

Begleitheft

Die Käfer sind los

- Alles rund um das Programmieren von Robotern -

Prof. Dr. Stephanie Schuler, Dr. Ulrike Dreher

Universität Koblenz-Landau



Deutsche Telekom Stiftung



PRiMA
LERNWERKSTATT

Inhalt

Vorwort	3
1 Das digitale Werkzeug BlueBot	4
1.1 Technische Einführung und Steuerung	4
1.2 Der BlueBot – ein Lernroboter	5
2 Ziele und Kompetenzen.....	6
2.1 Digitale Bildung.....	6
2.2 Algorithmisches Denken.....	8
2.3 Räumliches Vorstellungsvermögen	9
2.4 Prozessbezogene mathematische Kompetenzen.....	11
Problemlösen	11
Darstellen, Kommunizieren und Argumentieren	11
3 Didaktisch-methodische Hinweise	13
Gemeinsamer Beginn	14
01 Befehle einführen und visualisieren.....	16
02 Fahrtwege unterschiedlich darstellen.....	16
03 Algorithmen entwickeln	19
04 Fehler finden und beheben	19
05 Muster erkennen und verallgemeinern	19
06 Probleme lösen.....	20
Gemeinsamer Abschluss	21
Literatur.....	22
Anhang	25
Technische Anleitung zur Verbindung und Steuerung des BlueBot mit der Programmierleiste und der App auf dem Tablet:.....	25
Material für die Lernumgebung	27

Vorwort

Im Rahmen des DigitalPakts Schule „wollen Bund und Länder für eine bessere Ausstattung der Schulen mit digitaler Technik sorgen“ (BMBF, 2019). Außerdem folgt der DigitalPakt dem Grundsatz „Keine Ausstattung ohne Konzept“ (BMBF, 2019). Es obliegt also den Schulen Konzepte zu entwickeln, wie bereits Grundschul Kinder auf die Anforderungen in einer digitalen Welt vorzubereiten sind. Die Lernumgebung *Die Käfer sind los* kann dabei ein Baustein sein. Die programmierbaren Käferroboter haben das Potential, Kindern im Grundschulalter die Grundideen algorithmischen Denkens aufzuzeigen (Angeli et al., 2016, S. 49): durch das unterschiedliche Darstellen von Fahrtwegen in Form von Befehlsfolgen, Wegen im Plan und verbalen Beschreibungen, durch das Wechseln zwischen den unterschiedlichen Darstellungen, durch die Vernetzung dieser Darstellungen, durch das Finden und Korrigieren von Fehlern in diesen Darstellungen und das Erkennen und Nutzen von Mustern in diesen Darstellungen.

Darüber hinaus kann die hier vorgestellte Lernumgebung *Die Käfer sind los* sowohl inhaltsbezogene Kompetenzen, wie das räumliche Vorstellungsvermögen (Leitidee Raum und Form) (TRP Mathematik RLP, 2014, S. 19), als auch prozessbezogene mathematische Kompetenzen wie das Kommunizieren, Argumentieren, Darstellen und Problemlösen fördern (TRP Mathematik RLP, 2014, S. 12).

Im ersten Abschnitt stellen wir das digitale Werkzeug BlueBot, seine Funktionsweise und Steuerungsmöglichkeiten vor. Im zweiten Abschnitt geben wir Ihnen einen Überblick über den fachdidaktischen Hintergrund der Lernumgebung und beschreiben die mit dem Einsatz angestrebten Kompetenzen. Im dritten Abschnitt werden der Aufbau der Lernumgebung und die inhaltlichen Schwerpunktsetzungen erläutert. Im Anhang finden Sie Material- und Kopiervorlagen sowie eine Beschreibung, wie Sie mit dem frei im Netz verfügbaren Programm *GeoGebra* weitere Vorlagen erstellen können.

Das vorliegende didaktische Begleitheft wird durch ein Aufgabenheft ergänzt, das Sie in Ihrem Unterricht entweder ausgedruckt für jede Schülerin und jeden Schüler oder in Form von Stationen einsetzen können.

Wir laden Sie ein, gemeinsam mit Ihren Schülerinnen und Schülern mit dem BlueBot erste Schritte auf dem Weg zum algorithmischen Denken zu gehen.

Viel Freude dabei wünschen

Prof. Dr. Stephanie Schuler und Dr. Ulrike Dreher

1 Das digitale Werkzeug BlueBot

1.1 Technische Einführung und Steuerung

Der BlueBot ist ein programmierbarer Käferroboter mit einer übersichtlichen Anzahl an Funktionen. Auf dem Rücken befinden sich vier Richtungstasten (vorwärts, rückwärts, 90-Grad-Drehung nach links und nach rechts) und die Befehlstasten GO (um eine Fahrt zu starten), X (um Befehlsabfolgen zu löschen) und II (Pause-Taste). Mit Hilfe dieser Programmier Tasten lassen sich einfache Wege eingeben. Erst durch die Bedienung des GO-Befehls führt der BlueBot die Programmierung aus. Somit ist es möglich, dass Befehlsfolgen von den Schülerinnen und Schülern vorab geplant und eingegeben werden können, bevor sich der BlueBot in Bewegung setzt. Eine Vorwärts- bzw. Rückwärtsbewegung entspricht einer Länge von ungefähr 15 cm.



Abbildung 1: Befehlstasten auf dem Blue-Bot (eigene Abb.)

Die Programmierung von Befehlsfolgen ist auch mit dem Zubehör, der Programmierleiste, möglich. Mittels Bluetooth wird der BlueBot mit der Programmierleiste verbunden. Die Steuerung des BlueBots erfolgt über Befehlskarten, die in der gewünschten Reihenfolge in die Programmierleiste gelegt werden. Auch hier werden die Befehle erst ausgeführt, wenn die grüne Taste (der Programmierleiste) gedrückt wird. Einen Vorteil bei der Nutzung der Programmier-



Abbildung 2: Programmierleiste (eigene Abb.)

leiste ist die Visualisierung. Die Befehlsabfolge ist während des gesamten Prozesses sichtbar. Eine Korrektur kann durch den Austausch einzelner Befehlskarten erfolgen. In die Programmierleiste können zehn Befehlskarten gelegt werden, es können jedoch drei Leisten miteinander verbunden werden, sodass sich die Anzahl der Befehlskarten auf 30 erhöht. Zudem kann ein Erweiterungsset für die Befehlskarten erworben werden, sodass Wiederholungen eingebaut werden können. Diese werden allerdings in dieser Handreichung nicht berücksichtigt.

Die Steuerung des BlueBots ist außerdem mit der frei verfügbaren App des Herstellers auf einem Tablet möglich. Die Befehlskarten werden hier untereinander angeordnet und die Darstellung der Befehlskarten hat ein anderes Design, was bei der Einführung und Nutzung berücksichtigt werden muss.

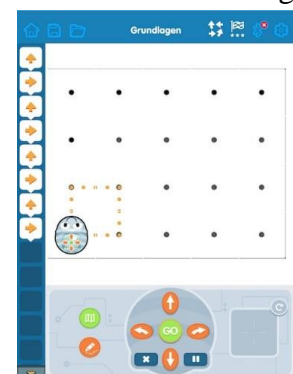


Abbildung 3: Ansicht in der App (eigene Abb.)

Eine technische Anleitung zur Verbindung des BlueBot mit den Programmierleisten und der App befindet sich im Anhang.



Abbildung 4: Setzleiste mit Befehlskarten
(eigene Abb.)

Um die Befehlsfolge bei der Eingabe über die Befehlstasten auf dem Rücken des BlueBot sichtbar zu machen, können die Schülerinnen und Schüler laminierte Befehlskarten in eine Setzleiste stecken, welche die analoge Form einer Programmierleiste darstellt. Um auf dem Rücken des BlueBot getätigte Eingaben zu korrigieren, muss die komplette Befehlsfolge über die Richtungstasten erneut eingegeben werden. Eine Veränderung einzelner Befehle ist nur über die Programmierleiste oder die App möglich.

Um die Fahrtwege nachvollziehen zu können, wird in der Lernumgebung ein quadratisches Punkteraster (4x5 Punkte) verwendet. Im großen Plan entspricht der Abstand zwischen den Punkten der Länge einer Vor- bzw. Rückwärtsbewegung des BlueBots, also 15 cm. Auf diesem Plan legt der BlueBot die programmierten Wege zurück.

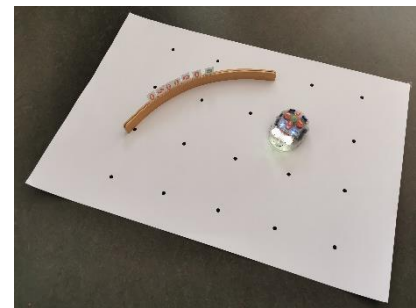


Abbildung 5: BlueBot und Setzleiste auf
einem großen Plan (eigene Abb.)

Verkleinerte Pläne werden zur Visualisierung der Fahrtwege genutzt. Hier können Fahrtwege zeichnerisch festgehalten werden. Die Verwendung eines



Abbildung 6: Kinder zeichnen Fahrtwege in
verkleinerten Plänen ein (eigene Abb.)

Punkterasters hat gegenüber einem Gitterraster (Karo-papier) den Vorteil, dass die Länge der Wegstrecke von Punkt zu Punkt in verkleinerten Plänen durch die Kinder einfach dargestellt und erfasst werden kann und die Startposition eindeutig durch einen Punkt beschrieben ist.

1.2 Der BlueBot – ein Lernroboter

Der BlueBot ist den sogenannten *Educational Robots* (Lernrobotern) zuzuordnen. Dabei kann zwischen *Build Bots* und *Use Bots* unterschieden werden (Catlin et al., 2018, S.6-8). Während *Build Bots* (z. B. Lego Mindstorms, Vex. Fischertechnik; Catlin et al., 2018, S. 7) von den Schülerinnen und Schülern selbst gebaut werden und somit die im Roboter verwendete Technologie fokussiert wird, stellen *Use Bots* (z. B. Bee-Bot, Valiant Turtle; Catlin et al., 2018, S. 8) fertige Roboter dar, mit denen Aufgaben direkt bearbeitet werden können. Erste *Use Bots* wurden bereits in den 1980er Jahren entwickelt und in unterrichtlichen Kontexten eingesetzt.

Papert (1980, S.11) stellte seinen *Turtle Bot*, der über die Sprache LOGO programmiert wird, als ein „object-to-think-with“ vor, also ein Objekt, mit dem das Denken unterstützt wird. *Educational Robots* ermöglichen den Übergang von abstrakten Denkweisen zu konkreten Umsetzungen (Caballero-Gonzalez et al., 2019, S. 1). *Use Bots* wie der BlueBot können bereits bei Schulanfängern und systematisch ab Klasse 2 eingesetzt werden. International konnte in einigen Studien bereits belegt werden, dass der Einsatz von Lernrobotern zur Förderung verschiedener mathematischer und informatischer Kompetenzen im Grundschulalter beitragen kann (Bartolini Bussi & Baccaglini-Frank, 2015; Caballero-Gonzalez et al, 2019; Città et al, 2019; Highfield et al., 2008; Yanik et al., 2017).

2 Ziele und Kompetenzen

2.1 Digitale Bildung

Ausgelöst durch den DigitalPakt Schule (BMBF, 2019) findet aktuell eine verstärkte Auseinandersetzung mit den Möglichkeiten der digitalen Bildung statt. Bereits 2012 gab die Kultusministerkonferenz eine Empfehlung zur Medienbildung in der Schule heraus (KMK, 2012). Darin stellte sie die Medienbildung als Querschnittsaufgabe dar, die es fächerübergreifend sowohl in der Primarstufe als auch in der Sekundarstufe zu fördern gilt. Schulische Medienbildung soll die Schülerinnen und Schüler auf den Umgang mit den neuen Medien in einer digitalisierten Welt vorbereiten, und sie dazu befähigen, selbstbestimmt und reflektiert mit Medien in ihrem Alltag umzugehen (KMK, 2012, S. 3f). Im Jahr 2017 wurden die zu erreichenden Medienkompetenzen in der Strategie „Bildung in der digitalen Welt“ (KMK, 2017) ausformuliert. Dort finden sich Kompetenzbereiche, die in den allgemeinbildenden Schulen Berücksichtigung finden sollen. Unter anderem werden im Kompetenzbereich *Problemlösen und Handeln* Problemlösekompetenzen im Umgang mit digitalen Werkzeugen angestrebt. Dieser Bereich steht bei der Lernumgebung *Die Käfer sind los* im Zentrum und kann durch die bereit gestellten Aufgaben bereits in der Primarstufe bei den Schülerinnen und Schülern angebahnt werden. Es geht darum, die Funktionsweisen digitaler Werkzeuge kennen zu lernen, diese zur Lösung von Problemen zu nutzen und algorithmische Strukturen in den Programmierungen zu erkennen. Dies kommt in der Lernumgebung dann zum Tragen, wenn für den BlueBot beispielsweise ein optimaler Weg (i.S. eines kürzesten Weges) zum Ziel gefunden werden muss, aber auch wenn einzelne Bausteine zu einer Bausteinfolge zusammengefügt werden und diese Folge zur Programmierung eines komplexeren Fahrtwegs genutzt wird.

Auch vom Ministerium für Bildung Rheinland- Pfalz wird der digitalen Bildung ein besonderes Augenmerk durch die Herausgabe der Richtlinie zur Digitalen Bildung in der Primarstufe zuteil (Ministerium RLP, 2018). Digitale Bildung wird dort wie folgt definiert: „Unter digitaler Bildung sind das Wissen und die Kompetenzen zu verstehen, welche die Schülerinnen und Schüler dazu befähigen, eigenverantwortlich an einer medialen und technischen Welt teilzuhaben. Digitale Bildung berücksichtigt das Lehren und Lernen *mit und über* [Hervorhebung d. Verf.] digitale Medien.“ (Ministerium RLP, 2018, S. 1). In der Lernumgebung werden beide Aspekte berücksichtigt:

- Die Schülerinnen und Schüler lernen *über* das digitale Werkzeug BlueBot, indem sie seine Funktionsweise kennenlernen,
- sie lernen aber auch *mit* dem digitalen Werkzeug, indem sie mit Hilfe des BlueBot Probleme lösen z. B. durch das Entwickeln eines einfachen Algorithmus.

In der Richtlinie zur Digitalen Bildung in der Primarstufe (Ministerium RLP, 2018), die den Rahmenplan Grundschule in Rheinland-Pfalz ergänzt, werden unter anderem folgende Kompetenzen im Bereich *Anwenden und Handeln* formuliert, die bis Ende Klasse 4 erreicht werden sollen.

