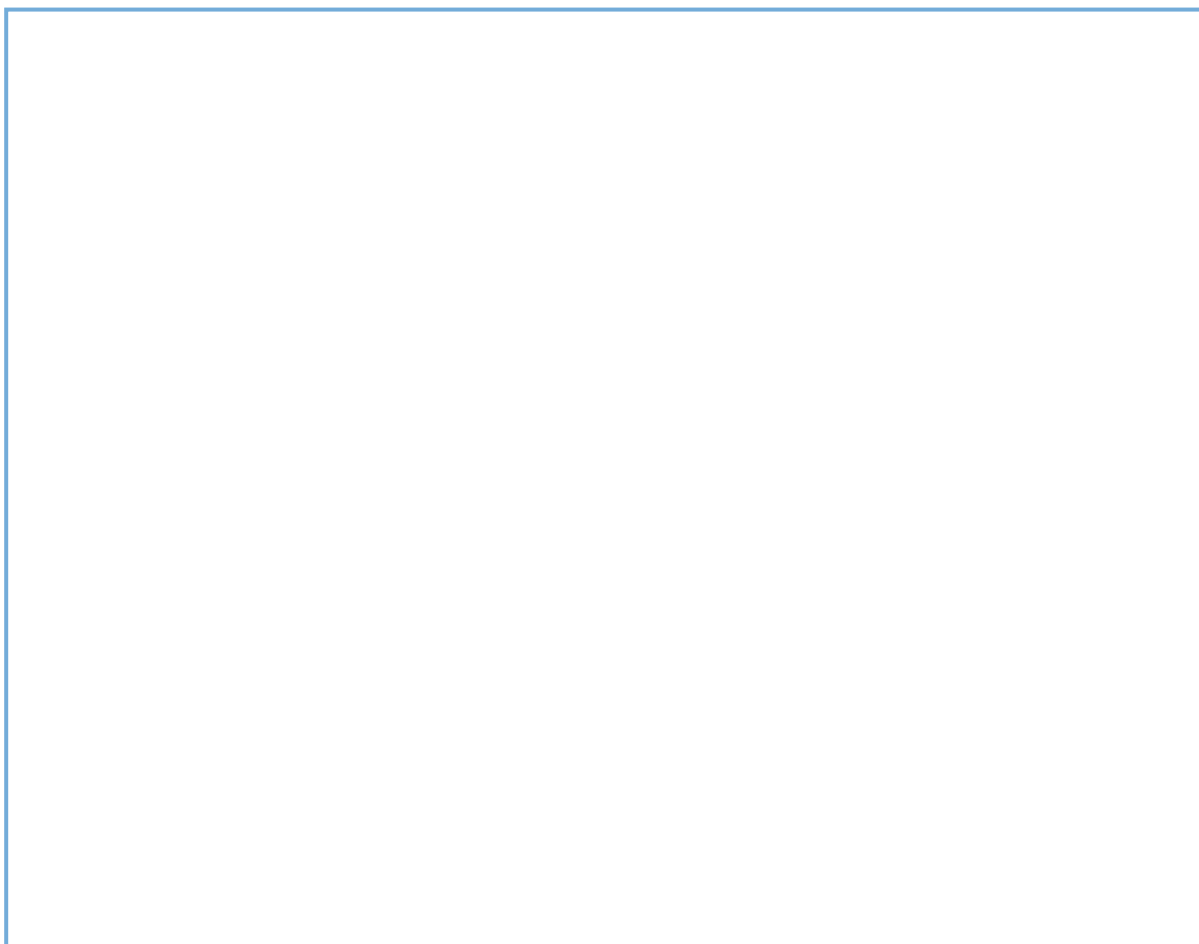


Aufgabe 2: Abbiegen mit Verzweigungen

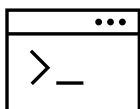


Erweitert euren Linienfolger um Verzweigungen.

Der Roboter soll anhand der Sensorwerte erkennen, ob die Linie nach links oder rechts verläuft und seine Fahrtrichtung entsprechend anpassen.



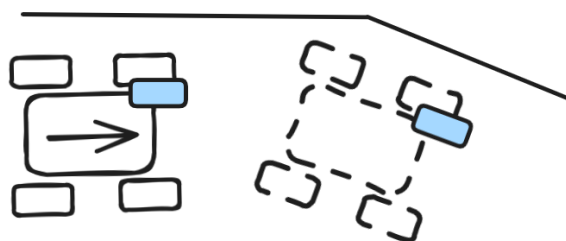
Viel Erfolg!



Einführung in die Robotik – Grundlagen der Programmierung

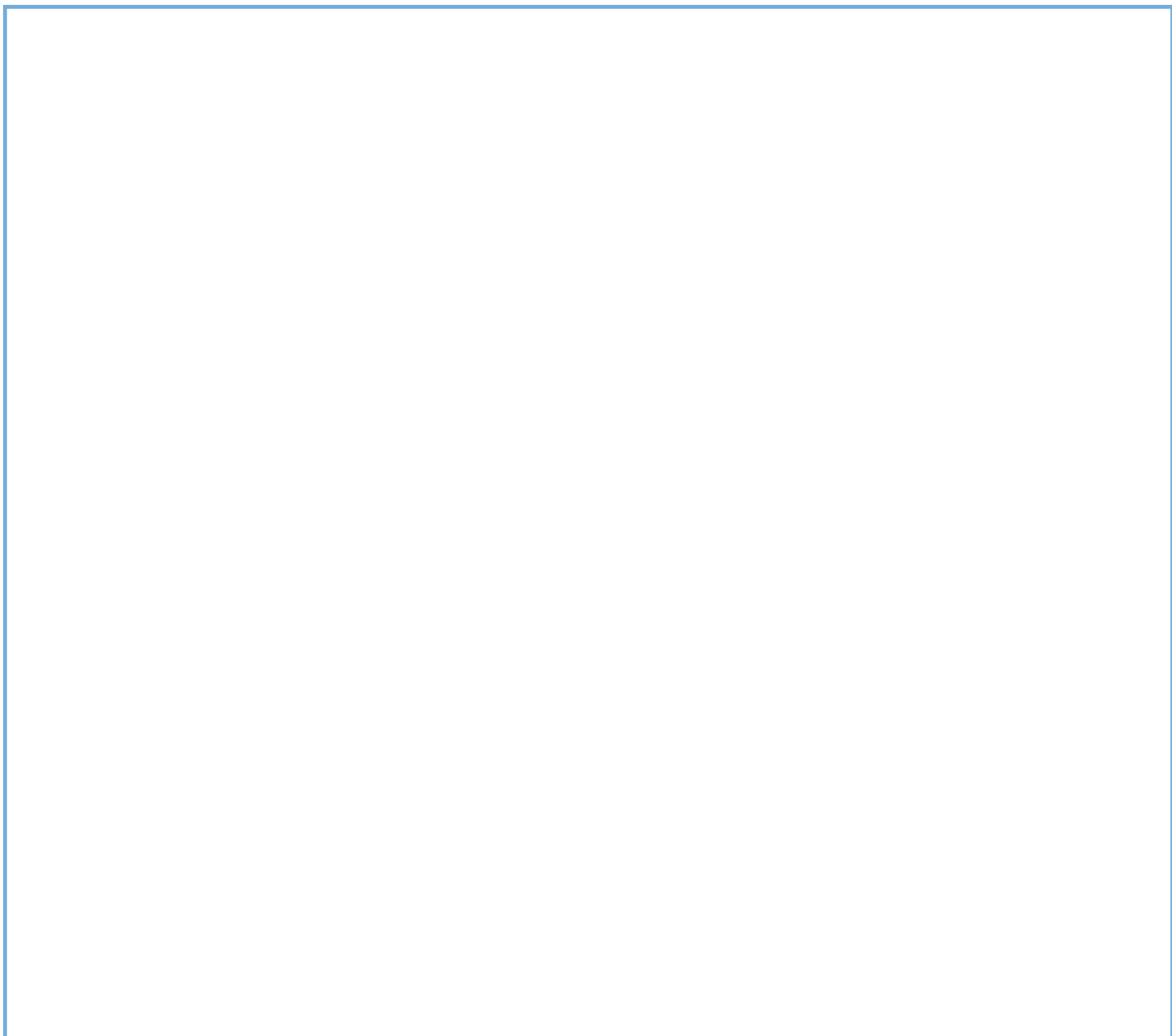
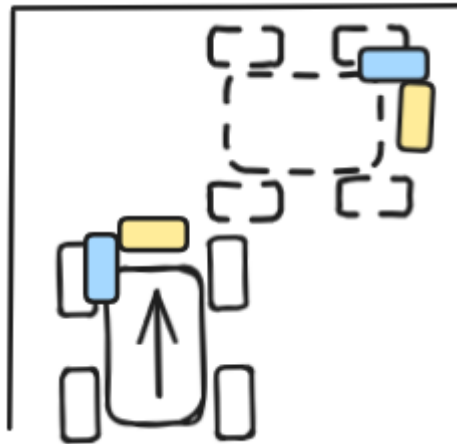
Wandverfolger

