

Wir programmieren einen Roboter

Drei Unterrichtsreihen für die Klassen 2-4 zur Abdeckung des MKR-NRW Bereich 6

Manuel Lammers

(manuellammers@icloud.com)

Lehrerhinweise

Bei den Abbildungen von Programmierblöcken und Schaltflächen handelt es sich um Screenshots aus der Programmieroberfläche ozoblockly.com.

Inhaltsverzeichnis

HINTERGRUND	4
SCHÜLER- ODER LEHERZENTRIERT?	4
INHALTE KLASSE 3.....	5
INHALTE KLASSE 4.....	5
SCHWERPUNKTE NACH MKR NRW	6
BENÖTIGTE MATERIALIEN.....	6
DER KOMPETENZBEREICH 6.4.....	6
QR-CODES.....	7
<u>KLASSE 2</u>	<u>8</u>
EINHEIT 1: ÜBERALL COMPUTER	8
AB 1	8
EINHEIT 2: EINEN ROBOTER KENNENLERNEN.....	9
AB 2	11
AB 3	11
AB4 AUFGABE 1	12
AB4 AUFGABE 2	13
EINHEIT 3: ANWEISUNGEN GEBEN	13
AB 5	13
AB 6	14
EINHEIT 4: ANWEISUNGEN GEZIELT EINSETZTEN.....	15
PLAKATVORLAGE:	15
AB 9	15
ABSCHLUSS DER UNTERRICHTSREIHE:	15
<u>KLASSE 3</u>	<u>16</u>
EINHEIT 1: EINEN ROBOTER KENNEN LERNEN	16
AB 1	16
AB 2	16
AB 3 UND 4	16
AB 5	17

AB6.....	18
EINHEIT 2: ANWEISUNGEN GEBEN	18
AB 7-8.....	18
AB 10	19
AB 11-12.....	19
EINHEIT 3: PROBLEMSTELLUNGEN LÖSEN	19
AB 13-15.....	19
AB 16	20
AB 17-18.....	20
EINHEIT 4: FREIE FAHRT	21
<u>KLASSE 4</u>	<u>22</u>
STUFE 1 (SEITEN 1-12):.....	22
STUFE 2 (SEITE 13-28):	23
STUFE 3 (SEITE 29-53)	23
NEUERUNGEN AUF STUFE 3.....	23
ZUM KOMPETENZBEREICH 6.4	24

Hintergrund

Seit 2020 gilt der Medienkompetenzrahmen NRW verbindlich. Von den sechs Kompetenzbereichen lassen sich die Bereiche 1-5 gut in den Alltagsunterricht integrieren, wenn das Schulleben die digitale Welt aktiv in das Unterrichtsgeschehen einbindet. Praktisch alle Anforderungen aus diesen Bereichen lassen sich dabei fächerübergreifend quasi nebenher erfüllen.

Anders sieht es für den Bereich 6 (Problemlösen und Modellieren) aus. Diese Unterrichtsreihen haben daher den Zweck, den Kompetenzbereich 6 des MKR-NRW in der 2. Klasse der Grundschule abzudecken.

Es liegen Materialien für die 2.-4. Klasse vor.

Die empfohlenen Zeitpunkte für die Durchführungen im Laufe der Schuljahre sind so gewählt, dass sie als zusätzlicher Unterrichtsstoff im Bereich Sachunterricht möglichst wenig weitere Unterrichtszeit belegen.

Schüler- oder Lehrerzentriert?

Prinzipiell sind zwei Arten von Material vorhanden:

- herkömmliche Arbeitsblätter
Ab der 3. Klasse sind diese auch zur eigenständigen Erarbeitung geeignet. Beispielsweise könnten einzelne Aufgaben Teil von Wochen- / oder Tagesarbeitsplänen sein.
- Unterrichtspräsentationen
Für schwächere Zielgruppen hat sich die eigenständige Erarbeitung nicht bewährt. Daher sind die Unterrichtspräsentationen und die zugehörigen Arbeitsblätter neu erstellt worden. Das Material ist für den lehrerzentrierten Unterricht gedacht. Die Arbeitsblätter enthalten in dieser Version nicht mehr alle benötigten Informationen. Sie können nur in Verbindung mit der Präsentation sinnvoll eingesetzt werden. Die Präsentationsdateien enthalten in der Moderatorenansicht Hinweise für Lehrer.
Die Inhalte sind identisch. Das Material für Klasse 4 enthält jedoch nicht die Aufgaben der „Forderstufe“.

**Der nachfolgende Text bezieht sich auf das herkömmliche Material,
die Inhalte sind jedoch identisch!**

Klasse 2:

Die Unterrichtsreihe für die 2. Klasse sollte am Ende des Schuljahres kurz vor den Sommerferien platziert werden. Das Thema und die Arbeit mit den Robotern sind erfahrungsgemäß so attraktiv, dass die Schüler sich auch bis zum letzten Tag vor den Sommerferien einbringen.

Die Unterrichtsreihe behandelt folgende Themen:

- Wo gibt es überall Computer / digitale Datenverarbeitung?
- Computer benötigen Anweisungen
- Erste Programmierung über Farbcodes

Klassen 3 und 4:

Die Unterrichtsreihen für die Klassen 3 und 4 bauen stark aufeinander auf. Die Reihe für die 3. Klasse ist für den Zeitraum unmittelbar vor den Sommerferien gedacht. Die Unterrichtsreihe für die 4. Klasse sollte direkt nach den Sommerferien anschließen. Eine aufwendige Wiederholung kann so entfallen.

Zudem sind die Aufgaben der Unterrichtsreihe für das 3. Schuljahr teilweise komplex und anders als die Reihe für den 2. Jahrgang nicht nur für den Klassenunterricht, sondern auch zur weitgehend eigenständigen Erarbeitung geeignet. Sie können im Block oder blattweise z.B. als Bestandteil von Wochenplänen herausgegeben werden.

Zu allen Aufgaben und Experimenten steht die Lösung jeweils auf der nächsten Seite.