- „Funktionsweisen und grundlegende Strukturen digitaler Werkzeuge kennen und einfache Algorithmen verstehen“,
- „Lösungswege entwickeln, Ergebnisse mit anderen teilen“ (Ministerium RLP, 2018, S. 8)

Es geht konkret darum „durch Versuch und Irrtum zur Lösung einfacher bzw. spielerischer Programmier-Aufgaben (Coding) [zu] kommen“ (Ministerium RLP, 2018, S. 8) und Problemlösestrategien mit dem digitalen Werkzeug anzuwenden wie beispielsweise „etwas [zu] zeichnen oder [zu] bauen und mit dem Computer [oder anderen digitalen Werkzeugen] in Bewegung [zu] bringen und [zu] steuern“ (ebd., S. 8). Unter Coding wird das Aufstellen einfacher Algorithmen in Form von Handlungsabfolgen verstanden. Andererseits geht es bei der Entwicklung von Lösungswegen unter anderem darum, „gemeinsam mit anderen ein Projekt unter Einbeziehung digitaler Medien durch[zuführen“ (Ministerium RLP, 2018, S. 8). Diese Kompetenzen können durch die vorliegende Lernumgebung gefördert werden, da sie dazu anleitet, die Funktionsweise des BlueBot kennen zu lernen, Lösungswege zu gestellten Problemen zu entwickeln, diese zu überprüfen und gemeinsam zu diskutieren. Außerdem wird der BlueBot durch die Programmierung in Bewegung gebracht.

2.2 Algorithmisches Denken

Algorithmisches Denken, im englischsprachigen Raum *Computational Thinking* genannt, umfasst das Erstellen von Algorithmen und das Finden und Korrigieren von Fehlern in (Computer-)Programmen (Wing, 2006). Algorithmisches Denken meint aber auch grundsätzlich das Zerlegen von Problemen des Alltags in Teilprobleme, um sie mit Hilfe eines Computers oder anderen digitalen Werkzeugen lösen zu können (Angeli et al., 2016, S. 49), wobei die Lösungsschritte nicht zwingend in ein Computerprogramm übertragen werden müssen (Sengupta et al., 2018, S. 67). Algorithmisches Denken bzw. Computational Thinking bezieht sich also vorrangig auf die Art des Denkens. Dieses Verständnis knüpft an die ursprüngliche Bedeutung des Begriffs *Algorithmus* an, der „eine endliche, eindeutige Folge von Handlungsanweisungen“ (Knöß, 1989, S. 57) darstellt, die zunächst beschrieben werden muss, bevor sie ausgeführt werden kann und dann im Weiteren automatisiert wird. Angeli et al. (2016) benennen fünf Komponenten des Denkprozesses beim Computational Thinking:

1. *Abstrahieren*: Irrelevante Eigenschaften ausschließen und relevante beibehalten.
2. Umgang mit *Algorithmen*: Aktionen Schritt für Schritt ausführen. Hierzu gehören zwei Unterfertigkeiten: Das Sequenzieren, welches das Planen der richtigen Reihenfolge der Teilschritte umfasst, und der Kontrollfluss, der während und nach dem Planen die Reihenfolge der Teilschritte kontrolliert und damit evaluiert.
3. Umgang mit *Fehlern*: Fehler in einem Algorithmus finden und beheben.
4. *Verallgemeinern von Mustern*: Lösung eines Problems so verallgemeinern, dass sie sich auf andere Probleme übertragen lässt.
5. *Zerlegung von Problemen* in Teilprobleme. (vgl. Angeli et al. 2016, S. 50)

In der vorgestellten Lernumgebung arbeiten die Kinder mit dem BlueBot und lernen zunächst die verschiedenen Befehle sowie unterschiedliche Darstellungen von Fahrtwegen kennen. Sie lernen, wie man Algorithmen, also Befehls- und Bausteinfoolgen entwickelt, wie man Fehler in Befehls- und Bausteinfoolgen findet und diese behebt, wie man Muster in Befehls- und Bausteinfoolgen erkennt und verallgemeinert und komplexe Probleme, durch die Zerlegung in Teilprobleme (z. B. in Befehls- und Bausteinfoolgen), löst. Die Lernumgebung umfasst vornehmlich drei der genannten Komponenten des Computational Thinking: den Umgang mit Algorithmen, den Umgang mit Fehlern und das Verallgemeinern von Mustern.

2.3 Räumliches Vorstellungsvermögen

Die Lernumgebung eignet sich außerdem zur Förderung des räumlichen Vorstellungsvermögens, da die Kinder die Fahrtwege des BlueBot zunächst in der Vorstellung planen, diese dann durch das Zusammenfügen von Befehlen zu Befehlsfolgen programmieren und diese anschließend durch das Starten der Programmierung evaluieren. Bei der Planung und Durchführung des Fahrtwegs lernen die Schülerinnen und Schüler Richtungen und Bewegungen zu beschreiben, aber auch Begriffe wie (Viertel-)Drehung nach links bzw. nach rechts, nach oben, nach unten, nach links/rechts oben/unten, vorwärts und rückwärts kennen und sachgerecht zu verwenden. Die Begriffe unterstützen dabei den Planungsprozess und werden in der Kommunikation über die Lösungswege gemeinsam genutzt.

Ein Blick in die Bildungsstandards für das Fach Mathematik an Grundschulen (KMK, 2004) und den TRP RLP zeigt, dass unter der Leitidee *Raum und Form* das räumliche Vorstellungsvermögen als eine wesentliche Kompetenz gesehen wird. Schülerinnen und Schüler sollen „über räumliches Vorstellungsvermögen verfügen, räumliche Beziehungen erkennen, beschreiben und nutzen (Anordnungen, Wege, Pläne, Ansichten)“ (ebd. S. 10). Weiter sollen „Bewegungen eines Objekts mit Richtung und Entfernung“ (TRP Mathematik RLP, 2014, S. 19) beschrieben werden. Um die Orientierung im Raum geht es auch im Teilrahmenplan Sachunterricht (TRP Sachunterricht RLP, 2006/2015). Unter der Leitperspektive „Umgebungen erkunden und gestalten – Perspektive Raum“ (ebd., S. 26) sollen die Schülerinnen und Schüler „Raumwahrnehmungen nach Lagebeziehungen und Perspektive zunehmend differenzieren“ (ebd., S. 26).

Für das räumliche Vorstellungsvermögen findet man in der Literatur keine einheitliche Definition. Es wird aber stets in mehrere Teilkomponenten unterteilt (z. B. Besuden 1999; Maier, 1999). Maier (1999, S. 14) versteht unter dem räumlichen Vorstellungsvermögen die „Fähigkeit, in der Vorstellung räumlich zu sehen und räumlich zu denken“. Er unterscheidet fünf Komponenten des räumlichen Vorstellungsvermögens: Veranschaulichung, mentale Rotation, räumliche Orientierung, räumliche Beziehungen und räumliche Wahrnehmung (Maier, 1999, S. 50 ff.). Für die Lernumgebung *Die Käfer sind los* sind insbesondere die mentale Rotation und die räumliche Orientierung relevant.

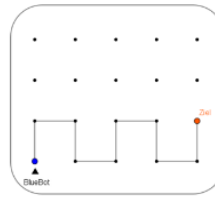
Die *mentale Rotation* wird dann benötigt, wenn im Plan vorgegebene Fahrtwege (s. Abb. 7) programmiert bzw. über Befehlskarten dargestellt werden sollen. Dann müssen die Bewegungen des Roboters (Vorwärts / Drehungen rechts / links um 90°) zunächst mental vollzogen werden, um die passenden Befehle auswählen zu können.

Die *räumliche Orientierung* spielt bei der Programmierung, der Darstellung und bei der Beschreibung von Wegen bzw. Befehlsfolgen

eine Rolle, da hier zwischen rechts und links oder auch vor- und rückwärts unterschieden werden muss. Außerdem muss die Perspektive des Roboters eingenommen werden können, die von der eigenen Perspektive in der Mehrzahl der Fälle abweicht. Die Befehlsfolge ist stets aus der Perspektive des BlueBot zu betrachten und nicht aus der eigenen. Um einen Weg zu programmieren, muss das Kind sich also in die Perspektive des BlueBot hineinversetzen, um die entsprechenden Befehle eingeben zu können, oder es muss die eigene Perspektive zur Perspektive des BlueBot in Beziehung setzen. Das Hineinversetzen ist dann einfach, wenn der Roboter in dieselbe Richtung ausgerichtet ist wie das Sichtfeld des Kindes. Eine Drehung um 90° in die eine oder andere Richtung muss in der Vorstellung vorgenommen werden (wenn der BlueBot zur Seite fährt, entspricht z. B. eine Drehung nach unten je nach Fahrtrichtung des BlueBot einer Links- bzw. Rechtsdrehung– mentale Rotation) oder kann durch Drehen des eigenen Körpers erreicht werden. Am schwierigsten ist der Perspektivwechsel, wenn der BlueBot auf das Kind „zufährt“ (Drehung um 180°), was in den Aufgabenstellungen „Abfahren eines Quadrats“ (Abb. 8) sehr gut deutlich wird: Eine Rechtsdrehung aus der eigenen Perspektive entspricht dann einer Linksdrehung aus der Perspektive des BlueBot. Hier muss aber nicht zwingend ein Perspektivwechsel stattfinden, sondern die Befehlsabfolge kann auch durch Wissen um diese Beziehung programmiert werden.

Aufgabe 6: Girlanden

Hier seht ihr eine Girlande.



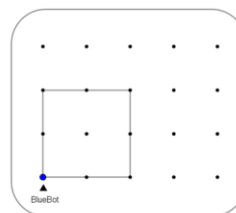
1. Findet einen Baustein und eine Bausteinfole für die Girlande.

2. Überprüft eure Bausteinfole mit dem BlueBot.

Abbildung 7: Beispiel einer Aufgabenstellung (Muster erkennen und verallgemeinern: Aufgabe 6) (eigene Abb.)

Aufgabe 2b

Hier seht ihr zwei Wege im kleinen Plan.



1. Steckt die passende Befehlsfolge mit Befehlskarten in die Setzleiste.
2. Markiert den Start- und den Endpunkt mit einem Klebezettel im großen Plan.
3. Überprüft eure Befehlsfolge mit dem BlueBot.
4. Hier könnt ihr eure Befehlsfolge notieren:

Abbildung 8: Aufgabenstellung mit Quadrat (Befehlsfolgen unterschiedlich darstellen: Aufgabe 2b) (eigene Abb.)

2.4 Prozessbezogene mathematische Kompetenzen

Weiterhin fördert die Lernumgebung prozessbezogene mathematische Kompetenzen, insbesondere das mathematische Problemlösen, das Kommunizieren und Argumentieren und das Darstellen.

Problemlösen

„Problemlösen bezeichnet die Fähigkeit, Lösungen von mathematischen Problemen zu erarbeiten, dabei verschiedene Wege zu erproben und schließlich Lösungsversuche und Lösungsverfahren planen zu können“ (TRP Mathematik RLP, S. 12).

Bei der Bearbeitung der Aufgaben in der Lernumgebung müssen die Lernenden vielfach eigene Lösungswege für gestellte Probleme bzw. Hürden entwickeln, bei welchen sie von einem unsystematischen Probieren zu einem systematischen Probieren gelangen können. Bei der Suche nach dem kürzesten Weg zwischen Start- und Zielpunkt können verschiedene Wege zunächst über einfaches Probieren, also unabhängig voneinander entwickelt werden. Es können aber auch durch das systematische Ersetzen einzelner Befehle weitere Wege entwickelt werden. Überdies haben die Schülerinnen und Schüler die Möglichkeit, Gemeinsamkeiten zwischen Aufgaben, sog. Muster, zu erkennen und zu nutzen, um bereits erstellte Lösungen auf andere Problemstellungen zu übertragen (KMK, 2004, S. 7).

Darstellen, Kommunizieren und Argumentieren

„Die prozessbezogenen *Kompetenzen Darstellen, Kommunizieren und Argumentieren* verweisen darauf, dass Mathematik nicht allein als eine individuelle Befassung zu sehen ist, sondern auch in der Gemeinschaft von Lernenden als Ergebnis gemeinsamer geistiger Konstruktion.“ (TRP Mathematik RLP, 2014, S. 11).

Während es beim Kommunizieren darum geht, „eigene Vorgehensweisen beschreiben, Lösungswege anderer verstehen und gemeinsam reflektieren, [aber auch] mathematische Fachbegriffe und Zeichen sachgerecht verwenden, Aufgaben gemeinsam bearbeiten [...] [und] dokumentieren“ zu können (TRP Mathematik RLP, 2014, S. 11), legt das Argumentieren das Hauptaugenmerk darauf, „mathematische Aussagen hinterfragen und auf Korrektheit prüfen, mathematische Zusammenhänge erkennen und Vermutungen entwickeln [und] Begründungen suchen und nachvollziehen“ zu können (TRP Mathematik RLP, 2014, S. 11).

Durch die Bearbeitung der Aufgaben in Kleingruppen oder Tandems werden die Kinder dazu angehalten, mathematisch zu kommunizieren, also ihre Vorgehensweise zu beschreiben und verschiedene Lösungswege miteinander zu vergleichen und zu reflektieren (KMK, 2004, S. 8).

Weiter lernen sie mathematische Fachbegriffe, wie z. B. eine „Vierteldrehung“, und Zeichen, wie die Pfeilsymbole auf den Befehlskarten, sachgerecht zu verwenden. Das Argumentieren wird dann gefördert, wenn sie Vermutungen aufstellen, diese prüfen und begründen, zwischen den einzelnen Lösungen Zusammenhänge herstellen und Muster erkennen.

Darüber hinaus lernen die Kinder, verschiedene Darstellungen von Fahrtwegen des BlueBot zu erstellen, diese miteinander zu vergleichen und zueinander in Beziehung zu setzen (KMK, 2004, S. 8). Beim Darstellen geht es darum „für das Bearbeiten mathematischer Probleme geeignete Darstellungen entwickeln, auswählen und nutzen, eine Darstellung in eine andere übertragen [und] Darstellungen miteinander vergleichen und bewerten“ zu können (TRP Mathematik RLP, 2014, S. 11).

Bei der Arbeit mit dem BlueBot werden verschiedene Darstellungsebenen nach Bruner (1974) bzw. Bönig (1995) berücksichtigt. Die Schülerinnen und Schüler müssen verschiedene Übersetzungen zwischen den vier Darstellungsebenen – enaktiv, ikonisch, symbolisch und sprachlich – vornehmen.

Zwischen den verschiedenen Darstellungen kann auf vielfältige Weise übersetzt werden. So sind prinzipiell Übersetzung von jeder der vier Darstellungsebenen in jede andere möglich (s. Abb. 9). Je nach Art der Darstellung handelt es sich um einen intermodalen (zwischen verschiedenen Darstellungsebenen) oder um einen intramodalen (innerhalb einer Darstellungsebene) Transfer (Bönig, 1995, S. 60). Die Übergänge zwischen ikonisch und symbolisch sind dabei fließend. Pfeildarstellungen (Abb. 13) ordnen wir der symbolischen Ebene zu, obwohl sie noch Beziehungen zu den Bewegungen des Roboters aufweisen.