Aufgabe 1: Wand verfolgen mit schräger Wand



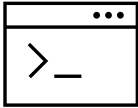
Programmiert euren Roboter so, dass er die Wand verfolgt. Benutzt dafür einen Ultraschallsensor und die Methoden, die ihr in den letzten Stunden gelernt habt.

Aufgabe 2: Wand verfolgen mit rechtem Winkel





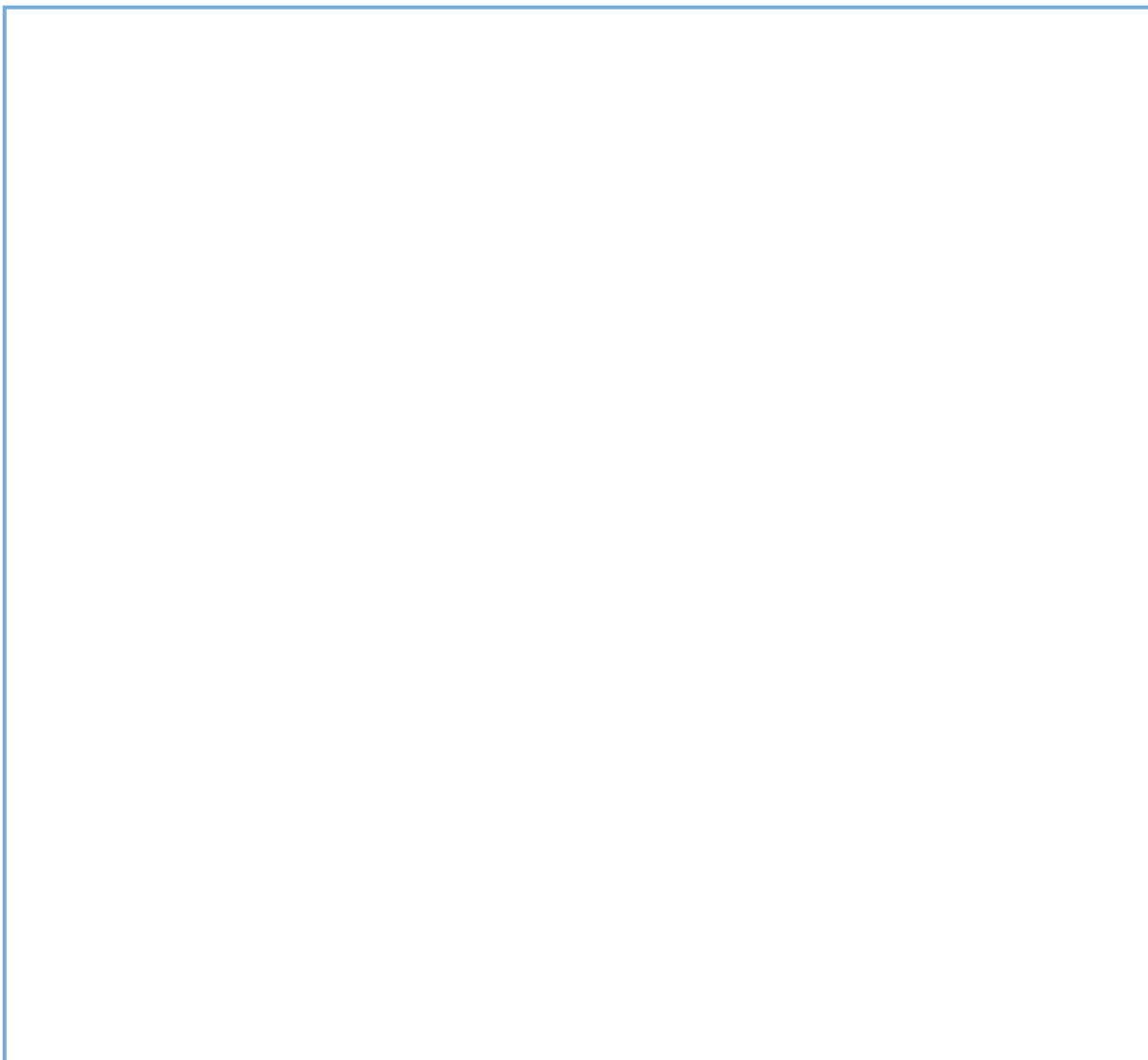
Rüstet euren Roboter nun so auf, dass auch rechtwinklige Wände verfolgt werden können.



Schlusswort

Die Programmierung von Robotern ist ein spannendes und zukunftsweisendes Thema, das nicht nur technische Fertigkeiten vermittelt, sondern auch kreatives Problemlösen, logisches Denken und Teamarbeit fördert. Mit diesem Leitfaden erhalten Sie als Lehrkräfte eine Grundlage, um Ihren Schülerinnen und Schülern den Einstieg in diese Welt zu erleichtern. Ziel ist es, Motivation und Faszination für die Robotik zu wecken.

Der Ansatz dieses Leitfadens ist es, theoretische Grundlagen mit praxisnahen Übungen zu verknüpfen und zu ermöglichen, flexibel auf die individuellen Gegebenheiten Ihrer Lerngruppe einzugehen. Dabei ist das Unterrichtsmaterial so gestaltet, dass Sie die





Möglichkeit haben, es an die vorhandenen Ressourcen und Vorkenntnisse Ihrer Schülerinnen und Schüler anzupassen.

Ein besonders wertvoller Aspekt des Lernprozesses ist die Anwendung des Erlernten in einem praxisnahen Kontext. Mit der Teilnahme am RoboCupJunior erhalten die Schülerinnen und Schüler die Möglichkeit, ihr Wissen in einem motivierenden Wettbewerb anzuwenden, weiterzuentwickeln und sich mit anderen SchülerInnen zu messen. Dies ist nicht nur eine hervorragende Lerngelegenheit, sondern auch eine Chance, Begeisterung für MINT-Fächer zu wecken und ein langfristiges Interesse an

Anhang

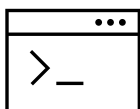
Arenenbeispiele mit dem Material aus dem Set
„School Kit One“

Technik und Informatik zu fördern.

Die Welt der Robotik bietet vielseitige Möglichkeiten, um technisches Verständnis, kreatives Denken und Problemlösungskompetenzen zu fördern. Wir hoffen, dass dieser Leitfaden eine hilfreiche Unterstützung für Ihren Unterricht ist und Ihnen praxisnahe Impulse für die Vermittlung von Robotik und Programmierung bietet. Gleichzeitig sind wir uns bewusst, dass jede Lerngruppe eigene Herausforderungen und Bedürfnisse mitbringt. Daher freuen wir uns über konstruktive Rückmeldungen, um das Unterrichtsmaterial kontinuierlich weiterzuentwickeln und bestmöglich an die Praxis anzupassen.

Ihnen und Ihren Schülerinnen und Schülern wünschen wir viel Erfolg bei der Umsetzung und vor allem viel Freude beim Entdecken, Experimentieren und Programmieren!

