

Abb. (1)

HUMANOIDE ROBOTER



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Ein Projekt Der Veranstaltung ‚Informatische Bildung am Gymnasium‘
Professur: Didaktik der Informatik

Katja Michalowski, Markus Sprenger



Verwendet Inhalte des LEGO®-Boost-Sets



INHALT

Kapitel	Thema	Seite
Einführung	Robotik	2
Grundlagen	Algorithmen	3
	Motoren & Sensoren	4
Stationen	Station 0: Eröffnungstanz	5
	Station 1: Parkour-Duell	7
	Station 2: Robo-Drama	9
	Station 3: Companion	11
Hilfe	Befehls-Glossar	13
	Notizen	19

WORUM GEHT ES?

In dieser Broschüre findest du einige Ergänzungen zum Workshop ‚Humanoide Roboter‘. Neben einigen Infos zum Thema Robotik dient sie als Referenz für den **Anwendungspart** der Veranstaltung. Du findest hier eine Kurzanleitung zur LEGO: Boost App, Vernie dem Roboter, sowie die Stations-aufgaben, die du im Workshop erledigen kannst. Als Hilfestellung haben wir ein Glossar erstellt, in dem du alle Befehle der App noch einmal nachlesen kannst.

VERNIE DER ROBOTER

Vernie ist ein humanoide Roboter aus dem LEGO®-Bausystem des **LEGO: Boost Sets**, und gewissermaßen dein **Partner** für die kommenden Stationen. Mit einer symbolbasierten Oberfläche wirst du ihn frei programmieren, und Aktionen ausführen lassen können. Der gerade mal 27cm große Vernie verfügt über Farb-, Abstands- und Neigungssensoren, sowie über programmierbare Motoren, um seine Aufgaben auszuführen.

DIE STATIONEN

Nach Beendigung der Einführungsstation (Station 0) werdet ihr in **Gruppen von je drei Schüler/innen** eine der **drei Hauptstationen** wählen können. Jede Station ist auf einen Aufgabenbereich zugeschnitten, den humanoide Roboter in der Gesellschaft einnehmen können. Traut ihr euch einiges an Programmierfertigkeiten zu, versucht ruhig eine Station mit höherem Schwierigkeitsgrad! **Kreativität** wird euch allerdings bei allen Stationen abverlangt.

WAS DU LERNST

Du gewinnst einen Einblick in die Disziplin der Robotik und ihrer Arbeitsweise.

Du wendest Algorithmische Grundstrukturen zum Programmieren eines Roboters an.

Du kannst Sensoren Anwendungsbereichen zuordnen, und sie zielgerichtet einsetzen.

Du diskutierst die Bedeutung humanoider Roboter in der Gesellschaft.



Abb. (3)



Abb. (2)

ETWAS MENSCHLICHES 'ERSCHAFFEN'

Die Definition eines ‚Roboters‘ variiert stark je nach Gesellschaft, Firma oder Land. Allgemein versteht man unter einem Roboter einen Apparat, der durch Computerprogramme gesteuert wird. Das Ziel ist stets dem Menschen (körperliche) Arbeit abzunehmen. Dabei ist nicht jeder Roboter humanoid! Ein **humanoider Roboter** bedient sich **dem physischen Aufbau und den Eigenschaften menschlicher Körper**, um seine Aufgaben zu verrichten.

In diesem Workshop dient **Vernie** aus dem **LEGO: Boost** Set als Beispiel für einen humanoiden Roboter. Auf den folgenden Seiten im Kapitel ‚Grundlagen‘ findest du kurze Erklärungen zum Thema Algorithmen und Sensoren, die anhand von Vernie und der LEGO: Boost App erklärt werden.

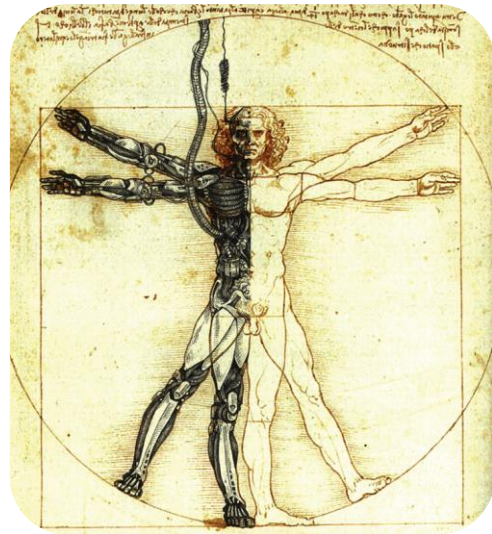


Abb. (4)