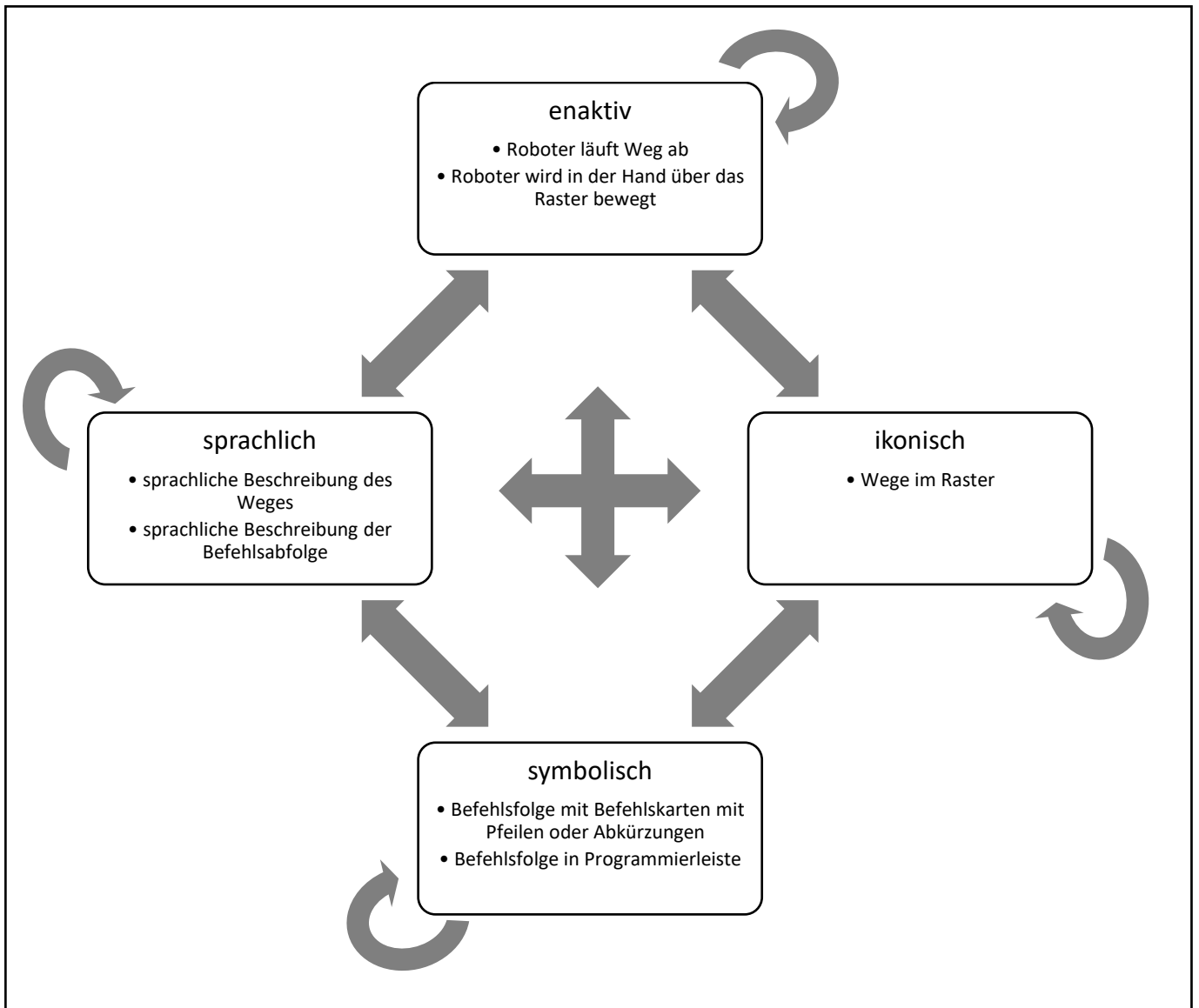


Abbildung 9: Übersetzung zwischen Darstellungsebenen am Beispiel des BlueBot (eigene Darstellung in Anlehnung an Bönig, 1995, S. 60)

3 Didaktisch-methodische Hinweise

Die Lernumgebung *Die Käfer sind los* umfasst sechs Aufgabenmodule:

- 01 Befehle einführen und visualisieren
- 02 Fahrtwege unterschiedlich darstellen
- 03 Algorithmen entwickeln
- 04 Fehler finden und beheben
- 05 Muster erkennen und verallgemeinern
- 06 Probleme lösen

Diese Aufgabenmodule können den Schülerinnen und Schülern in Form eines ausgedruckten Arbeitshefts oder in Form von Aufgabenkarten in wieder beschreibbaren Einsteckhüllen (z. B. Lerntheke, Stationenarbeit) zur Verfügung gestellt werden. Es bietet sich an, die Kinder für die

Bearbeitung der sechs Aufgabenbereiche in Tandems oder Kleingruppen (2-3 Kinder) einzuteilen. Alle Gruppen sollten zunächst die Aufgabenmodule *01 Befehle einführen und visualisieren* und *02 Fahrtwege unterschiedlich darstellen* bearbeiten, da diese grundlegend für die weitere Arbeit sind. Die vier anderen Aufgabenmodule können grundsätzlich in beliebiger Reihenfolge im Anschluss an die beiden ersten Bereiche bearbeitet werden.

Im Folgenden werden Hinweise zur unterrichtlichen Umsetzung der sechs Aufgabenbereiche gegeben, sowie Vorschläge für einen gemeinsamen Beginn und einen gemeinsamen Abschluss gemacht.

Gemeinsamer Beginn

Bevor die Schülerinnen und Schüler die einzelnen Aufgabenbereiche bearbeiten, wird in einem Unterrichtsgespräch, beispielsweise im Stuhlkreis, die Funktionsweise und Verwendung von Robotern thematisiert. Leitfragen könnten sein:

- Was ist ein Roboter? Was zeichnet ihn aus? Was unterscheidet ihn von einem Menschen?
- Welche Beispiele für Roboter kennt ihr?
- Woher weiß der Roboter, was er zu tun hat?

Manche Roboter können sich selbstständig bewegen und übernehmen bestimmte Tätigkeiten wie beispielsweise das Staubsaugen oder das Rasenmähen. Sie übernehmen wiederkehrende, mechanische Arbeiten, oft auch Arbeiten, die für Menschen sehr anstrengend oder gefährlich sind und werden von Computerprogrammen gesteuert, d. h. ein Roboter kann nur Tätigkeiten ausführen, für die er vorab programmiert wurde. Es gibt aber auch ortsfeste Roboter beispielsweise in der Autofertigung, wo sie Menschen dabei unterstützen, schwere Bauteile zusammenzusetzen, zu montieren, zu schweißen, zu kleben oder zu schneiden. Den ersten industriellen Roboter gab es bereits in den 1960er Jahren, also vor fast 60 Jahren, am Fließband eines Autoherstellers. Heute werden Roboter fast überall eingesetzt. Es gibt Medizin-, Haushalts-, Raumfahrt-, Erkundungs- und Spielzeugroboter. Dabei muss ein Roboter nicht unbedingt wie ein Mensch aussehen.

Um die Funktionsweise eines Roboters zu verdeutlichen wird das Roboterspiel *Unplugged* (Kotsopoulos et al. 2017) gespielt:

Dazu ist ein Punkteraster mit Klebestreifen auf dem Fußboden aufgeklebt (Für den Abstand der Punkte eignen sich 30 cm. und die Anzahl sollte 4x5 oder größer sein) (Abb. 10). Ein Kind (Kind 1), welches den Roboter spielen wird, verlässt den Raum. Im Raum bestimmen die übrigen Kinder einen Start- und einen Endpunkt sowie einen Weg vom Start- zum Endpunkt. Sie überlegen, welche Befehle man Kind 1 geben muss, damit es vom Startpunkt aus den Endpunkt erreicht. Die genannten Befehle werden von der Lehrkraft als Pfeile festgehalten (verdeckt, so dass Kind 1 sie beim Ablufen auf



Abbildung 10: Punkteraster 8x7
(eigene Abb.)

dem Punkteraster nicht sehen kann). Es wird ein weiteres Kind (Kind 2) bestimmt, welches die



Abbildung 11: Roboter-Spiel
(eigene Abb.)

Rolle des Programmierers übernimmt. Kind 1 wird zurückgeholt und am Startpunkt platziert (Abb. 11). Kind 2 blickt auf die festgehaltenen Pfeile und diktiert die Befehle Kind 1. Dieses führt diese ohne nachzufragen aus. Kind 2 darf dabei keine weiteren Erklärungen äußern. Es werden an dieser Stelle also bewusst keine Konventionen vorgegeben.

Anschließend wird reflektiert, wie eine „gute“ Programmierung formuliert werden muss, damit Kind 1 den gewünschten Weg abläuft und das Ziel erreicht. Abschließend wird darauf eingegangen, woher der Roboter weiß, was er zu tun hat, also wie er die Befehle des Programmierers versteht. Wichtig ist, dass Programmierer und Roboter die gleiche Sprache sprechen. Programmiersprachen gibt es viele verschiedene und man muss sie lernen. Wie auch in anderen Sprachen, können Fehler passieren. Dann weiß der Roboter nicht, was er tun soll und es kommt zu einer Fehlermeldung (ERROR) oder eine Reaktion bleibt aus. Dem Roboter müssen verschiedene Befehle übermittelt werden, z.B. „Schritt vor“ oder „wenn Hindernis, dann drehen...“. So eine Aneinanderreihung von Befehlen heißt in der Fachsprache Algorithmus. Ein Algorithmus ist eine eindeutige Handlungsvorschrift zur Lösung eines Problems. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. Sie können in einer Programmiersprache aber auch in unserer Alltagssprache verfasst werden. So ist auch ein Rezept oder die Inbetriebnahme eines Geräts ein Algorithmus, also eine feste, klar definierte Abfolge von Handlungsanweisung, die genau in dieser Reihenfolge ausgeführt werden müssen.

01 Befehle einführen und visualisieren

Im ersten Aufgabenmodul (01 Befehle einführen und visualisieren) lernen die Schülerinnen und Schüler die Befehle des digitalen Werkzeugs BlueBot kennen. Insgesamt finden sich auf dem BlueBot sieben verschiedene Befehlstasten, mit denen er gesteuert werden kann: Vorwärts (VW), Rückwärts (RW), Rechtsdrehung (RD), Linksdrehung (LD), Pause, Löschen und die Starttaste. Es handelt sich also um ein sehr einfaches digitales Werkzeug. Da die Befehle als Piktogramme dargestellt sind, können auch bereits sehr junge Kinder Vermutungen zu den Aktionen aufstellen, die der BlueBot beim Drücken der Tasten ausführen wird. Diese Vermutungen sollen verschriftlicht werden und anschließend durch das Drücken der Tasten geprüft werden (01 Befehle einführen und visualisieren, Aufgabe 1 und 2). Die Festigung der Bedeutung der Befehlstasten erfolgt durch ein einfaches Zuordnungsspiel (01 Befehle einführen und visualisieren, Aufgabe 3). Dabei sollen Piktogramm, verbale Beschreibung und Abkürzung einander zugeordnet werden (s. Anhang 2). Alle Karten werden gemischt und auf dem Tisch verteilt. Reihum sollen passende Paare bzw. Tripel gefunden werden. Das Spiel kann auch als Memory gespielt werden.



Abbildung 12: Vermutungen über Befehle überprüfen (eigene Abb.)

02 Fahrtwege unterschiedlich darstellen

Im zweiten Aufgabenmodul (02 Fahrtwege unterschiedlich darstellen) lernen die Kinder die verschiedenen Darstellungen der Fahrtwege des BlueBot kennen:

1. Symbolisch: Eine Befehlsfolge mit Befehlskarten in die Setzleiste stecken.



Abbildung 13: Setzleiste mit Befehlskarten (eigene Abb.)

2. Symbolisch: Eine Befehlsfolge mit Befehlskarten in die Programmierleiste legen.



Abbildung 14: Programmierleiste (eigene Abb.)

3. Ikonisch: Den Weg des BlueBot in einen verkleinerten Plan einzeichnen.

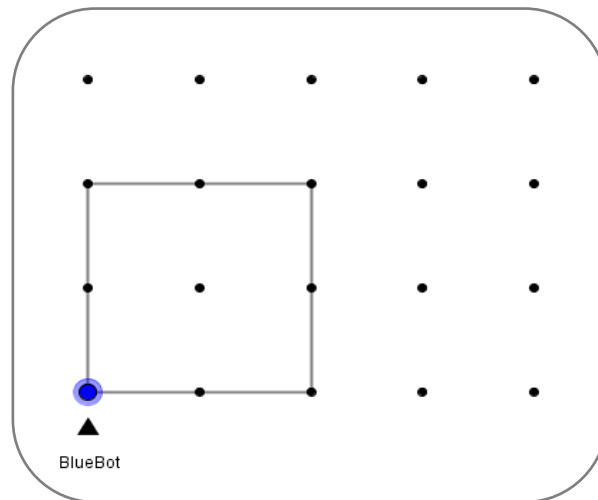


Abbildung 15: Weg eingezeichnet in kleinem Plan (eigene Abb.)

4. Sprachlich/Verbal: Den Weg des BlueBot mit Worten beschreiben.
- a. Mithilfe einer „geometrischen Beschreibung“

Der BlueBot fährt ein Quadrat mit dem Flächeninhalt von vier kleinen Quadraten ab.

Abbildung 16: geometrische Beschreibung (eigene Abb.)

- b. Mithilfe einer „Navi-Beschreibung“

Der BlueBot soll vom Startpunkt aus dreimal vorwärts fahren. Dann soll er sich nach rechts drehen und einmal nach vorne fahren. Danach dreht er sich nach rechts und fährt dreimal vorwärts. Zum Schluss soll er sich wieder nach rechts drehen und einmal nach vorne fahren.

Abbildung 17: Navi Beschreibung (eigene Abb.)

Die insgesamt neun Aufgaben dieses Moduls fördern verschiedene Kompetenzen im Bereich des Darstellungswechsels. Ziel ist es, dass die Kinder flexibel zwischen den verschiedenen Darstellungsarten wechseln können, Übersetzungsprozesse selbstständig durchführen können und die verschiedenen Darstellungen miteinander vernetzen (Prediger & Wessel, 2012, S. 30). Weiterhin lernen die Kinder, vorgegebene Darstellungen einander zuzuordnen, indem sie beispielsweise das Zuordnungsspiel 2 spielen (02 Fahrwege unterschiedlich darstellen, Aufgabe 4). Hier müssen sie erkennen müssen, welche Darstellungen (ikonisch, symbolisch, sprachlich)

zusammenpassen. Fährt der BlueBot die Befehlsfolge ab, wird auf der enaktiven Ebene gearbeitet. Diese wird stets bei der Überprüfung der Programmierung auf Korrektheit angesprochen.

