Kommentar Basismodul Segment A

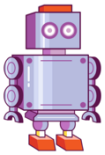
Das Basismodul besteht aus drei Segmenten (A, B, C). Zum erste Segment A werden hier Hinweise gegeben, die Hinweise zu den weiteren Modulen finden sich weiter unten im Text. Im Modul A wird zunächst „unplugged“ – also ohne Computer und digitaler Programmierumgebung – gearbeitet, später in diesem Segment erfolgt dann ein erster Kontakt zur Programmierumgebung *Scratch*.

Unplugged Einführung in Scratch

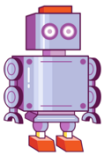
Im Folgenden wird ein erprobter Unterrichtsverlauf dargestellt. Sie können ihn selbstverständlich passend für Ihre Lernszenarien adaptieren.

Material aus *BM_Vorlagen (PDF)*: Groß (z.B. in A3 auf dem Schul-Farbkopierer ausdrucken) ausgedruckte und rechteckig ausgeschnittene Codeblöcke, ausgedruckte Sprechblasen, Bild von Robo (2 Größen zur Auswahl).

1. Das Bild von Robo hängt an der Tafel. Die Lehrkraft stellt Robo vor und erklärt, dass er nur tut, was man im sagt. Aber er versteht nicht alles, was gesagt wird. Er kann nur die Befehle umsetzen, die in seinem Befehlsvorrat angelegt sind.

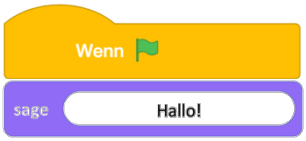


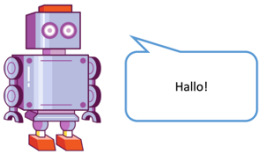
2. Bevor Robo etwas tut, muss man ihm ein Startsignal geben, auf die grüne Fahne tippen – neben Robo wird der Start-Block an die Tafel gehängt.



3. Robo braucht noch Anweisungen, ein <u>Programm</u>, was er tun soll. Eine <u>Anweisung</u>, die er versteht, ist „sage“ – der <u>Block</u> „sage (Hallo!)“ wird dem Tafelbild hinzugefügt (unterstrichene Wörter sind termini technici und sollen verwendet werden).

4. Nun wird das Programm getestet. Ein Kind kommt nach vorne und tippt auf die grüne Fahne. Zeitgleich hält die Lehrkraft die Sprechblase mit „Hallo“ neben Robo und nimmt sie nach 2 Sekunden wieder weg.

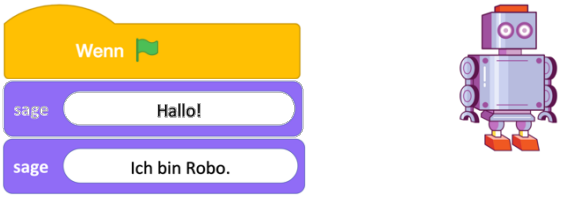




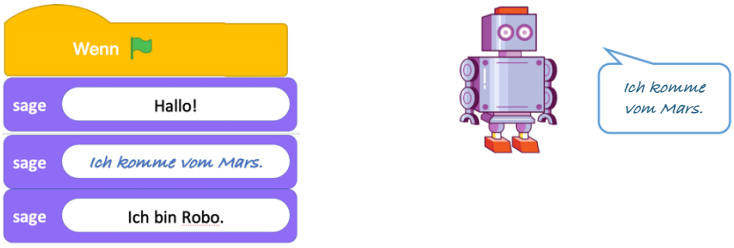
5. Die Lehrkraft informiert: Robo sagt immer das, was im Block drinsteht, man kann es ändern. Er kann auch mehrere Blöcke nacheinander ausführen.

6. Ein zweiter Block „sage“ wird dem Tafelbild hinzugefügt:

CC BY NC SA



7. Nun lassen zwei Kinder das Programm an der Tafel ablaufen: Kind 1 tippt auf die grüne Fahne und zeigt blockweise den Programmfortschritt an – Kind 2 hält passend die beiden vorbereiteten Sprechblasen 2 Sekunden lang neben Robo.
8. Zum Abschluss wird einer der leeren „sage“ Blöcke mit einem Text nach Kinderidee beschriftet. (Beispiel: Ich komme vom Mars.), ebenso passend eine leere Sprechblase. Zudem wird dieser Block zwischen die beiden bereits an der Tafel hängenden „sage“ Blöcke geschoben.