LuL benötigen in Sachen Programmierung KEINEN Wissensvorsprung vor den Schülern und können sich den Stoff gemeinsam mit ihnen erarbeiten.

Inhalte Klasse 3

Die Unterrichtsreihe behandelt folgende Themen:

- Wiederholung der Grundlagen
- Erweiterung der Anweisungen
- Lösen von Problemen mit Hilfe der gelernten Anweisungen

Inhalte Klasse 4

- Programmierung mit OzoBlockly (Scratch)
- Anweisungen
- Parameter

- Schleifen
- Verschachtelte Schleifen
- Bedingte Anweisungen

Schwerpunkte nach MKR NRW

1.1, 1.2, 6.1, 6.2, 6.3, 6.4

Benötigte Materialien

Benötigte Materialien: Neben diesen Lehrerhinweisen und den zugehörigen Arbeitsblättern werden Ozobot Bit oder Ozobot Evo Lernroboter benötigt. Schulen aus Hamm (NRW) können diese beim Medienzentrum der Stadt ausleihen.

Klasse 2:

Es sollte ein Roboter und ein Set sehr dicker Filzstifte (Schwarz, Rot, Blau, Grün) für je zwei SuS vorhanden sein. Die Farben müssen genau den Stiften der originalen Ozobot Stiften entsprechen! Für einige Übungen können zusätzliche Materialien benötigt werden, die aber in jedem Klassenzimmer vorhanden sind (Blätter, kleine Gegenstände als Hindernisse usw.)

Klasse 3 / 4:

Neben dem Roboter sollte für jedes Kind ein Tablet oder PC mit Internetzugang vorhanden sein. Partnerarbeit ist möglich, die Arbeitsgeschwindigkeit ist jedoch erfahrungsgemäß sehr unterschiedlich. Zudem gibt es oft unterschiedliche Lösungsansätze. Sich in einen fremden Code hineinzudenken ist schon für erfahrenen Programmierer nicht ganz einfach. Zudem besteht die große Gefahr, dass die stärkeren Schüler die Programmierung übernehmen. Es ist daher deutlich sinnvoller, wenn alle SuS individuell an ihren eigenen Programmen arbeiten können.

Klasse 4: Zusätzlich zu den zuvor genannten Materialien werden ein Set sehr dicker Filzstifte (Schwarz, Rot, Blau, Grün) benötigt.

Der Kompetenzbereich 6.4

„Bedeutung von Algorithmen Einflüsse von Algorithmen und Auswirkung der Automatisierung von Prozessen in der digitalen Welt beschreiben und reflektieren“

Dieser Bereich sollte in jeder Unterrichtsreihe mindestens einmal aufgegriffen werden.

Erfahrungsgemäß nennen die SuS häufig spontan eigene Beobachtungen, die sie zu Inhalten aus der Unterrichtsreihe vergleichbare Techniken im Alltag gemacht haben. Daher sind Inhalte zu diesem Bereich nicht fest im Material verankert. Der Bereich 6.4 sollte immer dann spontan mit einbezogen

werden, wenn Schülerbeiträge auf diesen eingehen. In diesem Dokument werden ebenfalls Anknüpfungspunkte zur Thematisierung unter dem Stichwort „zum Kompetenzbereich 6.4“ genannt.

QR-Codes

Das überarbeitete Material für die Klassen 2 und 3 enthält nun QR-Codes. Diese Codes enthalten den nebenstehenden Text als Code, so dass auf iPads Siri diesen für Kinder mit Schwierigkeiten beim Lesen vorlesen kann. Dafür muss folgender Kurzbefehl angelegt werden:

- QR-/Balkencode scannen
- Sprechen

Dem fertigen Kurzbefehl sollte dann ein eingängiger Name wie „vorlesen“ gegeben werden und auf den Homescreen verlinkt werden.

In der zweiten Hälfte dieses Videos zeige ich das Vorgehen:

<https://www.youtube.com/watch?v=jOqdhCtWAYg>

Klasse 2

Einheit 1: Überall Computer

Kompetenzbereich: 6.1, 6.4

Kompetenzerwartung: Die SuS unterscheiden Gegenstände, die keinen Computer enthalten (analoge Welt) von Gegenständen, die einen Computer enthalten.

Materialien: Analoge und digitale Alltagsgegenstände (z.B. Taschenrechner, Handy, Armbanduhr (analog oder digital), Taschenlampe, Anspitzer, Stift, Kamera....)

Achten Sie darauf, dass auch elektrische Geräte dabei sind, die keine Computer enthalten, damit nicht der falsche Eindruck entsteht, dass alle elektrischen Geräte digital seien.

1. Stundentransparenz: Wir unterscheiden Gegenstände mit und ohne Computer.
2. Gespräch: Was ist ein Computer? (Sitzkreis, Theaterkreis o.Ä.)
Unter „Computer“ wird hier verstanden, dass ein Gegenstand eingegebene Daten erfasst, digital berechnet und eine Ausgabe generiert. Beim Flugzeug wären die Eingaben die Steuerungseingaben des Piloten und Sensordaten, die Ausgabe die Steuerung der Flugzeugkomponenten und Ton-, Bild- und Textausgaben im Cockpit.
Beim PC wären die Eingaben beispielsweise die Benutzereingaben an der Tastatur und/oder Maus, per Spracheingabe (über das Mikrofon) oder Stift. Die Ausgaben wären Bildschirmanzeigen, Töne/Musik und Ausgaben auf dem Drucker.
3. Gegenstände zuordnen: Welche Alltagsgegenstände enthalten Computer?
Dazu die realen Gegenstände gemeinsam zuordnen.
4. Arbeitsblatt 1 (s.u.)
5. Gespräch: Überlegen, wie viele verschiedene Computer jeder täglich benutzt.
(digitaler Wecker, digitale Personenwaage, Toaster mit Bräunungssensor, moderner Kaffeevollautomat, elektrische Zahnbürste mit Sensoren und/oder Timer, Brotbackmaschine, Waschmaschine, Spülmaschine, Handy, Tablet, PC, TV, Spielekonsole, CD/MP3-Player, Bluetooth-Lautsprecher, Taschenrechner, digitale Spielzeuge, W-Lan-Accesspoint, Telefon, Auto, Fahrradacho /-computer, digitale Medizinprodukte, Alarmanlage, digitale Armbanduhr). (Kompetenzbereich 6.4)
6. Aussicht auf die nächste Einheit: Vorstellung Ozobot Bit Lernroboter

AB 1

Kompetenzerwartung:

Die SuS unterscheiden Gegenstände, die keinen Computer enthalten (analoge Welt) von Gegenständen, die einen Computer enthalten (digitale Welt).