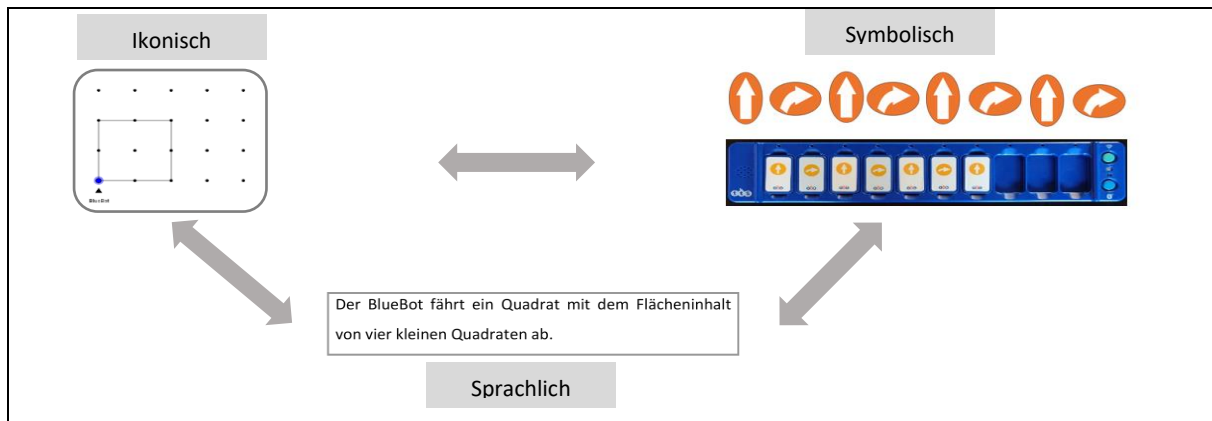


Abbildung 18: Darstellungswechsel (eigene Abb.)

Tabelle 1 gibt Aufschluss darüber, welche Darstellungswechsel mit den Aufgaben in diesem Modul jeweils gefördert werden können. Bei allen Aufgaben wird zum Schluss die Programmierung auf der enaktiven Ebene mit dem BlueBot überprüft.

Aufgabe	Darstellungswechsel	
	von (Ausgangsdarstellung)	zu (Zieldarstellung)
1a	(Symbolisch → Ikonisch)	
	Befehlsfolge	Weg im Plan
1b	(Symbolisch → Ikonisch)	
	Befehlsfolge (Programmierleiste)	Weg im Plan
2a	(Ikonisch → Symbolisch)	
	Weg im Plan	Befehlsfolge (Setzleiste)
2b	(Ikonisch → Symbolisch)	
	Weg im Plan	Befehlsfolge (Setzleiste)
2c	(Ikonisch → Symbolisch)	
	Weg im Plan	Befehlsfolge (Setzleiste)
2d	(Ikonisch → Symbolisch)	
	Weg im Plan	Befehlsfolge (Programmierleiste)
3a	(Sprachlich → Ikonisch → Symbolisch)	
	Verbale Beschreibung	Weg im Plan und Befehlsfolge
3b	(Sprachlich → Ikonisch → Symbolisch)	
	Verbale Beschreibung	Weg im Plan und Befehlsfolge
4	Zuordnungsspiel: symbolische, sprachliche und ikonische Darstellungen werden einander zugeordnet	

Tab. 1: Darstellungswechsel

03 Algorithmen entwickeln

Im dritten Aufgabenmodul (03 Algorithmen entwickeln) lernen die Schülerinnen und Schüler, wie man Befehlsfolgen aus Bausteinen erstellt und dass sich durch die Wiederholung von Bausteinen, auch Wiederholungen und Figuren in den Fahrtwegen ergeben. Werden beispielsweise die Befehle *Vorwärts* und *Rechtsdrehung* zu einem Baustein kombiniert und dieser Baustein viermal wiederholt, ergibt sich ein Quadrat (siehe 03 Algorithmen entwickeln, Aufgabe 1). Dabei werden zunächst zwei (oder drei) Befehle zu einem Baustein gruppiert. Das geschieht in „03 Algorithmen entwickeln, Aufgabe 2“ durch die Aufforderung Befehle mit Tesafilm zusammenzukleben. Bei laminierten Befehlskarten lassen sich diese anschließend wieder trennen. Dieser Baustein wird dann mehrfach hergestellt und aneinandergereiht. Die wiederholte Verwendung eines Bausteins vermittelt den Kindern die Idee des Sequenzierens.



Abbildung 19: Bausteine durch Zusammenkleben herstellen (eigene Abb.)

04 Fehler finden und beheben

Im vierten Aufgabenmodul (04 Fehler finden und beheben) lernen die Schülerinnen und Schüler, wie man Fehler erkennt und den Fehlerort identifiziert. Es sollen Fehler in einer vorgegebenen Befehlsfolge gefunden werden. Dazu muss die Befehlsfolge analysiert und die verschiedenen Darstellungen, wie beispielsweise Befehlsfolge und Weg im verkleinerten Plan, miteinander verglichen werden. So kann der Fehlerort und die Art des Fehlers identifiziert werden und der Fehler korrigiert werden. Dazu müssen Befehle oder ganze Bausteine ausgetauscht oder ergänzt werden.

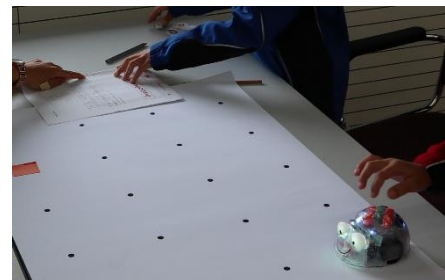


Abbildung 20: Fehler in selbst erstellten Aufgaben finden (eigene Abb.)

05 Muster erkennen und verallgemeinern

Das fünfte Aufgabenmodul (05 Muster erkennen und verallgemeinern) beschäftigt sich mit dem Erkennen und Verallgemeinern von Mustern. Die Schülerinnen und Schüler haben hier die Möglichkeit, sechs Aufgaben zu bearbeiten.

Die Kinder lernen, wie man Muster erkennt, indem sie Gemeinsamkeiten innerhalb von und zwischen Figuren nutzen, um Bausteine und Bausteinfolgen zu erstellen. Beides ist beispielsweise bei der Aufgabe „Treppauf, treppab“ (05 Muster erkennen und verallgemeinern, Aufgabe 5) möglich. Zunächst kann



Abbildung 21: Bausteine und Bausteinfolgen für Quadrate erstellen (eigene Abb.)

ein Muster innerhalb der Treppen-Figur zum Erstellen einer Bausteinfolge für die vorgezeichnete Treppe genutzt werden. Wenn erkannt wird, dass die Treppe aus mehreren gleichen Abschnitten („Stufen“) besteht, kann dieses Muster genutzt werden, um einen Baustein für eine Stufe zu erstellen und diesen mehrfach zu wiederholen. Anschließend sind andere Treppen gefordert. Um deren Bausteinfolgen zu erstellen, können Gemeinsamkeiten und Unterschiede zu vorherigen Treppen erkannt und genutzt werden, um die bereits erstellten Bausteinfolgen lediglich anzupassen anstatt sie neu zu erstellen. Dabei lernen die Kinder außerdem, wie sie verschiedene Probleme auf ein gemeinsam zugrundeliegendes Muster zurückführen können.

06 Probleme lösen

Im sechsten Aufgabenmodul (06 Probleme lösen) werden den Kindern komplexe Problemlöseaufgaben präsentiert. Hierbei geht es um die Bewegung des BlueBot im Punkteraster mit bestimmten Einschränkungen. Beispielsweise soll der BlueBot vom Start zum Ziel so steuern, dass er sich maximal zweimal dreht (06 Probleme lösen, Aufgabe 1) oder maximal 7 Befehlskarten verwendet werden dürfen (06 Probleme lösen, Aufgabe 2). Außerdem werden Aufgaben mit Hindernisparcours bearbeitet (06 Probleme lösen, Aufgabe 3-6). Diese Aufgabenstellungen können bearbeitet werden, indem Problemlösestrategien, wie das Zerlegen in Teilprobleme, die Suche nach Mustern, das Vorwärts- und Rückwärtsarbeiten, das systematische Probieren oder der Darstellungswechsel angewendet werden (Bruder & Collet, 2011).

Während der Darstellungswechsel bereits in dem Modul 02 ausführlich thematisiert wurde, kann das systematische Probieren hier als neue Strategie entwickelt werden (06 Probleme lösen, Aufgabe 1 und 2). Dabei muss bei einer entwickelten Befehlsfolge systematisch einer der Befehle ausgetauscht werden, bis der Fahrtweg korrekt ist. Das Vorwärtsarbeiten ist den Kindern auch durch die anderen Aufgabenformate geläufig, da sie hier meist mit dem Startpunkt in der Planung und Durchführung beginnen. Bei Rückwärtsarbeiten geht es darum rückwärts, d. h. von Ziel aus zu planen. Diese Strategie kann bei Aufgaben mit Hindernissen angewandt werden, wenn die Kinder zunächst vom Ziel aus planen.

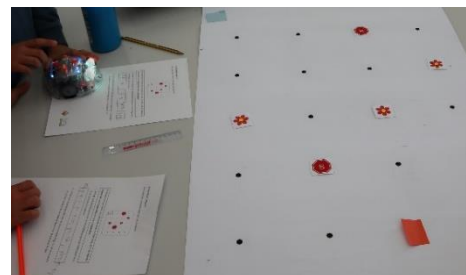


Abbildung 22: Aufgabe mit Hindernisparcours (eigene Abb.)

Eine weitere Strategie stellt das Zerlegen in Teilprobleme dar. Dabei wird nicht der ganze Weg geplant, sondern schrittweise von einem Hindernis zum anderen. Alle drei Strategien sind bei Aufgabe 3, 4, 5 und 6 hilfreich, da Stopp-Schildern ausgewichen und Blumen auf dem Weg zum Ziel eingesammelt werden müssen.

Gemeinsamer Abschluss

In einer gemeinsamen Abschlussrunde wird mit den Kindern erarbeitet, auf was bei einer Programmierung der BlueBot geachtet werden muss. Mögliche Aspekte könnten sein, genau zu wissen was der BlueBot bei welchem Befehl tut oder aus der Perspektive des BlueBot anstatt aus der eigenen zu denken. Auch Aspekte der Gruppenarbeit, wie die Einigung auf einen Fahrtweg oder gegenseitige Unterstützung bei der Eingabe, können an dieser Stelle angesprochen werden. Typische Hürden im Programmierprozess werden besprochen und zusammengeführt. Beispielsweise, dass bei der Programmierung auf dem Rücken des BlueBot Tasten doppelt gedrückt worden sind, oder es zu Verwechslungen zwischen der Drehung nach links/rechts und einem Schritt zur Seite nach links/rechts verwechselt wurden. Zuletzt wird gemeinsam eine Spiegelbildaufgabe bearbeitet. Aufgabe ist es, den gegebenen Weg an einer vorgegebenen Achse zu spiegeln. Hierbei handelt es sich um eine neue Problemstellung, bei der die Kinder das erworbene Wissen anwenden können und gleichzeitig das räumliche Vorstellungsvermögen herausgefordert wird. Durch die Spiegelung, entsteht eine Transferaufgabe. Die Kinder müssen überlegen, welche Befehle bei der Spiegelung erhalten bleiben und welche durch andere ausgetauscht werden müssen.

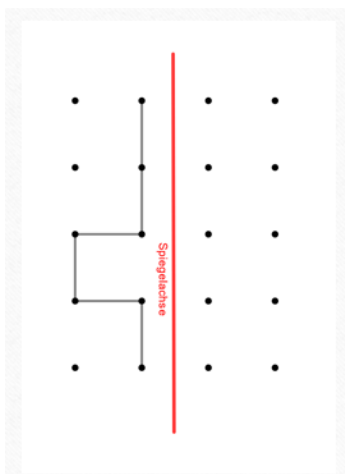


Abbildung 23: Spiegelungsaufgabe
(mit einem Plan) (eigene Abb.)

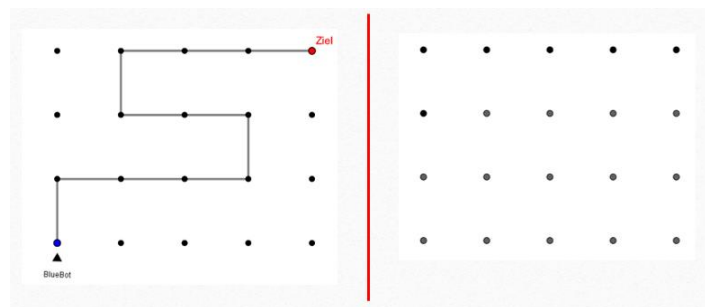


Abbildung 24: Spiegelungsaufgabe (mit zwei Plänen) (eigene Abb.)

Danksagung

In diese Broschüre sind die Erkenntnisse mehrerer Bachelor – und Masterarbeiten eingeflossen. Wir bedanken uns bei den Studierenden Sina Klomann, Melissa Löhle, Kristin Müller, Julia Sauer, Julian Thinner und Katrin Vollmar (alphabetische Reihenfolge).

Literatur

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19 (3), 47–57.
- Bartolini Bussi, M. G. & Baccaglioni-Frank, A. (2015). Geometry in early years: sowing seeds for a mathematical definition of squares and rectangles. *ZDM*, 47(3), 391–405.
- Besuden, H. (1999). Raumvorstellung und Geometrieverständnis. *Mathematische Unterrichtspraxis*, 3, 1-10.
- BMBF – Bundesministerium für Bildung und Forschung (2019). DigitalPakt Schule. Das sollten Sie jetzt wissen. <https://www.bmbf.de/de/wissenswertes-zum-digitalpakt-schule-6496.php> [23.09.2020]
- Bönig, D. (1995). Multiplikation und Division. Empirische Untersuchungen zum Operationsverständnis bei Grundschulern. Münster: Waxmann.
- Bruder, R., & Collet, C. (2011). *Problemlösen lernen im Mathematikunterricht*. Berlin: Cornelsen Scriptor.
- Bruner, J. S. (1974). Entwurf einer Unterrichtstheorie. Buchreihe *Sprache und Lernen*, Bd. 5. Berlin: Berlin-Verlag.
- Caballero-Gonzalez, Y., García-Valcárcel, A., & García-Holgado, A. (2019). Learning computational thinking and social skills development in young children through problem solving with educational robotics. *TEEM '19: Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 19–23. <https://doi.org/10.1145/3362789.3362874>
- Catlin, D., Holmquist, S. & Kandlhofer, M. (2018). *EduRobot Taxonomy: A Provisional Schema for Classifying Educational Robots* [Poster Session], Malta.
- Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., Reale, F. & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, 141, 1-11.