9. Wiederum simulieren zwei Kinder vorne an der Tafel den Programmlauf. Die Lehrkraft macht noch einmal deutlich, dass die Reihenfolge der Blöcke und damit die Reihenfolge der Anweisungen veränderbar ist, ebenso ist der Inhalt beliebig änderbar (um das Missverständnis zu vermeiden, Robo können nur fest im Repertoire verankerte Sätze sprechen).
10. Damit ist die Einführungsstunde beendet.

Individuelle Arbeit mit dem Kurskartensatz A

Material *BM_Kartensatz (PDF)*, darin mit „A“ gekennzeichnete orange umrandete Karten. Bitte einen Kartensatz für jedes Kind als Ausdruck bereitstellen, zur Herstellung siehe *READ ME FIRST (PDF)*. Die Kinder sollen in festen Partnerteams arbeiten, für jedes Team wird ein digitales Endgerät mit Internetzugang benötigt (nicht zeitgleich, verteiltes Arbeiten etwa im Rahmen eines Wochenplans ist möglich).

1. Vor der Arbeit am Kurskartensatz muss die Lehrkraft kurz das Kartensystem erklären:
   a. Jedes Kind arbeitet mit seinem eigenen Kartensatz, jedoch immer zu zweit in Partner-Teams. Zudem arbeitet das Team gemeinsam an einem Gerät, wenn es benötigt wird.
   b. Alle Karten, die als Lernabschnitt zusammengehören, haben die gleiche Farbe (im Segment A orange) und sind fortlaufend nummeriert. Sie sollen nacheinander bearbeitet werden.
   c. Die einzelnen Lernabschnitte bestehen aus einführenden Karten (Pflichtaufgabe mit dem Partnerarbeits-Piktogramm 🗣🗣) und dazwischen weiteren Aufgabenkarte (mit dem Probiere-es-aus-Piktogram 💡). Die einführenden Karten (🗣🗣) sollten alle Kinder bearbeiten, die Probiere-es-aus-Aufgabenkarten (💡) soweit man kommt und so gut es geht.
   d. Zu einigen Karten gibt es eine vorbereitete Lernumgebung in *Scratch*. Wenn die Kinder auf der Karte dazu aufgefordert werden, öffnen sie einen Webbrowser und tippen die Kurz-URL *3yp.de* ein (am PC bitte *in die Adresszeile*, nicht in das häufig vorhandene Suchmaschinen-Suchfeld.) und drücken die Eingabetaste (bzw. tippen auf Go oder → am iPad). Auf Karten, die auf eine Lernumgebung

CC BY NC SA

zugreifen, steht im Text und oben rechts auf der Karte der Code, den man auf der Seite 3yp.de dort dann anklicken muss – es öffnet sich dann die vorbereitete Lernumgebung in einem neuen Tab.

e. Auf manchen Karten sollen oder dürfen beide Kinder etwas schreiben.

2. Nach Ausgabe Kurskarten können die Kinder nun mit der ersten Karte des Kurses A beginnen.
3. **Karte A1 – A3 „Programme“ (🗣🗣):** Auf der Karte A1 wird das Aufrufen der Kurz-URL *3yp.de* eingeführt. Es öffnet sich eine kurze Erklär-Animation in Scratch, welche die wesentlichen Bestandteile der unplugged-Einführungsphase noch einmal kompakt darstellt. Je nach System brauchen die Kinder vielleicht einmalige Hilfe beim Einstellen des Browsers (Cookie Consent, möglichst im Vollbild arbeiten, …). Auf den Karten A2 und A3 können die Kinder dann unplugged ihr Wissen anwenden.
4. **Karte A4 „Programme“ (💡):** Auf dieser ersten Probiere-es-aus-Karte dürfen die Kinder dann noch einmal unplugged selbst kreativ werden. Vielleicht kann es sinnvoll sein, den Unterschied der beiden Kartentypen noch einmal deutlich zu machen.
5. **Karte A5 „Scratch“ (🗣🗣):** Die Kinder kommen hier über *3yp.de* und Klick auf A5 zum ersten Mal mit der Programmierumgebung Scratch in Kontakt. Sie dürfen zunächst beliebig klicken und alles ansehen, um die Erfahrung zu machen, dass man nichts falsch machen kann. Jederzeit kann das Fenster geschlossen und über 3yp.de und Klick auf A5 wieder im Ursprungszustand geöffnet werden. In seltenen Fällen ist die Scratch Programmierumgebung beim Erstaufruf auf Englisch – bitte dann einmal helfen und unter der Weltkugel im Menü „Deutsch“ als Sprache auswählen.
6. **Karte A6 „Scratch“ (💡):** Nun sollen die Kinder der Abbildung von Scratch die passenden Begriffe zuordnen (Karte A5 unten). Erfahrungsgemäß schaffen sie das ohne vorherige Erklärung, rein aus der Logik der Abbildung.
7. Damit ist die Arbeit am Segment A des Basismodules abgeschlossen. Wenn Sie einen Ausblick geben wollen: Im Segment B werden die Kinder selbst Programme in Scratch schreiben lernen.