EINSATZGEBIETE HUMANOIDER ROBOTER

ASSISTENTEN UND HELFER

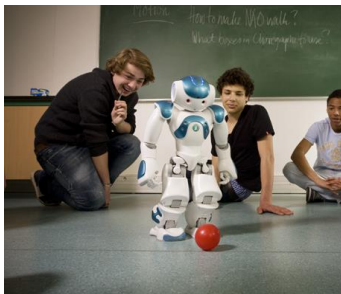


Abb. (5)

Der Roboter **NAO** der französischen Hersteller Softbank Robotics ist beispielsweise in der Bildung tätig. Neben dem aufrechten Gang und Balanceregulation, kann NAO mittels Frontkameras und Mikrofone seine Umgebung erkennen. Er wird vereinzelt in **Universitäten**, aber auch in **Schulen** als Lehrerhilfe oder Lerngegenstand für Kinder mit Lernstörungen eingesetzt.

KÜNSTLICHE ATHLETEN

Humanoide Roboter machen sich insbesondere die Umgebungsstrukturen zu nutze, die ohnehin auf Menschen ausgelegt sind. Ein Beispiel für State-of-the-Art Mobility ist der Roboter **ATLAS** von Boston Dynamics. Mit Fähigkeiten wie das Öffnen von Türen, das Bedienen von Fahrzeugen, Treppensteigen und Balancehaltung soll er vor allem in **Katastrophenfällen** zur Rettung eingesetzt werden.



Abb. (6)

ROBOTER UND EMOTIONEN



Abb. (7)

Auf einer Human Interaction Persuasive Observation Platform (**HIPOP**) der Universität Pisa werden unter anderem Emotionale Interaktionen zwischen Menschen und Roboter untersucht. Dazu nutzt die Forschergruppe den Androiden **FACE**, der emotionale, non-verbale Kommunikation mit präzisen Gesichtsausdrücken ermöglicht. Als **Androiden** bezeichnet man humanoide Roboter, die einen Menschen ästhetisch nachbilden.

KÜNSTLICHE INTELLIGENZ

So wie der Mensch einen Körper und einen Denkapparat besitzt, nutzt ein Roboter seine maschinelle Hardware, **Sensoren**, und eine Basis aus **Algorithmen** zum Verarbeiten der Daten seiner Umwelt. Je komplexer diese Algorithmen aufgebaut sind, umso intelligenter wirkt ein Roboter. Dazu zählt ebenfalls die Lernfähigkeit eines Programmes. So können Roboter sich ihrer Umgebung anpassen, oder mit Unsicherheiten umgehen. Moderne Lösungen nutzen sogenanntes **Deep-Learning**, um ganze neuronale Netze zu modellieren.

Künstliche Intelligenzen, die von heutigen Programmen genutzt werden, bezeichnet man allesamt noch als ‚**schwache KI**‘. Sie konzentrieren sich auf das Lösen eines konkreten Problems. Eine ‚**starke KI**‘, die aus eigenem Antrieb agiert, oder ein Selbstbewusstsein besitzt, ist bis heute nur Gegenstand von Fiktion.



Abb. (8)

ALGORITHMEN

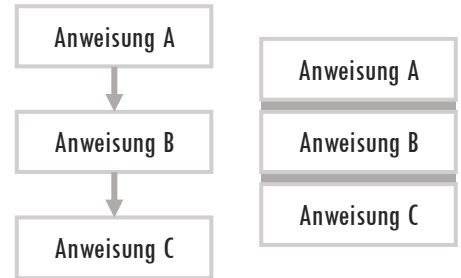
Algorithmen sind Verarbeitungsvorschriften zur Lösung von Problemen, die so exakt formuliert sind, dass sie von Maschinen abgearbeitet werden können. Algorithmen sind eine wesentliche Grundlage von Computerprogrammen. Sie legen letztlich fest, wie der Computer die jeweiligen Aufgaben erledigen soll. Algorithmen kommen aber auch in anderen Bereichen vor. Selbst das tägliche Aufwachen und Aufstehen gleicht so bei vielen Menschen einem Algorithmus.

INFORMATION

„**Sequenzen**“ bezeichnen eine endliche Folge von Anweisungen. In LEGO: Boost wird eine Sequenz immer durch den Programmierbefehl „Start“ ausgeführt, und durch den Befehl „Stopp“ beendet.



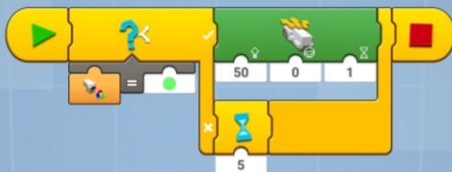
STRUKTOGRAMM



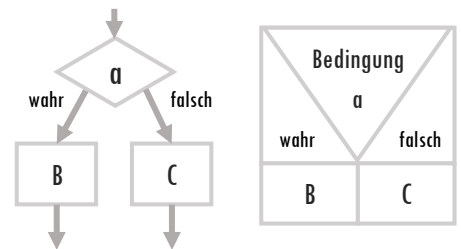
Eine einfache Befehlsfolge, bei der Vernie zunächst gerade aus fährt, und sich anschließend nach rechts dreht.