- Highfield, K., Mulligan, J. & Hedberg, J. (2008). Early mathematics learning through exploration with programable toys. In O. Figueras (Hg.), *International group for the psychology of mathematics education: Proceedings of the joint meeting of PME 32 and PME-NA XXX (S. 169–176)*. Cinvestav-UMSNH.
- KMK – Beschlüsse der Kultusministerkonferenz (2004). Bildungsstandards im Fach Mathematik für den Primarbereich. München: Luchterhand.
- KMK – Beschluss der Kultusministerkonferenz (2012). Medienbildung in der Schule. Beschluss vom 8. März 2012.
https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2012/2012_03_08_Medienbildung.pdf [08.09.2020]
- KMK – Beschluss der Kultusministerkonferenz (2017). Strategie der Kultusministerkonferenz „Bildung in der digitalen Welt“. Beschluss vom 8.12.2016, Stand: 09.11.2017
https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2017/Digitalstrategie_KMK_Weiterbildung.pdf [08.09.2020]
- Knöß, Petra (1989). *Fundamentale Ideen der Informatik im Mathematikunterricht*. Wiesbaden: Deutscher UniversitätsVerlag.
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J. & Yiu, C.(2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, 3(2), 154–171.
- Maier, P.H. (1999). Räumliches Vorstellungsvermögen. Ein theoretischer Abriß des Phänomens räumliches Vorstellungsvermögen. Mit didaktischen Hinweisen für den Unterricht (1. Aufl.). Donauwörth: Auer.
- Ministerium RLP – Ministerium für Bildung, Wissenschaft, Weiterbildung und Kultur Rheinland-Pfalz (2018). Richtlinie zur digitalen Bildung in der Primarstufe. Verwaltungsvorschrift des Ministeriums für Bildung vom 30. Oktober 2018. https://grundschule.bildung-rp.de/fileadmin/user_upload/grundschule.bildung-rp.de/Downloads/Aktuelles/Richtlinie_Digitale_Bildung_Primarstufe.pdf [10.08.2020]

- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books. *ResearchGate Link*. (n.d.). Retrieved February 18, 2021, from [https://www.researchgate.net/publication/264043999 Educational Robots and Computational Thinking](https://www.researchgate.net/publication/264043999_Educational_Robots_and_Computational_Thinking)
- Prediger, S., & Wessel, L. (2012). Darstellungen vernetzen – Ansatz zur integrierten Entwicklung von Konzepten und Sprachmitteln. *PM: Praxis der Mathematik in der Schule 54 (Heft 45)*, 28-33.
- Sengupta, P., Dickes, A. C. & Farris, A. V. (2018). Toward a phenomenology of computational thinking in K-12 STEM. In M. S. Khine (Hrsg.), *Computational thinking in STEM discipline: Foundations and research highlights*. Berlin: Springer.
- TRP Mathematik RLP - Ministerium für Bildung, Wissenschaft, Weiterbildung und Kultur Rheinland-Pfalz (2014). Rahmenplan Grundschule. Teilrahmenplan Mathematik. [https://grundschule.bildung-rp.de/fileadmin/user_upload/grundschule.bildung-rp.de/Downloads/Rahmenplan/Rahmenplan Grundschule TRP Mathe 01 08 2015.pdf](https://grundschule.bildung-rp.de/fileadmin/user_upload/grundschule.bildung-rp.de/Downloads/Rahmenplan/Rahmenplan_Grundschule_TRP_Mathe_01_08_2015.pdf) [10.08.2020]
- TRP Sachunterricht RLP - Ministerium für Bildung, Wissenschaft, Weiterbildung und Kultur Rheinland-Pfalz (2006/2015). Rahmenplan Grundschule. Teilrahmenplan Sachunterricht. [https://grundschule.bildung-rp.de/fileadmin/user_upload/grundschule.bildung-rp.de/TRP Sachunterricht f. Bildungsserver 2 .pdf](https://grundschule.bildung-rp.de/fileadmin/user_upload/grundschule.bildung-rp.de/TRP_Sachunterricht_f. Bildungsserver_2_.pdf) [10.08.2020]
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Yanik, B., Kurz, T., & Memis, Y. (2017). Learning from Programming Robots: Gifted Third Graders Explorations in Mathematics Through Problem Solving. In H. Ozcinar, G. Wong, & T. Ozturk (Eds.), *Teaching Computational Thinking in Primary Education* (pp. 230–255). IGI GLOBAL.

Anhang

Technische Anleitung zur Verbindung und Steuerung des BlueBot mit der Programmierleiste und der App auf dem Tablet:

1. Benötigte Programme und Materialien



Abbildung 25: Unterseite des BlueBot (eigene Abb.)

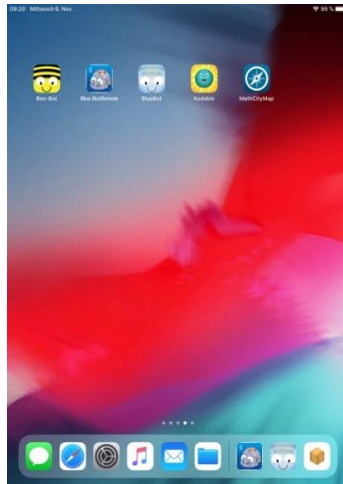


Abbildung 26: App auf Tablet (eigene Abb.)



Abbildung 27: Programmierleiste und Steuerungapp (eigene Abb.)

2. Steuerung über Programmierleiste

Hinweis: Über die Programmierleisten ist keine eindeutige Zuordnung möglich. Es kann nur funktionieren, indem immer ein BlueBot und eine Leiste nacheinander angeschaltet werden.

Verbindungsschritte:

1. BlueBot anschalten (Rückseite)
2. Programmierleiste anschalten (Rückseite)
3. Bluetooth am BlueBot aktivieren (Rückseite)
4. Sobald die Verbindung hergestellt ist leuchten die Augen des BlueBots und das Verbindungslicht blau.



Abbildung 29: Unterseite Programmierleiste (eigene Abb.)



Abbildung 28: zwei verbundene Programmierleisten (eigene Abb.)

Sollen mehrere Programmierleisten aneinandergereiht werden, müssen die Leisten über Kabel miteinander verbunden werden. Erst dann den BlueBot starten. Eine der Leisten symbolisiert über das blaue Licht die hergestellte Verbindung, an dieser Leiste wird dann auch die Befehlsabfolge gestartet.



Abbildung 30: Unterseite BlueBot (eigene Abb.)

Wird keine Verbindung gefunden erneut den BlueBot ausschalten und dann neu verbinden.

3. Steuerung über die App auf dem Tablet

Hinweis: Es ist sinnvoll die BlueBots zu nummerieren und sie in der App mit Nummern anzulegen. Die BlueBots sind beispielsweise unter den Namen BlueBot 1-10 zu finden. So kann nachdem ein BlueBot einer Programmierleiste zugeordnet wurde einem Tablet zugeordnet werden.

1. Öffnen der App (siehe Screenshot 1.)
2. BlueBot auswählen (BlueBot 1, 2, 3, ... oder 10)
(auf dem Bild ist nur ein BlueBot benannt. Sollen die BlueBots **umbenannt** werden, muss man zunächst die Verbindung herstellen und dann über *rename* umbenennen)

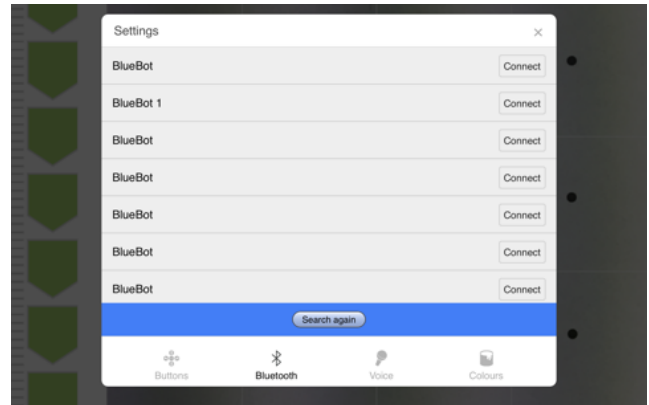


Abbildung 31: BlueBot mit dem Tablet verbinden (eigene Abb.)

Zudem ist in der App das Einfügen von Aufgabenstellungen möglich.

Generell können Aufgabenblätter eingefügt werden, über Dokumente oder Foto

1. Auf die Karte klicken und dann auf das Fotosymbol.
2. Foto machen und Foto in das Raster ziehen.
3. Auf Save klicken.
4. Raster kann unter gespeicherten Dateien gefunden werden (oder über „Karten“-Symbol, ganz unten)

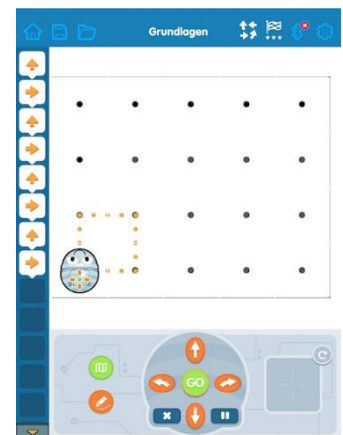






































Abbildung 32: Ansicht in der App (eigene Abb.)

Material für die Lernumgebung








Für jedes Paar/jede Gruppe bzw. jede Aufgabe werden die folgenden Materialien benötigt.

- Arbeitsheft oder Aufgabenkarten in Einsteckhüllen (s. separate Datei)
- BlueBot
- Programmierleiste und dazugehörige Befehlskarten
- ggf. Tablet mit App
- Setzleiste
- Ausgeschnittene und folierte Befehlskarten (s. Anhang 1)
- Großer Plan (separate Datei, zu drucken auf DIN A1 bzw. 92 cm x 62 cm)
- Kleine Pläne (s. Anhang 3)
- Zuordnungsspiel 1 (s. Anhang 2)
- Zuordnungsspiel 2 (s. Anhang 4)
- Stoppschilder und Blumenkarten (s. Anhang 5)
- Tesafilm
- Post-Its (7,6cm x 7,6cm) oder Kreppbandklebepunkte
- Bleistifte
- Radiergummi
- Stoppuhr

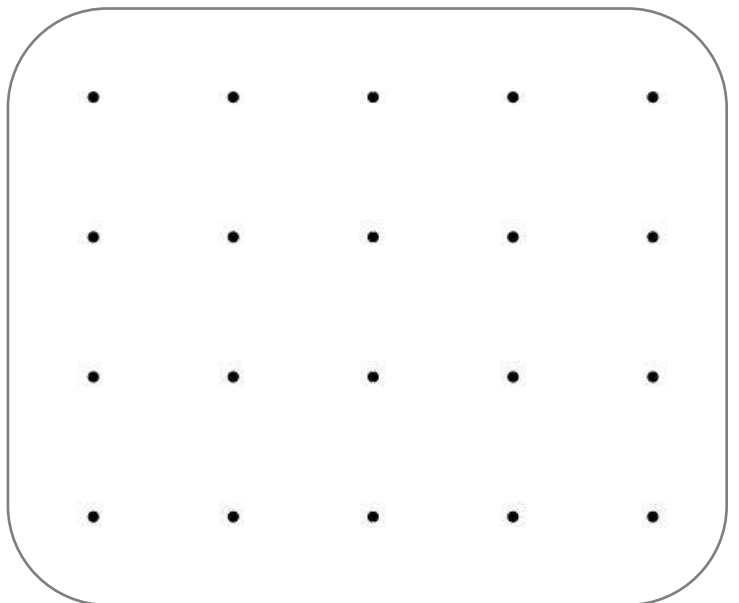
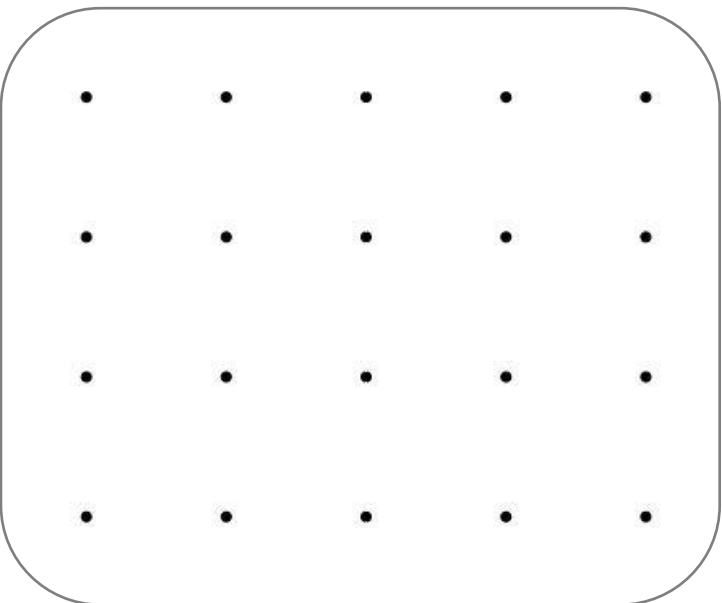
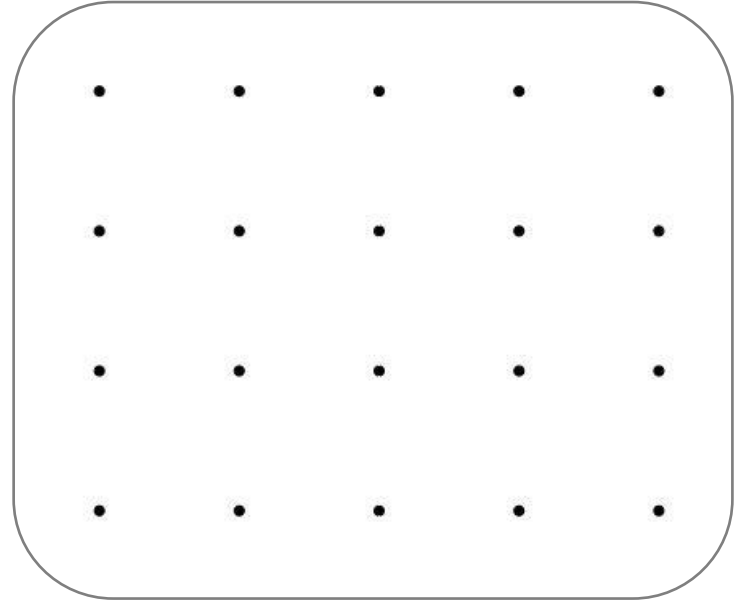
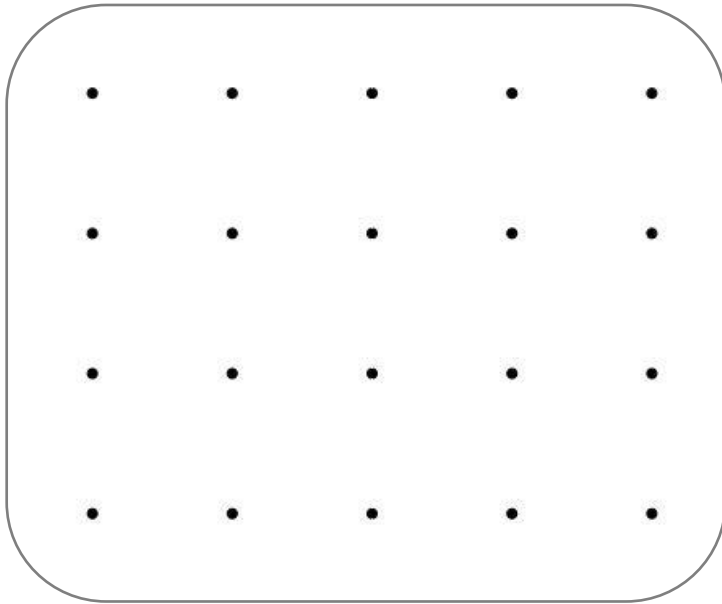
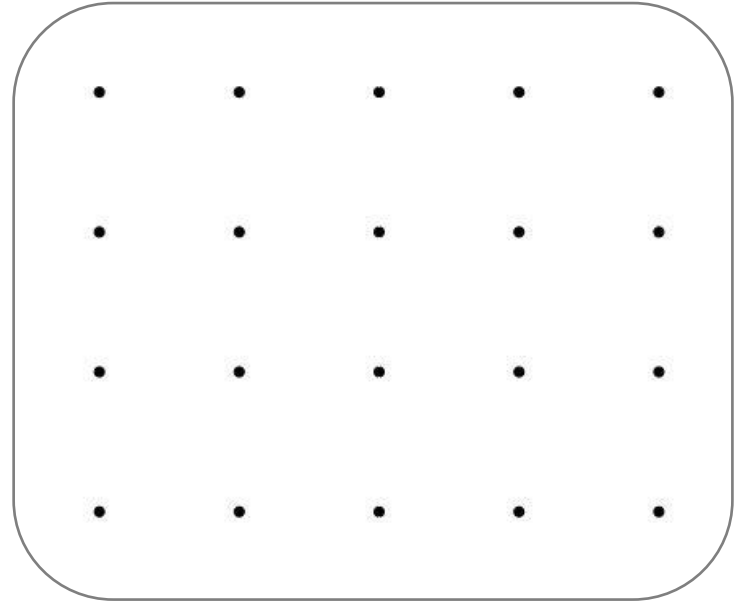
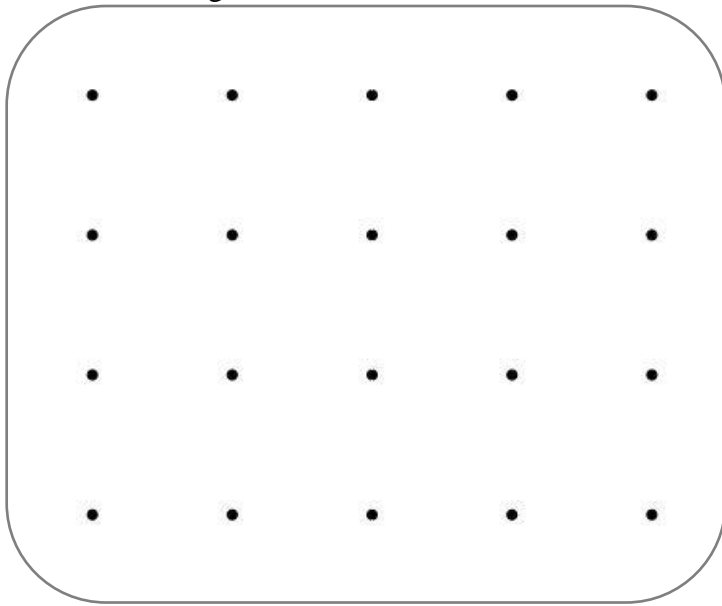
Anhang 1: Ausgeschnittene und folierte Befehlskarten – (2x pro Schülergruppe ausdrucken)