CC BY NC SA

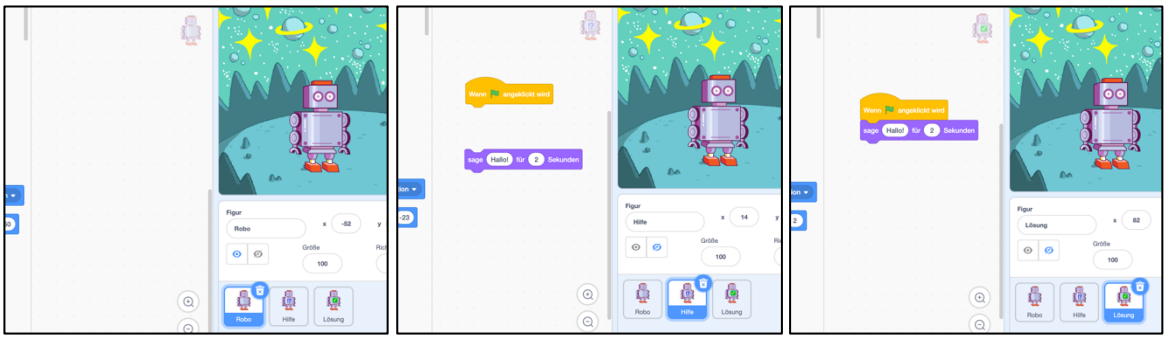
Kommentar Basismodul Segment B

Das Basismodul besteht aus drei Segmenten. Hier wird nun das zweite Segment B beschrieben. Die Arbeit findet in Partnerteams mit dem Kurskartensatz B und in der Programmierumgebung *Scratch* statt.

Individuelle Arbeit mit dem Kurskartensatz B

Material *BM_Kartensatz (PDF)*, darauf der Kartensatz = Kurskartensatz B (rote Umrandung), bitte wieder für jedes Kind als Ausdruck bereitstellen. Zudem wird für jedes Team ein digitales Endgerät mit Internetzugang benötigt. Erneut ist verteiltes Arbeiten möglich.

1. **Karte B1 – B2 „Hallo!“ ( ):** Die Kinder lernen die vorgefertigte Programmierumgebung mit Robo kennen. Es wird die Benutzung des Start- und des „sage“-Blocks eingeführt, als Übertragung aus den unplugged-Elementen der Einführung. Zudem tritt hier zum ersten Mal das dreistufige Hilfesystem auf:



Ausgewählte Figuren: Robo Hilfe Lösung

Hinweis für die Lehrkraft: In der Programmierumgebung Scratch gehört Code im Programmierbereich immer der ausgewählten Figur. Theoretisch können mehrere Figuren Code sogar parallel ausführen. Hier ist diese Funktion genutzt, um verschiedene Stufen der Hilfe zu repräsentieren. Wenn die Kinder eine Programmierumgebung aufrufen, dann ist immer die Figur *Robo* ausgewählt, ggf. finden sich hier auch schon Codeblöcke im Programmierbereich, aber dargestellt ist immer die Arbeitsumgebung, mit der sich die Kinder zunächst beschäftigen sollen. Wenn sie die Figur *Hilfe* (Robo mit Fragezeichen) anklicken, sehen sie dort weitere vorarrangierte Codeblöcke, die zur Lösung führen können, die Lösung selbst aber ist noch nicht dargestellt. Wenn sie die Figur *Lösung* (Robo mit grünem Haken) anklicken, dann sehen sie einen Vorschlag, wie das fertige Programm aussehen könnte. Der Code der Figuren *Hilfe* und *Lösung* ist im Programm ohne Funktion – so sind auch Änderungen dort nicht wirksam und daher auch nicht sinnvoll. *Zum Programmieren müssen die Kinder immer wieder zurück auf die Figur* Robo *gehen und dort weiterarbeiten!* Entscheiden Sie als Lehrkraft selbst, wann und wie sie dieses Hilfesystem den Kindern präsent machen, und wie Sie damit umgehen wollen. Noch ein abschließender Tipp, wenn einmal etwas völlig schiefläuft: über *3yp.de* + Kartencode lässt sich jede zur Karte gehörende Lernumgebung neu laden, die Arbeit muss dann aber von vorne beginnen.

2. **Karte B3 „Robo spricht“ ( ):** In dieser offeneren Aufgabe reproduzieren die Kinder zunächst eigenständig die Ausgabesequenz *„Hallo! | Ich bin Robo. | Es freut mich sehr, dich kennenzulernen.“* ähnlich der unplugged-Einführung. Danach dürfen sie kreativ werden und eigene Ausgabesequenzen ausprobieren,

CC BY NC SA

zudem werden sie aufgefordert, den Unterschied zwischen den Blöcken „sage“ und „sage … für x Sekunden“ und analog bei den „denke“ Blöcken auszuprobieren.

Hinweis für die Lehrkraft: „sage“ ohne Zeitvorgabe lässt die Sprechblase so lange stehen, bis die nächste Ausgabe-Anweisung erfolgt – ggf. auch so schnell, dass es nicht mehr sichtbar ist, wenn mehrere „sage“-Blöcke ohne Zeitangabe hintereinandergeschaltet werden. Mit Zeitangabe dagegen werden sie immer für die Zeitdauer eingeblendet und nach Ablauf der Zeit ausgeblendet.

3. **Karte B4 „FREiWILLIG“ (💡):** Diese Karte ist freiwillig und als Zusatzaufgabe für schnelle Kinder oder die Weiterarbeit zu Hause gedacht. Die Inhalte sind nicht notwendig für Folgearbeiten. Die Kinder lernen in dieser offenen Aufgabe, das Aussehen von Robo („Kostüme“) und des Hintergrundes („Bühnenbilder“) zu verändern. Für Robo stehen drei, für die Bühne acht Varianten zur Verfügung, bereits vorausgewählt aus dem Fundus von Scratch. Auch hier können die Blöcke mehrfach verwendet werden. Neu hinzu kommt ebenfalls der Block „warte (1) Sekunde“. Gegebenenfalls lohnt es sich, einzelne besonders gelungene Bearbeitungen der beiden offenen Aufgabenkarten festzuhalten (siehe dazu „Wie präsentiere ich Lösungen“ unter Tipp und Tricks auf *promaprim.de*). Obwohl diese Karte eine freiwillige Aufgabe ist, halten sich Kinder hier gern sehr lange auf. Geben Sie ggf. einen Impuls zur Weiterarbeit!

4. **Karte B5 – B6 „Robo fragt“ (🗨🗨):** Nun lernen die Kinder die Benutzung des „frage“-Blocks als ersten neuen Blocktyp kennen. Erfahrungsgemäß fällt es einigen Kindern schwer, den Block „Antwort“ zu verstehen – es wird eben nicht das Wort ‚Antwort‘ ausgegeben, sondern der Inhalt der Systemvariable *Antwort* – also das, was als Antwort auf die Frage eingetippt wurde. Darum haben wir den Inhalt der Systemvariable Antwort auch auf der Bühne eingeblendet (Haken neben Antwort im Blockbereich).

