



Sommer Camp

mit Robo Wunderkind

vorstellen, bauen, programmieren, spielen!



Dieses Werk ist unter der Creative Commons Attribution 4.0 International License lizenziert. Um eine Kopie dieser Lizenz anzusehen, besuchen Sie <https://creativecommons.org/licenses/by/4.0/deed.de> oder senden Sie einen Brief an Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Willkommen zum Robo Wunderkind Sommer Camp! SchülerInnen und TeilnehmerInnen, die zuvor wenig oder keine technischen Erfahrungen gemacht haben, werden im Laufe des Programms zu selbstbewussten Programmierern und Erfindern. Das Erlernen der Grundlagen des Programmierens, sowie zahlreiche fortgeschrittene Programmier-Tricks werden hier abgedeckt, als auch Möglichkeiten für benutzerdefinierte und kreative Gestaltung beim Bauen von Robotern. Unser sofort einsetzbarer Lehrplan unterstützt Sie bei der Vorbereitung und dem gesamten Verlauf des Camps mit Robo Wunderkind. Sie können gemeinsam mit Ihren SchülerInnen mit Robo spielen: gemeinsam lernen, programmieren und sich Neues überlegen!

Unser sofort einsetzbarer Lehrplan beinhaltet folgendes:

- Sowohl alle **Kerninformationen** und Details um das Sommerlager zu organisieren, als auch ausformulierte **Lernergebnisse** und **Übersichten** von verschiedenen **Konzepten**.
- **10** übersichtliche und einfach nachvollziehbare Unterrichtsplanungen.
- **6 Zusatzaufgaben** um die Komplexität zu erhöhen oder die Unterrichtszeit zu verlängern.
- **Zusatzmaterialien** um Sie beim Unterrichten zu unterstützen.

1. Kerninformation	3
2. Kurzübersicht über das Programm	4
3. Überblick: Konzepte der Robotik und Computerwissenschaften	5
4. Lernergebnisse	7
5. Aufbau einer Unterrichtseinheit mit dem Robo Wunderkind Baukästen	9
6. Stundenplanungen	10
7. Zusatzmaterialien	
• Keywords	39
• Programmierbefehle und Symbole zum Ausdrucken	47
• Karten mit den Schritten des Design Prozesses	51

Themen: MINT Fächer

Alter der Kinder: 6-8, 9-12

Gruppengröße: 6 – 12 SchülerInnen



Dauer: 5 Tage

- **Tag 1-4:** zwei Projekte pro Tag; jedes dauert 1,5 Stunden; insgesamt 12 Stunden Programmieren;
- **Tag 5:** Vorbereitung und Präsentation des Abschlussprojektes für die Eltern; insgesamt 3 Stunden.



Alter der Kinder:

Wir empfehlen zwei Gruppen von SchülerInnen, im Alter von: 6-8 oder 9-12.



Schwierigkeitsgrad: Jede Einheit beinhaltet unterschiedliche Komplexitätsstufen. Die Grundlagen sind für alle SchülerInnen gleich. Durch die Zusatzaufgaben, kann die Komplexität der Aufgaben an das Niveau der SchülerInnen angepasst werden. Jede Stunde beinhaltet sowohl die Grundlagen, als auch mögliche Anpassungen für Fortgeschrittene.

Zusatzaufgaben: Diese ermöglichen es, die Komplexität der Stunden an die Niveaus der SchülerInnen anzupassen, von sehr einfachem Programmieren hin zu komplexeren Projekten und Herausforderungen.

Empfohlene Vorkenntnisse: Die SchülerInnen benötigen keine Vorkenntnisse. Sie werden die grundlegenden Begriffe und Prinzipien der Robotik und des Programmierens während dem Verwenden des Robo Wunderkind Baukasten lernen.



Benötigte Materialien:

- Robo Wunderkind Baukästen;
- Tablet(s);
- Ergänzende Materialien um individuelle Roboter zu gestalten: Lego™ Bausteine, Farbpapier, Karton, etc;
- Zusatzmaterialien: Ausdruckbare Programmier Symbole und Robo Wunderkind Module, Keywords und Design Prozess Cards.

Kurzübersicht über das Programm

Tage		Schwierigkeitsgrad	Konzepte
Tag 1	I. Robo kennenlernen!	☆	Robotik, Ingenieurwesen, Steuerungen und mechanisches Design
	II. Robo erkundet seine Umgebung	☆	Programmieren; Sequenzielle Logik; RW visuell basiertes Programmieren: Befehl, Verbindung, Schleife
Tag 2	I. Robo verwandelt sich in Geräte	☆☆	Elektronische Geräte, mechanisches Design, Problemlösung; RW visuell basiertes Programmieren: Zustand, Parallel, Ausführung
	II. Robo ist eine Taschenlampe	☆☆	Algorithmus-, Taster-, mechanisches und Code-Design, RW visuell basiertes Programmieren: Übergang, Bedingung
Tag 3	I. Robo ist ein intelligenter Wecker	☆☆☆	Intelligente Geräte, mechanisches und Code Design, RW visuell basiertes Programmieren: Übergang, Bedingung
	II. Robo ist ein intelligentes Haustier	☆☆☆	Intelligente Geräte, Sensor, mechanisches und Code Design, RW Visual Based Programming: Übergang, Bedingung
Tag 4	I. & II. Projekt mit Robo für die Endpräsentation	☆☆☆	Konstruktionsprozess, mechanisches und Code Design
Tag 5	I. & II. Endpräsentation	☆☆☆	Konstruktionsprozess, mechanisches und Code Design

Konzepte	Tag 1		Tag 2		Tag 3		Tag 4		Tag 5	
	Einheit 1	Einheit 2	Einheit 3	Einheit 4	Einheit 5	Einheit 6	Einheit 7	Einheit 7	Endpräsentation	
Robotik										
1. Robotik, Ingenieurwesen	+	+	+	+	+	+	+	+	+	+
2. Elektrizität:										
• Elektrischer Strom	+	+	+	+	+	+	+	+	+	+
• Örtliche Verbindungen	+	+	+	+	+	+	+	+	+	+
3. Kabellose Kommunikation	+	+	+	+	+	+	+	+	+	+
4. Fernbedienung	+						+	+	+	+
5. Design Thinking Prozesse:										
• Steuerungs Design	+								+	+
• (mechanisches) Design	+	+	+	+	+	+	+	+	+	+
• Code Design		+	+	+	+	+	+	+	+	+
6. Nutzer Input: Verwendung der Steuerelemente (Geräusche, Licht, Motoren, Servomotor)	+						+	+	+	+
7. Outputs: Funktionen der Module										
• Outputs: Geräusche	+	+			+	+	+	+	+	+
• Outputs: (RGB) Licht	+	+	+	+	+	+	+	+	+	+
• Outputs: DC Motoren	+	+	+		+	+	+	+	+	+
• Outputs: Servo Motor	+		+			+	+	+	+	+

Computerwissenschaften										
1. Programmieren		+	+	+	+	+	+	+	+	+
2. State-Machine basiertes Programmieren:										
• Befehl		+	+	+	+	+	+	+	+	+
• Verbindung		+	+	+	+	+	+	+	+	+
• Zustand			+	+	+	+	+	+	+	+
3. Sequenzielle Logik		+	+	+	+	+	+	+	+	+
4. Schleifen			+	+	+	+	+	+	+	+
5. Parallele Ausführung			+	+	+	+	+	+	+	+
6. Nutzer Input										
• Software Input: Parameter von Befehlen	+	+	+	+	+	+	+	+	+	+
• Sensor Input: Taster				+	+		+	+	+	+
• Sensor Input: Abstandssensor						+	+	+	+	+
• Sensor Input: Geräuschsensor						+	+	+	+	+
7. Digitale Bildung	+	+	+	+	+	+	+	+	+	+
8. Problemlösung			+	+	+	+				
9. Konstruktionsprozess							+	+	+	+
10. Präsentation								+	+	+

Lernergebnisse der Robotik und Computerwissenschaften in Zusammenhang mit den RW Baukästen:

I. Robotik:

- SchülerInnen verstehen und befolgen die Regeln der Kasse bei der Verwendung von Technologien - **Digitale Bildung**.
- SchülerInnen verstehen was Roboter sind und wie diese im Alltag eingesetzt werden, was Robotik ist.
- SchülerInnen wissen was elektrischer Strom und kabellose Kommunikation (Bluetooth) ist und warum Roboter diese benötigen.
- SchülerInnen verstehen, wie Roboter gesteuert werden. Sie kennen den Unterschied zwischen Steuerung und Programmierung.
- SchülerInnen können die Konzepte des mechanischen Designs und Programmier Designs bei Robo Projekten anwenden.

Robo Wunderkind Baukasten:

- SchülerInnen kennen die RW Module und Verbindungen und verstehen deren Funktionen - **Outputs**.
- SchülerInnen verstehen die Funktion des Hauptblocks und können erklären, warum dieser für jedes Projekt notwendig ist.
- SchülerInnen verstehen und berücksichtigen die grundlegende Baulogik der RW Baukästen.
- SchülerInnen verwenden die Funktionen der einzelnen Module um konkrete Projekte umzusetzen. - Anwendung mechanischen Designs.

Robo Live App:

- SchülerInnen können die Robo Live App verwenden: Sie können den Hauptblock verbinden, neue Projekte erstellen und steuern.

II. Computerwissenschaften:

- SchülerInnen wissen, was ein Programm und Programmieren sind.
- SchülerInnen verstehen die Begriffe des State-Machine Based Programmierens wie Befehl, Verbindung, Zustand, Schleife und können diese in der visuell gestalteten Programmiersprache der Robo Code App umsetzen.
- SchülerInnen verstehen den Unterschied zwischen sequenziellen und parallelen Befehlen und können diese programmieren.
- SchülerInnen kennen die Eigenschaften von Befehlen (Nutzer Input) und können diese für die Nutzung eines Roboters verwenden.
- SchülerInnen verstehen und berücksichtigen den konstruktiven Design Prozess um ein Robo Projekt zu erstellen.

Robo Code App:

- SchülerInnen können die Robo Code App verwenden: Sie können den Hauptblock verbinden, ein neues Projekt erstellen und das Befehlsmenü bedienen.
- SchülerInnen können die visuell aufgebaute Programmiersprache der Robo Code App verwenden um in einfachen Projekten Problemstellungen zu lösen.

Kognitive und Verhaltensbezogene Inhalte:

- SchülerInnen können ihre Aufmerksamkeit den Informationen zuwenden, die für das Erstellen der Aufgaben nötig ist (dauerhafte, selektive, abwechselnde und geteilte Aufmerksamkeit).
- SchülerInnen können die Konzentration während der für die Erledigung von Aufgaben / Projekten vorgesehenen Zeit beibehalten.
- SchülerInnen können Informationen für kurzfristige Aufgaben sowie für längere Zeiträume auswendig lernen.
- SchülerInnen können ihre Fähigkeit zum räumlichen Denken anwenden, um nach ihren Vorstellungen funktionierende Roboter zu bauen.
- SchülerInnen können ihr logisches und analytisches Denken anwenden, um verschiedene Roboter zu steuern und zu programmieren.
- SchülerInnen können Problemstellungen durch ihr kritisches Denkvermögen lösen: Ziele setzen, planen, handeln, reflektieren, verbessern, bewerten und dabei Kritik akzeptieren und daran wachsen.
- SchülerInnen können ihre Fantasie und Kreativität einsetzen, um ein eigenes Projekt zu erarbeiten.
- SchülerInnen können in Paaren / kleinen Gruppen zusammenarbeiten und soziale Fähigkeiten nutzen, um ein gemeinsames Projekt zu erarbeiten.
- SchülerInnen können erarbeitete Projekte vor ihrer Klasse präsentieren, konstruktives Feedback annehmen und geben.

Aufbau:

- Fokus auf bestimmte Konzepte in Bezug auf die RW Module, die Robo Live oder Robo Code App.
- Lernziele, Lernergebnisse und Keywords.
- Zusatzmaterial zum Ausdrucken.