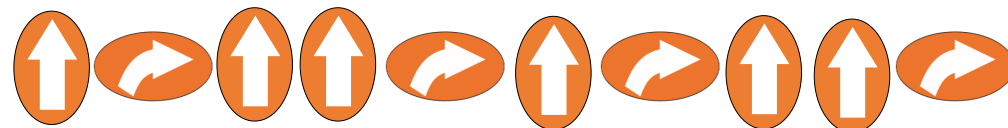
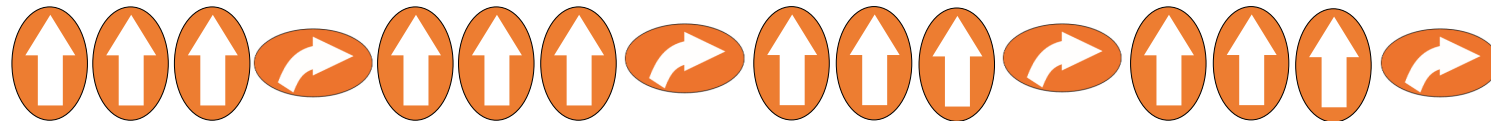
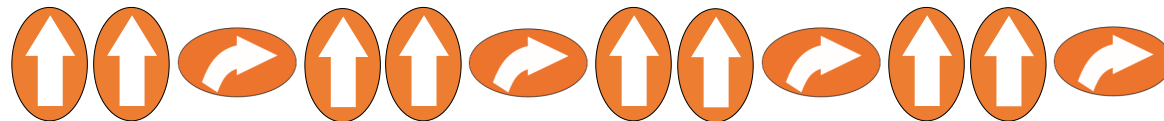
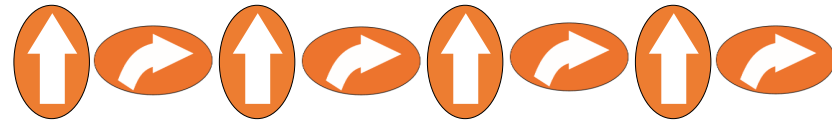
Anhang 2: Zuordnungsspiel 1 (1x pro Schülergruppe ausdrucken und ausschneiden)

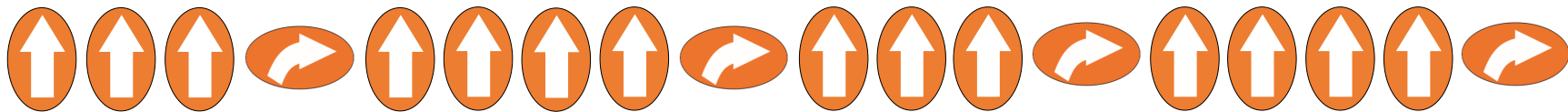
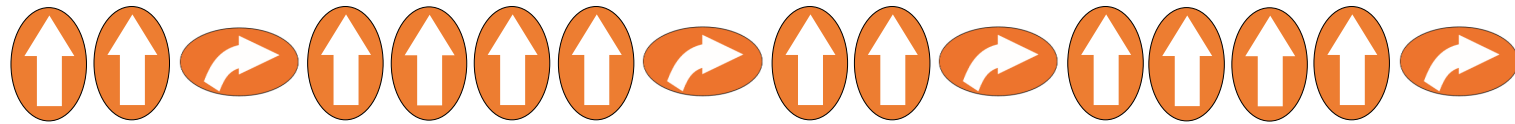
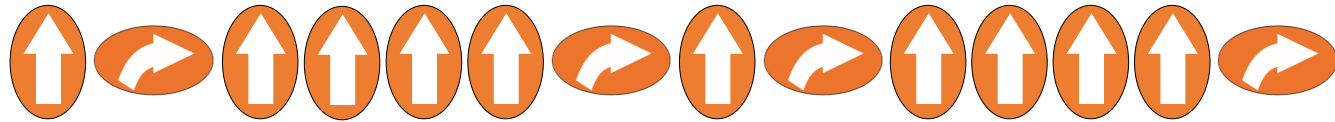
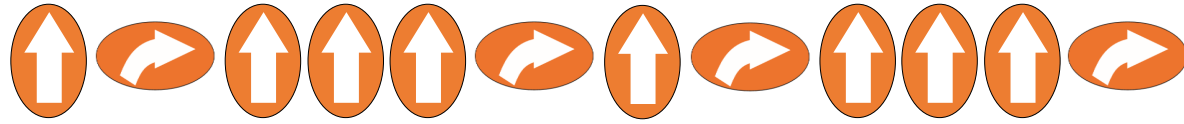
						
Vorwärts	Rückwärts	Links- drehung	Rechts- drehung	Löschen	Pause	Start
VW	RW	LD	RD			

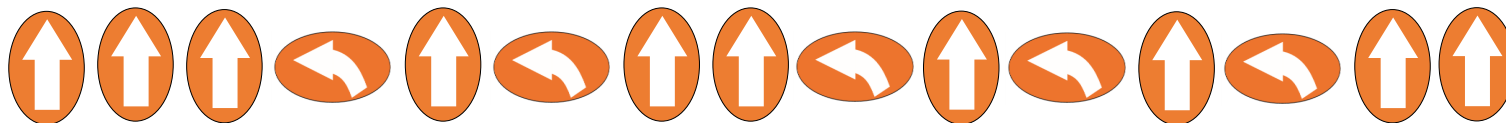
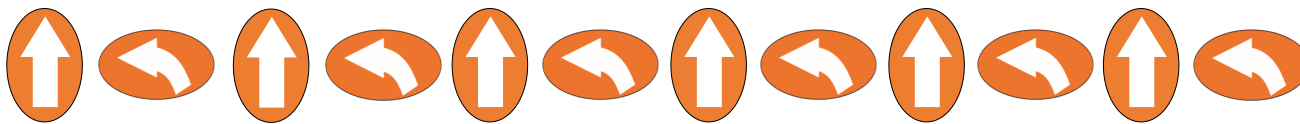
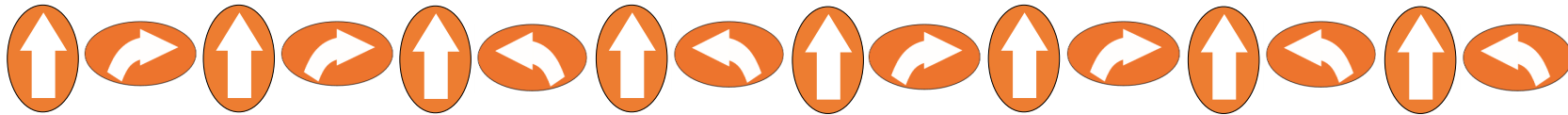
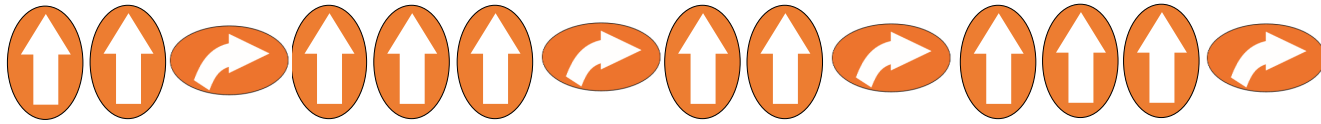
Anhang 3: Verkleinerte Pläne



Anhang 4: Zuordnungsspiel 2 (1x pro Schülergruppe ausdrucken und ausschneiden)







Der BlueBot fährt ein Rechteck, das kein Quadrat ist und den Flächeninhalt von vier Quadraten hat.

Der BlueBot fährt das kleinstmögliche Quadrat ab.

Der BlueBot fährt das kleinstmögliche Rechteck ab, das kein Quadrat ist.

Der BlueBot fährt das Rechteck ab, das zwei Einheiten hoch und vier Einheiten breit ist.

Der BlueBot fährt ein Quadrat mit dem Flächeninhalt von vier Quadraten ab.

Der BlueBot fährt ein Quadrat mit dem Flächeninhalt von neun Quadraten ab.

Der BlueBot fährt so, dass ein Rechteck aus sechs Quadraten entsteht.