Das rosslight robotics – Team



Line Entry Arenen

Schwierigkeitseinstufung der Arenen

	Spur	Lücke	Rampe	Kreuzung	Hindernis	Wippe	Evacuation Zone
1	☆☆☆	☆☆☆	-	-	☆☆☆	-	☆☆☆
2	☆☆☆	☆☆☆	☆☆☆	-	-	-	☆☆☆
3	☆☆☆	☆☆☆	☆☆☆	-	-	-	☆☆☆
4	☆☆☆	☆☆☆	-	☆☆☆	-	-	☆☆☆
5	☆☆☆	☆☆☆	-	☆☆☆	-	-	☆☆☆
6	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	-
7	☆☆☆	☆☆☆	-	☆☆☆	☆☆☆	-	☆☆☆
8	☆☆☆	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
9	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	-
10	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	-

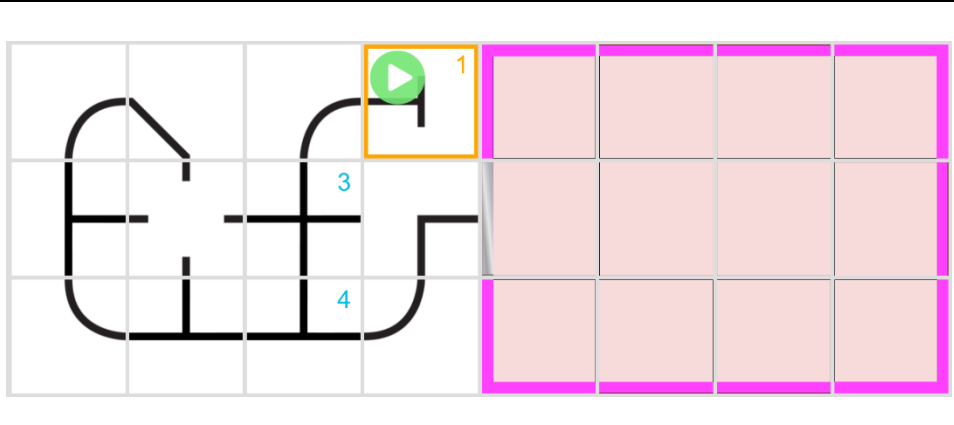
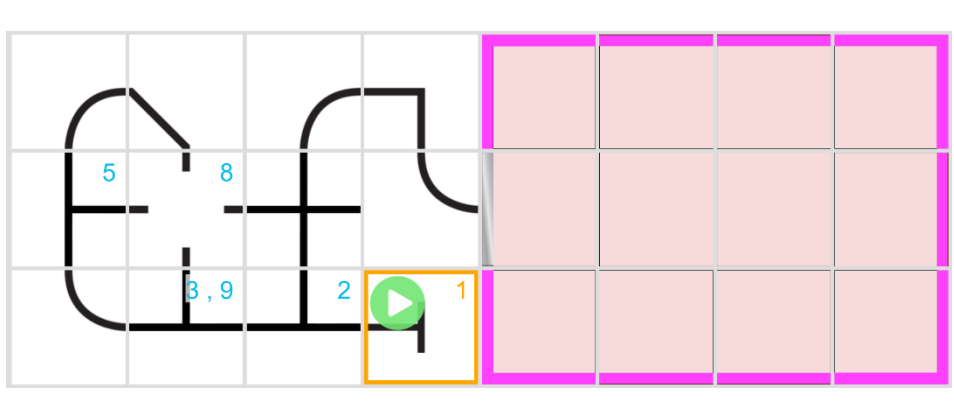
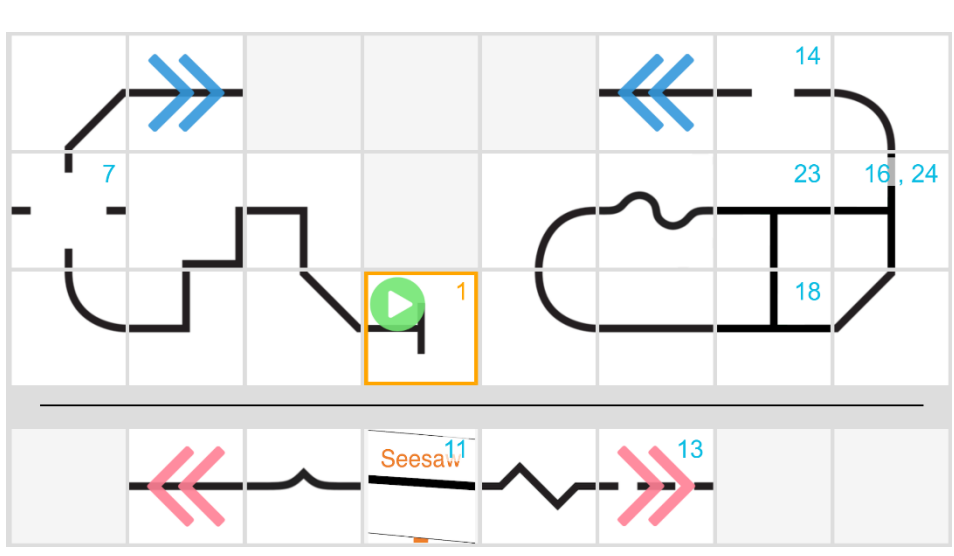
Hinweis:

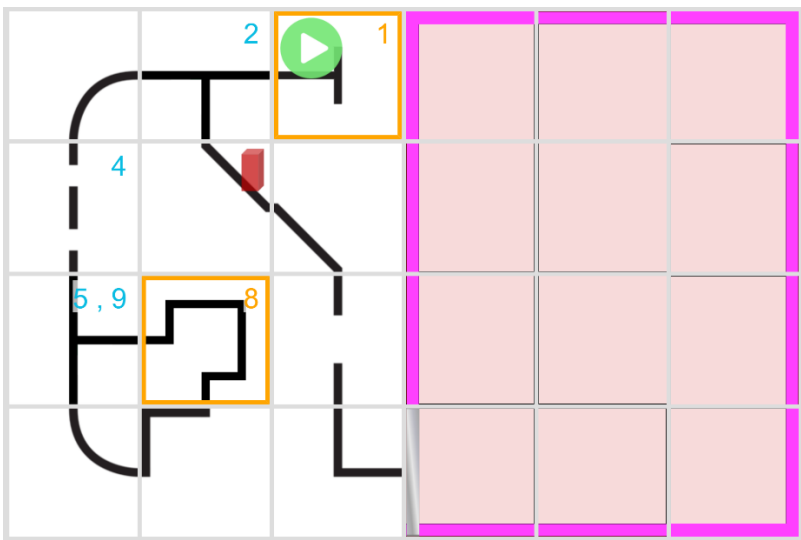
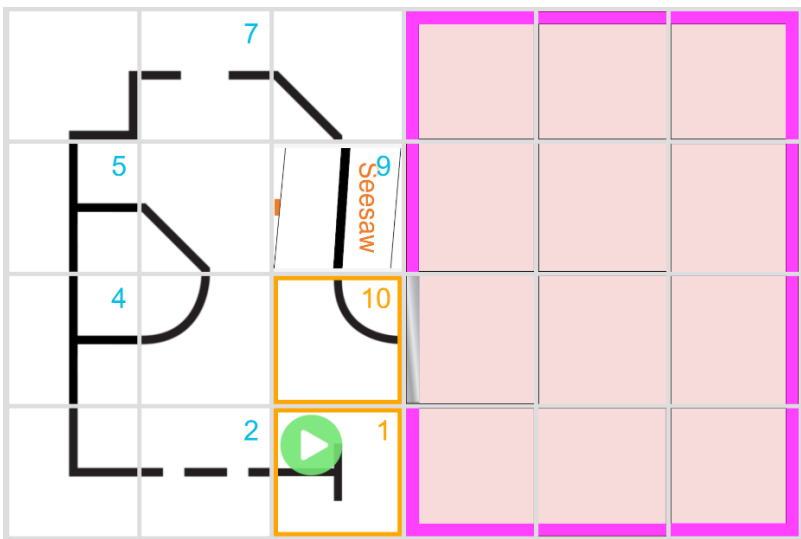
Die Schwierigkeit wurde vor allem nach den folgenden Kriterien ermittelt:

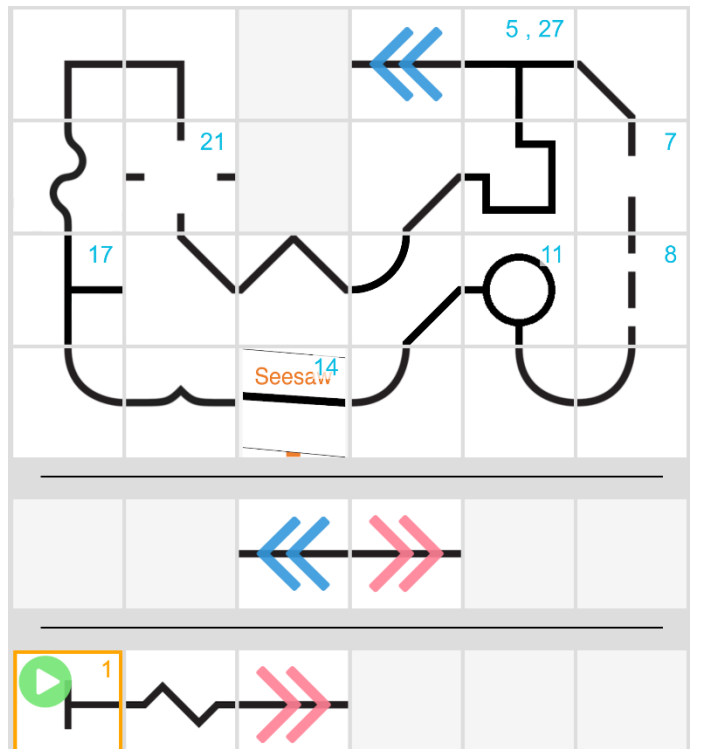
- Aufkommen auf Kachel (e.g. Schräge vor Lücke)
- Kombination mehrerer Elemente
- Seltenheit der Kombinationen

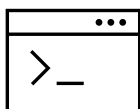
In manchen Kategorien werden keine 3/3 Sterne erreicht. In den Plänen sind keine „Speed Bumper“ eingezeichnet. Diese können frei auf den Platten platziert werden und machen einzelne Kacheln deutlich schwieriger (bsp. „Speed Bumper“ auf einer Rampe / Kreuzung / in der Evacuation Zone).

Die Bilder der Arenen zeigen kleine Zahlen auf. Diese können ignoriert werden.

Arena 4: Schwierigkeit: ★★★☆☆ Fahrtrichtung: Rechts	
Arena 5: Schwierigkeit: ★★★☆☆ Fahrtrichtung: Links	
Arena 6: Schwierigkeit: ★★★☆☆ Fahrtrichtung: Links/Rechts	

Arena 7: Schwierigkeit: ★★★★☆ Fahrtrichtung: Rechts	
Arena 8: Schwierigkeit: ★★★★☆ Fahrtrichtung: Links/Rechts	

Arena 9: Schwierigkeit: ★★★★ Fahrtrichtung: Links/Rechts	
Arena 10: Schwierigkeit: ★★★★ Fahrtrichtung: Links/Rechts	



Line Arenen

Schwierigkeitseinstufung der Arenen

	Spur	Lücke	Rampe	Kreuzung	Hindernis	Wippe	Evacuation Zone
0	☆☆☆	☆☆☆	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-
1	☆☆☆	☆☆☆	-	☆☆☆	☆☆☆	☆☆☆	-
2	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	-
3	☆☆☆	☆☆☆	-	☆☆☆	☆☆☆	-	☆☆☆
4	☆☆☆	-	-	☆☆☆	-	☆☆☆	☆☆☆
5	☆☆☆	☆☆☆	-	☆☆☆	-	☆☆☆	-
6	☆☆☆	☆☆☆	-	☆☆☆	-	☆☆☆	-
7	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	-
8	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	☆☆☆
9	☆☆☆	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	-
10	☆☆☆	☆☆☆	☆☆☆	☆☆☆	-	-	-

Hinweis:

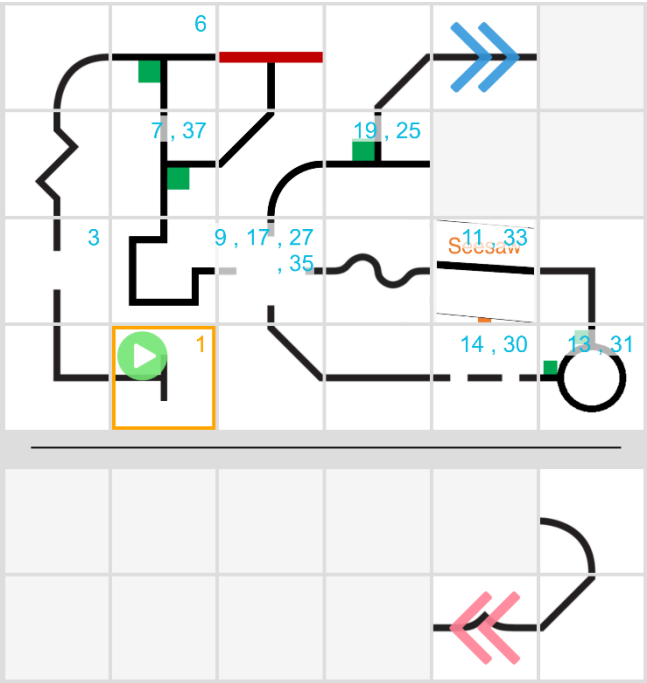
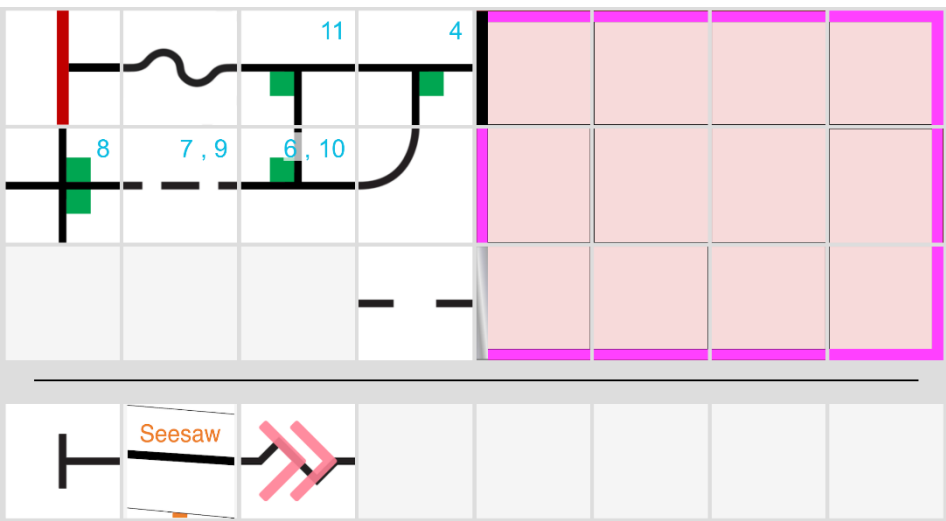
Die Schwierigkeit wurde vor allem nach den folgenden Kriterien ermittelt:

- Aufkommen auf Kachel (e.g. Schräge vor Lücke)
- Kombination mehrerer Elemente
- Seltenheit der Kombinationen

In manchen Kategorien werden keine 3/3 Sterne erreicht. In den Plänen sind keine „Speed Bumper“ eingezeichnet. Diese können frei auf den Platten platziert werden und