Unterrichtsabschnitte: 8 Schritte

Einleitung 7 – 10 min	<ol style="list-style-type: none">1 Aktivieren der SchülerInnen: Bezug zu Vorwissen und persönlichen Erfahrungen der SchülerInnen herstellen.2 Analysieren: Erzählung der Robo Story um die SchülerInnen emotional zu erreichen, Problemstellung erarbeiten, diskutieren und die Projektziele festlegen.
Angeleitete Aufgabe 15 – 25 min	<ol style="list-style-type: none">3 Vorbereitung: Erinnerung an zuvor erlernte Inhalte zu den RW Baukästen und den Apps; Verbinden des Hauptblocks mit den Tablets und Gestaltung eines neuen Projektes.4 Learn by doing – Gemeinsames Bauen und Programmieren: Die SchülerInnen lösen unterschiedliche Aufgaben und sammeln Wissen beim kooperativen Programmieren und Diskutieren.5 Zusammenfassen der neuen Informationen, bevor zur selbstständigen Aufgabe gewechselt wird.
Selbstständige Aufgabe 30 – 45 min	<ol style="list-style-type: none">6 Gestaltung eines eigenen Projektes: Die SchülerInnen arbeiten einzeln, paarweise oder in kleinen Gruppen, um ein eigenes Projekt zu erstellen. Die Lehrperson unterstützt SchülerInnen, welche Schwierigkeiten haben einzeln. Für die Gestaltung der Umgebung und individuelle Roboter, können unterschiedliche Materialien verwendet werden. Um Projekte mit zwei oder mehreren Robotern zu erstellen, können SchülerInnen in großen Gruppen zusammenarbeiten. Präsentation (optional): Die SchülerInnen stellen ihre Projekte der Klasse vor und geben sich gegenseitig konstruktives Feedback. * Variation: <i>Zusätzliche Aufgaben zur Steigerung der Komplexität für fortgeschrittene SchülerInnen.</i>
Reflexion & Feedback 8 – 10 min	<ol style="list-style-type: none">7 Zusammenfassung der gelernten Inhalte. Feedback über die Schwierigkeit der Aufgaben einholen.8 Aufräumen: Die SchülerInnen lernen den sorgfältigen Umgang mit Geräten und Baukästen kennen. Nach der Verwendung werden die elektronischen Geräte ausgeschaltet und die Module in die Box sortiert. Die Tablets werden sorgfältig eingesammelt.

Dauer: 60 – 90 min

Tag 1. Einheit 1. Robo kennenlernen!

Konzepte: Robotik, Ingenieurwesen,
Steuerung, Mechanisches Design

Schwierigkeitsgrad: ★☆☆



Robo Story:

Heute haben wir einen besonderen Gast in unserer Klasse! Das ist Robo, ein intelligenter Roboter, der zu unserem Sommerlager gekommen ist, um unser Freund zu sein!



Unterrichtsziele:

Wir werden Ingenieure, um unseren ersten Roboter zusammenzubauen und zu steuern!

Module:



Hauptblock



Motoren

x2



Servo Motor



RGB Licht



Taste



Entfernungssensor



Räder

x2



kleines Rad



Steckverbindungen

x2



Lego Adapter



Verbindungs-
kabel



Verbindungsblock

Fokus:

- **Robotik:** Roboter und deren Aufgaben im Alltag, Fernsteuerung als Möglichkeit zur Bedienung von Robotern.
- **Digitale Bildung:** Regeln für die Verwendung von Tablets und Bauteilen.
- **Robo Live App:** Schnittstelle und Logik.
- **Robo Wunderkind Baukasten:** Module, Verbindungsstücke und deren Funktionen.

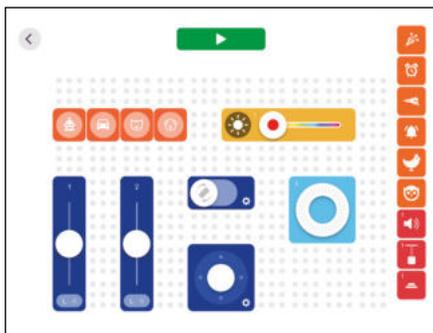
Lernziele:

Die SchülerInnen können einen Roboter unter Berücksichtigung der unterschiedlichen Funktionen der einzelnen Module zusammenbauen.

Lernergebnisse:

- Die SchülerInnen kennen verschiedene Module und deren Funktionen.
- Die SchülerInnen können erklären, warum jeder Roboter einen Hauptblock benötigt.
- Die SchülerInnen verstehen wie Module miteinander zu kombinieren sind um einen Roboter zu bauen.
- Die SchülerInnen können die Robo Live App verwenden, um verschiedene Module zu steuern.

Steuerung:



Keywords:

- Robotik, Ingenieurwesen, Roboter;
- Module: Hauptblock, Verbindungsstücke, Trennungs-Tool.

Unterrichtsabschnitte:

Einleitung
7 – 10 min

- 1 Fragen:** Wisst ihr was ein Roboter ist? Warum bauen Menschen Roboter? Wer von euch hat bereits Erfahrung mit Robotern? Wie werden Roboter gesteuert? Wer beschäftigt sich mit Robotern? Besprechen folgender Begriffe: Ingenieurwesen, Robotik, Roboter.
- 2 Analysieren:** Erzählen der Robo Story. Gemeinsames erarbeiten von Problemstellung und Lösungen zur Bewältigung.

Angeleitete Aufgabe
15 – 25 min

- 3 Öffnen** und erkunden der Robo Boxen: kurzes durchschauen der Module.
- 4 Entdecken der Robo Live App:**
 - Besprechen Sie die Verwendung des Tablets für die Steuerung des Roboters. Teilen Sie die Tablets aus und besprechen Sie die Benutzerregeln der Tablets.
 - Einschalten des Hauptblocks und Verbindung mit der Robo Live App; besprechen von: Mein Robo Bildschirm, Coding Lab und Neues Projekt-Taste;
 - Erstellen eines neuen Projekts und erkunden des Kontrollbildschirms, Steuerung und Kontrollmenü.
 - Weitere Module werden nun an den Hauptblock angesteckt. Die SchülerInnen können herausfinden, wie sich das in der App bemerkbar macht. Anschließend werden diese wieder auseinandergebaut. Besprechen Sie die Verbindungsstücke und das Trennungs-Tool mit Ihren SchülerInnen. Verwenden der Steuerungselemente um die einzelnen Module zu steuern. Besprechen Sie, wie diese funktionieren.
 - Führen Sie den Begriff **Fernsteuerung ein**.
- 5 Zusammenfassen** der neuen Inhalte bevor zur eigenständigen AUfgabe übergegangen wird.

Bemerkung: Die SchülerInnen müssen am Ende der Einheit nicht jedes einzelne Module perfekt verstehen und kennen. Vielmehr soll ihnen eine Möglichkeit geboten werden die Module durch selbstständiges Erkunden kennenzulernen. Dabei werden die Module berührt, verbunden und gesteuert um die Funktionen herauszufinden. Achten Sie jedoch darauf, dass jede/r SchülerIn die Funktionen des Hauptblocks kennt.

Selbstständige Aufgabe
30 – 45 min

- 6 Gestalte dein eigenes Projekt:** Die SchülerInnen können nun verschiedene Variationen von Robo bauen und diese mit Hilfe der Robo Live App steuern. Stellen Sie folgende Frage: Wieso hast du den Robo auf diese Art gebaut und inwiefern wird dir der Robo im Alltag helfen? Es können Zusatzmaterialien verwendet werden um einen individuellen Roboter zu gestalten.

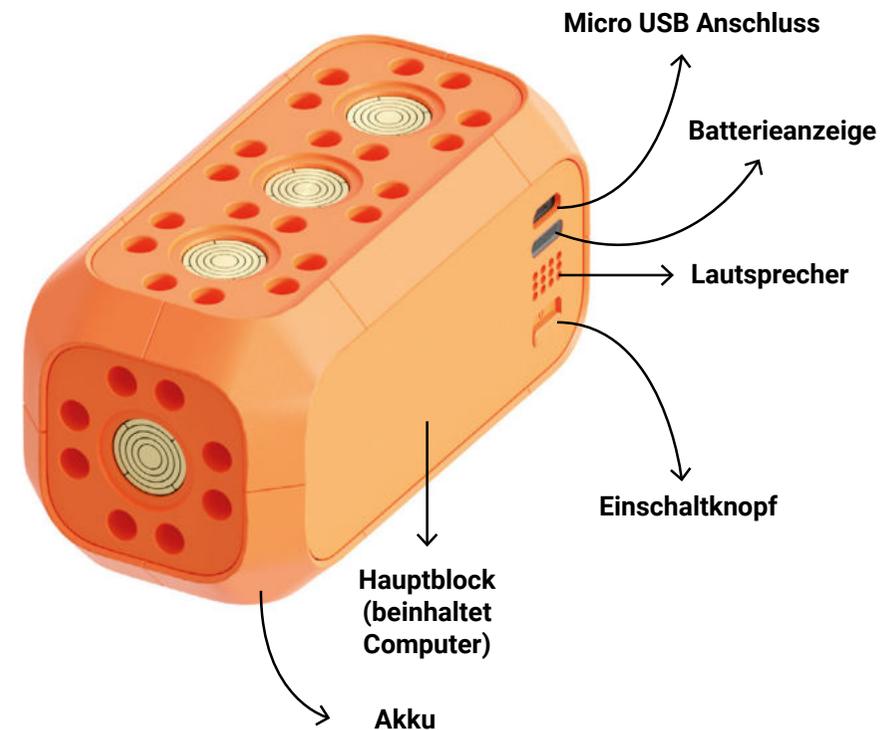
Reflexion & Feedback
8 – 10 min

- 7 Zusammenfassung:** Besprechen einiger Module und ihrer Funktionen, welche in den Projekten verwendet wurden. Feedback erhalten: Waren die Aufgaben schwierig oder einfach? Interessant oder langweilig? Welcher Teil der Einheit war am interessantesten? Wieso?
- 8 Aufräumen:** Die SchülerInnen lernen auf die Geräte die sie verwenden aufzupassen - RW Module und die Tablets.

Tag 1. Einheit 1. ✳ Zusatzaufgabe

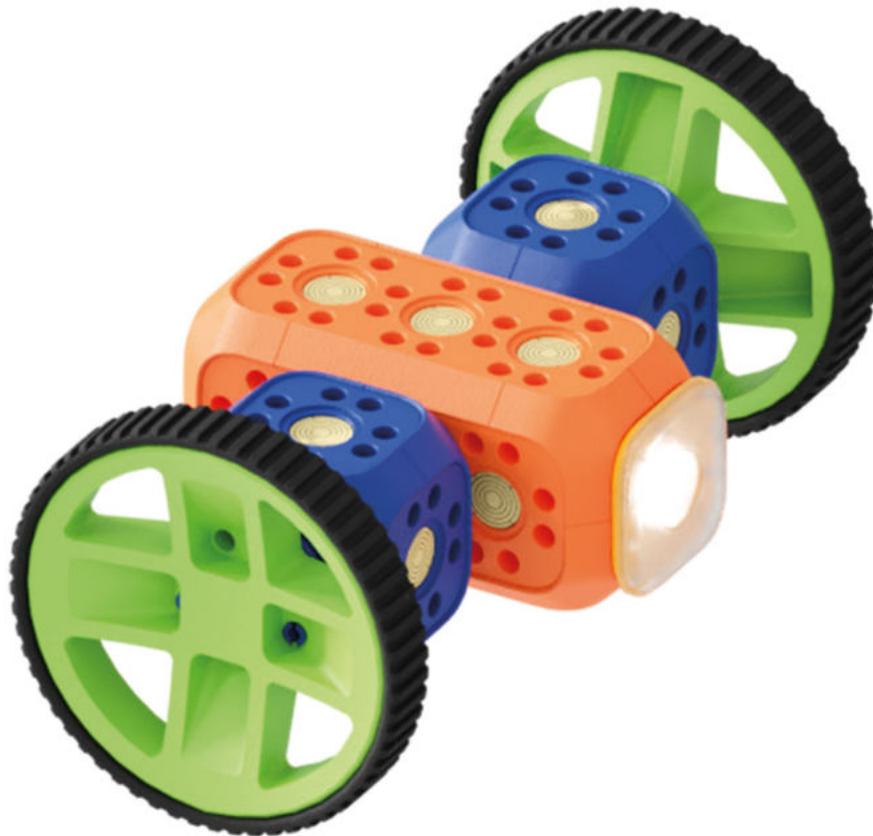
Ziel: Kennenlernen der technischen Details und Designs des Hauptblocks.

- 1 Frage:** Welches Modul ist das Wichtigste? Was denkst du?
- 2 Untersucht den Hauptblock und frage:** Was siehst du auf der Oberfläche? Was könnte in ihm drin sein?
- 3 Zusammenfassung:**
 - Das wichtigste Modul ist der Hauptblock, denn er ist das Gehirn von Robo.
 - Auf dem Hauptblock sehen wir den Netzschalter, die kleine Leuchtanzeige - Leistungsindikator, den Lautsprecher und den USB-Anschluss.
 - Im Hauptblock befinden sich eine Batterie und ein kleiner (eingebetteter) Computer.
- 4** Führe die Abbildung des Hauptblocks und seine technischen Details vor. Besprechen der Funktion von: Netzschalter, Leistungsindikator, Lautsprecher, USB Anschluss, Batterie und dem kleinen Computer.
- 5 Zusammenfassung:** Der Hauptblock versorgt die Module mit Strom, er kann mit dem Tablet verbunden werden und erhält über dieses seine Befehle. Der Hauptblock muss in jedem Projekt vorkommen, da die Model über ihn zum Laufen gebracht werden.