Achtung!

Einige der Gegenstände kann es in digitaler oder analoger Form geben! (z.B. Puppe mit Sprachmodul, Herd mit Timer, Bohrmaschine mit Sensoren und digitaler Steuerung, Uhren)

Einheit 2: Einen Roboter kennenlernen

Kompetenzbereiche:

1.1, 1.2, 6.1, 6.2

Kompetenzerwartungen:

- Die SuS unterscheiden einen Roboter von anderen Computern!
 - Die SuS lernen den Ozobot Bit kennen.
 - Computer sind nicht schlau, sie brauchen für alles Anweisungen (Befehle)!
1. Rückblick auf die letzte Stunde (analoge / digitale Geräte unterscheiden)
 2. Stundentransparenz: Wir klären, was ein Roboter ist. Danach stelle ich euch einen echten Roboter vor.
 3. Gespräch: Was kennt ihr für Roboter (real und aus TV/Literatur)?
Was unterscheidet einen Roboter von anderen Computern?

Sachanalyse:

Roboter (slawisch: robota -> Fronarbeit): Als Begriff für menschenähnliche Maschinen geprägt durch SciFi-Literatur (Isaac Asimov 1922-1992).

nach Thomas Christaller: sensomotorische Maschinen zur Erweiterung oder zum Ersatz menschlicher Handlungen. Bestehen aus mechanischen und sensorischen Elementen und werden über einen (digitalen) Rechner gesteuert.

Abgrenzung Automat: Automaten folgen eher einem strikten vorgegebenen Ablauf, der durch einen Auslöseimpuls initiiert wird. Automaten können oft auch ohne digitalen Rechner realisiert werden (Kaugummi-Automat). Die Sensorik ist oft weniger komplex (Erkennen der Münze). Roboter hingegen handeln in ihrem vorgegebenen Umfeld autonom. Der Übergang ist oft fließend. Ohne entsprechende Programmierung ist auch Bit nur ein Automat.

Abgrenzung Gerät / Maschine: Oberbegriff

Abgrenzung Computer: Die reine Rechenmaschine mit oder ohne HID (Human Interface

Device / Tastatur, Maus, Display). Mechanische Komponenten, die mit der Umwelt interagieren, Beweglichkeit im Raum und Sensorik fehlen.

4. Vorstellung Ozobot Bit - ein echt(er) kleiner Roboter

Rechtfertigung / Begründung: Mit einfachen Robotern können die Anfänge der Programmierung viel anschaulicher vermittelt werden als mit einem PC oder Tablet, bei dem das Gerät zur Programmierung und das programmierte Geräte identisch sind. Die Folgen der Programmierung sind zudem besser visualisierbar.

Material: Steckbrief Ozobot Bit

Hinweise zum Steckbrief:

Der Steckbrief ist absichtlich so formuliert, dass Bit wie eine Person vorgestellt wird. Dies bieten den Anlass über (vermeintlich) menschliche Züge von Robotern zu diskutieren: Haben Roboter Gefühle oder einen Willen? (Nein, natürlich nicht, sie arbeiten lediglich eine oft komplexe Programmierung ab.)

Hintergrundwissen zum Steckbrief:

Der **Name Bit**. Ein Bit ist die kleinste mögliche Informationseinheit in der Informatik. Es kann den Werte 0 oder 1 annehmen (Daher auch Bits Lieblingszahlen im Steckbrief). Ozobot ist der Markenname der Produktreihe, Bit das kleinste Modell der Serie. Vermutlich ist der Name eine Anspielung hierauf oder auf seine physische Größe.

Exkurs:

Alle Computerdaten werden in Form von 0 und 1 gespeichert. Egal ob Texte, Bilder, Musik oder Apps. Größere Informationseinheiten sind ein Byte (8 Bit) und entsprechende Präfixe/Multiplikatoren Kilo, Mega, Giga, Tera, Peta... Ein Kilobyte kann dabei 1000 Byte oder 1024 Byte groß sein, je nach genutzter Basis. Details würden hier zu weit führen und sind für den Grundschulbereich auch nicht relevant.

Interessant ist jedoch noch zu wissen, dass die Einheit 1 Byte = 8 Bit daher kommt, dass frühe Computer einen Datenbus von 8 Bit Breite hatten. Das bedeutet, dass sie mit jedem Rechenzyklus 8 Bit auf einen Schlag (Bissen/Happen -> Bit) verarbeiten konnten.

Korrekterweise müsste man bei heutigen (2020) 64 Bit Systemen also sagen, ein Byte = 64 Bit. Dies ist jedoch nicht so! Traditionellerweise bleibt es bei 1 Byte = 8 Bit!

Akku: Eingebauter Lithium-Ionen-Akku

Leuchtdioden: Die Leuchtdioden oben können alle Farben darstellen. Die Leuchtdioden auf der Unterseite dienen lediglich der Beleuchtung des Untergrundes, damit die Sensoren die Farbe des Untergrundes ermitteln können.

Sensoren: Hinter den Löchern auf der Unterseite befinden sich Sensoren, die die Farbe des Untergrundes ermitteln können. Bit benötigt sie, um Linien folgen (Linienverfolgungsmodus)

und Befehle über Farbcodes erfassen zu können. Die festgestellte Farbe kann zudem im Programmiermodus abgefragt werden. (Der Programmiermodus ist erst in der Unterrichtsreihe für die Klassen 3 und 4 vorgesehen.)

Über farbige Blink-Codes von einem Tablet oder PC können Bit zudem Programme übermittelt und Einstellungen an Bit vorgenommen werden.