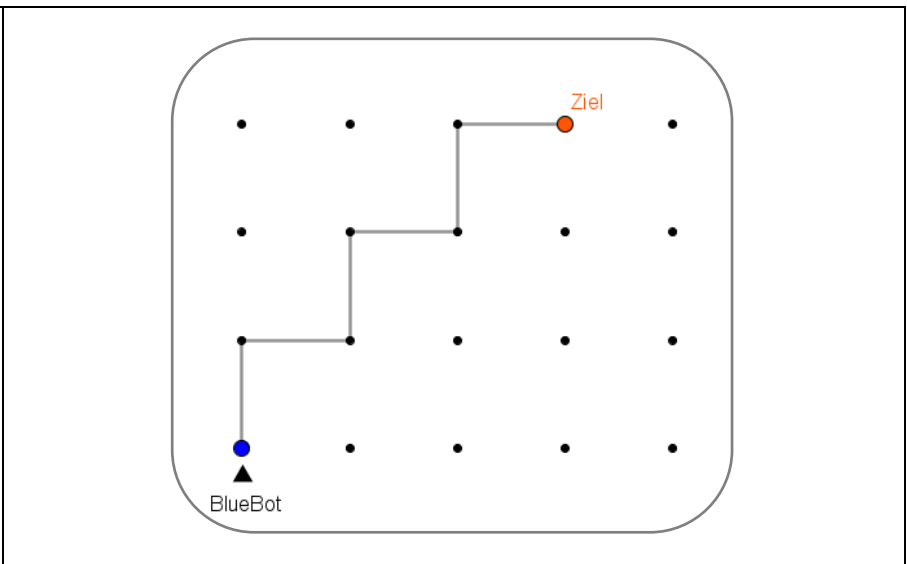
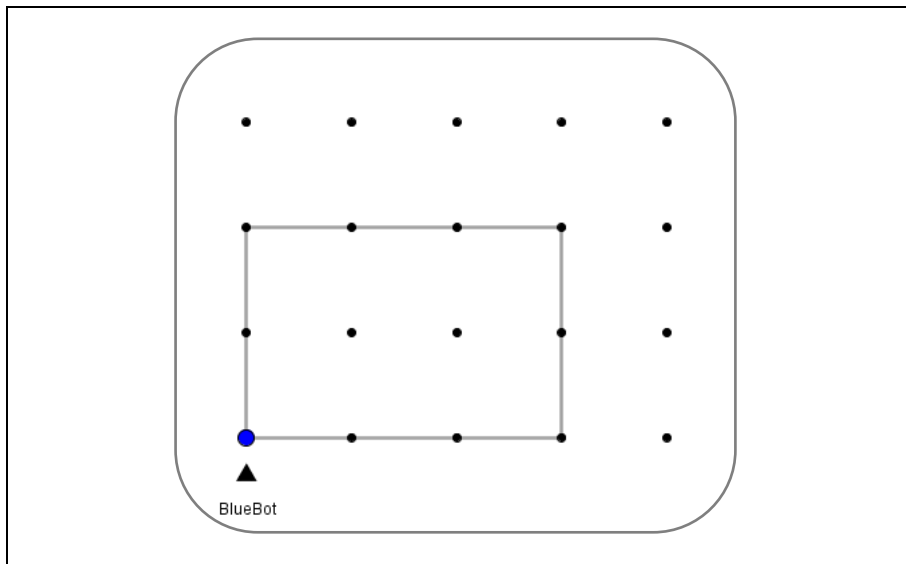
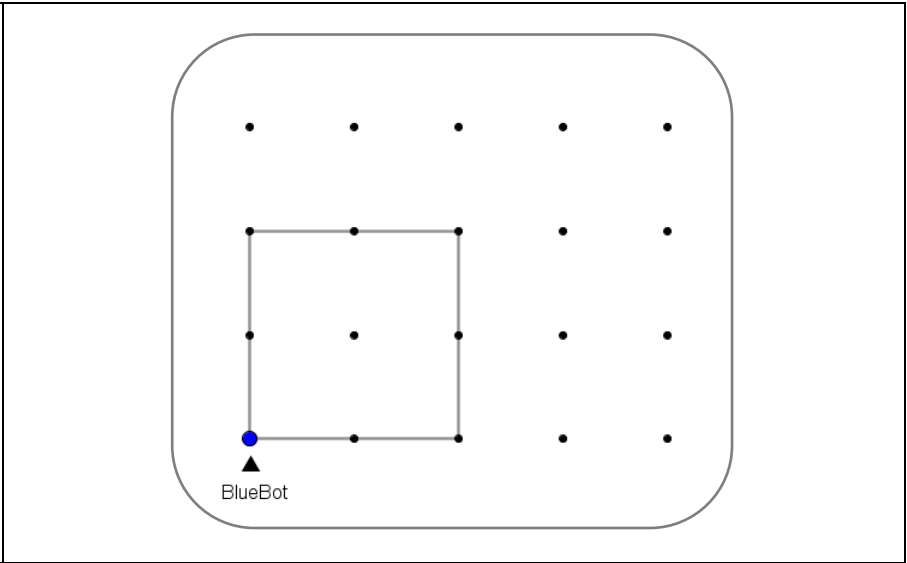
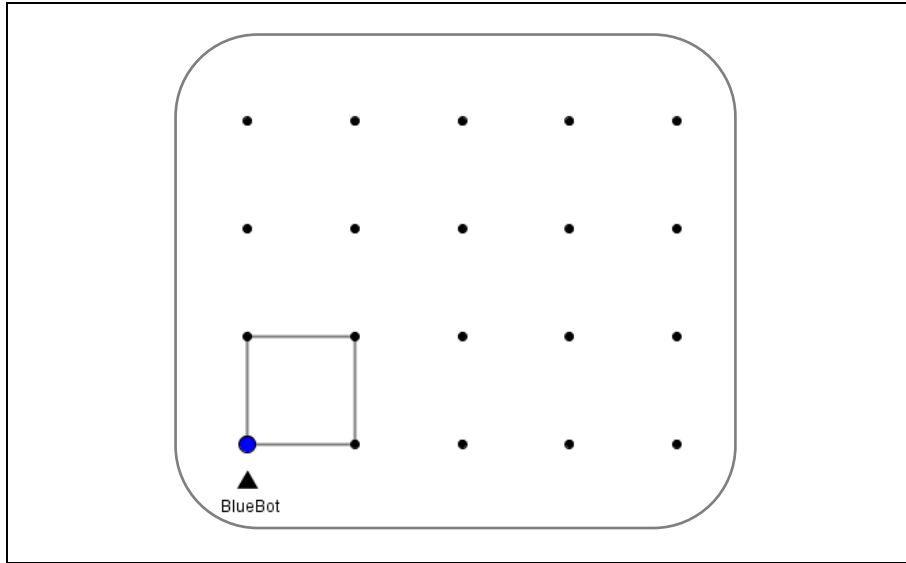
Der BlueBot fährt ein Rechteck aus drei Quadraten ab.

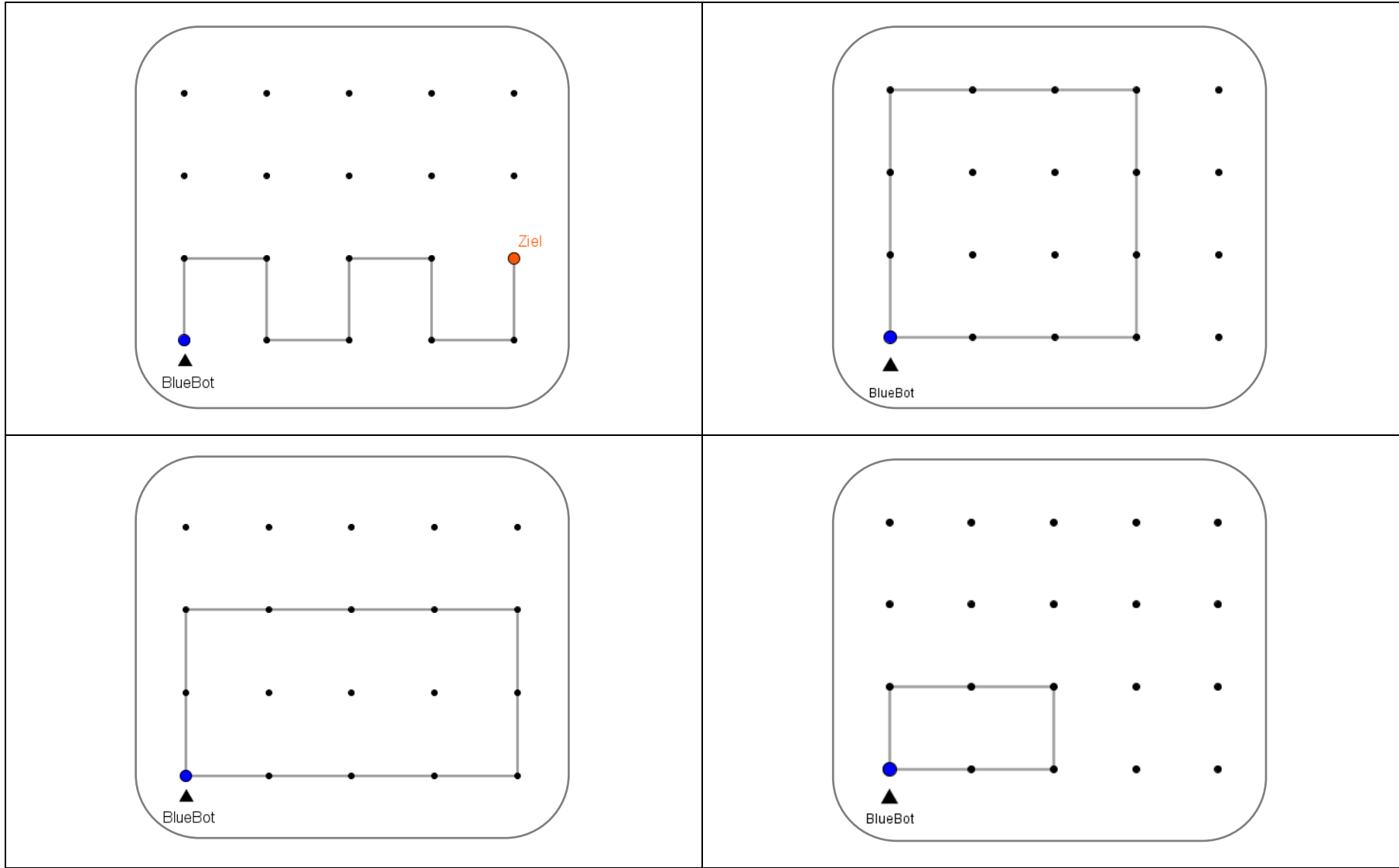
Der BlueBot fährt eine Girlande ab.

Der BlueBot fährt das größtmögliche Rechteck ab.

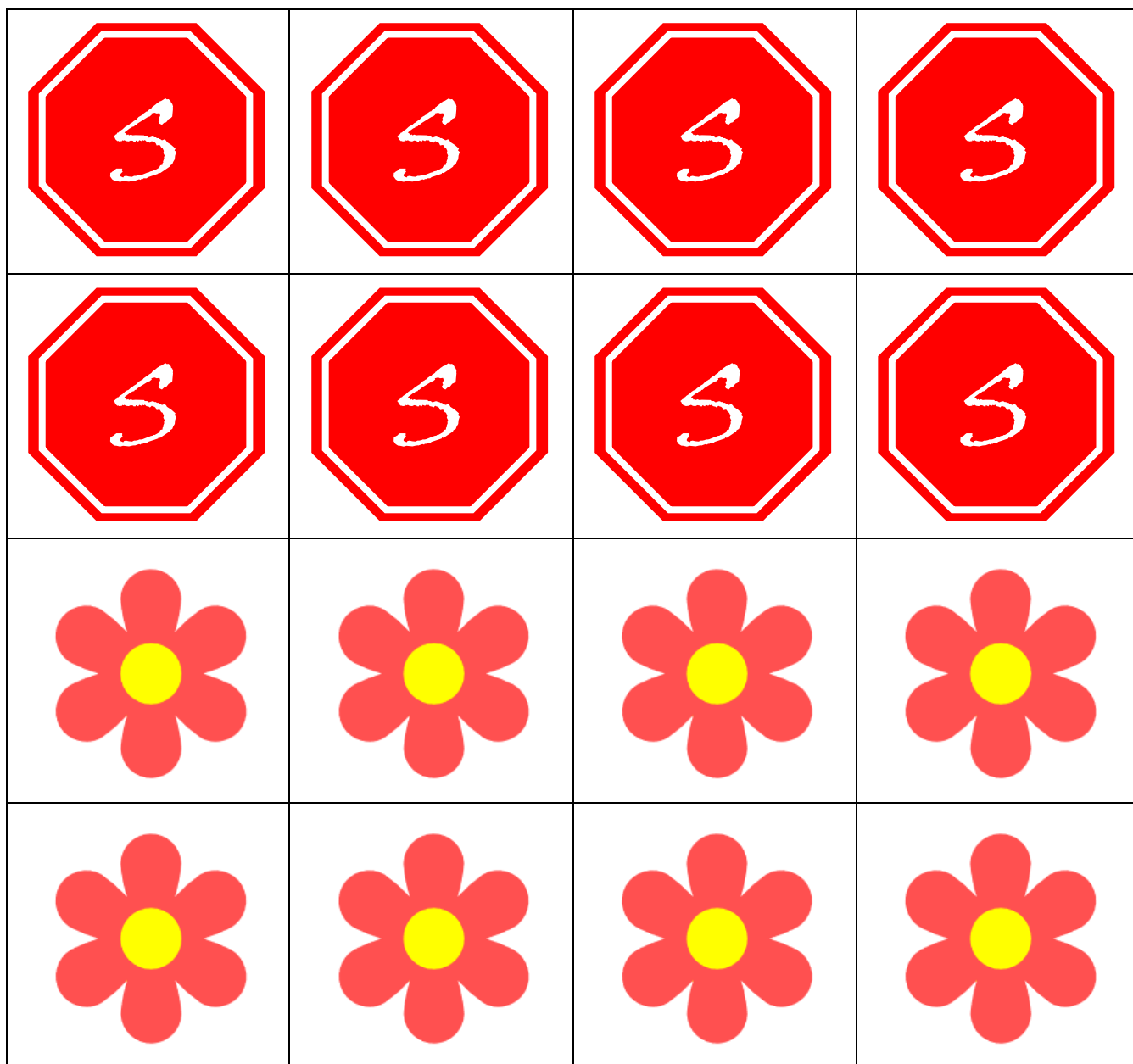
Der BlueBot fährt eine Treppe ab.

Der BlueBot fährt einen Weg, der aussieht wie der Buchstabe L.



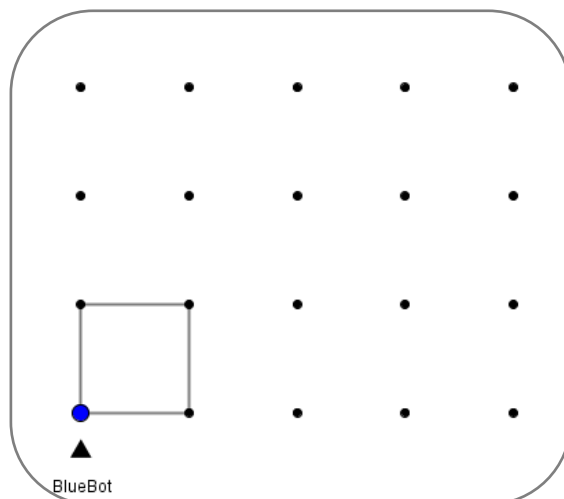


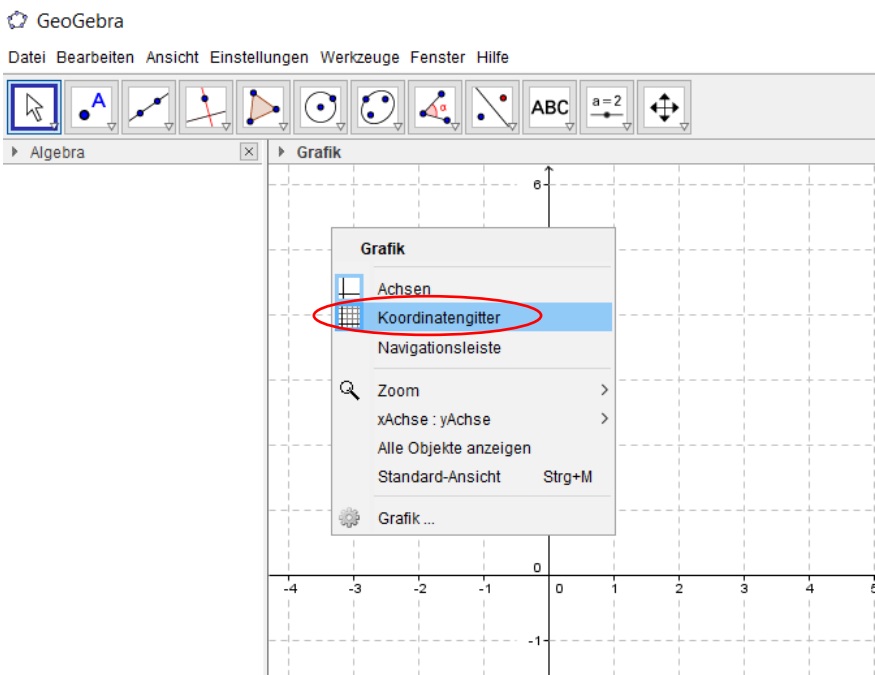
Anhang 5: Stoppschilder und Blumenkarten – (1x pro Schülergruppe ausdrucken)