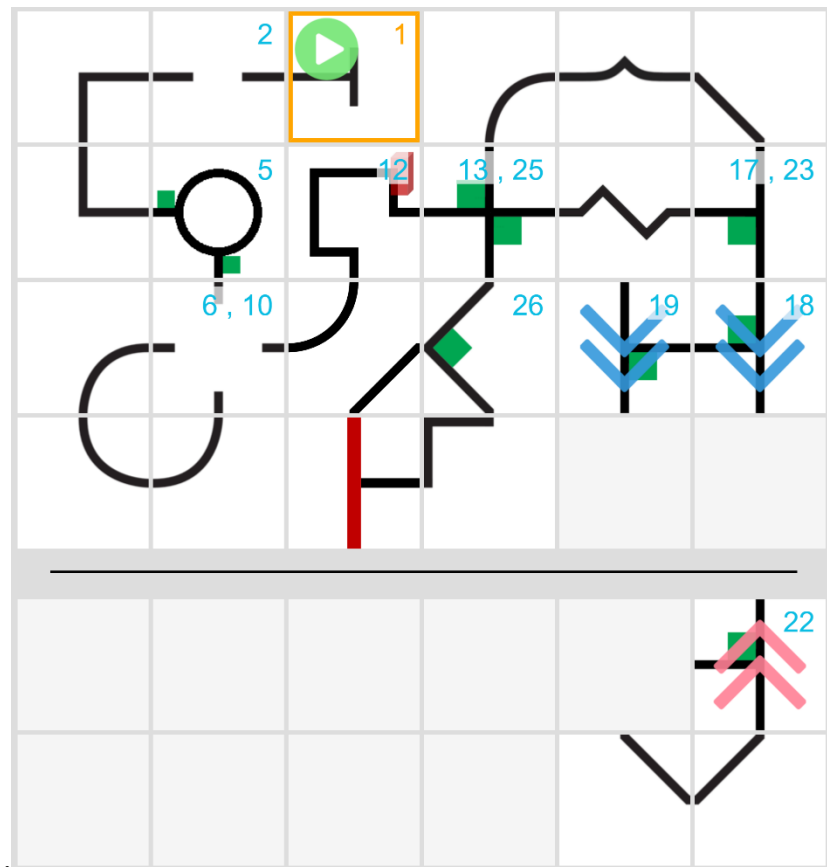
54

Arena 5: Schwierigkeit: ★★★★☆	
Arena 6: Schwierigkeit: ★★★★☆	

Arena 7: Schwierigkeit: ★★★★☆	
Arena 8: Schwierigkeit: ★★★★☆	

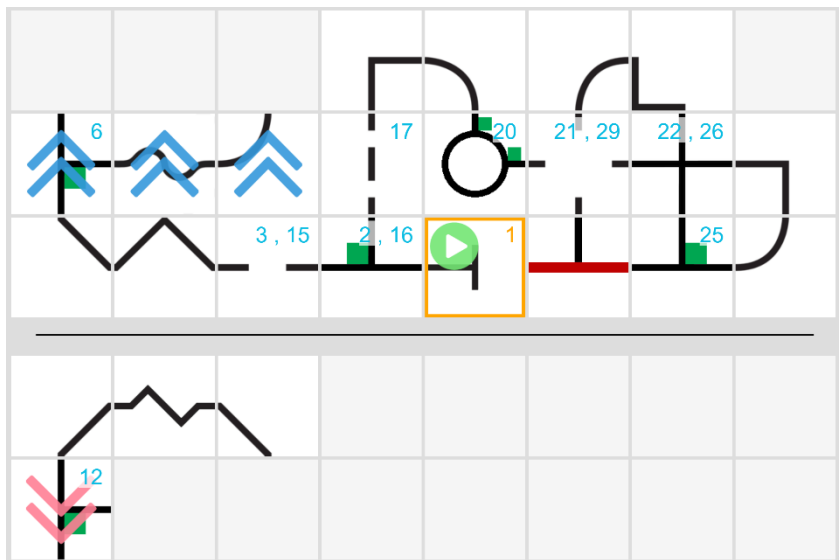
Arena 9:

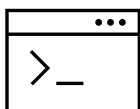
Schwierigkeit:



Arena 10:

Schwierigkeit:





Maze Entry Arenen

Schwierigkeitseinstufung der Arenen

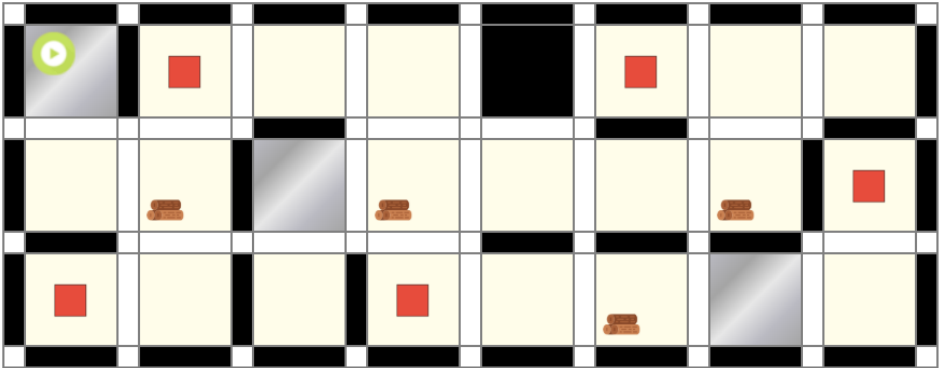
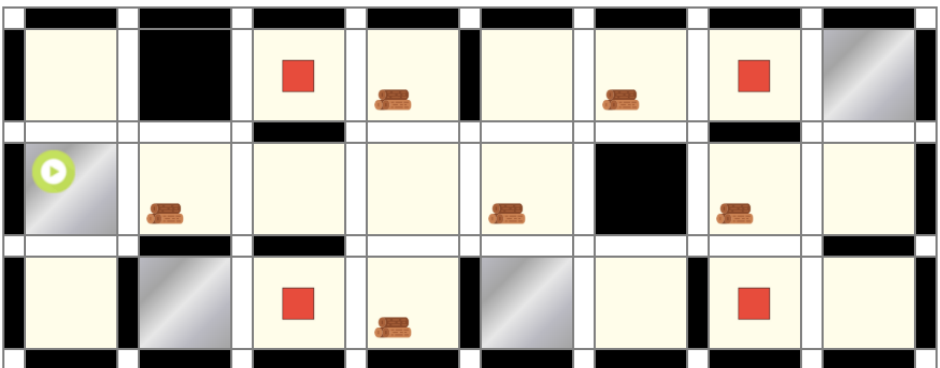
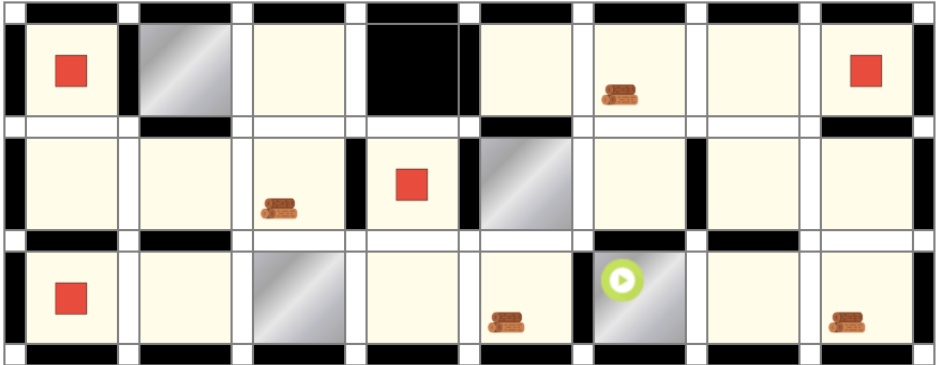
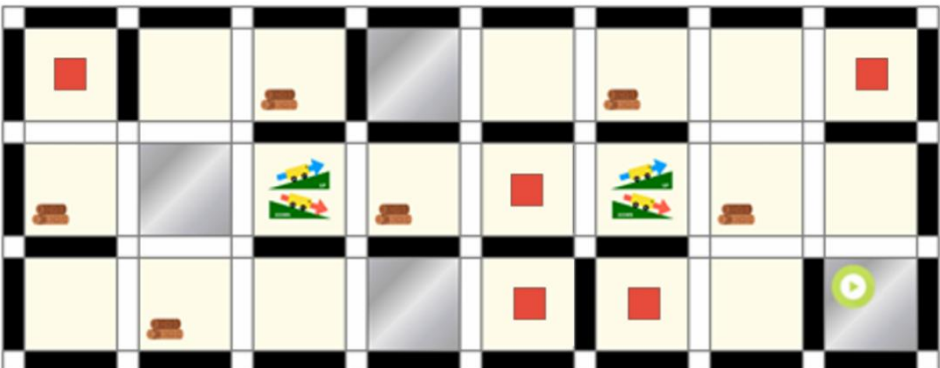
Arena	Labyrinth	Rampe	Schwarze Kacheln	Hindernis	Bumper	Opfer
1	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
2	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
3	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
4	☆☆☆	☆☆☆	-	-	☆☆☆	☆☆☆
5	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
6	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
7	☆☆☆	-	☆☆☆	-	☆☆☆	☆☆☆
8	☆☆☆	-	☆☆☆	☆☆☆	☆☆☆	☆☆☆
9	☆☆☆	☆☆☆	☆☆☆	-	☆☆☆	☆☆☆
10	☆☆☆	☆☆☆	-	-	☆☆☆	☆☆☆