Taster: Über den Taster wird Bit an- und ausgeschaltet. Ist Bit aus, wird er mit einem einzelnen Druck im Linienverfolgungsmodus angeschaltet. Mit einem doppelten Druck startet das abgespeicherte Programm.

5. **Hinweise zur Benutzung:**

Wenn du mit Bit arbeitest, dann achte also immer darauf, dass

- er niemals vom Tisch fällt oder für längere Zeit gegen schwere Hindernisse fährt.
- der Tisch oder der Boden auf dem du arbeitest sauber sind. Schon kleine Krümel können seiner winzigen Mechanik schaden!

6. Arbeitsblatt 2-4

7. Anwendung des Gelernten in eigenen Bahnen.

AB 2

Mögliche Beobachtungen:

Bit leuchtet (blau), dreht sich, bleibt stehen und beginnt (blau) zu blinken.

Sollte die Standardfarbe umgestellt worden sein, leuchtet und blinkt Bit in einer anderen Farbe.

Erläuterung:

Durch das Anschalten mit einem einfachen Tastendruck wird Bit mit Linienverfolgungsmodus angeschaltet. Bit dreht sich, um evtl. vorhandene Linien unter ihm zu finden. Findet er keine, bleibt er stehen und blinkt.

Schlussfolgerung:

Ohne Anweisungen durch Linien und Farben macht Bit nichts.

→ Computer brauchen Anweisungen, um etwas Sinnvolles zu tun.

Die restlichen AB dieser Einheit dienen nicht dazu, informationstechnische Grundlagen zu vermitteln, sondern Eigenschaften von Bit vorzustellen, die für die Durchführung weiterer Übungen hilfreich sind.

AB 3

Mögliche Beobachtungen:

Fall 1: Linie zu dünn:

Bit erkennt keine Linie und verhält sich so, als ob keine da wäre. Er dreht sich, stoppt und blinkt dann.

Fall 2: Linie zu dick:

Bit fährt los und leuchtet in der Farbe der Linie. Bei schwarzen Linien leuchtet er in der Standardfarbe (so lange nichts verstellt wurde: blau). Dabei „eiert“ er auf der Linie hin und her. Evtl. wendet er auch mehrfach.

Fall 3: Linie hat eine geeignete Dicke (ca. 5mm):

Bit verfolgt die Linie bis zum Ende und bleibt dann stehen. Dabei leuchtet er in der Farbe der Linie. Sollte die Linie ihre Farbe ändern, ändert sich auch die Farbe der oberen Leuchtdioden.

Erläuterung:

Bit erkennt die Linien. Linien sind für ihn die Anweisung, diesen zu folgen. Ohne neue Anweisungen bleibt Bit am Ende einer Linie stehen. Die Farbe der Linie ist für Bit die Anweisung, in der Linienfarbe zu leuchten.

Schlussfolgerung:

Bit kann Linien und Farben unter ihm erkennen. Hierfür hat er Sensoren (in den 5 Vertiefungen an der Unterseite, zusammen mit 5 Leuchtdioden)

Hinweis:

Sollte Bit der Linie nicht folgen, sind die Sensoren evtl. verdreckt oder Bit muss neu kalibriert werden:

- Bit anschalten
- Den Taster erneut betätigen (für 2 Sekunden) und auf einen schwarzen Punkt stellen.
- Bit beginnt zu blinken. Er bewegt sich kurz aus dem Kreis heraus. Dann sollte er grün blinken.
- Bit ist nun kalibriert.

Bit muss immer dann kalibriert werden, wenn sich der Untergrund ändert.

Sollte Bit noch immer nicht wie erwartet reagieren, bietet es sich an, ihn zu resettet. Dies geschieht über die App „Ozobot Bit“.

AB4 Aufgabe 1

Mögliche Beobachtungen:

Bit fährt die Hindernisse um.

Hinweis:

Bit sollte nicht zu lange mit durchdrehenden Rädern gegen schwere Hindernisse fahren! Dies kann ihn beschädigen!

Schlussfolgerung:

Bit weicht Hindernissen nicht aus.

Hintergrund:

Bit verfügt nur über Sensoren an der Unterseite. Das Modell „Ozobot Evo“ verfügt jedoch über weitere Ultraschallsensoren. Evo wird hier nicht thematisiert.

AB4 Aufgabe 2

Mögliche Beobachtungen:

Sehr spitzen Winkeln kann Bit nicht zuverlässig folgen. Manchmal bleibt er im Winkel stehen oder kehrt den Weg zurück, den er gekommen ist.

zum Kompetenzbereich 6.4:

Roboter, die in der Industrie selbstständig fahren, werden oft ebenfalls durch farbige Linien geleitet. Manchmal sind die Linien auch „unsichtbar“, weil sie aus Metallführungen unter der Fahrbahn stehen, die von Magnetsensoren abgetastet werden. Im Außenbereich wird diese Technik immer öfter durch GPS ersetzt.

Einheit 3: Anweisungen geben

Kompetenzbereiche: 1.1, 6.1-6.3

Kompetenzerwartungen:

- Die SuS vertiefen die Wichtigkeit präziser Anweisungen.
- Die SuS lernen die Folgen fehlender Anweisungen kennen.

1. Rückblick auf die letzte Stunde (Vorstellung Bit)
2. AB 5-6
3. Anwendung des Gelernten in eigenen Bahnen.
4. Ausblick auf die nächste Stunde (neue Anweisungen, viel ausprobieren)

AB 5

Mögliche Beobachtungen:

Bit biegt zufällig in eine Richtung ab oder fährt geradeaus.

Hintergrund:

Bit erkennt die Kreuzung, hat aber keine Anweisung, wie er sich verhalten soll. In diesem Fall wählt Bit zufällig eine Richtung aus (Zufallsgenerator).