Konzepte: Programmieren; Sequenzielle Logik;
RW visuell basiertes Programmieren: Befehl, Verbindung, Schleife

Schwierigkeitsgrad: ★☆☆



Robo Story:

Robo ist ein sehr neugieriges Wesen, das gerne die Welt erkundet! Robo möchte gerne reisen aber zuerst muss er lernen, wie er mit Gegenständen interagiert - und muss außerdem lernen wie er fährt!

Roboter können viele Sachen machen wenn du sie steuerst aber du kannst Robo auch so programmieren, dass er verschiedene Sachen macht. Es gibt eine eigene Welt - die Robo Code App - wo wir die Codes für Robos Auftritt erstellen.



Unterrichtsziele:

Um Robo dabei zu Helfen auf seine erste Reise zu gehen, werden wir Programmierer und erstellen unseren ersten Programm Code in der Robo Code App.

Module:



Programm:



Fokus:

- **Robotik:** Output - Geräusche, RGB Licht, DC Motoren.
- **Computerwissenschaften:** die Rolle des Programmierens in unserem täglichen Leben; Nutzer Input - Parameter von Befehlen.
- **Robo Code App:** Oberfläche, Befehl - Geräusche, Optik, Bewegung, Verbindung, Schleife.

Lernziele:

- Die SchülerInnen können ein einfaches sequentielles Programm erstellen, das Geräusche, visuelle Signale und Bewegungen beinhaltet: Fahren und Abbiegen sowie Verbindungen dazwischen, anschließend soll es so angepasst werden, dass der Roboter eine Schleife fährt.

Lernergebnisse:

- Die SchülerInnen verstehen was ein Programm ist.
- Die SchülerInnen kennen die Code Tasten und Befehle in der Robo Code App und können Geräusche programmieren.
- Die SchülerInnen verstehen folgende Befehle und können diese einsetzen: Geräusch, konstantes Licht, Blinker, Fahren und Drehen.
- Die SchülerInnen können zwei oder mehr Befehle miteinander verbinden um ein einfaches, sequentielles Programm zu erstellen.

Keywords:

- Sequenzieller Code;
- **Module:** Hauptblock, (DC) Motoren, (RGB) Licht;
- **Robo Code App:** Programmierbildschirm, **Befehl**, Startpunkt, Verbindung; Befehle: Geräusche, Optik - konstantes Licht und Blinker; Bewegung - Fahren und Drehen.

Unterrichtsabschnitte:

Einleitung
7 – 10 min

- 1 Fragen:** Wer ist unser neuer Freund? Was haben wir gestaltet? Wie können Menschen Roboter steuern? Was ist der Unterschied zwischen fernsteuern und programmieren? Habt ihr bereits zuvor programmiert? Was ist Programmieren? Was ist ein Programm? Besprechen der Begriffe: Programmieren, Code.
- 2 Analysieren:** Erzählen der Robo Story. Gemeinsames erarbeiten von Problemstellung und Lösungen zur Bewältigung.

Angeleitete Aufgabe
15 – 25 min

- 3 Öffnen** der Baukästen; Abrufen der Funktionen des Hauptblocks und deren Wichtigkeit.
- 4 Kennenlernen der Robo Live App:**
 - Verbinden des Hauptblocks mit der Robo Code App; Besprechen Sie: Mein Robo Bildschirm, Robo Projektbildschirm, Neues-Projekt-Taste. Die SchülerInnen teilen Informationen zu ihren Robo Menüs mit der Klasse.
 - **Erstellen Sie ein neues Projekts; Besprechen Sie:** Programmierbildschirm, Befehlsleiste.
 - **Programmieren von Geräuschen:** Robo lernt Geräusche zu machen!
 - Kennenlernen von Startpunkt, Start-Taste, Verbindungstaste und Verbindungs-Modus, Papierkorb und Lösch-Modus.
 - Fügen Sie das (RGB) Light hinzu, und programmieren Sie anschließend die Befehle: konstantes Licht und Blinklicht. Bringen Sie Ihrem Robo bei, verschiedene visuelle Signale zu erzeugen, um den Raum zu beleuchten oder möglicherweise mit anderen Robotern zu interagieren
 - Fügen Sie die (DC) Motoren hinzu und programmieren Sie anschließend Bewegungen: Fahren und Drehen. Bringen Sie Ihrem Robo bei, herumzufahren.
 - Probieren Sie aus: verbinden Sie alle Geräusche, visuelle Signale und Bewegungen und führen Sie den Begriff sequenzieller Code ein. Ändern Sie den Code, um eine Schleife zu bilden.
- 5 Zusammenfassung:** Was ist unsere Aufgabe, die wir lösen müssen? Welche Fähigkeiten und Module benötigt Robo für seine erste Reise? Welche Befehle haben wir programmiert? Wie werden sie Robo auf seiner Reise helfen? Sind wir bereit, mit Robo zu reisen?

Selbstständige Aufgabe
30 – 45 min

- 6 Gestalten eines eigenen Projektes:**
 - Entscheide, wohin der Robo-Reisende gehen wird. Erstelle eine Umgebung und programmiere den Robo-Reisenden speziell für diese Situation, indem du alle bekannten Befehle und Verbindungen verwendest, um einen sequenziellen Code zu erstellen. Verwende verschiedene Materialien, um deinen Robo anzupassen.
 - **Präsentation (optional):** Die SchülerInnen stellen ihre Projekte der Klasse vor und geben sich gegenseitig konstruktives Feedback.

Reflexion & Feedback
8 – 10 min

- 7 Zusammenfassung:** Was ist Programmieren, Code; the Robo Code App interface and coding buttons; Geräusch, konstantes Licht, Blinker, Fahr- und Drehbefehl sowie deren Verbindung, sequenzieller Code. Feedback über die Schwierigkeit der Aufgaben einholen.
- 8 Aufräumen:** Die SchülerInnen lernen auf die Geräte die sie verwenden aufzupassen - RW Module und die Tablets.

Tag 1. Einheit 2. * Zusatzaufgabe

Ziel: Besprechen von Programmiersprachen im Allgemeinen und der Programmiersprache der Robo Code App im Speziellen.

- 1 Frage:** Sind alle Programme gleich? Warum? Was ist eine Programmiersprache? Kennst du eine Programmiersprache?
- 2 Besprechen Sie:** Wie sieht die Programmiersprache von Robo aus?
 - Was ist ein Befehl?
 - Wo finden wir diese Befehle?
 - Haben alle Befehle die gleiche Farbe? Wieso haben sie verschiedene Farben?
 - Verbinde verschiedene Module mit dem Hauptblock um das Prinzip zu verstehen.
 - Wie verbinden wir Befehle zu einem sequenziellen Code?
 - Was ist wichtig beim Zeichnen einer Verbindung? Wie wirkt sich die Richtung der Verbindung auf den Code aus?
- 3** Vorführen des Screenshots vom Programmierbildschirm mit den verschiedenen Befehlen und Verbindungen dazwischen.
- 4 Zusammenfassung:**
 - Die Programmiersprache der Robo Code App ist eine visuelle, welche mit den Farben der Module von Robo abgestimmt ist.
 - Es gibt bunte Befehle - Programmiersymbole die wie Blasen aussehen, dabei handelt es sich um Befehle für Robo.
 - Wir können alle Befehle in der Befehlsleiste finden.
 - Befehle haben unterschiedliche Farben, weil sie mit den Funktionen verschiedener Module verbunden sind. Es gibt vier Arten von Befehlen in der Befehlsleiste: Geräusche sind (so wie der Hauptblock) orange, visuelle Signale sind gelb (wie das RGB Licht), Bewegung ist blau (wie die Motoren) und Specials sind violett. Ein bestimmter Befehl kann nur programmiert werden, wenn das Modul, das diesen Befehl ausführt, mit dem Hauptblock des Robos verbunden ist.
 - Um einen sequenziellen Code zu erstellen, müssen wir Verbindungen zeichnen.
 - Wir müssen aufpassen, in welche Richtung die Verbindung(en) zeigen, denn in dieser Reihenfolge wird Robo die Befehle ausführen. Wenn die Anordnung falsch ist, gibt es einen Fehler im Code.



Tag 2: Einheit 3. Robo verwandelt sich in Geräte: Robo-Ventilator, Robo-Lampe

Konzepte: Elektronische Geräte, mechanisches Design, Probleme lösen;
RW visuell basiertes Programmieren: Zustand, Parallele Ausführung

Schwierigkeitsgrad: ★★☆☆



Robo Story:

Roboter wurden von Menschen gebaut um ihnen bei Aufgaben zu helfen die zu schwierig, zu gefährlich oder einfach zu langweilig sind, um sie selbst zu erledigen. Können wir unseren Robo in ein Gerät verwandeln, das uns bei Aufgaben hilft?



Unterrichtsziele:

Robo wird in verschiedene Geräten verwandelt um uns im Alltag zu helfen.

Einheit #1: Robo-Ventilator

Stell dir einen schönen Sommertag vor, an dem das Klassenzimmer zu heiß wird. Wie können wir den Raum abkühlen? Welches Gerät bräuchten wir dazu?

Einheit #2: Robo-Lampe

Stell dir vor, du sitzt an einem regnerischen Tag zuhause und willst dein Zimmer mit Hilfe von hellem Lichtern warm und gemütlich machen. Welches Gerät könntest du verwenden?

Module:



Hauptblock



Motoren



Servo Motor



RGB Licht



Rad



Verbindungs-
block



Steckverbindungen

Programm:



Geräusche



visuelle Signale



konstantes Licht



Blinklicht



Bewegung



Motoren



Servo

Fokus:

- **Robotik:** Outputs - Geräusche, RGB Licht, DC, und Servo Motoren.
- **Computerwissenschaft:** Nutzer Input - Parameter von Befehlen; Parallele Ausführung.
- **Robo Code App:** Geräusche, visuelle Signale, Bewegung.

Lernziele:

Ein Programm mit einem sequenziellen Code oder einer parallelen Ausführung erstellen, das Geräusche, visuelle Signale und Bewegung umfasst.

Lernergebnisse:

- Die SchülerInnen wissen was ein elektronisches Gerät ist.
- Die SchülerInnen können Problemstellungen und Lösungen finden.
- Die SchülerInnen können Module so kombinieren, dass der Robo die gestellten Aufgaben erfüllen kann.
- Die SchülerInnen können mit der Robo Code App ein neues Projekt erstellen.
- Die SchülerInnen können ein einfaches sequenzielles Logikprogramm oder ein Programm mit paralleler Ausführung erstellen, das Geräusche, visuelle Signale und Bewegung umfasst.

Keywords:

- Elektronische Geräte, Problemlösungsprozess;
- **Module:** DC Motor, Servo Motor;
- **Robo Code App:** Programmieraste wie Start, Stopp, Papierkorb und Verbindungstaste, Startpunkt, Befehl, Verbindung, Schleife, Zustand.

- 1 Fragen:** Was haben wir beim letzten Mal gebaut? Was haben wir beim letzten Mal programmiert und warum? Welche Module haben wir verwendet und warum?
- 2 Analysieren:** Erzählen der Robo Story. **Frage:** Was ist ein elektronisches Gerät? Welche Geräte verwenden wir im Alltag? Wie helfen sie uns?