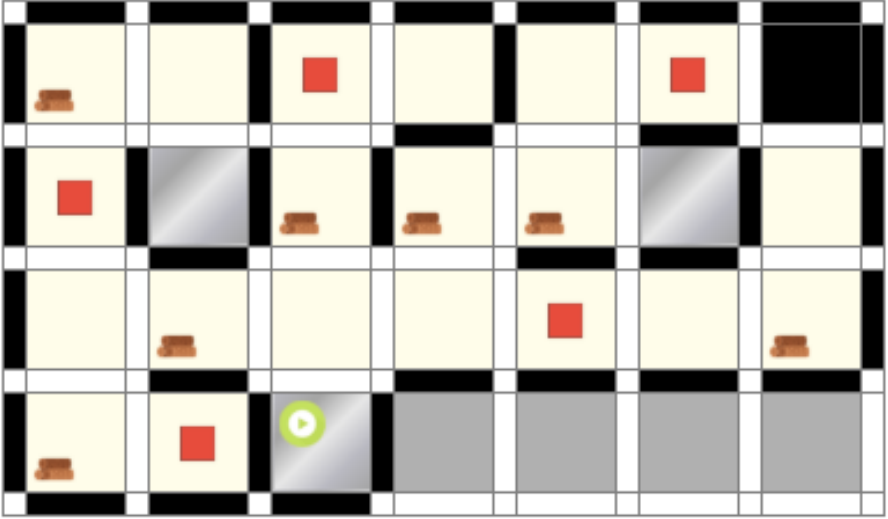
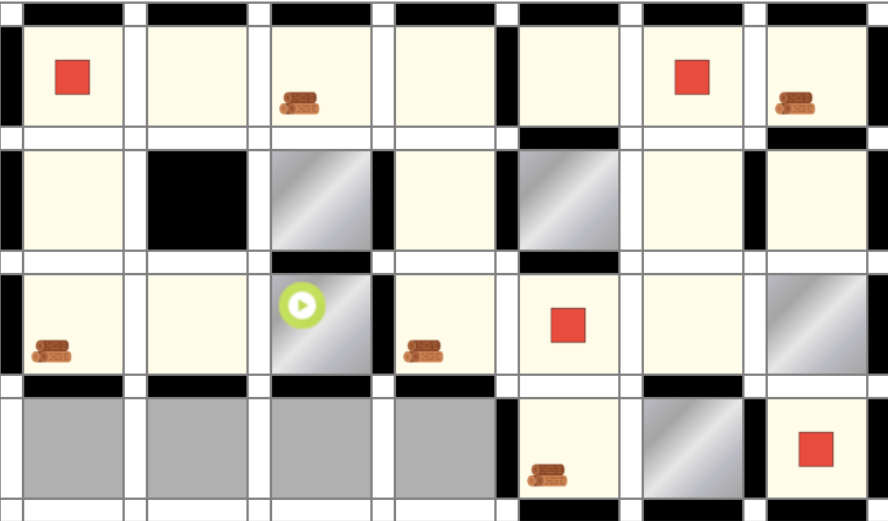
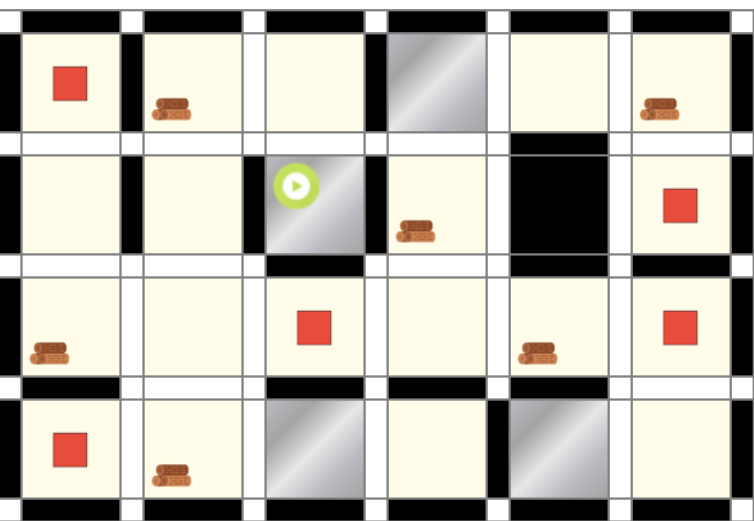
Hinweis:

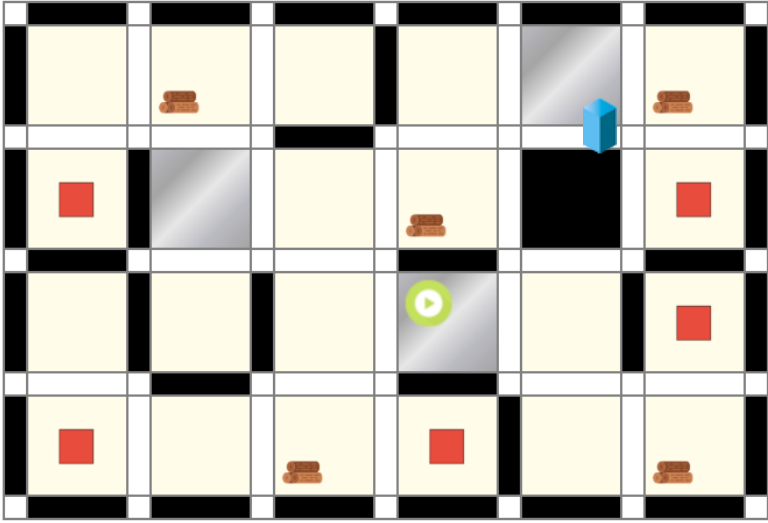
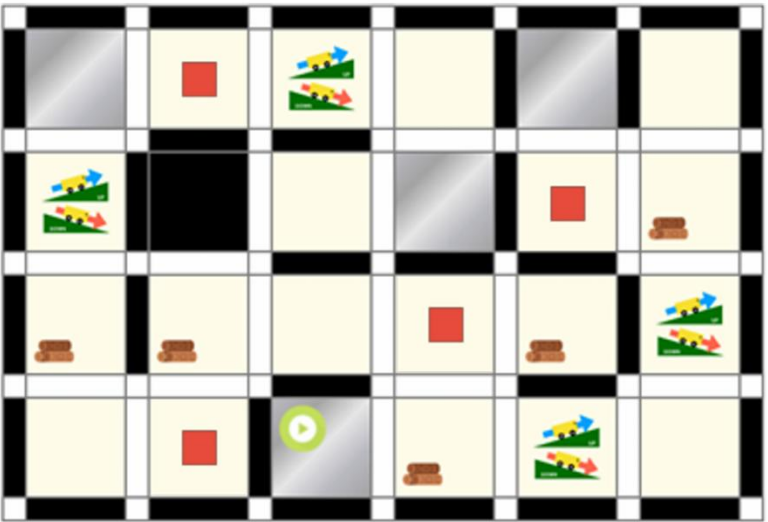
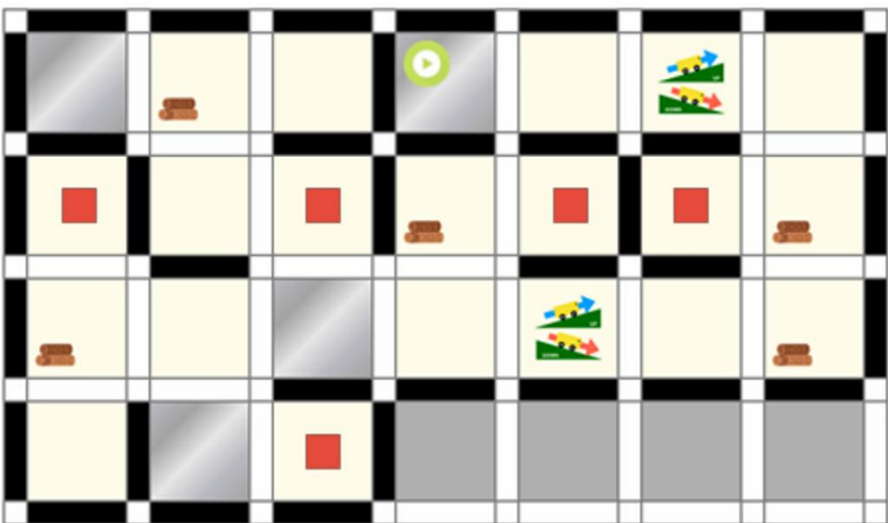
Die Schwierigkeit der Arenen wurde nach den folgenden Kriterien bewertet:

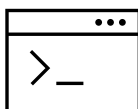
- Erreichbarkeit der Opfer
- Wegführung durch das Labyrinth
- Zusätzliche Elemente, die das Durchfahren erschweren (bspw. Rampen, Speed Bumper und schwarze Platten)

Die Einschätzung über die Schwierigkeit der Arenen kann individuell variieren. Die Arenen dienen zur Inspiration und können selbstverständlich an die eigenen Bedürfnisse angepasst werden.

Arena 1: Schwierigkeit: ☆☆☆	
Arena 2: Schwierigkeit: ☆☆☆	
Arena 3: Schwierigkeit: ☆☆☆	
Arena 4: Schwierigkeit: ☆☆☆	

Arena 5: Schwierigkeit: ★★★★☆	
Arena 6: Schwierigkeit: ★★★★☆	
Arena 7: Schwierigkeit: ★★★★☆	

Arena 8: Schwierigkeit: ★★★☆☆	
Arena 9: Schwierigkeit: ★★★★☆	
Arena 10: Schwierigkeit: ★★★★★	



Maze Arenen¹

Schwierigkeitseinstufung der Arenen

Arena	Labyrinth	Rampe	Treppe	Hindernis	Bumper	Opfer
1	☆☆☆	-	-	-	☆☆☆	☆☆☆
2	☆☆☆	-	-	-	☆☆☆	☆☆☆
3	☆☆☆	-	☆☆☆	☆☆☆	☆☆☆	☆☆☆
4	☆☆☆	☆☆☆	-	-	☆☆☆	☆☆☆
5	★★★★	☆☆☆	-	-	☆☆☆	☆☆☆
6	★★★★	☆☆☆	-	-	☆☆☆	☆☆☆
7	☆☆☆	☆☆☆	☆☆☆	☆☆☆	☆☆☆	★★★★
8	★★★★	-	-	-	☆☆☆	★★★★
9	★★★★	★★★★	☆☆☆	-	☆☆☆	★★★★
10	★★★★	★★★★	★★★★	-	☆☆☆	☆☆☆

Hinweis:

Die Schwierigkeit der Arenen wurde nach den folgenden Kriterien bewertet:

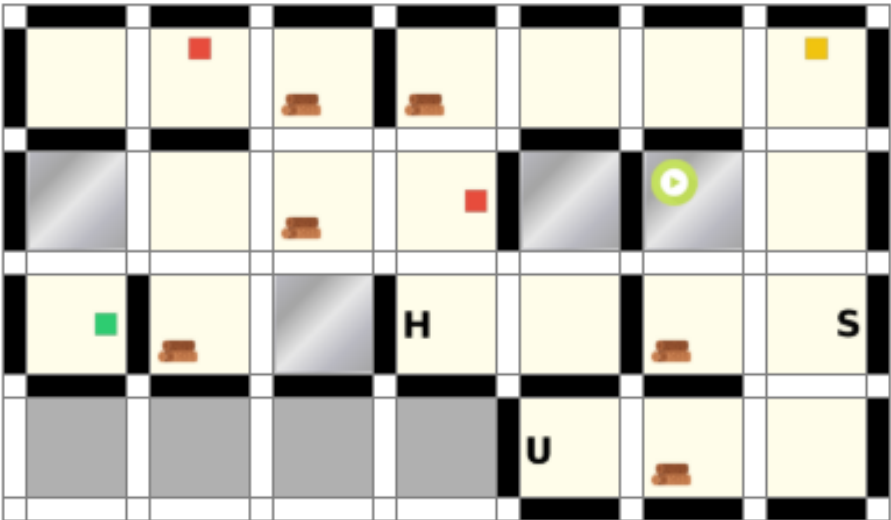
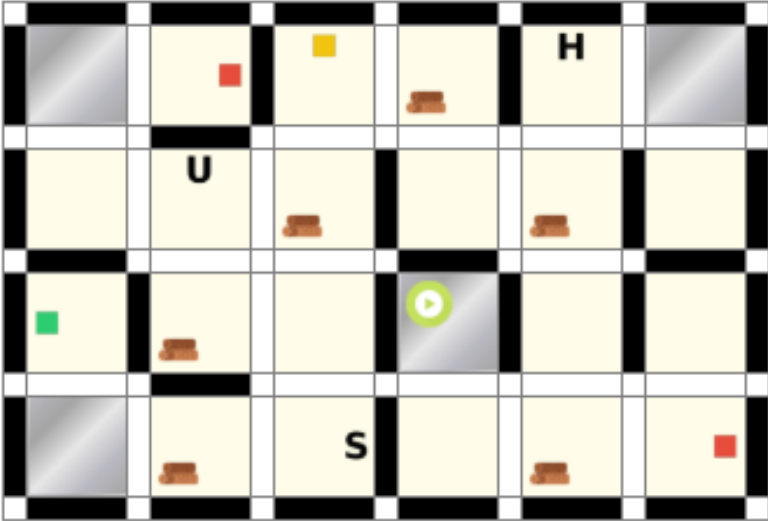
- Position der Opfer
- Position der linearen und „floating“ Kacheln
- Zusätzliche Elemente, die das Durchfahren erschweren (bspw. Rampen, Treppen, Speed Bumper, schwarze und blaue Kacheln)