Konsequenz:

Immer, wenn sich die Situation ändert (z.B. Möglichkeit abzubiegen, Ende der Linie) muss Bit eindeutige Anweisungen bekommen, um seine Reaktion nicht dem Zufall zu überlassen. Die

notwendige Anweisung wird immer unmittelbar vor dem Eintreten des Ereignisses gegeben (siehe AB6a und AB6b)

AB 6

Hintergrund:

Das AB ist identisch mit AB5, bis auf die Anweisung (Farbcode) vor der Kreuzung: „Rechts abbiegen“
Anweisungen erhält Bit über 3-4-stellige Farbcodes, die in die Bahn eingefügt werden. Eine vollständige Liste aller Anweisungen können Sie der Anleitung entnehmen bzw. liegt jedem Bit als Karte bei. Für diese Unterrichtsreihe notwendige Anweisungen werden im Material vorgestellt.

Mögliche Beobachtungen:

Bit biegt nun immer rechts ab.

Hintergrund:

Vor der Kreuzung ist der Farbcode für die Anweisung „rechts abbiegen“ eingefügt worden. Somit biegt Bit nicht mehr zufällig ab, da er nun eine eindeutige Anweisung hat.

Konsequenz:

Anweisungen bestehen aus Farbcodes, die in der Bahn integriert sind. Anweisungen, die sich auf eine vorausliegende Entscheidung beziehen, stehen immer vor dieser: z.B. vor einer Kreuzung.

Achtung! Farbcodes müssen immer auf geraden Streckenabschnitten eingetragen werden.

Hinweis: Für Kinder, die bei zukünftigen Übungen Probleme mit dem exakten Einzeichnen der Codes haben, kann auf Aufkleber (siehe Datei: „Ozobot farbcodes zweckform L4731.pdf“) zurückgegriffen werden. Alternativ bietet der Hersteller auch Holzpuzzleteile für die Erstellung von Bahnen an.)

zum Kompetenzbereich 6.4:

Optische Codes kennen die Kinder zum Beispiel in Form von Barcodes im Supermarkt oder in Form von QR Codes. Barcodes werden an der Kasse über einen Scanner gezogen, das Produkt wird anhand der im Code verschlüsselten Nummer erkannt. Die Kasse kann nun in der Datenbank den Preis und die Produktbezeichnung auslesen, damit rechnen und die Bezeichnung gegebenenfalls auf den Kassenzettel drucken. Ist das Kassensystem mit einem Warenwirtschaftssystem verbunden, kann auch der Verkauf eines Produktes registriert werden. Somit weiß das System immer, wie groß der Warenbestand noch ist und kann rechtzeitig Waren nachbestellen.

QR-Codes können Anweisungen oder Daten verschiedener Art enthalten. Häufig enthalten sie eine Internetadresse, auf die der Benutzer beim Scannen weitergeleitet wird. Dies ist besonders häufig bei Werbung der Fall. Kinder, die mit der Anton-App arbeiten, kennen QR-Codes von ihren Benutzerausweisen. Diese können anstelle einer Eingabe der Zugangsdaten gescannt werden.

Einheit 4: Anweisungen gezielt einsetzen

Kompetenzbereiche: 1.1, 6.1-6.4

Kompetenzerwartungen: Die SuS lernen weitere Anweisungen kennen und setzen diese ein. Die SuS planen Bahnen für Bit und setzen diese mit den gelernten Anweisungen um.

Umsetzung:

In Einheit 4 werden weitere Anweisungen eingeführt (Plakat). Die Anwendung soll auf AB 7-9 und eigenen Bahnen der Schüler geübt werden.

Plakatvorlage:

Die Geschwindigkeits-Anweisungen „Nitro“ und „Schneckentempo“ gelten nur für einen kurzen Augenblick. Zudem sind sie spiegelverkehrt aufgebaut. Wird die Anweisung „Schneckentempo“ von der anderen Seite durchfahren, so ergibt sich die Anweisung „Nitro“ und umgekehrt.

Alle anderen Geschwindigkeitsanweisungen können von beiden Seiten durchfahren werden.

AB 9

Gestalten Sie einen Wettbewerb aus dem Arbeitsblatt. Wer schafft es, Bit am schnellsten ins Ziel zu bringen?

Zusatzregel Alternative 1: Bit muss zuverlässig auch bei mehreren Versuchen den gleichen Weg fahren und darf nicht vom Weg abkommen (zu hohe Geschwindigkeit in der Kurve).

Zusatzregel Alternative 2: Zufallselemente sind erlaubt (Kreuzungen ohne Richtungsanweisung). Es zählt die schnellste Zeit von 3 Fahrten!

Klären Sie, warum bei den unterschiedlichen Regeln unterschiedliche Bahnen zum Gewinn führen können! (Anweisungsfelder, die sonst für Richtungssteuerungen genutzt werden müssten, stehen nur für Geschwindigkeitssteuerungen zur Verfügung. Der Weg ist dafür zufällig!)

Abschluss der Unterrichtsreihe:

Klären Sie, wo computergesteuerte Prozesse im Alltag wichtig sind:

- Schranke im Parkhaus
- Automatischer Stopp bei Überfahren roter Signale bei Zügen
- Autos mit Fahrassistenz
- hochwertiger Toaster: Auswurf des Toasts bei ausreichender Bräune
- Spielzeugdrohnen: Gleichgewicht halten beim Fliegen
- Rasenmäh-, Saug- oder Wischroboter

Klasse 3

Hinweise zur Durchführung: Anders als die Unterrichtsreihe für das Ende der 2. Klasse sind die Unterrichtsreihen für die 3. und 4. Klasse für die eigenständige Arbeit konzipiert. Zu jeder Aufgabe folgt die/eine Lösung auf dem jeweils nächsten Arbeitsblatt.

In der Programmierung noch unerfahrene LuL benötigen zur Durchführung keinen Wissensvorsprung vor den SuS, sie können sich den Stoff gemeinsam mit ihren Schülern erarbeiten.

Einheit 1: Einen Roboter kennen lernen

Kompetenzbereiche: 1.1, 1.2, 6.1, 6.2

- Kompetenzerwartungen: Die SuS kennen den Lernroboter Ozobot Bit.
- Die SuS kennen Regeln zum Umgang mit Bit.
- Die SuS wissen, dass Computer und damit auch Roboter Anweisungen für den Betrieb benötigen.