Angeleitete Aufgabe
15 – 25 min

- 3 Lösen der Aufgabe #1** - Baue und programmiere einen Robo-Ventilator: Führe die SchülerInnen durch den Problemlöseprozess.
 - **Erkennen:** Lautes Vorlesen von Situation **#1, erkennen** der Problemstellung und finden einer theoretischen Lösung.
 - **Planen:** Entscheide welche Module und Coding Actions für das Projekt benötigt werden und warum. Erstelle einen stufenweisen Plan: Algorithmus.
 - **Bauen des Robo-Ventilators:** Probiere unterschiedliche Bauarten. **Mechanische Gestaltung**, vergleiche den DC Motor mit dem Servo Motor. Verwende die Robo Live App um zu verstehen wie die Module funktionieren und wie die unterschiedliche Gestaltung Robos Auftritt beeinflussen. Entscheide dich für die beste Lösung.
 - **Programmieren des Robo-Ventilators:** Erkunde die Bewegungen des ersten Motors und des Servos und ihre Einstellungen; vergleiche sie miteinander. Bei was sind sie gleich und bei was verschieden?
 - **Reflexion:** Besprechen des Prozesses und der endgültigen Lösung. Achte dabei besonders auf den DC und Servo Motor und ihre Bewegungen.
- 4 Lösen der Aufgabe #2** - Baue und programmiere eine **Robo-Lampe:** Lass die SchülerInnen den Problemlöseprozess üben.
 - **Erkennen:** Lautes Vorlesen von Situation **#2, erkennen** der Problemstellung und finden einer theoretischen Lösung.
 - **Planen:** Entscheide welche Module und Coding Actions für das Projekt benötigt werden und warum. Erstelle einen stufenweisen Plan: Algorithmus.
 - **Bauen der Robo-Lampe:** Probiere unterschiedliche Bauarten. **Mechanische Gestaltung**, verwende die Robo Live App um zu verstehen wie die Module arbeiten wie verschiedene Gestaltungen Robos Auftritt verändern. Entscheide dich für die beste Lösung.
 - **Programmieren der Robo-Lampe:** Lerne wie die Parallele Ausführung funktioniert. Programmiere visuelle Signale, Geräusche, und Bewegungen als sequenziellen Code und wandle sie dann in einen Zustand um. Besprecht die Unterschiede in Robos Auftritten. Führe die Begriffe Zustand und parallele Ausführung ein.
 - **Reflexion:** Besprich den Prozess und die endgültige Lösung. Achte dabei besonders auf die parallele Ausführung und den Zustand.
- 5 Zusammenfassung:** Was ist das mechanische Design und warum ist es für die Projekte wichtig? Was ist der Unterschied zwischen einem sequentiellen Code und einer parallelen Ausführung?

Selbstständige Aufgabe
30 – 45 min

- 6 Einzel- oder Gruppenarbeit:**
 - Baue und programmiere ein Robo-Gerät das dir im Alltag hilft. Verwende den Hauptblock, das RGB Licht, den DC und den Servo Motor. Verwende sowohl sequenzielle Logik als auch die parallele Ausführung in einem Code. Plane, baue und programmiere um das Projekt auszuführen. Verwende zusätzliche Materialien um deinen Robo individuell anzupassen.
 - **Präsentation (optional):** Die SchülerInnen stellen ihre Projekte der Klasse vor und geben sich gegenseitig konstruktives Feedback.

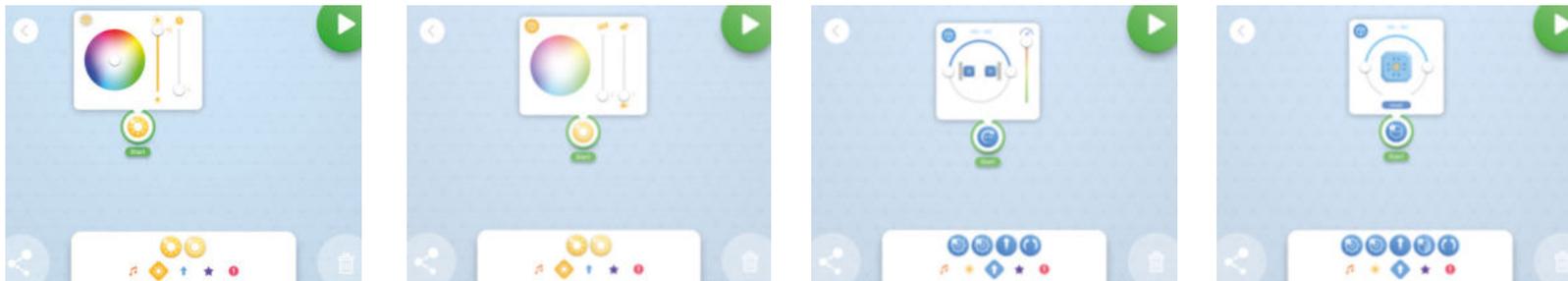
Reflexion & Feedback
8 – 10 min

- 7 Zusammenfassung:** Motor 1 und Servo Bewegungen und ihre Einstellungen, parallele Ausführung, Stadium und welcher Befehl zu welchem Stadium führt. Erhalte Feedback zur Komplexität der Aufgaben und den Aktivitäten.
Aufräumen: Die SchülerInnen lernen auf die Geräte die sie verwenden aufzupassen - RW Module und die Tablets.

Tag 2: Einheit 3. ✳ Zusatzaufgabe

Ziel: Erfahre mehr über die Zufalls Funktion.

- Suchen Sie in den Befehlseinstellungen für konstantes Licht nach der Zufallsfunktion.
 - **Probieren Sie** die Zufalls Funktion für konstantes Licht aus und besprechen Sie, wie sie den Robo-Auftritt beeinflusst.
 - **Frage: Was bedeutet "Zufall"? Wer legt die Einstellungen fest, wenn sie zufällig sind?**
 - **Finden Sie** die anderen Befehle, die über die Zufalls Funktion verfügen und probieren Sie sie aus: **konstantes Licht, Blinker, Drehen und Servo-Befehle.**



- Programmieren Sie die Zufalls Funktion** für verschiedene Befehle:
 - **Programmiere die Zufalls Funktion für visuelle Signale:** konstantes Licht und eine Blink-Befehl in unterschiedlichen Variationen.
 - Vergleiche sie die Befehle mit den eingestellten Farben.
 - **Programmiere die Zufalls Funktion für Bewegungen:** Dreh- und Servobefehle in verschiedenen Variationen.
 - Vergleichen Sie die Befehle mit einem festgelegten Winkel.
 - Vergleichen Sie visuelle Signale und Bewegung mit der Zufalls Funktion - welche Parameter werden davon gesteuert?

- Gestalte ein kleines Projekt:** Überlege dir, wofür die Zufalls Funktion programmiert werden könnte.

Beispiele für Projekte:

- **Visuelle Signale:**
 - Robo Story:** Robo verwendet verschiedene Farben um seine Stimmung auszudrücken.
 - Spiel:** Robo macht verschlüsselte Lichtsignale für andere Roboter. Sie müssen sich an sie erinnern und versuchen, sich zu wiederholen.
- **Movement:**
 - Robo Story:** Robo schüttelt seinen Kopf beim Tanzen.
 - Spiel:** Robo sieht zu verschiedenen Blickwinkeln, Sie müssen erraten, in welche Richtung er beim nächsten Mal den Kopf drehen wird.

Tag 2: Einheit 4. Robo ist eine Taschenlampe

Konzepte: Algorithmus, Taster, Mechanisches Design, Programmier Design,
RW visuell basiertes Programmieren: Übertragung, Bedingung

Schwierigkeitsgrad: ★ ★ ☆



Robo Story:

Stell dir vor du suchst etwas unter deinem Bett, wo es ganz dunkel ist. Welches Gerät würdest du verwenden?

Arbeitet dieses Gerät die ganze Zeit? Kann es an- und ausgeschaltet werden? Wie? Vielleicht mit einer Taste?



Unterrichtsziele:

Jetzt verwandeln wir Robo in ein neues Gerät mit einer Taste, eine Robo-Taschenlampe die an- und ausgeschaltet werden kann.

Module:



Hauptblock



RGB licht



Taste

Programm:



Befehle



Taste

Fokus:

- **Robotik:** Taster, Geräte mit Tasten.
- **Computerwissenschaften:** Nutzer Input - Parameter von Befehlen..
- **Robo Code App:** Taster-Bedingung.

Lernziele:

Die SchülerInnen können eine Robo-Taschenlampe bauen und ein einfaches Programm erstellen, das einige der grundlegenden Befehle und Taster-Bedingungen enthält.

Lernergebnisse:

- Die SchülerInnen wissen was ein Algorithmus ist, können den Begriff erklären und ihn dazu verwenden, ein Robo-Gerät zu programmieren.
- Die SchülerInnen wissen was eine Taste ist und können deren Funktion erklären.
- Die SchülerInnen verstehen, wie Module verbunden werden müssen, um eine Robo-Taschenlampe zu bauen.
- Die SchülerInnen wissen was Übertragung ist und wie es bei einem sequenziellen Code und bei paralleler Ausführung dazu kommt.
- Die SchülerInnen können ein einfaches Programm erstellen, das einige der grundlegenden Befehle und Verbindungen dazwischen enthält.
- Die SchülerInnen können den Code mit den Tasten-Bedingungen anpassen.

Keywords:

- Taster, Bedingung;
- Module: Taste;
- Robo Code App: Übertragung, Bedingungen, Taster-Bedingung.

Tag 2: Einheit 4. Robo ist eine Taschenlampe

Einleitung
7 – 10 min

- 1 Fragen:** Was ist ein Gerät? Welche Geräte haben wir letztes Mal gebaut und programmiert und warum? Welche Arten von Codes haben wir programmiert? **Wiederholen der Begriffe: sequenzieller Code, parallele Ausführung.**
- 2 Analysieren:** Erzählen der Robo Story, besprechen und festlegen der Projektziele zusammen mit den SchülerInnen.

- 3 Planen:** Entscheide welche Module und Programmierbefehle du für das Projekt benötigst und warum. Befolge die schrittweise Anleitung für den Auftritt der Robo-Taschenlampe, führe den Begriff "Algorithmus" ein.

1) Robo-Taschenlampe ist ausgeschaltet - nichts passiert. => 2) Taste wird gedrückt. => 3) Robo-Taschenlampe leuchtet.
4) Robo-Taschenlampe leuchtet. => 5) Taste wird gedrückt. => 6) Robo-Taschenlampe ist ausgeschaltet.

- 4 Baue eine Robo-Taschenlampe und programmiere sie so, dass sie den Algorithmus verwendet. Lass die SchülerInnen den richtigen Code selbst entdecken, indem du sie durch folgende Schritte führst: :**

- Füge den Taster hinzu. **Frage:** Welches neue Modul werden wir für die Robo-Taschenlampe brauchen? Was ist die Taste? Welche Geräte aus dem Alltag haben Tasten?
- Erfahre etwas über den Pausen-Befehl. **Fragen:** Wie können wir folgendes programmieren: 1) die Robo-Taschenlampe wird ausgeschaltet - nichts passiert. Was bedeutet das - Robo macht nichts?
- Füge dem Code visuelle Signale hinzu: 3) Robo-Taschenlampe leuchtet. Und verbinde die visuellen Signale mit des Pausen-Befehls.
- Erfahre mehr über den Übergang: Der Akt des Wechsels von einem Zustand in einen anderen. Probiere verschiedene Einstellungen für die Laufzeit aus, um zu sehen, wann der Übergang stattfindet. Programmiere eine unbegrenzte Laufzeit für den Pausen-Befehl. Besprechen Sie, dass der Übergang in diesem Fall nicht stattfindet.
- Erfahre mehr über die Bedingungen: Ein spezielles Symbol, das den Übergang zwischen zwei Befehlen ermöglicht. Wir müssen die Bedingung verwenden, wenn 1) wir eine Situation haben, in der der Übergang nicht automatisch erfolgt - die Lebensdauer ist unendlich oder 2) wir möchten, dass der Übergang nur in der konkreten Situation stattfindet.
- Bedingungen erkunden: Finden Sie sie in der Befehlsleiste: besprechen Sie ihr Design und wie sie sich von den Befehlen unterscheiden - sie sehen aus wie Aufkleber. Frage: Welche Art von Bedingung könnten wir in unserer Situation benötigen?
- Programmieren Sie die Tasten-Bedingung. Suche die Tasten-Bedingung in der Befehlsleiste und platziere sie auf der Verbindung zwischen zwei Befehlen. Besprechen Sie, dass die Bedingung auf der Verbindungen platziert werden muss, um anzugeben, welcher Übergang reguliert wird. Versuch es.
- Ändern Sie den Code entsprechend dem Algorithmus: 4) gleiches konstantes Licht + 5) Verbindung zurück zum Pausen-Befehl + 6) Gleicher Pausen-Befehl. Besprechen: Um den Algorithmus abzuschließen, haben wir den Code in die Schleife geändert. Die zweite Tasten-Bedingung fehlt jedoch für die zweite Verbindung.
- Programmieren Sie die zweite Tasten-Bedingung: Platzieren Sie sie auf der zweiten Verbindung in der Schleife. Versuch es.

- 5 Reflexion:** Besprechen Sie den Kodierungsprozess und die endgültige Lösung. Achten Sie auf die Taste als Robo-Modul und auf seine Funktionsweise.

Angeleitete Aufgabe
15 – 25 min

- 6 Einzel- oder Gruppenarbeit:**

- Überlegen Sie, welches Robo-Gerät mit der Taste gebaut werden kann und wie die Tasten-Bedingung im Code dafür verwendet werden kann. Plane, Baue und Programmieren Sie, um das Projekt durchzuführen. Verwenden Sie verschiedene Materialien, um Robo anzupassen.
- Präsentation (optional): Die SchülerInnen präsentieren ihre Projekte der Klasse und geben einander konstruktives Feedback.