5. **Karte B7 – B8 „Robo spricht anders“ (💡):** Hier benutzen wir zum ersten Mal einen selbstdefinierten Block. Er heißt „rede ( )“ und ist in der Blockpalette ganz unten unter den eigenen Blöcken zu finden. Wenn „rede ( )“ benutzt wird, dann werden die Textinhalte sukzessiv in die Sprechblase nachgeschoben, und nicht mehr nacheinander eingeblendet.

   Erklärung für die Lehrkfraft: Der Block hängt die übergebenen Textinhalte an den Inhalt der eigens dazu deklarierten Variable *Redetext* an und gibt den kompletten Variableninhalt aus. Die Deklaration von „rede ( )“ befindet sich ganz unten im Programmfenster, zum Ansehen einfach herunterscrollen. Da Scratch beim Programmstopp Variableninhalte nicht löscht, muss der Inhalt von *Redetext* immer erst gelöscht werden. Dies tut der zweite eigene Block „rede nichts“. Dies alles ist für die Kinder fertig vorprogrammiert– sie brauchen es nicht selbst herzustellen oder zu verstehen. Den Block „rede ( )“ werden wir in den folgenden mathematischen Modulen häufig brauchen, wenn z.B. Schleifen einzelne Ergebnisse liefern und diese in die Sprechblase nachgeschoben werden.

6. Damit ist die Arbeit am Segment B des Basismodules abgeschlossen. Wenn Sie einen Ausblick geben wollen: Im Segment C werden die Kinder eine erste mathematische Problemstellung in Scratch umsetzen.

CC BY NC SA

KOMMENTAR BASISMODUL

SEGMENT C

Das Basismodul besteht aus drei Segmenten. Hier wird nun das dritte Segment C beschrieben. Die Arbeit findet in Partnerteams mit dem Kurskartensatz C und in der Programmierumgebung *Scratch* statt.

INDIVIDUELLE ARBEIT MIT DEM KURSKARTENSATZ C

Material *BM_Kartensatz (PDF)*, daraus die blau umrandete „C"-Karten, wieder bitte für jedes Kind als Ausdruck bereitstellen. Zudem wird für jedes Team ein digitales Endgerät mit Internetzugang benötigt. Auch ist verteiltes Arbeiten erneut möglich.

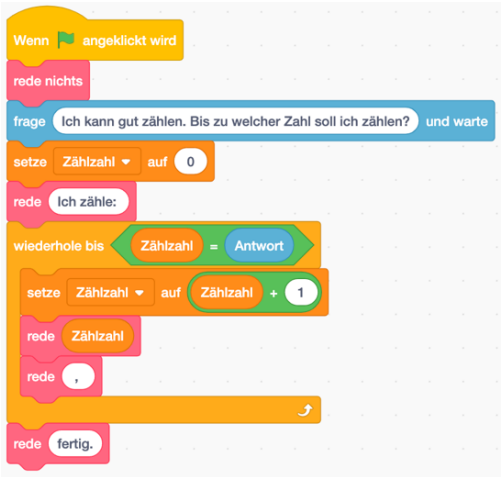
1. **Karte C1 – C2 „Neue Blöcke" (🗨🗨):** Diese Karten thematisieren drei fundamentale Grundideen der Informatik: Sequenz, Schleife und Bedingung. Dazu kommt noch die Einführung von (Programm-)Variablen als Speicherort für Werte. Die Karte C1 ist die einzige Karte des Basismoduls, zu der es zwei Lernumgebungen (C1a und C1b) auf *3yp.de* gibt. Zunächst wird in Aufgabe 1 eine einfache Sequenz – linear aufeinander folgende Programmbefehle – zum Programmieren eines „zählenden" Robos benutzt. Nachdem in Aufg. 2 Variablen eingeführt sind (Hinweis: Der Codeblock „Antwort" ist im Grunde genommen auch eine Variable, die als Wert den im Fragefeld eingegebenen Wert enthält), werden diese in der gleichen Zählsequenz benutzt, um deren Funktion kennen zu lernen. Aufgabe 4 führt dann eine Rechenoperation ein. Karte C2 thematisiert die Idee der Schleife als Abkürzung für immer gleiches sequenzielles Vorgehen. Damit die Schleife nicht endlos läuft, braucht sie eine Abbruchbedingung, die hier ebenfalls eingeführt wird. Ggf. werden die Kinder hier Hilfe benötigen beim Umbau des Programmcodes von C2 zur Schleifenvariante (Tipp: Hilfesystem).