INFORMATION

Mit „**Verzweigungen**“ kann eine Bedingung an eine Sequenz geknüpft werden. Das gleicht im Grunde einer „Wenn/Dann/Sonst“-Struktur. Wird die Bedingung erfüllt, wird Aktion A ausgeführt, ansonsten wird Aktion B ausgeführt.



STRUKTOGRAMM



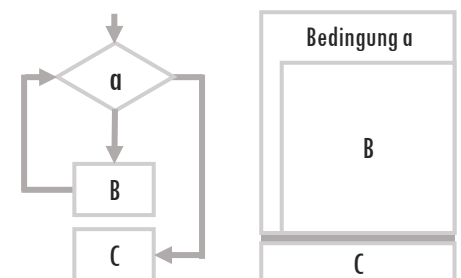
In dieser Verzweigung wird die Sequenz „Geradeausfahren“ nur dann ausgeführt, wenn Vernie grün sieht. Andernfalls wartet er 5 Sekunden.

INFORMATION

„**Schleifen**“ dienen dazu eine Sequenz zu wiederholen, solange eine bestimmte Bedingung erfüllt wird. In der Informatik unterscheiden wir zwischen kopf- und fußgesteuerten Schleifen.



STRUKTOGRAMM



Vernie schüttelt verneinend seinen Kopf, und wartet zwei Sekunden. Dies wiederholt er, so lange er kein grünes Licht sieht. Registriert sein Farbsensor grünes Licht, dann gibt er einen Bestätigungston aus und fährt los.

MOTOREN & SENSOREN

VERNIES MÖGLICHKEITEN

Da Vernie weder über Augen, Ohren oder Gliedmaßen verfügt, benutzt er Sensoren, um sich in seiner Umgebung zurechtzufinden. Mit Sensoren können Maschinen Input aus ihrer Umwelt aufnehmen, und diese in verwertbare Daten umwandeln. So kann Vernies Abstandssensor über Infrarotwellen feststellen, ob sich vor ihm ein Gegenstand befindet oder nicht!

VERNIES MOTOREN & SENSOREN

Einführung

Grundlagen

Stationen

Hilfe

Algorithmen

Motoren

Sensoren

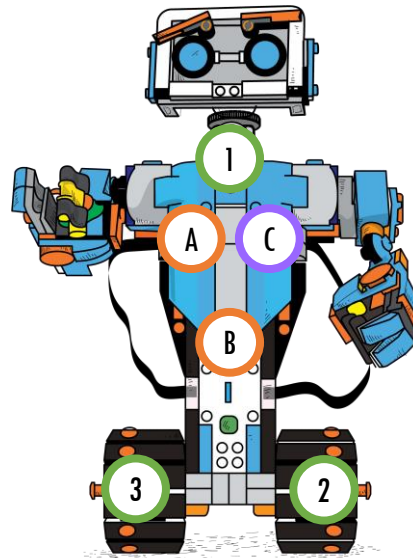


Abb. (9)

1 Kopf-Motor



Vernie bewegt seinen Kopf nach links oder rechts mit einer Geschwindigkeit um eine Gradzahl.

Intervall für Geschwindigkeit:
-100 (links) bis 100 (rechts)
Intervall für Grad:
-9999 (links) bis 9999 (rechts)

2 Linker Fuß-Motor A



Vernie dreht sich nach rechts mit einer einzugebenden Geschwindigkeit für eine vorzugebende Dauer.

Intervall für Geschwindigkeit: -100 bis 100
Intervall für Zeit: 0 bis 9999
Input für 90°: v=30, t=1
Input für 180°: v=65, t=1
Input für 360°: v=35, t=3

3 Rechter Fuß-Motor B



Vernie dreht sich nach links mit einer einzugebenden Geschwindigkeit für eine vorzugebende Dauer.

Intervall für Geschwindigkeit: -100 bis 100
Intervall für Zeit: 0 bis 9999
Input für 90°: v=30, t=1
Input für 180°: v=65, t=1
Input für 360°: v=35, t=3

2 & 3 Antriebsbasis: Beide Fuß-Motoren



Vernie fährt mit einer einzugebenden Geschwindigkeit und einer festzulegenden Lenkung mit einer festzulegenden Dauer.