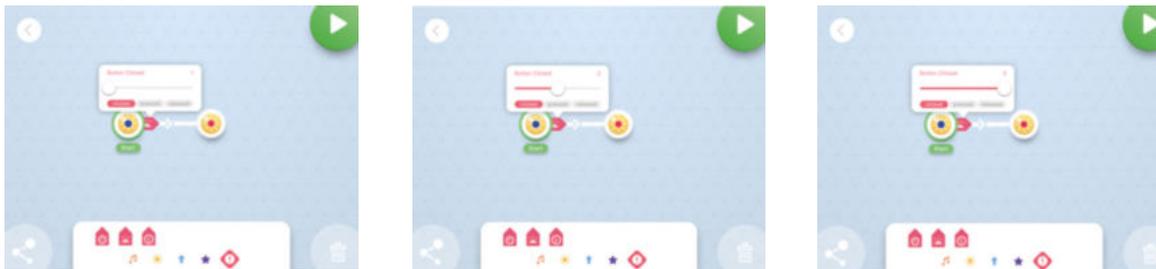
Selbstständige Aufgabe
30 – 45 min

Ziel: Erkunde die Einstellungen der Knop-Bedingung.

1 **Frage:** Welche Geräte mit Tasten kennst du? Wie funktioniert eine Taste? Sind alle Tasten gleich oder verschieden? Können wir auch die Taste für Robo verschieden programmieren?

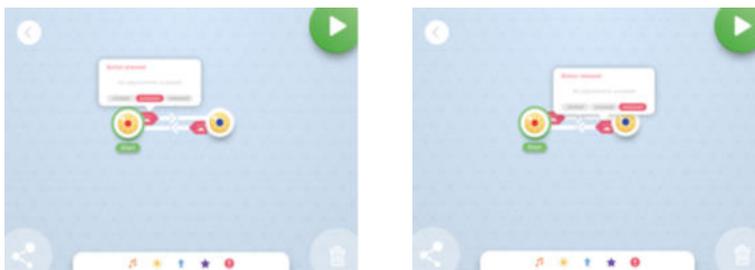
2 **Programmieren** Sie die Knopf-Bedingung und ändern Sie die Einstellungen für die Anzahl der Klicks. Der Übergang erfolgt, wenn:

- Die Taste wird einmal geklickt;
- Die Taste wird zweimal geklickt;
- Die Taste wird dreimal geklickt.



3 Programmieren Sie die Schleife und fügen Sie die Tasten-Bedingung bei beiden Übergänge hinzu:

- Der erste Übergang geschieht, wenn die Taste gedrückt wird;
- Der zweite Übergang geschieht, wenn die Taste losgelassen wird.



4 **Spielerisches Ausprobieren** - Beispiele für Aufgaben:

- Die Robo-Taschenlampe macht Geräusche wenn sie ein- und ausgeschaltet wird.
- Die Robo-Taschenlampe ändert ihre Farbe wenn die Taste einmal gedrückt wird und schaltet sich ein und aus, wenn die Taste zweimal gedrückt wird.
- Die Robo-Taschenlampe verändert den Modus - konstantes Licht / Blinklicht, wenn die Taste gedrückt / losgelassen wird.

5 **Zusammenfassung:** Die Funktion der Taste (Modul), die Funktion der Tasten-Bedingung im Code, seine Einstellungen und wie sie Robos Auftritt beeinflussen.

Konzepte: Intelligente Geräte, mechanisches und Code Design, RW visuell basiertes Programmieren: Übergang, Bedingung.

Schwierigkeitsgrad: ★★ ★



Robo Story:

Wie du weißt, kann sich unser Robo in verschiedene Geräte verwandeln! Dadurch kann er bei alltäglichen Aufgaben helfen. Was genau ist ein intelligentes Gerät? Lass uns diese Frage zusammen beantworten.

Stell dir vor, du musst früh aufstehen um in die Schule zu gehen, wie können wir das lustiger machen? Was ist, wenn du Angst hast zu verschlafen?



Unterrichtsziele:

Wir lernen etwas über intelligente Geräte und verwandeln unseren Robo in eines davon: einen intelligenten Robo-Wecker.

Module:



Hauptblock



RGB Licht



Motoren



Taste



Rad

Programm:



Befehle



Uhr



Stoppuhr



Taste

Fokus:

- **Robotik:** Outputs - Geräusche, RGB Licht, DC Motor.
- **Computerwissenschaften:** Algorithmus.
- **Robo Code App:** Uhr-, Stoppuhr- und Tasten-Bedingungen..

Lernziele:

Die SchülerInnen können einen intelligenten Robo-Wecker bauen und ein einfaches Programm erstellen, das einige der grundlegenden Befehle sowie die Uhr-, Stoppuhr- und Tasten-Bedingungen enthält.

Lernergebnisse:

- Die SchülerInnen wissen was ein Algorithmus ist und können ihn dazu verwenden einen Code für einen intelligenten Robo-Wecker zu erstellen.
- Die SchülerInnen verstehen wie Module verbunden werden müssen, um einen intelligenten Robo-Wecker zu bauen.
- Die SchülerInnen können ein einfaches Programm erstellen, das einige der grundlegenden Befehle und Verbindungen zwischen ihnen enthält.
- Die SchülerInnen können den Code mit Hilfe von Uhr-, Stoppuhr- und Tasten-Bedingung anpassen.

Keywords:

- Algorithmus;
- Robo Code App: Stoppuhr- und Tasten-Bedingung.

Einleitung
7 – 10 min

- 1 Frage:** Warum bauen Menschen Roboter? Was ist ein intelligentes Gerät? Wie ist ein intelligente Gerät mit anderen Geräten verbunden? Hast du schon ein intelligentes Gerät verwendet - wenn ja, welches? Welche Geräte haben wir letztes Mal gebaut und programmiert? Welche Bedingungen haben wir in dem dafür entwickelten Code verwendet? Wie haben wir unseren Code geplant? Wiederhole: Wir haben stufenweise Anleitungen - Algorithmus.
- 2 Analysiere:** Erzählen der Robo Story, besprechen und festlegen der **Projektziele** zusammen mit den SchülerInnen.

Angeleitete Aufgabe
15 – 25 min

- 3 Plane:** Entscheide welche Module und **Programmierbefehle** für das Projekt benötigt werden und warum. Befolge die stufenweise Anleitung - **Algorithmus** für den Robo-Wecker:
1) Robo-Wecker wartet - nichts passiert => 2) Zeit aufzuwachen => 3) Robo-Wecker erzeugt einen Alarm Code (unendlich)
4) WENN die Taste gedrückt wird => 5) Robo-Wecker schaltet sich aus.
- 4** Baue einen Robo-Wecker und programmiere ihn so, dass der Algorithmus verwendet wird. Die SchülerInnen entdecken den richtigen Code, indem Sie sie durch die einzelnen Schritte führen:
 - Baut einen intelligenten Robo-Wecker und bespricht welche Module dazu verwendet wurden und warum?
 - **Programm 1)** Robo-Wecker wartet - nichts passiert - Pausen-Befehl. Abfragen wie dieser Befehl funktioniert.
 - **Programm 3)** Robo-Wecker erzeugt einen Alarm Code (unendlich) - **Geräusche, Visuelle Signale, Bewegung als sequenzielle Logik oder parallele Ausführung.**
 - **Frage:** Wie wird Schritt 2) Zeit aufzuwachen programmiert? **Abrufen:** Was ist eine Bedingung? Wie beeinflussen Bedingungen den Übergang zwischen zwei Befehlen? Wie sehen die Bedingungen in der Robo Code App aus und wie funktionieren sie? Welche Bedingung haben wir beim letzten Mal programmiert?
 - Erfahre mehr über die Uhr- und Stoppuhr- Bedingungen: Robo weckt dich zu einer bestimmten Zeit auf. Probiere beide Bedingungen aus und bespricht den Unterschied.
 - Füge die Taste hinzu und programmiere folgendes: 4) WENN die Taste gedrückt wird - 5) Robo-Wecker schaltet sich aus.
 - **Ausprobieren:** Verwende verschiedene Einstellungen für die Befehle und Bedingungen im Code.
- 5 Reflexion:** Bespricht den Prozess des Bauens und Programmierens, sowie die endgültige Lösung. Achte dabei besonders auf die Stoppuhr und die Uhr und wie sie arbeiten.

Selbstständige Aufgabe
30 – 45 min

- 6 Einzel- oder Gruppenarbeit:**
 - Überlegt, welche intelligenten Robo-Geräte gebaut werden können und wie Tasten-, Stoppuhr- oder Uhr-Bedingungen dafür im Code verwendet werden können. Plane, baue und programmiere um das Projekt umzusetzen. Verwende verschiedene Materialien um deinen Robo individuell anzupassen.
 - *Präsentation (optional) : Die SchülerInnen präsentieren ihre Projekte der Klasse, geben und erhalten konstruktives Feedback.*

Reflexion & Feedback
8 – 10 min

- 7 Zusammenfassung:** Übergang, Pause-Befehl, Tasten-, Stoppuhr- oder Uhr- Bedingungen. Erhalte Feedback zur Schwierigkeit der Aufgaben.
- 8 Aufräumen:** Die SchülerInnen lernen auf die Geräte die sie verwenden aufzupassen - RW Module und die Tablets.

Ziel: Erfahre etwas über die **Push-Benachrichtigung**.

- 1 Frage:** Wie können uns intelligente Geräte benachrichtigen? Weißt du, was eine Push-Benachrichtigung ist? Hast du schon mal eine Push-Benachrichtigung gesehen? Welche Geräte senden Push-Benachrichtigungen? Kann unser Robo eine Push-Benachrichtigung senden?
- 2 Entdecke die Push-Benachrichtigung:**
 - Finde die Push-Benachrichtigungsbefehl in der Befehlsleiste und besprich wie sie aussieht: Farbe und Symbol.
 - Programmiere die erste Push-Benachrichtigung: schreibe den Text und probier es aus; besprich anschließend was auf dem Bildschirm passiert ist.
 - Programmiere verschiedene Push-Benachrichtigungen: verändere den Text.
 - Füge vor und nach dem Push-Benachrichtigungs-Befehl andere Befehle ein und verbinde sie zu einem sequenziellen Code.
 - Fügen Sie den Push-Benachrichtigungsbefehl und einen weiteren Befehl zu einem Zustand hinzu, um die parallele Ausführung zu bilden.



- 3 Zusammenfassung:**
 - Eine Push-Benachrichtigung ist eine Nachricht die auf einem Gerät wie einem Handy oder Tablet aufscheint. Verschiedene Apps können sie zu jeder Zeit schicken. User müssen die App nicht geöffnet haben oder das Gerät in dem Moment verwenden um sie zu bekommen.
 - Die Robo Code App kann auch Push-Benachrichtigungen schicken: sie scheinen auf dem Bildschirm auf, wenn der Code läuft.
 - 4 Fügen Sie den **Push-Benachrichtigungsbefehl zu dem Code hinzu:****
 - 5 Fügen Sie den **Push-Benachrichtigungsbefehl zu eigene Projekten hinzu:****
- Programmiere Robo so, dass er dir eine Nachricht schickt, wenn er dich aufweckt, zum Beispiel: "Guten Morgen!"
 - Überlege welches intelligente Robo-Gerät eine Push-Benachrichtigung schicken könnte. Plane, baue und programmiere um dein Projekt umzusetzen.

Konzepte: Intelligente Geräte, Sensor (Input), mechanisches und Code Design, RW visuell basiertes Programmieren: Übergang, Bedingung

Schwierigkeitsgrad: ★★ ★



Robo Story:

Manche Leute können, aus den verschiedensten Gründen, kein Haustier haben. In der heutigen Zeit können sich Roboter in verschiedene Wesen verwandeln. Können sie auch zu Haustieren werden?



Unterrichtsziele:

Heute bauen und programmieren wir ein intelligentes Robo-Haustier.

Module:



Hauptblock



RGB Licht



Motoren



Servo Motor



Entfernungssensor



Taste



Rad



kleines Rad

Programm:



Befehle



Hindernis



Geräusch

Fokus:

- **Robotik:** Mechanisches Design, Outputs: Geräusche, RGB Licht, DC und Servo Motoren.
- **Computerwissenschaften:** Algorithmus, Sensor Input.
- **Robo Code App:** Hindernis- und Geräusch-Bedingungen.

Lernziele:

Die SchülerInnen können ein Robo-Haustier bauen und ein einfaches Programm erstellen, das einige grundlegende Befehle sowie die Geräusch- und Hindernis-Bedingungen enthält.