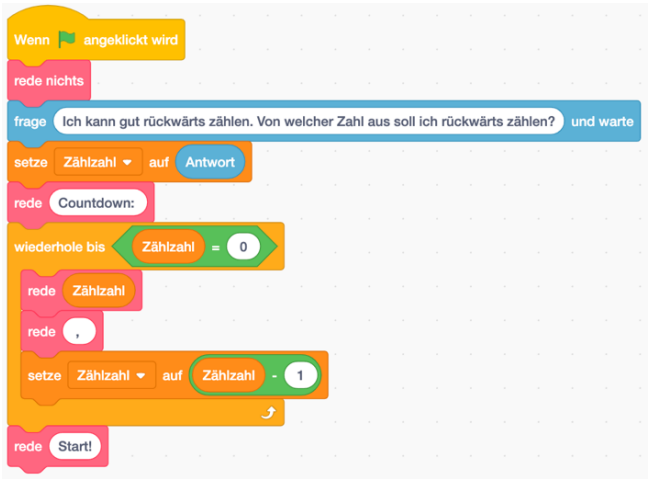
2. **Karte C3 – C4 „Robo zählt" (🗨🗨):** Diese Karten beginnen mit einem unplugged-Element. Die Kinder sollen auf der Karte C3 zunächst unplugged schriftlich von 1 bis 10 zählen und dann von 10 rückwärts bis 1. Die benutzten Start- und Stop-Textsequenzen sind jene, die Robo dann auch später benutzt. In Aufgabe 2 sollen die Kinder nun die Schleife einbauen und mit Blöcken bestücken, so dass Robo vorwärts zählt. Diese Aufgabe reproduziert damit das unplugged-Element aus Aufgabe 1, ist aber prinzipiell auf verschiedene Arten lösbar. Die Kinder werden daher mehrfach testen und optimieren müssen, so dass man hier von konstruktivem, entdeckenden Programmieren sprechen kann. Es ist wichtig, dass die Kinder auch die Variante „bis 15 zählen" testen, um zu überprüfen, ob ihre gefundene Lösung nur für das Zählen bis 10 gilt, oder allgemeingültig ist. Zudem weisen Sie ggf. auf den Tipp hin, innerhalb der Schleife die Reihenfolge der Blöcke zu testen – erfahrungsgemäß haben die Kinder hier Scheu und manipulieren eher die Startbedingungen, wenn ihr Programm falsch zählt. Unter Hilfe und Lösung ist ein Weg zum fertigen Programm zu finden.

   Anmerkung für die Lehrkraft: Die Größe der Sprechblase in Scratch bzw. ihr Inhalt ist limitiert. Daher bricht die Ausgabe ab, wenn Zahlen größer als 80 eingegeben werden. Erfahrungsgemäß probieren die Kinder gern auch sehr große Zahlen – steuern Sie hier ggf. gegen.

   Die Karte C4 dagegen ist erneut konstruktives, entdeckendes Programmieren: Das Programm soll so umgebaut werden, dass Robo rückwärts zählt. Zunächst erfolgt der Rückbezug auf den unplugged-Countdown. Dann sind einige Operationen notwendig, die offensichtlich werden, wenn man beide Programme mit den intendierten Lösungen vergleicht:

CC BY NC SA





Die Kinder müssen immer wieder testen, ob ihre Veränderungen greifen – dazu helfen auch die beiden eingeblendeten Werte von *Antwort* und *Zählzahl* auf der Bühne. Der fertige Code für den Countdown ist diesmal nicht in der Hilfe enthalten, lassen Sie die Kinder ruhig intensiv probieren. Die Folgekarte C5 jedoch beginnt mit dem Countdown als fertiges Programm – falls Kinder keine Lösung finden, sollen sie mit C5 weitermachen.