AB 1

reines Leseblatt

AB 2

Mögliche Beobachtungen:

Bit leuchtet (blau), dreht sich, bleibt stehen und beginnt (blau) zu blinken.

Sollte die Standardfarbe umgestellt worden sein, leuchtet und blinkt Bit in einer anderen Farbe.

Erläuterung:

Durch das Anschalten mit einem einfachen Tastendruck wird Bit mit Linienverfolgungsmodus angeschaltet. Bit dreht sich, um evtl. vorhandene Linien unter ihm zu finden. Findet er keine, bleibt er stehen und blinkt.

Schlussfolgerung:

Ohne Anweisungen durch Linien und Farben macht Bit nichts.

→ Computer brauchen Anweisungen um etwas Sinnvolles zu tun.

AB 3 und 4

Mögliche Beobachtungen:

Linie: Bit verfolgt die Linie bis zum Ende. Dort bleibt er blinkend stehen.

Punkt: Sollten SuS auf die Idee kommen, Bit auf den großen Kalibrierungspunkt zu stellen: Bit fährt innerhalb des Punktes hin und her, verlässt ihn aber nicht.

Erläuterung:

Linie: Bit erkennt die Linien. Linien sind für ihn die Anweisung, diesen zu folgen. Ohne neue Anweisungen bleibt Bit am Ende einer Linie stehen. Die Farbe der Linie ist für Bit die Anweisung, in der Linienfarbe zu leuchten.

Punkt: Bit arbeitet im Linienverfolgungsmodus und erwartet Anweisungen in Form farbiger Linien unter sich. Er versucht, den Punkt als Linie zu interpretieren und dieser zu folgen (was aber nur bedingt funktioniert).

Schlussfolgerung:

Bit kann Linien und Farben unter ihm erkennen. Hierfür hat er Sensoren (in den 5 Vertiefungen an der Unterseite, zusammen mit 5 Leuchtdioden)

Hinweis:

Sollte Bit der Linie nicht folgen, sind die Sensoren evtl. verdreckt oder Bit muss neu kalibriert werden:

- Bit anschalten
- Den Taster erneut betätigen (für 2 Sekunden) und auf einen schwarzen Punkt stellen.
- Bit beginnt zu blinken. Er bewegt sich kurz aus dem Kreis heraus. Dann sollte er grün blinken.
- Bit ist nun kalibriert.

Bit muss immer dann kalibriert werden, wenn sich der Untergrund ändert.

Sollte Bit noch immer nicht wie erwartet reagieren, bietet es sich an, ihn zu resettet. Dies geschieht über die App „Ozobot Bit“.

AB 5

Mögliche Beobachtungen:

Bit leuchtet in der Farbe der gezeichneten Linie und verfolgt diese. Ausnahme: Schwarze Linien. Hier leuchtet Bit blau. Bit reagiert so, als wenn er auf ein weißes Blatt gesetzt wird: Linien zu kontrastarm oder zu dünn (Ideal: ca. 4mm Breite – wie Textmarker)

Bit schlingert auf der Linie hin und her, dreht möglicherweise um: Linie viel zu dick.

Bit verfolgt Kurven und Winkel. Extrem engen Kurven und spitzen Winkeln kann er nicht folgen.

Erläuterungen:

Bei schwarzen Linien leuchtet Bit in der eingestellten Standardfarbe, im Lieferzustand blau (über die Ozobot Bit App einstellbar).

Schlussfolgerung:

Bit erkennt Anweisungen in Form von Farben nur innerhalb gewisser Spezifikationen. Sie müssen die richtige Stärke aufweisen, kontrastreich sein und keine zu engen Kurven oder spitze Winkel

aufweisen. → Computer können nicht interpretieren, was der Benutzer wohl meinen könnte:
Anweisungen müssen präzise sein, in diesem Fall: Die Ausführung der Linien muss sorgfältig sein!

AB6

Mögliche Beobachtungen:

Bit fährt das Hindernis um.

Hinweis:

Bit sollte nicht zu lange mit durchdrehenden Rädern gegen schwere Hindernisse fahren! Dies kann ihn beschädigen!

Schlussfolgerung:

Bit weicht Hindernissen nicht aus.

Hintergrund:

Bit verfügt nur über Sensoren an der Unterseite. Das Modell „Ozobot Evo“ verfügt jedoch über weitere Ultraschallsensoren. Evo wird hier nicht thematisiert.

Einheit 2: Anweisungen geben

Kompetenzbereiche: 1.1, 6.1-6.3

Kompetenzerwartungen:

Die SuS erkennen die Notwendigkeit genauer Anweisungen, lernen Anweisungen (Farbcodes) kennen und setzen diese ein.

AB 7-8

Mögliche Beobachtungen:

blau-rot-blau:

Bit wendet an dieser Anweisung und fährt zurück in die Gegenrichtung. Die Anweisung kann von beiden Seiten mit derselben Wirkung durchfahren werden.

grün-schwarz-grün:

Bit ändert die Geschwindigkeit auf „normal“. Die Anweisung kann von beiden Seiten mit der gleichen Wirkung durchfahren werden.

blau-grün-rot:

Bit ändert für einen kurzen Augenblick auf „Nitro-Geschwindigkeit“ und kehrt dann zur Normalgeschwindigkeit zurück. Wird die Anweisung in Gegenrichtung durchfahren, so wechselt Bit für einen kurzen Augenblick auf „Schnecken tempo“ und kehrt dann zur Normalgeschwindigkeit zurück.

Schlussfolgerung:

Viele Anweisungen werden in Form von Farbsequenzen an Bit übermittelt.

Hinweise: Die einzelnen Farben einer Sequenz müssen ca. 4-5mm lang sein und sauber aneinander anschließen, damit Bit sie richtig lesen kann. Zwischen den einzelnen Farben dürfen beispielsweise keine weißen Bereiche enthalten sein. Sauber zeichnen!

Farbsequenzen werden oft nicht richtig ausgelesen, wenn sie sich innerhalb von Kurven oder zu nahe an diesen befinden.