Lernergebnisse:

- Die SchülerInnen wissen was ein Algorithmus ist, können erklären worum es sich dabei handelt und können ihn dazu verwenden, ein intelligentes Robo-Gerät zu programmieren.
- Die SchülerInnen verstehen, wie sie die Module verbinden müssen, um ein Robo-Haustier zu bauen.
- Die SchülerInnen wissen was Übergänge sind und wie sie sich im sequenziellen Code und in der parallelen Ausführung äußern.
- Die SchülerInnen können einfache Programme erstellen, die die grundlegenden Befehle und Verbindungen dazwischen beinhalten.
- Die SchülerInnen können den Code mit Hindernis- und Geräusch-Bedingungen anpassen.

Keywords:

- Intelligente Geräte, Sensor (Input), Abstand, Dezibel,
- Module: Entfernungssensor, Verbindungskabel
- Robo Code App: Hindernis- und Geräusch-Bedingungen.

Einleitung
7 – 10 min

- 1 Frage:** Welches intelligente Gerät haben wir das letzte Mal gebaut und programmiert? Welche Bedingungen haben wir dafür im Code verwendet? Besprechen: Hast du ein Haustier? Kümmerst du dich um dein Haustier? Ist das manchmal schwierig? Wieso haben manche Menschen kein Haustier, auch wenn sie sie mögen? Passen alle Menschen mit allen Haustieren zusammen? Können wir ein Robo-Haustier erschaffen? Hast du schon einen Roboter gesehen, der wie ein Haustier aussieht?
- 2 Analytiere:** Erzählen der Robo Story, besprechen und festlegen der Projektziele mit den SchülerInnen.

Angeleitete Aufgaben
15 – 25 min

- 3 Planen und Durchführen des ersten Projekts**
 - Erstelle den Algorithmus - eine schrittweise Anleitung, was das Robo-Haustier machen soll.
 - 1) Robo-Haustier geht nach vorne => 2) Robo-Haustier sieht die Wand => 3) Robo-Haustier hält an
 - Entscheide welche Module und Befehle du für das Projekt brauchen wirst und warum, achte besonders auf den **Verbindungswürfel** und seine Funktionen.
 - Programmiere die Schritte 1) Robo-Haustier geht nach vorne - **Fahrbewegung** und 3) Robo-Haustier hält an - **Pause-Befehl**.
 - Finde und erkunde die Hindernis-Bedingung: das Robo-Haustier erkennt und reagiert auf die Bedingung. Spiele herum und probiere verschiedene Einstellungen aus. Führe den Begriff - Sensor ein und besprich seine Funktionen.
- 4 Planen und Durchführen des ersten Projekts**
 - Erstelle den Algorithmus - eine schrittweise Anleitung, was das Robo-Haustier machen soll.
 - 1) Robo-Haustier schläft => 2) Robo-Haustier hört das Geräusch => 3) Robo-Haustier wacht auf
 - Entscheide welche Module und Befehle du für das Projekt brauchen wirst und warum, achte besonders auf den **Verbindungswürfel** und seine Funktionen.
 - Programmiere die Schritte 1) Robo-Haustier schläft - Pause-Befehl und 3) Robo-Haustier wacht auf - einige grundlegende Befehle wie Geräusche, visuelle Signale oder Bewegung
 - Finde und erkunde die Geräusch-Bedingungen: das Robo-Haustier reagiert auf das Geräusch. Spiele herum und probiere verschiedene Einstellungen aus. Abrufen: der Sensor und seine Funktionen.
- 5 Reflexion:** Besprechen Sie den Kodierungsprozess und die endgültige Lösung für jedes Projekt. Achten Sie auf die Hindernis- und Geräusch-Bedingungen und deren Funktionsweise.

Selbstständige Aufgabe
30 – 45 min

- 6 Einzel- oder Gruppenarbeit:**
 - Überlege was dein Robo-Haustier machen soll und wie die Hindernis- und Geräusch-Bedingungen dafür in dem Code verwendet werden können. Plane, erstelle einen Algorithmus, baue und programmiere um das Projekt umzusetzen. Verwende verschiedene Materialien um deinen Robo anzupassen.
 - *Präsentation (optional) : Die SchülerInnen präsentieren ihre Projekte der Klasse, geben und erhalten konstruktives Feedback.*

Reflexion & Feedback
8 – 10 min

- 7 Zusammenfassung:** Übergang, Pause-Befehl, Tasten-, Stoppuhr- oder Uhr- Bedingungen. Erhalte Feedback zur Schwierigkeit der Aufgaben.
- 8 Aufräumen:** Die SchülerInnen lernen auf die Geräte die sie verwenden aufzupassen - RW Module und die Tablets.

Ziel: Erkunde den Abstands- und Geräuschsensor und programmiere fortgeschrittene Einstellungen für die Hindernis- und Geräusch-Bedingungen.

- 1 Frage:** Was ist ein Sensor? Welche unterschiedlichen Arten gibt es? Wie verwenden Menschen sie? In welchen Geräten können diese Sensoren verwendet werden? Besprich verschiedene Sensoren und ihre Einsatzmöglichkeiten:
 - Temperatursensor
 - Näherungssensor
 - Beschleunigungsmesser
 - Drucksensor
 - Lichtsensor
 - Ultraschallsensor
 - Rauch- und Gassensor
 - Alkoholsensor
- 2** Erkunde den Abstandssensor und programmiere die Hindernis-Bedingung:
 - Besprechen Sie Zentimeter als eine Maßeinheit für Entfernung.
 - Miss die Entfernung zwischen verschiedenen Objekten, indem du die Steuerung in der Robo Live App verwendest.
 - Öffnen Sie die Robo-Code-App und besprechen Sie die "kleiner und größer als" Symbole in den Hindernis-Bedingungseinstellungen.
 - Programmiere ein Robo-Gerät, um den genauen Abstand anzuzeigen (durch einen Ton, ein Lichtsignal oder eine Bewegung). Achten Sie auf das "kleiner und größere als" Symbol.
 - Programmiere ein Robo-Fahrzeug so, dass es zu einer bestimmten Entfernung vor ein Hindernis fährt und dann anhält oder umdreht.
 - Verwende mehr als eine Hindernis-Bedingung in einem Code.
- 3** Erkunde den Geräuschsensor und programmiere die Geräusch-Bedingungen :
 - Dezibel als Maßeinheit für die Lautstärke wird besprochen.
 - Miss die Lautstärke verschiedener Geräusche im Klassenzimmer, indem du die Steuerung der Robo Live App verwendest.
 - Öffne die **Robo Code App** und besprich die "lauter und leiser als" Symbole in den Einstellungen der Geräusch-Bedingung.
 - Programmiere ein Robo-Gerät, um die Lautstärke anzuzeigen (durch einen Ton, ein Lichtsignal oder eine Bewegung).
 - Programmiere einen Robo-Fahrzeugantrieb, bis ein bestimmter Geräuschpegel erzeugt wird und halte an oder drehe um.
 - Verwende mehr als eine Geräusch-Bedingung in einem Code.
- 4** Erschaffe eigene Projekte: ein intelligentes Gerät mit einem Abstands- oder Geräuschsensor: plane, baue und programmiere um das Projekt umzusetzen.

Tag 4: Einheit 7 - I & II Teil. die SchülerInnen gestalten ihre eigenen Projekte für die Abschlusspräsentation.

Konzepte:

Konstruktionsprozess, mechanisches und Code Design

Schwierigkeitsgrad: ★★ ★



Robo's Story:

Jetzt entscheidest du, in welches Gerät oder welchen Charakter sich dein Robo verwandeln soll! Erstelle deinen eigenen Roboter mit allen Modulen und Verbindungsstücken die du willst, einschließlich zusätzlicher Materialien zum Erstellen einer Umgebung. Verwende die Robo Live and Robo Code App um deinen Robo zu steuern und zu programmieren. Denke daran: Um den Robo zu bauen und zu programmieren, müssen wir auch einen Grund haben, warum wir dieses bestimmte Robo-Gerät oder diesen Charakter benötigen und in der Lage sein zu erklären, wie es uns helfen kann.

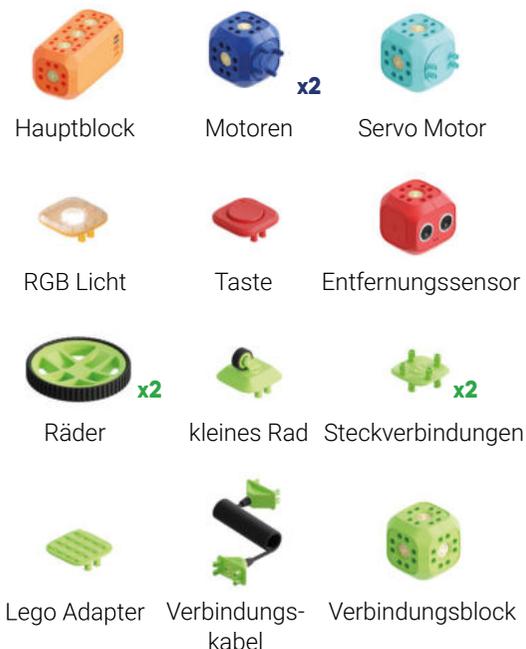


Unterrichtsziele:

Um Robo nach unseren Vorstellungen umzusetzen, befolgen wir die Schritte des Konstruktionsprozesses.

Tag 4: Einheit 7 - I & II Teil. die SchülerInnen gestalten ihre eigenen Projekte für die Abschlusspräsentation.

Module:



Fokus:

- **Robotik:** Outputs - Geräusche, RGB Licht, DC und Servo Motoren.
- **Computerwissenschaften:** Nutzer Input bezüglich der eingestellten Aufgabe, mechanisches und Code Design, Konstruktionsprozess..

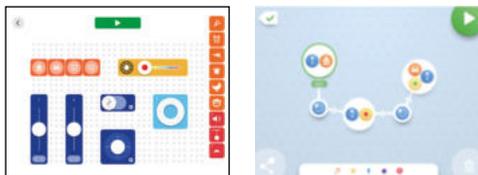
Lernziele:

Die SchülerInnen können ihr eigenes Projekt mit dem RW Robotik Baukasten erstellen, den Roboter zusammenbauen und programmieren entsprechend seiner Funktion, indem sie die gelernten Module und die RW visuell basierte Programmiersprache nützen.

Lernergebnisse:

- Die SchülerInnen können sich eine Geschichte und einen Grund ausdenken, um einen bestimmten Roboter zu bauen.
- Die SchülerInnen kennen die Schritte des Konstruktionsprozesses und können diesen folgen, um die gestellte Aufgabe zu bearbeiten.
- Die SchülerInnen können ihr eigenes Robo-Projekt mit allen gelernten Modulen und der RW visuell basierten Programmiersprache erstellen und programmieren.
- Die SchülerInnen können mit verschiedenen Materialien eine geeignete Umgebung schaffen.
- Die SchülerInnen können ihr Projekt der Klasse vorstellen und ihren MitschülerInnen konstruktives Feedback geben.

Steuerung und Programm:



Keywords:

- Projekt, Konstruktionsprozess, mechanisches und Code Design.

Zusätzliche Materialien:

- Karten zum Ausdrucken mit den Schritten des Konstruktionsprozesses.
- *Optional: Lego™ Steine, Farbpapier und andere Materialien.*

Unterrichtsabschnitte:

Einleitung
7 – 10 min

- 1 Frage:** Wie viele Projekte haben wir mit Robo gemacht? Was war der Grund für jedes Projekt? Wie haben wir jedes Projekt erstellt? Hast du schon mal dein eigenes Projekt in der Schule oder im täglichen Leben erstellt? Wie hast du das gemacht? Ist dieser Prozess ähnlich wie das, was wir mit Robo gemacht haben? Denkst du, dass es einen zusammenhängenden Plan für die Erstellung eines Projekts gibt? Einführen des Begriffs: Konstruktionsprozess.
- 2 Analytiere:** Erzählen der Robo Story, besprechen und festlegen der Projektziele mit den SchülerInnen.