Anmerkung für die Lehrkraft: Wir hätten auch die Blöcke „ändere (Zählzahl) um (1)“ anstelle von „setze (Zählzahl) auf (Zählzahl +1)“ und „wiederhole (Antwort) mal“ anstelle von „wiederhole bis <Zählzahl = Antwort>“ wählen können. Wir haben uns bewusst für „setze (Zählzahl) auf (Zählzahl + 1)“ entschieden, denn beim Countdown können die Kinder dann grundschulgerecht „setze (Zählzahl) auf (Zählzahl - 1)“ benutzen, dagegen wäre „ändere (Zählzahl) um (- 1)“ noch sehr abstrakt im Grundschulstoff. Die Variante der Schleife mit Abbruchbedingung haben wir gewählt, um eben diese Abbruchbedingung einzuführen (es hätte ja auch „wiederhole ( ) mal“ gereicht). In den späteren mathematischen Modulen werden wir sie des Öfteren brauchen.

3. **Karte C5 und C6 „FREIWILLIG“ (💡):** Diese Karten sind freiwillig und als Zusatzaufgabe für schnelle Kinder oder die Weiterarbeit zu Hause gedacht. Die Inhalte sind nicht notwendig für Folgearbeiten. Bereits auf der ebenfalls freiwilligen Karte B4 im Kartensatz B haben die Kinder gelernt, Kostüme und Bühnenhintergründe zu ändern. Auf C5 können sie nun mit Bewegungen experimentieren. Die Vorgaben dienen als initialer Impuls – der Kreativität sind keine Grenzen gesetzt. Wir haben als eigenen Block „Robo zurücksetzen“ unter den Startblock hineingeschummelt – hier wird alles wieder auf das Ursprungsaussehen zurückgesetzt, egal was die Kinder probieren.

   Anmerkung für die Lehrkraft: Auf die Anwendung der Blockpalette Klang haben wir bewusst verzichtet. Zum einen haben nicht alle Computer in Schulen Lautsprecher, zum anderen haben wir einmal in einem PC-Raum mit 15 Rechnern das Sound-Erlebnis erfahren, das entsteht, wenn Kinder hier auf Entdeckungstour gehen.

   Auf der Karte C6 können die Kinder zum Abschluss nun die Programmierung eines Spiels versuchen. Zunächst einmal wird wieder unplugged wiederholt, was die Zeichen <, > und = bedeuten, immer in Relation zur Zahl 351, denn im vorgefertigten Programm ist Robo auch zunächst 351 Jahre alt. Im Programm müssen die passenden Bedingungen den Steuerungsblöcken „wiederhole, bis < >“ und „wenn < >, dann“ zugeordnet werden. Wenn das Programm funktioniert, also in Relation zu 351 die richtigen Antworten gibt, dann kann es zu einem echten Spiel umgewandelt werden: Robos Alter wird mit dem Operatoren-Block „Zufallszahl von 100 bis 500“ (Zahlenwerte müssen geändert werden) nun wirklich zum Gegenstand des Ratens!

CC BY NC SA

4. **Karte „Urkunde“:** Die Kinder haben das Basismodul Programmieren mit Scratch erfolgreich abgeschlossen. Sie haben sich nun mit allen wichtigen Programmelementen beschäftigt, auf welche die sich nun anschließenden mathematischen Module aufbauen werden. Das ist doch eine kleine Urkunde wert!

5. **Karte „Zu Hause programmieren?“:** Hier geben wir ein paar Hinweise zum Programmieren zu Hause. Diese Karte können Sie auch vorab ausgeben, wenn zum ersten Mal die Frage aufkommt, ob die Kinder auch zu Hause an den Inhalten arbeiten dürfen.

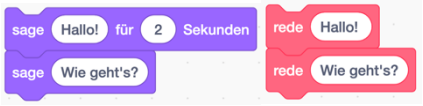
Gemeinsame Abschlussreflexion des Basismoduls

Abschließend ist es Zeit für eine gemeinsame Abschlussreflexion des Basismoduls. Wir fokussieren hier auf wichtige Blöcke in Scratch, weniger auf die Inhalte, die damit bearbeitet wurden. In den sich anschließenden mathematischen Modulen wird dagegen vor allem der Blick auf die mathematischen Inhalte gelenkt werden.

Material aus *BM_Vorlagen (PDF)*: Ausgedruckte und zugeschnittene Codeblöcke in groß.