AB 10

Mögliche Beobachtungen:

Bit biegt entweder ab oder fährt geradeaus.

Schlussfolgerungen: Ohne Anweisungen können unerwartete oder unvorhersehbare Reaktionen auftreten.

Hintergrund:

Bit ist so programmiert, dass er für den Fall, dass er vor einer Abzweigung oder Kreuzung keine genaue Anweisung bekommen hat, zufällig einen Weg auswählt.

AB 11-12

Mögliche Beobachtungen:

Bit fährt an der Kreuzung immer geradeaus und somit eine Acht, wenn an beiden Platzhaltern die Anweisung „geradeaus“ eingetragen wurde.

Hinweis: Die Anweisung „geradeaus“ kann nicht von beiden Seiten durchfahren werden. Es ist daher auf die richtige Reihenfolge der Farbsequenz zu achten: blau-schwarz-rot in Fahrtrichtung.

Einheit 3: Problemstellungen lösen

Kompetenzbereiche: 1.1, 6.1-6.3

Kompetenzerwartungen:

Die SuS lernen weitere Anweisungen kennen.

Die SuS nutzen Anweisungen, um gestellte Probleme zu lösen.

AB 13-15

Es ist eine Kombination aus Richtungs- und Geschwindigkeitsanweisungen nötig, um so schnellstmöglich zum Ziel zu gelangen.

Die Aufgabe ist erfüllt, wenn Bit zuverlässig immer auf dem gleichen Weg das Ziel erreicht.

Diese Übung eignet sich als Wettbewerb zwischen Schülern oder Arbeitsgruppen.

Alternative Regel: Es gewinnt, wer mit seinem Bit innerhalb von drei Fahrten am schnellsten zum Ziel kommt.

Erklärung: Die alternative Regel erlaubt also Zufallselemente (keine Richtungsanweisung vor einer Kreuzung). Die dadurch freigewordenen Anweisungsfelder können zur Geschwindigkeitsregelung genutzt werden.

Zu schnelle Geschwindigkeiten können Bit aus der Bahn werfen.

Die AB 14-15 sind zwei alternative Lösungen, die beide gleichwertig sind. Ein Weg ist länger, Bit fährt aber schneller. Der andere Weg ist kürzer, Bit fährt aber langsamer.

AB 16

Mögliche Beobachtungen:

Nach der Anweisung „springe nach links“ (grün-rot-grün“) dreht Bit sich um 90° nach links. Danach verlässt er die Strecke und fährt so lange geradeaus, bis er auf eine neue trifft. Auf der neuen Strecke fährt er in zufälliger Richtung weiter.

Dies ist auch der Grund dafür, dass es zwei Ziele gibt.

Hinweise:

Es kann passieren, dass Bit, wenn er seine Strecke verlässt, nicht richtig geradeaus fährt. Muss Bit längere Strecken frei fahren, ohne dass er von einer Linie geleitet wird, so kann dies im Extremfall dazu führen, dass Bit im Kreis fährt und sein Ziel daher nicht erreicht.

Sollte dieses Problem bei einem Bit zu groß werden, sollten seine Räder gesäubert und seine Motoren neu kalibriert werden. Dies kann in der „ozoboz Bit App“ erledigt werden. Die Räder werden gereinigt, indem Bit vorsichtig etwas seitlich über ein frisches Blatt Papier gezogen wird.

Hintergrund:

Die Räder sind sehr klein, so dass schon geringste Abweichungen in der Drehzahl eines Rades dazu führen, dass Bit nicht mehr geradeaus fährt, wenn er nicht geleitet wird. Ist ein Rad feucht oder fettig, verändert sich die Reibung des Rades, was den gleichen Effekt hat.

AB 17-18

Hinweise:

Nach der Kurve ist die Anweisung „springe vorwärts“ einzufügen: „grün-blau-grün“.

An der gegenüberliegenden Seite sollte eine Linie quer zur Fahrtrichtung gezeichnet werden, um Bit aufzufangen.

Da Bit in eine zufällige Richtung fährt, sobald er auf die „Auffanglinie“ trifft, müssen am Linienende die Anweisungen „wenden“ (blau-rot) eingefügt werden.

Damit sichergestellt wird, dass Bit irgendwann zum Ziel abbiegt, müssen die Anweisungen zum Abbiegen eingesetzt werden (links abbiegen → grün-schwarz-rot, rechts abbiegen → blau-rot-grün).

Schlussfolgerung:

Häufig müssen viele Anweisungen genutzt werden, um eine scheinbar einfache Aufgabe zu lösen.

AB 18 zeigt diese Lösung.

Einheit 4: Freie Fahrt

Kompetenzbereiche: 1.1, 1.2, 6.1-6.3

Auf den folgenden AB werden alle für Bit verfügbaren Anweisungen vorgestellt. Geben Sie den SuS ausreichend Gelegenheit, eigene Bahnen zu entwerfen.

Auf AB 22 werden Vorschläge gemacht, es wird auf Material zur Weiterarbeit verwiesen und die Fortführung der Arbeit mit Bit und „echten“ Programmieranweisungen in Aussicht gestellt (Material Klasse 4).

zum Kompetenzbereich 6.4:

Klären Sie abschließend, wo Techniken aus der Unterrichtsreihe im Alltag vorkommen:

Roboter, die in der Industrie selbstständig fahren, werden oft ebenfalls durch farbige Linien geleitet. Manchmal sind die Linien auch „unsichtbar“, weil sie aus Metallführungen unter der Fahrbahn stehen, die von Magnetsensoren abgetastet werden. Im Außenbereich wird diese Technik immer öfter durch GPS ersetzt.

Optische Codes kennen die Kinder zum Beispiel in Form von Barcodes im Supermarkt oder in Form von QR-Codes. Barcodes werden an der Kasse über einen Scanner gezogen, das Produkt wird anhand der im Code verschlüsselten Nummer erkannt. Die Kasse kann nun in der Datenbank den Preis und die Produktbezeichnung auslesen, damit rechnen und die Bezeichnung gegebenenfalls auf den Kassenzettel drucken. Ist das Kassensystem mit einem Warenwirtschaftssystem verbunden, kann auch der Verkauf eines Produktes registriert werden. Somit weiß das System immer, wie groß der Warenbestand noch ist und kann rechtzeitig Waren nachbestellen.