Angeleitete Aufgabe
15 – 25 min

- 3 Erfahre mehr über den Konstruktionsprozess.** Bitten Sie die SchülerInnen zu beschreiben, wie sie ein Projekt gestalten würden. Definieren Sie jeden Schritt gemeinsam mit den Schülern. Hängen Sie die Karte für diesen Schritt an die Tafel.
 - **Schritt 1. Identifiziere einen Grund: Was ist das Problem oder die Idee?** Definiere den Zweck eines bestimmten Robo-Projekts. Erörtere es, besprich es oder schreibe es auf; sei in der Lage den Grund zu erklären.
 - **Schritt 2. Brainstorm: Was sind die Lösungen?** Überlege so viele Lösungen wie möglich, aber bewerte sie nicht. In diesem Schritt müssen die Lösungen nicht unbedingt gut sein.
 - **Schritt 3. Evaluieren und auswählen: Was würde passieren, wenn...?** Denke über die Vor- und Nachteile jeder Lösung nach, diskutiere und ordne sie. Wähle die beste Lösung.
 - **Schritt 4. Skizziere und plane: Was brauche ich?** Mache eine Skizze und entscheide, welche Robo-Module oder andere Materialien du für das Projekt benötigst.
 - **Schritt 5. Erarbeite eine Lösung: Bauen und programmieren, testen, wiederholen!** Arbeite an deiner Idee und probiere es aus. Wenn die erste Lösung nicht funktioniert, besprecht den Grund und fahre mit einer anderen fort. Es ist wichtig, es so lange zu versuchen, bis das Problem gelöst ist. Verliere nicht die Motivation - nicht alle Probleme sind leicht zu lösen!
 - **Schritt 6. Abschluss: Ist alles bereit?** Sobald du die beste Lösung für dein Projekt gefunden hast, schließe sie ab. Erstelle eine Umgebung für deinen Robo oder passe ihn an und prüfe, ob alles für eine Präsentation bereit ist.
 - **Schritt 7. Präsentiere die Lösung.** Zeige der Klasse dein Projekt und bitte um Feedback.
 - **Schritt 8. Reflexion: Wie war es?** Sobald das Problem gelöst ist, reflektiere den Prozess und überlege: Was hat funktioniert? Was nicht? Was kannst du beim nächsten Mal anders machen?

Selbstständige Aufgabe
30 – 45 min

- 4 Übe den Konstruktionsprozess: Schritt 1 - 4.** Die SchülerInnen folgen den Schritten des Konstruktionsprozesses an der Tafel, um ihr eigenes Robo-Projekt zu planen. Sie können selbstständig, paarweise, in kleinen Gruppen arbeiten oder in einem gemeinsamen Projekt mit zwei oder mehr Robotern zusammenarbeiten.
- 5 Reflexion:** Besprechen Sie die Projektideen und -pläne gemeinsam. Geben Sie einander konstruktives Feedback, um die Projektpläne zu verbessern.

Unterrichtsabschnitte:

Einleitung
7 – 10 min

- 1 **Wiederhole:** Den gesamte Plan für die Erstellung eines Projekts - den Konstruktionsprozess.
Analysiere: Wiederhole die Robo Story und den Stundenplan.
- 3 Wiederholen Sie die ersten 4 Schritte des Konstruktionsprozesses und bitten Sie die SchülerInnen, die in Teil 1 erstellten Projektideen vorzustellen:
 - **Schritt 1.** Identifizieren eines Grundes: Was ist das Problem oder die Idee?
 - **Schritt 2.** Brainstorm: Was sind die Lösungen?
 - **Schritt 3.** Evaluieren und wählen einer Lösung.
 - **Schritt 4.** Skizziere und plane: Was brauche ich?
- 4 Wiederholen Sie die weiteren Schritte des Konstruktionsprozesses und konzentrieren Sie sich dabei auf die Schritte 5-6: Hängen Sie die Karten für diese Schritte auf.
 - **Schritt 5. An einer Lösung arbeiten: Bauen und programmieren, testen, wiederholen!** Arbeite an deiner Idee und probiere es aus. Wenn die erste Lösung nicht funktioniert, besprechen Sie den Grund und fahren Sie mit einer anderen fort. Es ist wichtig, es so lange weiter zu probieren, bis das Problem gelöst ist. Verliere nicht die Motivation! Nicht alle Probleme sind leicht zu lösen.
 - **Schritt 6. Abschluss: Ist alles fertig?** Sobald du die beste Lösung für dein Projekt gefunden hast, schließe es ab: erstelle eine Umgebung für deinen Robo oder passen ihn an und prüfe, ob alles bereit ist für eine Präsentation.
 - **Schritt 7.** Präsentieren der Lösung.
 - **Schritt 8.** Reflexion: Wie war es?

Selbstständige Aufgabe
30 – 45 min

- 5 **Üben des Konstruktionsprozesses: Schritt 5 - 6.** Die SchülerInnen folgen den Schritten des Konstruktionsprozesses an der Tafel, um an der Lösung zu arbeiten und ihre eigenen Robo-Projekte abzuschließen. Sie können alleine, paarweise, in kleinen Gruppen arbeiten oder in einem gemeinsamen Projekt mit zwei oder mehr Robotern zusammenarbeiten. Verwenden Sie verschiedene Materialien, um eine Umgebung zu erstellen und Roboter anzupassen.
 - *Bereiten Sie die Präsentation vor (optional): Helfen Sie den SchülerInnen, Fotos vom Robo zu machen, machen Sie Screenshots des Codes und erstellen Sie die Präsentation für den letzten Tag.*
- 6 Präsentation vor der Klasse: Die SchülerInnen stellen der Klasse ihre Projekte vor und geben einander konstruktives Feedback, um sich auf die Abschlusspräsentation vorzubereiten.

Reflexion &
Feedback
8 – 10 min

- 7 **Reflexion:** Besprechen Sie den Prozess des Programmierens und die endgültige Lösung für jedes Projekt. achten Sie auf Schritt 6 - Fertigstellen: Ist alles bereit?

Konzepte:

Konstruktionsprozess, mechanisches und Code Design

Schwierigkeitsgrad: ★★ ★



Unterrichtsziele:

Am allerletzten Tag des Camps präsentieren die SchülerInnen ihre Projekte ihren Eltern. Sie werden ihre persönlichen Erfahrungen während des Prozesses des Erschaffens und Programmierens teilen, Geschichten erzählen, ihre Projekte vorstellen und ihr neu erworbenes Wissen zeigen.

I. Robotik

- Roboter –** eine Maschine, die in der Lage ist, eine komplexe Reihe von Befehlen automatisch auszuführen. Roboter können von einem externen Steuergerät (z.B. einer Fernbedienung) geführt oder vorher programmiert werden. Roboter werden von Menschen gebaut, um bei vielen verschiedenen Aufgaben zu helfen, die manchmal zu komplex, zu gefährlich oder einfach zu langweilig sind.
-
- Robotik –** ein interdisziplinärer Zweig der Ingenieurwissenschaften und der Wissenschaften, der eingesetzt wird, um sich mit dem Entwurf, dem Bau, dem Betrieb und der Verwendung von Robotern sowie der Steuerung, sensorischen Rückmeldungen und Informationsverarbeitung von Computersystemen zu befassen.
-
- Ingenieurwesen –** der Prozess des Erstellens und Aufbaus von technologischen Lösungen und Produkten mit Hilfe von Mathematik und Naturwissenschaften. Ein Ingenieur ist eine Person, die das Ingenieurwesen anwendet und weiterentwickelt. Ingenieure lösen Probleme mit ihren Erfindungen. Es gibt mehrere Bereiche des Ingenieurwesens.
-
- Elektrizität –** Eine Form von Energie, die an einem Ort entsteht und zu einem anderen fließt und zum Betreiben von elektronischen Geräten verwendet wird. Die Robo Wunderkind Module werden von einer Batterie im Hauptblock mit Strom versorgt. Die Stromzufuhr von Modul zu Modul erfolgt über die Verbindungsstücke (und den darauf befindlichen Pogo-Pins) und die Steckoberflächen jedes Moduls.
-
- Kabelgebundene (lokale) Kommunikation –** Eine Art elektrische Kommunikation zwischen elektronischen Geräten, die Informationen über drahtgebundene Kommunikationstechnologie (Steckverbinder) überträgt. Im Robo Wunderkind Robotik-Baukasten gibt es zwei Arten von Verbindern: Verbindungsstücke und Kabelverbindungen, die die Signale übertragen - Befehle vom Hauptblock an andere Module.
-
- Drahtlose Kommunikation –** Die Art der Kommunikation, bei der Informationen über eine Distanz ohne Verwendung einer "fest verdrahteten" Verbindung (wie Pogo-Pins auf den Robo Wunderkind-Modulen) übertragen werden. Die Entfernungen können kurz sein (bei Verwendung einer Fernbedienung) oder sehr lang (Tausende oder sogar Millionen von Kilometern für die Funkkommunikation). Die Robo Code App und die Robo Live App senden die Informationen - Befehle, die Robo per drahtloser Kommunikation über Bluetooth ausführen kann.
-
- Bluetooth Verbindung –** Ist eine drahtlose Verbindung für den Datenaustausch über kurze Entfernungen.

Computer –	Ein Gerät zum Arbeiten mit Informationen. Die Informationen können Zahlen, Wörter, Bilder, Filme oder Töne sein. Computerinformationen werden auch als Daten bezeichnet. Computer können sehr schnell sehr große Datenmengen verarbeiten. Sie speichern und zeigen auch Daten an. Menschen benutzen Computer jeden Tag bei der Arbeit, in der Schule und zu Haus. Computer werden in Fabriken verwendet, um zu kontrollieren, wie Dinge hergestellt werden und in Büros, um beispielsweise Aufzeichnungen zu führen. Im Hauptblock des Robos befindet sich ein kleiner Computer, der über eine drahtlose Verbindung mit einem Gerät (Tablet) kommunizieren kann. Deshalb muss der Hauptblock in jedem Projekt immer vorhanden sein, damit alle anderen Module funktionieren.
Elektronisches Gerät –	Ein Gerät, das seinen Zweck mit Hilfe von Elektrizität erreicht. Es gibt eine große Auswahl an elektronischen Geräten, die täglich verwendet werden, wie beispielsweise Laptops, Mobiltelefone, Kameras, Lüfter, Öfen, Waschmaschinen, Spielekonsolen, Drucker, Radios und natürlich Robo!
Intelligentes Gerät –	Ein elektronisches Gerät, das im Allgemeinen mit anderen Geräten oder Netzwerken (z.B. über Bluetooth) verbunden ist und bis zu einem gewissen Grad interaktiv und autonom arbeiten kann.
Sensor –	Ein Gerät, das ein physikalisches Signal oder einen Reiz (wie Geräusch, Druck oder Licht) empfängt und auf unterschiedliche Weise darauf reagiert. Es gibt einen Geräuschsensor, einen Abstandssensor und einen Taster im Robo Wunderkind Education Kit und einen Bewegungssensor und einen Lichtsensor im Advanced Kit.
Taster (Druckknopf) –	Ist ein einfacher Schaltmechanismus zum Steuern verschiedener Aspekte einer Maschine oder eines Prozesses. Im Robo Wunderkind Robotik-Baukasten befindet sich ein roter Taster, der in der Robo Code App als Taster-Bedingung programmiert werden kann.
Fernbedienung –	Eine Methode zur Fernsteuerung einer Maschine mit drahtlosen Signalen. Dies geschieht in Echtzeit. Zum Beispiel die Fernbedienung für den Fernseher, wenn Sie zum Wechseln der Kanäle Tasten drücken.
Digitale Kompetenz –	Die Fähigkeit, mithilfe verschiedener digitaler Plattformen klare Informationen zu finden, zu bewerten, zusammenzustellen und klare Informationen zu erstellen. Es umfasst sowohl praktische Software-Kenntnisse als auch kritisches Denken, um Online-Sicherheit zu gewährleisten. Die digitale Kompetenz wird anhand der Grammatik, der Komposition, der Schreibfertigkeiten und der Fähigkeit einer Person bewertet Schriften, Bilder, Audiodateien und Designs mithilfe von Technologie zu erstellen. Im Robo Wunderkind Lehrplan werden die zahlreichen, mit der digitalen Kompetenz im Zusammenhang stehenden, Fähigkeiten behandelt, z. B. Regeln für den sicheren Einsatz von Geräten, den Gebrauch von Informationen durch digitale Medien und vor allem die Fähigkeit, Technologie durch Codierung zu erzeugen.

Problemlösung – Ist das Finden einer Lösung für ein Problem. Die Schritte für die Problemlösung unterscheiden sich je nach Disziplin und Strategie geringfügig. Sie umfassen jedoch immer die Definition eines Problems, das Erkennen, Priorisieren und Auswählen von Alternativen für eine Lösung, die Implementierung einer Lösung und das Nachdenken über diese Lösung.

Im **Robo Wunderkind Lehrplan** durchlaufen die SchülerInnen 4 Schritte, um ein Problem mit dem Robo Wunderkind Robotik-Baukasten und mit der Robo Code App oder der Robo Live App zu lösen:

1) Identifiziere ein Problem => 2) Plane die Lösung(en) => 3) Arbeite an der Lösung: bauen und programmieren => 4) Reflektiere

II. Computerwissenschaften

Algorithmus – Schrittweise Lösung einer komplexen Aufgabe. Jeder Schritt ist eine klare Anweisung. Ein einfaches Beispiel für einen Algorithmus ist ein Kochrezept, bei dem Sie in ihrer Reihenfolge geordnete Anweisungen haben, um letzten Endes ein Gericht zu kochen.