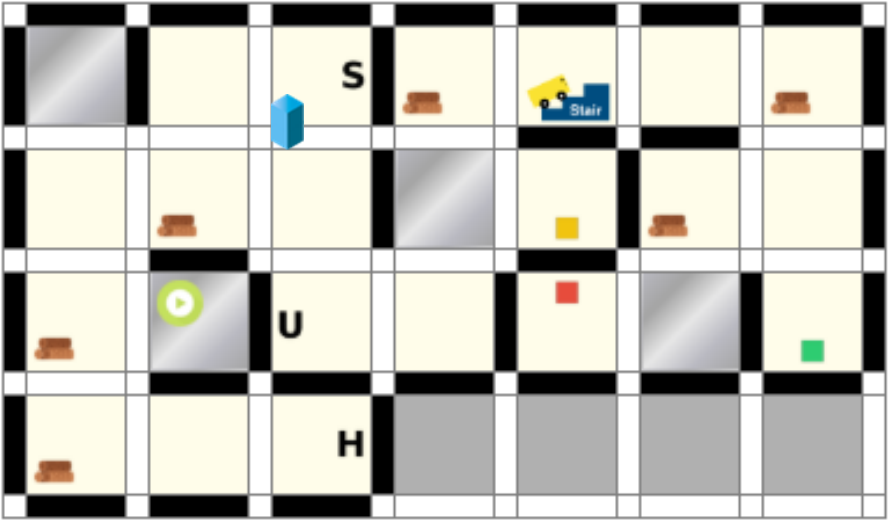
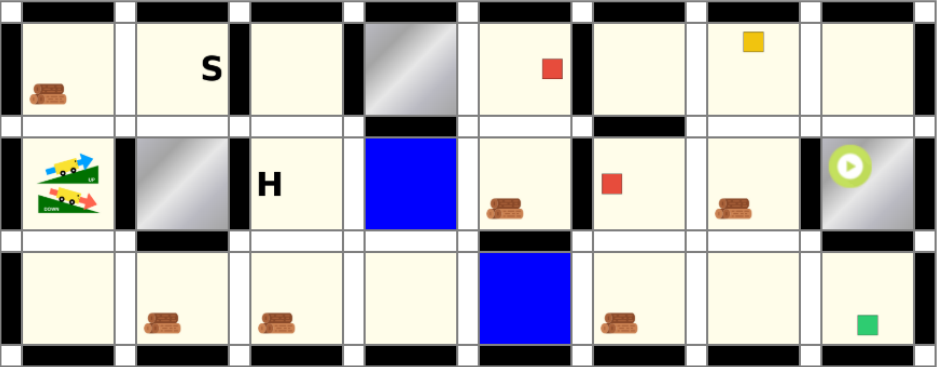
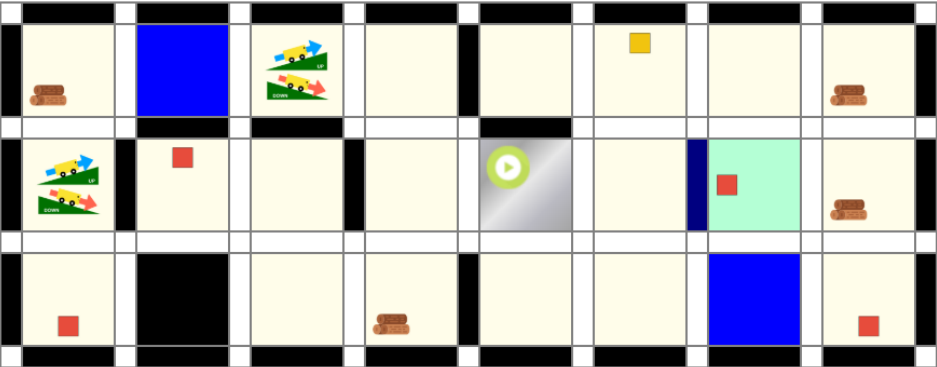
Die Einschätzung über die Schwierigkeit der Arenen kann individuell variieren.

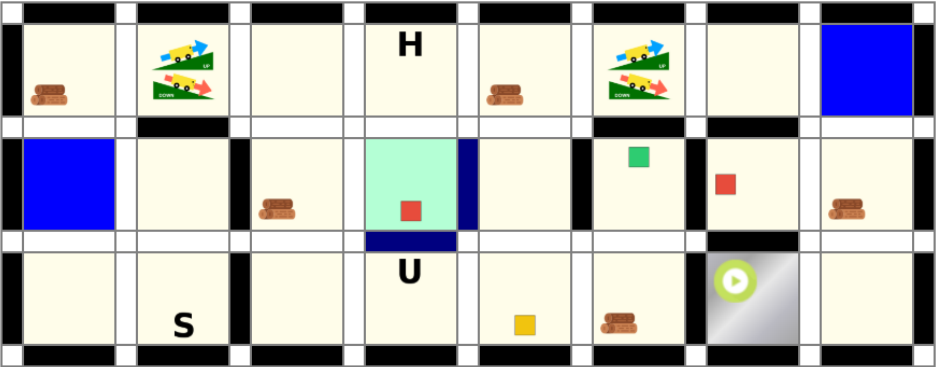
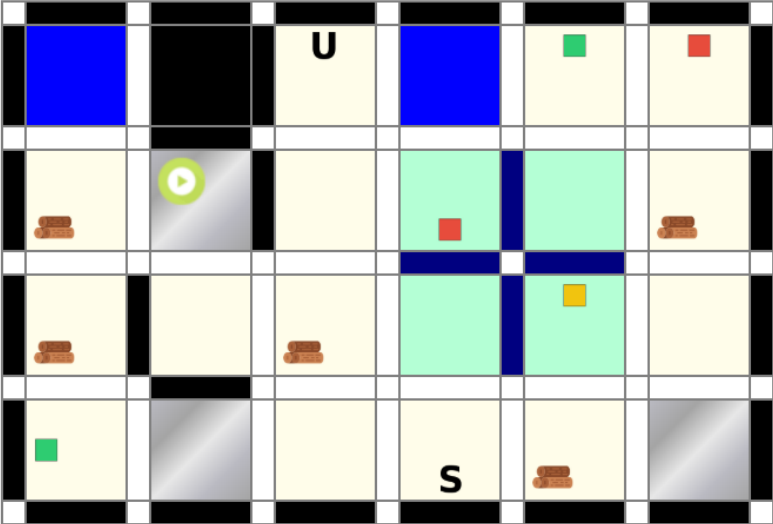
Insgesamt sind Arenen mit „floating“ Wänden allerdings schwerer bewertet, da diese in

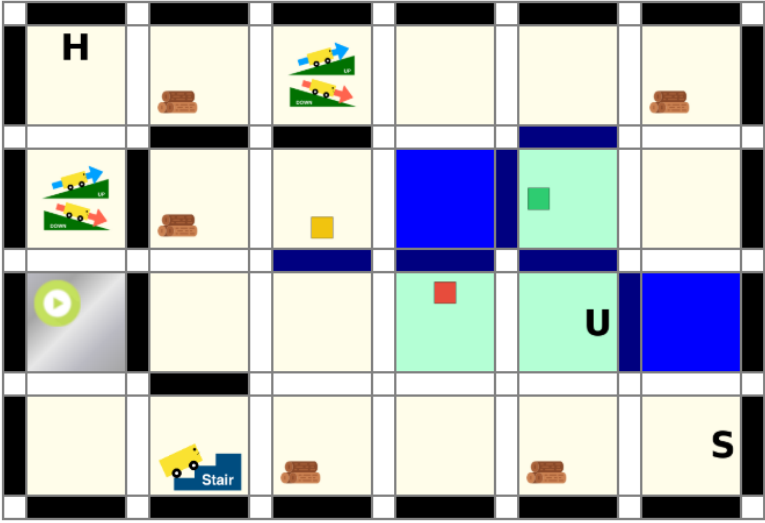
¹ Die grün gekennzeichneten Kacheln stellen „floating“ Kacheln dar. Diese Kacheln sind nicht erreichbar, indem man der rechten oder linken Wand folgt und somit geben die Opfer auf diesen Kacheln mehr Punkte.

der Regel nur mit einer vom Roboter erstellen Karte zu erreichen sind. Die Arenen dienen zur Inspiration und können selbstverständlich an die eigenen Bedürfnisse angepasst werden.

Arena 1: Schwierigkeit: ☆☆☆	
Arena 2: Schwierigkeit: ☆☆☆	

Arena 3: Schwierigkeit: ☆☆☆	
Arena 4: Schwierigkeit: ☆☆☆	
Arena 5: Schwierigkeit: ☆☆☆	

Arena 6: Schwierigkeit: ★★★☆☆	
Arena 7: Schwierigkeit: ★★★☆☆	
Arena 8: Schwierigkeit: ★★★★☆	

Arena 9: Schwierigkeit: ★★★★	
Arena 10: Schwierigkeit: ★★★★	