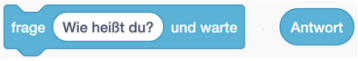
1. Es wurden vier große Karten mit unterschiedlichen Blockarten zur Reflexion ausgewählt. Es bietet sich an, diese Kindern anzubieten, die bereits früh fertig mit dem Basiskurs sind. Ein Partnerteam könnte sich dann darauf vorbereiten, eine dieser Karten vor der Klasse noch einmal vorzustellen. Die anderen Kinder können dann immer noch Fragen stellen, Ergänzungen anbieten, und so weiter. Die Angebote sind von unterschiedlichem Schwierigkeitsgrad.

2. Codeblöcke:



   Hier kann dargestellt werden:
   - Unterschied zwischen den beiden „sage“ Blöcken (Dauer der Sprechblase)
   - Unterschied zwischen „sage“ und „rede“. (Erscheinungsform des Textes in der Sprechblase)

3. Codeblöcke:



   Hier kann dargestellt werden:
   - Unterschied zwischen „sage“ und „frage“ Blöcken (Warten auf eine Antwort)
   - Funktion der Systemvariable *Antwort* (beinhaltet die eingegeben Antwort, nicht das Wort ‚Antwort‘)

4. Codeblöcke:

   Hier kann dargestellt werden:
   - Was passiert in der Schleife? (Es wird immer wieder gefragt und eine Antwort erwartet)
   - Wann wird die Schleife verlassen? (wenn 351 eingegeben wurde, also 351 in *Antwort* steckt)

5. Codeblöcke:

CC BY NC SA

Hier kann dargestellt werden:
- Was ist eine Variable (sie beinhaltet den zugewiesenen Wert)
- Was passiert beim Rechnen mit Variablen (Wert -> Rechnung -> Ergebnis wird wieder zugewiesen)

Mit dem Ende der Reflexionsphase ist die Arbeit am Basismodul abgeschlossen. Es können sich nun mathematische Module anschließen.

HINWEISE ZUM COPYRIGHT

Dieser Kurs stellt eine mathematikdidaktische Adaption des Kurses von Katharina Geldreich, Didaktik der Informatik, Technische Universität München dar, der unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen inkl. remixen, verändern und darauf aufbauen 4.0 international) unter

Geldreich, K. (o. J.): Programmierzirkus - Professur für Didaktik der Informatik. Abgerufen 8. Oktober 2020, von https://www.edu.tum.de/ddi/forschung/laufende-projekte/programmierzirkus/

veröffentlich ist. Einige Materialien enthalten Screenshot von Scratch/CC-BY-SA-4.0 Creative Commons. Scratch ist ein Projekt der Scratch Foundation und der Lifelong Kindergarten Group am MIT Media Lab. Es steht kostenlos unter *https://scratch.mit.edu* zur Verfügung.

Urheber dieses Kurses ist das Projekt ProMaPrim, vertreten durch Dr. Ulrich Schwätzer, Universität Duisburg-Essen, Didaktik der Mathematik, Thea-Leymann-Strasse 9, D-45127 Essen. Die Urheberrechte an diesem Programmierkurs stehen unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 international). Sie dürfen:

- Teilen — das Material in jedwedem Format oder Medium vervielfältigen und weiterverbreiten
- Bearbeiten — das Material remixen, verändern und darauf aufbauen

Unter folgenden Bedingungen:

- Namensnennung — Sie müssen angemessene Urheber- und Rechteangaben machen, einen Link zur Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Diese Angaben dürfen in jeder angemessenen Art und Weise gemacht werden, allerdings nicht so, dass der Eindruck entsteht, der Lizenzgeber unterstütze gerade Sie oder Ihre Nutzung besonders.
- Nicht kommerziell — Sie dürfen das Material nicht für kommerzielle Zwecke nutzen.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das Material remixen, verändern oder anderweitig direkt darauf aufbauen, dürfen Sie Ihre Beiträge nur unter derselben Lizenz wie das Original verbreiten.
- Keine weiteren Einschränkungen — Sie dürfen keine zusätzlichen Klauseln oder technische Verfahren einsetzen, die anderen rechtlich irgendetwas untersagen, was die Lizenz erlaubt.

Das ProMaPrim Icon, das Partnerarbeitsicon und das Aufgabenicon wurde erstellt unter Verwendung der Icons pc.png, brainstorming.png und die-gluhbirne.png von Freepik auf www.flaticon.com (freie Lizenz bei Namensnennung).