QR Codes können Anweisungen oder Daten verschiedener Art enthalten. Häufig enthalten sie eine Internetadresse, auf die der Benutzer beim Scannen weitergeleitet wird. Dies ist besonders häufig bei Werbung der Fall. Kinder, die mit der Anton-App arbeiten, kennen QR-Codes von ihren Benutzerausweisen. Diese können anstelle einer Eingabe der Zugangsdaten gescannt werden.

Klasse 4

Kompetenzbereiche: 1.1, 1.2, 6.1-6.4

In der Unterrichtsreihe für die vierten Klassen wird von der Programmierung über Farbcodes auf Programmierung über Ozoblocky gewechselt. OzoBlockly ist eine spezielle Version der für Kinder entwickelten Block-Programmiersprache „Scratch“, die auf die Nutzung mit Ozobot Bot und Evo abgestimmt wurde.

Die Sprache ist über die Webseite <https://Ozobot.com/OzoBlockly> abrufbar.

Details zur Nutzung werden innerhalb des Arbeitsmaterials geklärt. Ein Internetzugang ist während des Unterrichts unerlässlich.

Unterteilt ist die Unterrichtsreihe für die 4. Klasse wiederum in drei Abschnitte, die verschiedenen Niveaustufen der Programmierung entsprechen. Sie unterscheiden sich in der Anzahl verfügbarer Anweisungen und Parameter, die ihrerseits bestimmen, welche logischen Aufgabenstellungen sich abbilden lassen.

Die Programmierumgebung und die Programmiersprache sind Englisch! Bei der Programmierumgebung fällt dies kaum ins Gewicht. Es müssen evtl. einzelnen Wörter gelernt werden.

Auf einzelne Übungen und deren Lösung wird hier nicht weiter eingegangen, da das Material genau wie das Material für die 3. Klasse die Lösungen und Erklärungen bereits enthält und ebenfalls auch für die eigenständige Erarbeitung geeignet ist.

LuL sollten den jeweiligen Teil komplett durchgearbeitet haben, bevor sie den SuS dieses zur Verfügung stellen!

Stufe 1 (Seiten 1-12):

Die Programmierung wird weitgehend über Anweisungen in Form grafischer Symbole umgesetzt, welche untereinander angeordnet werden. Jedes Symbol entspricht einer Anweisung inklusive evtl. benötigter Parameter: z.B.

Anweisung: Fahre zehn Schritte geradeaus. „Fahre“ ist dabei die eigentliche Anweisung, „zehn Schritte“ und „geradeaus“ stellen die Parameter dar. Beides wird auf dieser Ebene zu folgendem Anweisungssymbol zusammengefasst:



Der erste Teil umfasst zudem eine kurze Einführung in die Programmieroberfläche (Seiten 1-6)

Stufe 2 (Seite 13-28):

Ab der 2. Stufe ersetzen englische Anweisungen die Symbole. Insgesamt reduziert sich die Anzahl der Anweisungen gewaltig, da zu den meisten Anweisungen nun Parameter hinzukommen. Somit kommt die 2. Stufe bereits deutlich näher an „echte“ Programmiersprachen heran:



„Move“ ist die eigentliche Anweisung. Als Parameter können die Richtung, die Entfernung und die Geschwindigkeit eingegeben werden.

Da die Programme nun aufwendiger werden, ist es sinnvoll, diese auch abzuspeichern und wieder laden zu können. Auch dies wird im 2. Teil erarbeitet.

Als wichtigste Neuerung kommen jedoch Schleifen (Loop) hinzu. Programmteile können so wiederholt aufgerufen werden. Dies ermöglicht den eigentlichen Schwerpunkt: Das Erkennen und Gestalten von Algorithmen zur Problemlösung.

Stufe 3 (Seite 29-53)

Hinweis: Diese Stufe ist für den Grundschulbereich ganz klar Fordermaterial! Die Ansprüche an planerische Denken und auch das Niveau der Abstraktion ziehen deutlich an!

Das Material kann optional stärkeren Schülern zur Verfügung gestellt werden. Der Bedarf an Hilfestellungen durch den Lehrer wird bei vielen Schülern steigen. Der dritte Teil kann auch ersatzlos weggelassen werden. Die Kompetenzen des MKR-NRW werden auch durch die beiden ersten Teile der Unterrichtsreihe für die 4. Klasse gut abgedeckt.

Neuerungen auf Stufe 3

Schleifen (Loops) werden nun auch ineinander verschachtelt.

Bedingte Anweisungen werden eingeführt. Linien, Farberkennung und Programmierung in OzoBlockly werden kombiniert. Eigentlich kann man erst ab hier von „Robotik“ reden, da erst jetzt Sensorik und Programmierung kombiniert werden und gemeinsam zu einer Aktion von Bit führen. Zuvor war nur Anhand der Programmierung Bits Handlung weitgehend vorhersehbar und von außen sonst nicht zu beeinflussen. Bit wurde also bisher mehr als „Automat“ und nicht als „Roboter“ eingesetzt.

zum Kompetenzbereich 6.4

Computergesteuerte Anlagen, die mittels Sensoren auf die Umwelt reagieren, gibt es wie Sand am Meer im Alltag:

Einige Beispiele:

- Rasenmäherroboter überfahren die abgesteckten Grenzen nicht und weichen (im besten Fall) Hindernissen aus (oft über Ultraschall- oder Kontaktsensoren).
- Ampelanlagen reagieren auf Kontaktschleifen in der Fahrbahn oder über kleine Kameras am Ampelmast auf die Verkehrslage.
- Drohnen (oft auch als Spielzeug verbreitet) bemerken jede Schräglage und passen die Rotorgeschwindigkeiten blitzschnell so an, dass die Drohne wieder in eine normale Fluglage kommt. Das funktioniert auch dann, wenn man sie anschubst oder sie gegen ein Hindernis fliegen.