Anhang 6: Anleitung zum Erstellen von Raster- und Wegevorgaben mit GeoGebra

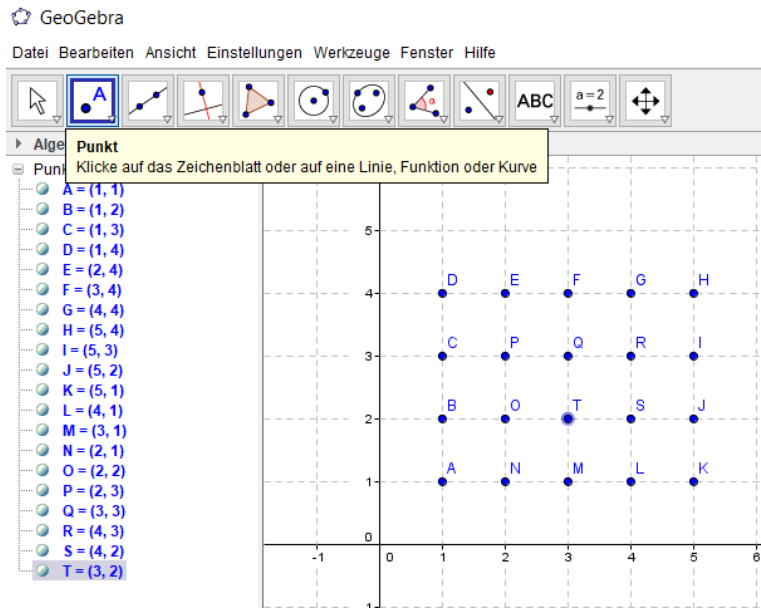
Die verkleinerten Raster und Wege im Raster wurden mithilfe des Programms GeoGebra (<https://www.geogebra.org/>) erstellt. Im Folgenden finden Sie eine Anleitung zum Erstellen von Vorlagen am Beispiel des Quadrates.



Schritt	Erklärung
1. Koordinatengitter sichtbar machen	<p>Es ist hilfreich, in der Programmansicht das Koordinatengitter sichtbar zu machen, da es so einfacher ist, die Punkte zu erstellen. Um das Koordinatengitter sichtbar zu machen, muss mit einem Rechtsklick auf die weiße Fläche geklickt werden und der Reiter <i>Koordinatengitter</i> muss betätigt werden (rotes Oval).</p>  <p>The screenshot shows the GeoGebra software interface. At the top, there is a menu bar with 'Datei', 'Bearbeiten', 'Ansicht', 'Einstellungen', 'Werkzeuge', 'Fenster', and 'Hilfe'. Below the menu bar is a toolbar with various icons for drawing and editing. The main workspace is divided into two panes: 'Algebra' and 'Grafik'. The 'Grafik' pane shows a coordinate system with a grid. A context menu is open over the grid, and the 'Koordinatengitter' option is highlighted with a red oval. The menu items include 'Achsen', 'Koordinatengitter', 'Navigationsleiste', 'Zoom', 'xAchse : yAchse', 'Alle Objekte anzeigen', 'Standard-Ansicht Strg+M', and 'Grafik ...'.</p>

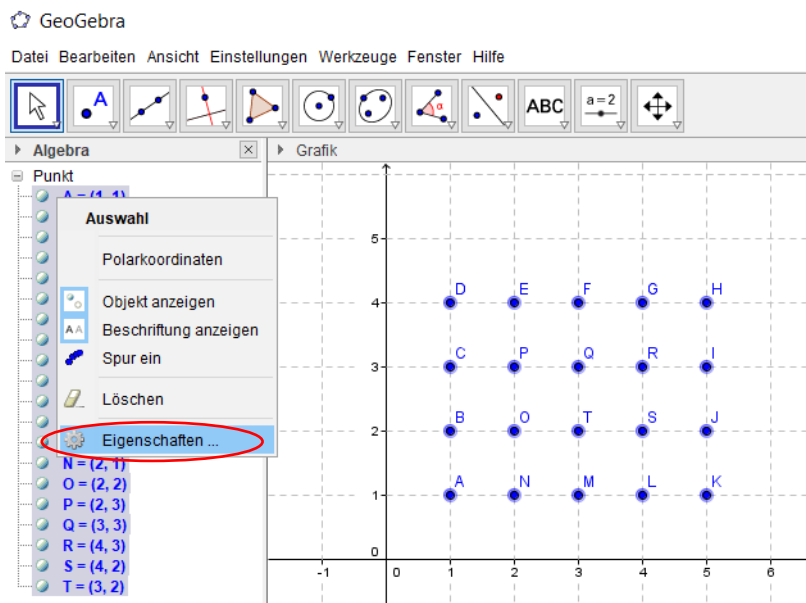
2. Erstellen eines Rasters

Um ein Raster zu erstellen, müssen 20 Punkte (4x5) im Programm eingefügt werden, diese können dann in einem nächsten Schritt formatiert werden (Farbe, Größe, Beschriftung, Form).

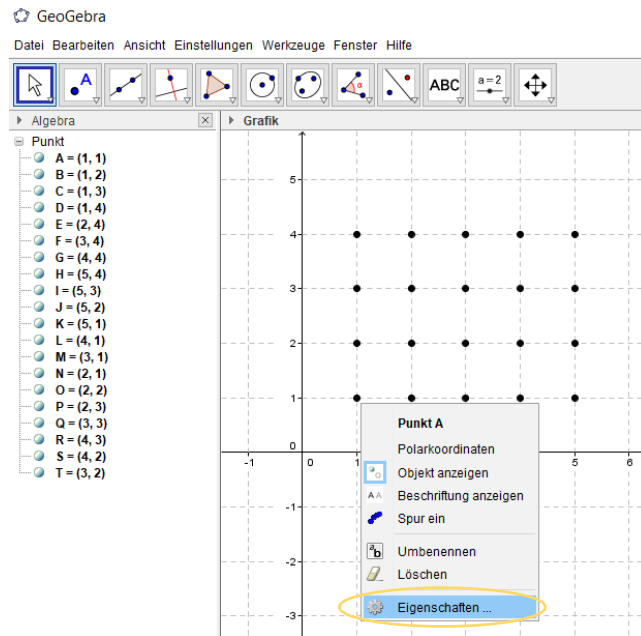


3. Formatierung der Punkte

Um die Punkte zu formatieren, müssen alle Punkte unter dem Reiter *Algebra* markiert werden. Mithilfe eines Rechtsklicks auf die Punkte öffnet sich die Auswahl. Unter dem Reiter *Eigenschaften* können die Punkte entsprechend formatiert werden (rotes Oval). Hier werden die Punkte auf Größe 3 und die Farbe Schwarz gestellt. Die Beschriftung wird ausgestellt.

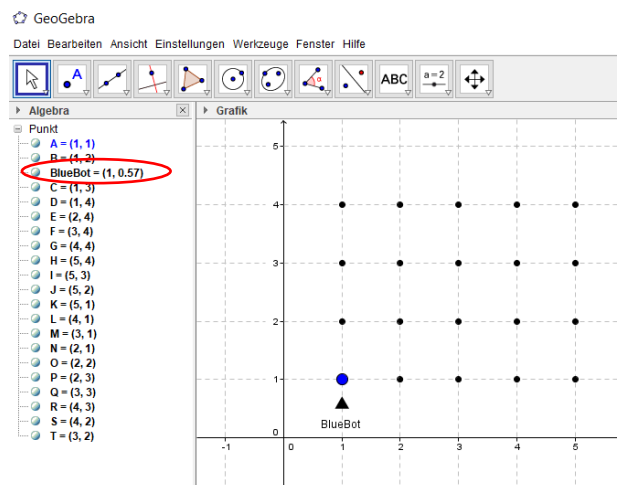


Der Start- und Endpunkt des Quadrates wird auf Größe 6 gestellt und blau formatiert. Die Formatierung kann mit einem Rechtsklick auf den jeweiligen Punkt manuell vorgenommen werden (gelbes Oval). Falls sich der Endpunkt vom Startpunkt unterscheidet, kann dieser ebenfalls entsprechend formatiert werden, indem dieser umbenannt und die Größe und Farbe angepasst wird.



4. Einfügen des Schriftzugs *BlueBot*

Der Schriftzug *BlueBot* und das Dreieck, das dessen Ausrichtung angibt, wird mithilfe eines neuen Punktes eingefügt. Der Punkt wird zu einem Dreieck der Größe 6 und der Farbe Schwarz formatiert (Anleitung zum Formatieren von Punkten siehe Schritt 3). Hierbei ist es wichtig, den Punkt genau unter den Start- und Endpunkt zu setzen und diesen Punkt mit *BlueBot* zu benennen (roter Kreis).

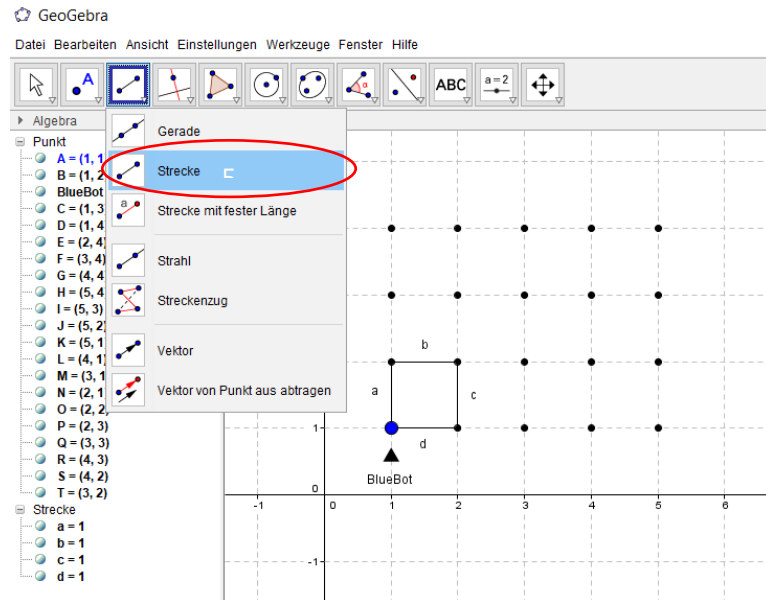


Hinweis: In der linken Spalte können die Punkte manuell verschoben werden, indem die genauen Koordinaten eingegeben werden, dafür genügt ein Doppelklick

auf den jeweiligen Punkt unter der Spalte „Algebra“. So kann der Punkt weiter nach oben versetzt werden (rotes Oval).

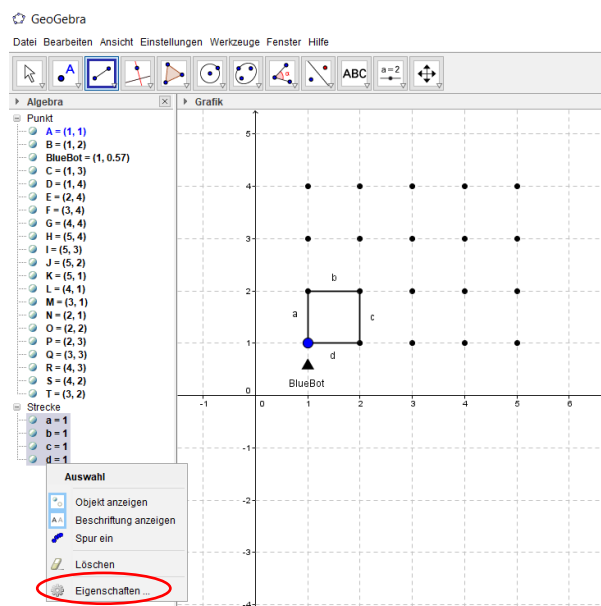
5. Erstellen der Seiten

Im nächsten Schritt werden die Seiten des Quadrats eingefügt. Dafür werden entsprechende Strecken (keine Gerade!) eingefügt (rotes Oval).



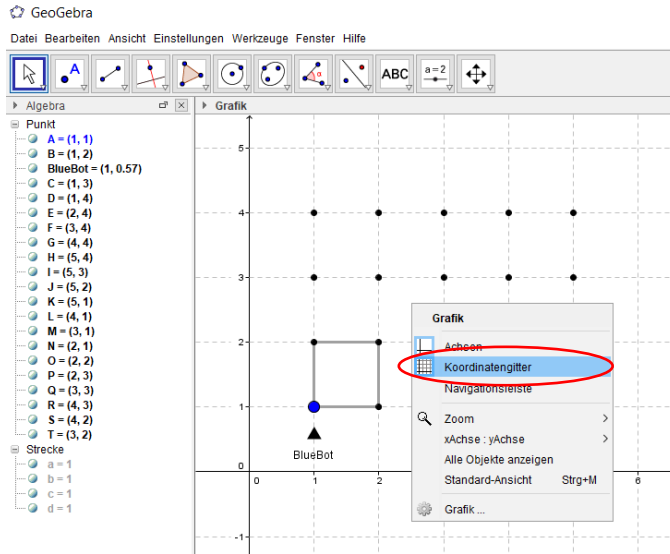
6. Formatierung der Seiten

Um die Seiten zu formatieren müssen die Seiten unter dem Reiter *Algebra* markiert werden. Mithilfe eines Rechtsklicks auf die Seiten öffnet sich die Auswahl. Unter dem Reiter *Eigenschaften* können die Strecken entsprechend formatiert werden (rotes Oval). In diesem Beispiel werden die Beschriftungen der Strecken ausgeschaltet, die Strecken grau gefärbt und die Stärke 6 festgelegt.



7. Ausblenden des Koordinatengitters

Um die Screenshots zu erstellen, muss das Koordinatengitter wieder ausgeblendet werden. Dies gelingt wieder mit einem Rechtsklick auf die Zeichenfläche und dem anschließenden Klicken des Reiters *Koordinatengitter (rotes Oval)*



8. Weiterverarbeitung in Word

Ist der Screenshot erstellt, kann dieser in Word eingefügt werden. Vorher sollte noch eine Form eingefügt werden, welche als Rahmen für den Weg im verkleinerten Plan dient. Hier wurde die Form *Rechteck: abgerundete Ecken* (roter Kreis) eingefügt und entsprechend formatiert, hier Fülleffekt: keine Füllung; Formkontur: grau (rotes Oval). Ist die Form eingefügt, kann der Screenshot in die Form geschoben werden. Fertig ist der Weg im verkleinerten Plan.