Programmcode – Eine Reihe von Anweisungen, die einem Computer mitteilen, was zu tun ist. Eine Folge von kurzen Befehlen nacheinander oder parallel zueinander.

Programmiersprache Damit Sie mit einem Computer kommunizieren können (und dieser Ihre Anweisungen ausführt), müssen Sie seine Sprache sprechen. Es gibt verschiedene Programmiersprachen, einige sind sehr kompliziert, während andere der gesprochenen Sprache ähnlich sind. In der **Robo Code App** verwenden wir eine spezielle visuelle Programmiersprache. Die 3 Begriffe (**Algorithmus, Programm, Programmiersprache**) stehen miteinander in Verbindung. Um den SchülerInnen ein besseres Verständnis hierzu zu vermitteln, können wir folgendes sagen:

- Wenn wir eine komplexe Aufgabe haben, können wir sie in kleinere, individuelle Anweisungen aufteilen - einen Algorithmus erstellen.
- Wir können eine Programmiersprache verwenden, um diese Anweisungen in der Sprache zu schreiben, die der Computer versteht. Daher erstellen wir einen Code.

Robo Wunderkind visuell basiertes Programmieren – Eine einzigartige und intuitive Programmierschnittstelle, die für jüngere Kinder entwickelt wurde, um State-Machine-basierte Programme zu erstellen.

State-Machine Based Programming – Ein Stadium führt Befehle aus und führt zu einem anderen Stadium, basierend auf Ereignissen.

- Befehl - ein Symbol, das eine Output-Aufgabe ausführt und verschiedene Parameter hat: Laufzeit, Geschwindigkeit, Entfernung etc
- Zustand - ein Set von Befehlen, der aus einem oder mehreren Befehlen bestehen kann,
- Verbindung - zeigt den möglichen Übergang zu anderen Stadien an (könnte passieren/könnte nicht passieren),
- Übergang - der Akt des Wechsels von einem Stadium in den anderen (= der Akt des Geschehens),
- Bedingung - ein Symbol, das zwei Zahlen vergleicht und feststellt, dass die Ergebnisse richtig oder falsch sind und anzeigt, ob es richtig ist - der Übergang passiert oder falsch ist - nichts passiert.

Sequenzielle Logik – Eine Reihenfolge von Stadien, wobei ein Stadium zum nächsten führt, bis das Programm beendet ist.

Schleife – Eine Folge von Stadien, die von einem zum nächsten führt, die das Programm nicht beendet, sondern den Übergang zurück in den Startzustand (wiederholt N-mal) zur Folge hat.

Parallele Ausführung – Die Ausführung mehrerer Befehle zur gleichen Zeit in einem Stadium.

User Input – Die Daten werden vom Benutzer an das Gerät übermittelt. Dies kann eine digitale Eingabe sein, wie z. B. Text, der auf einem Bildschirm angezeigt werden soll oder er kann physisch sein, wie ein Klicken des Tasters durch den Benutzer oder das Drücken einer Taste auf einer Tastatur.

In **der Robo Live App** - Verwenden der entsprechenden Steuerung zur Steuerung des betreffenden Moduls

In **der Robo Code App**

1) Software Input:

- Parameter von Befehlen wie Laufzeit des Befehls, Helligkeit, Geschwindigkeit, Winkel oder Entfernung,
- Verbindungen die zwischen Befehlen gezogen werden,
- Bedingungen und Parameter von Bedingungen.

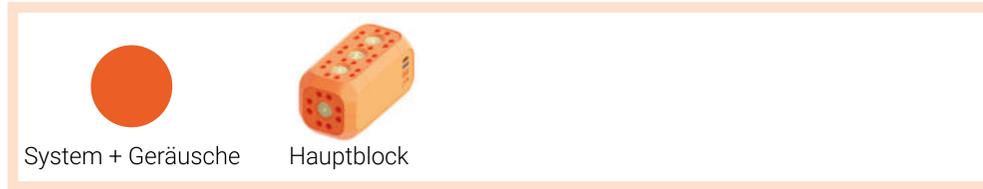
2) Sensoreingang - Daten, die von Sensoren empfangen werden, z. B. Geräuschpegel, Entfernung vor einem Hindernis oder Druck auf den Taster.

User Output – Bei Robo Wunderkind-Modulen wären die Outputs die Antworten auf die Fragen:

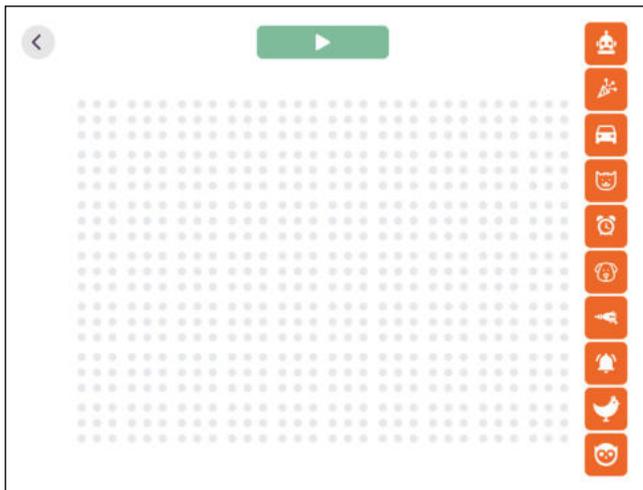
- Was macht das Modul?
- Wie unterscheiden sich Module voneinander?
- Wie arbeiten sie zusammen?
- Wie kombiniere ich die Module, um die Projektziele zu erreichen?

Robo Wunderkind Robotics kit:

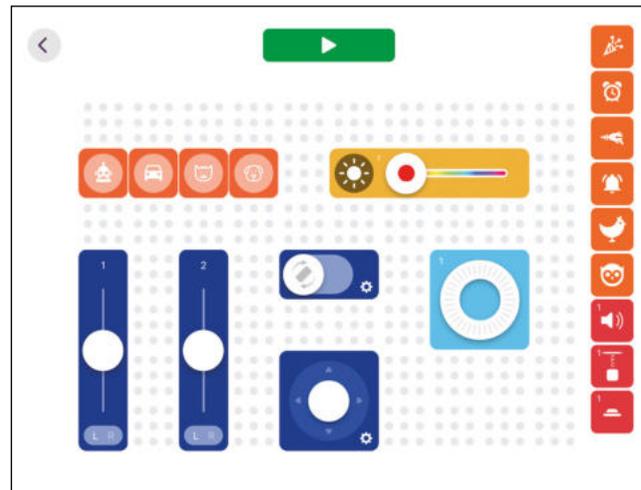
RW Module im Überblick:



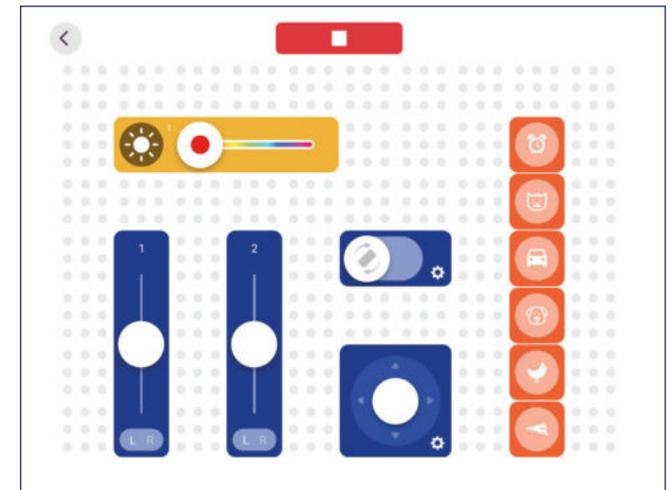
①



②



③



① **Steuerbildschirm** Hier sehen Sie, welche Steuerelemente bereits zu der Steueroberfläche hinzugefügt wurden.

Steuermenü: Hier sehen Sie alle Steuerelemente, die Sie auf zu der Steueroberfläche hinzufügen können. Durch Drag-and-Drop können Sie neue Steuerelemente hinzufügen.

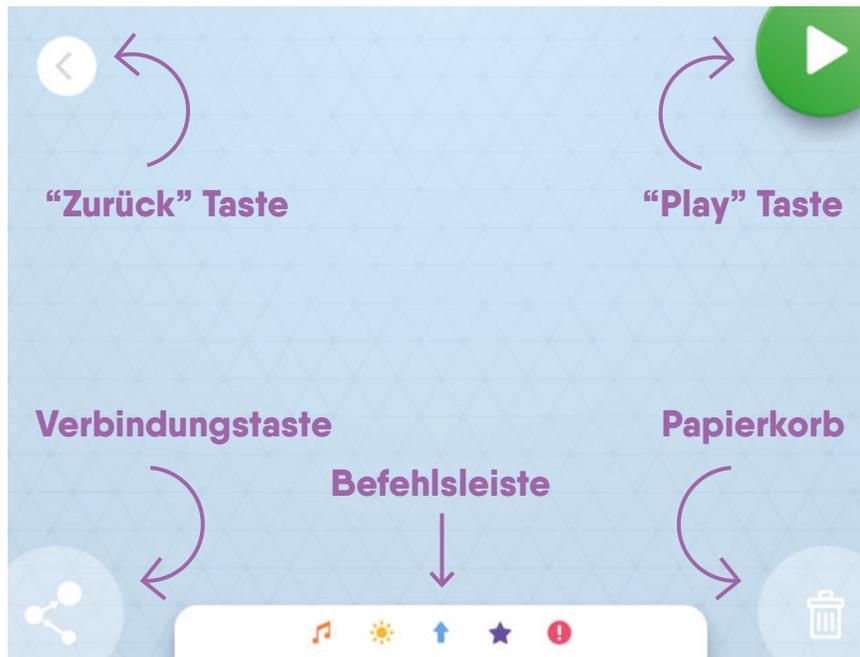
Play / Bearbeitungs - Taste: Wenn Sie bereit sind Ihren Roboter live zu steuern, drücken Sie die Play Taste. Drücken Sie die Bearbeitungs Taste um in den Bearbeitungsmodus zurückzukehren.

Speichern: Drücken Sie auf den "Zurück" Taste um zurück zum "Meine Projekte" Bildschirm zu gelangen. Ihr Projekt wird automatisch gespeichert.

② **Steuerelemente**

- Wenn keine Module mit dem Hauptblock verknüpft sind, können nur Geräusche auf den Steuerbildschirm gezogen werden
- Sobald neue Module mit dem Hauptblock verbunden werden, können die jeweiligen Steuerelemente der einzelnen Module auf den Steuerbildschirm gezogen werden
- Um Steuerelemente zu löschen, können diese einfach wieder zurückgezogen werden

③ **Play Modus** Wenn Sie bereit sind Ihren Roboter live zu steuern, drücken Sie auf die Play Taste. Drücken Sie die Bearbeitungs Taste um in den Bearbeitungsmodus zurückzukehren.



Programmier Bildschirm

Unbegrenzt viel Raum für Programme.

Befehlsleiste: Hier finden Sie alle verfügbaren Befehle.

Verbindungs Taste: Verwenden Sie den Verbindungstaste um einzelne Befehle zu einer Befehlskette (Programm) zu verbinden.

Lösch Modus: Aktivieren und deaktivieren Sie den Lösch-Modus durch das Antippen des Papierkorbs.

"Start" Taste: Starten Sie Ihr Programm hier!

Stop Taste: Wenn ein Programm abgespielt wird, wird aus dem "Play" Taste ein "Stop" Taste. Damit können Sie den Robo jederzeit stoppen.

"Zurück" Taste: Drücken Sie auf den Speichertaste um zurück zum "Meine Projekte" Bildschirm zu gelangen. Das alte Projekt wird automatisch gespeichert.









Karten mit den Schritten des Design Prozesses

Schritt 1. Identifizieren einer Problemstellung:
Was ist das Problem oder die Idee?

Schritt 2.
Brainstorming: Welche Lösungen gibt es?

Schritt 3. Evaluiere und wähle eine Lösung.

Schritt 4. Skizziere und plane: Was wird benötigt?

**Schritt 5. Arbeite an deiner Lösung: Baue,
Programmiere, Teste, Wiederhole!**

Schritt 6. Der Feinschliff:
Ist alles fertig und bereit?

Schritt 7. Präsentation der Lösung.

Schritt 8. Reflektiere: Wie war das Projekt für dich?