Intervall für Geschwindigkeit:
-100 bis 100
Intervall für Lenkung:
-100 (links) bis 100 (rechts)
Intervall für Zeit:
0 bis 9999

Beispiel:

Input für 90°:
v= 50, Lenkung -35, t= 1
Input für 360° mit Durchm. (außen) 63 cm:
v= 52, Lenkung -20, t= 6
→ Entsprechend 180°: t=2
Oder: v=20, Lenkung=30, t=12

A Abstandssensor



Mit diesen Aktionen wartet Vernie, bis ein Gegenstand in die Nähe seines Abstandssensors kommt. Auch Helligkeit kann er hierüber wahrnehmen. Die Entfernung, ab der der Abstandssensor reagiert lässt sich einstellen.

B Erschütterungs/Neigungssensor



Mittels des Neigungssensors im Move-Hub kann Vernie reagieren, wenn er aus seiner Lage gebracht wird – Erwa wenn er bewegt wird, umfällt oder der Boden unter ihm zittert. Die Empfindlichkeit lässt sich einstellen.

C Farbsensoren und Lampen



Mit lälfarbenen Befehlen könnt ihr u.a. die Farbe von Vernies zwei Lampen einstellen. Sie verfügen ebenfalls über einen Farbsensor, mit denen Vernie Befehle ausführt, sobald er eine bestimmte Farbe erkennt. Achtet auf genügend Umgebungslicht!

Sensoren des Tablets



Der Vernie-Roboter selbst verfügt über keine Soundausgabe. Er kann sich jedoch den Sensoren eines Tablets oder Smartphones bedienen, und über es Sound aufnehmen oder dessen Neigung wahrnehmen.

Beachtet, dass auch Vernies Motoren als Sensoren funktionieren! Vernie kann Variablen wie die Neigung, Position und Geschwindigkeit seiner einzelnen Motoren wiedergeben. So lassen sich interessante Wenn/Dann-Szenarien bauen! Eine genaue Liste der Variablen und Befehle von Motoren findet ihr ab Seite 16!

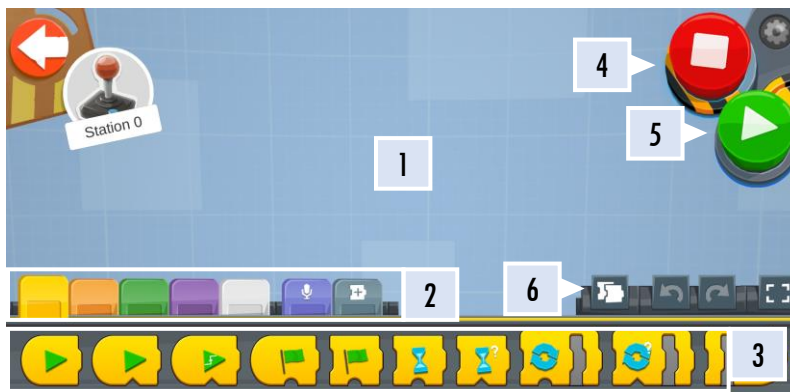
★ ★ ★ STATION 0

BEVOR ES LOSGEHT


- Öffne die **LegoBoost-App** und warte bis die Oberfläche gezeigt wird, in der Vernie und seine Freunde zu sehen sind.
- Scrolle in diesem Fenster ganz rechts und klicke auf den Rollladen mit den Puzzleteilen.
- Hinter der Tafel befindet sich ein Portal. Klicke das Portal an.
- Klicke im folgenden Menü auf das große Plus  und erstelle so ein neues Projekt. Das Projekt öffnet automatisch.
- Um Zugriff auf alle Befehle zu bekommen, stelle sicher, dass die fortgeschrittenen Befehle aktiviert sind. (siehe Abbildung rechts)
- Die App möchte sich nun mit Vernie verbinden. Drücke den grünen Knopf an Vernies Hub und warte bis die Verbindung zustande gekommen ist.





DIE PROGRAMMIEROBERFLÄCHE



1	Arbeitsfläche Hier werden die Sequenzen zusammengesetzt und gestartet
2	Befehlsreiter Hier findest du die Gruppierung der Befehle
3	Befehlszeile Hier findest du die einzelnen Befehlsblöcke
4	Stopp-Button Stoppt alle laufenden Sequenzen
5	Start-Button Startet alle Programme, die sich in der Arbeitsfläche befinden
6	Kopier-Button Kopiert die nächste Sequenz, die angetippt wird

! Vernie schaltet sich automatisch ab, wenn er ein paar Minuten keine Befehle bekommen hat. Um ihn neu zu verbinden, tippe in der App oben rechts auf das Zahnrad , um das Einstellungsfenster zu öffnen. Drücke den grünen Knopf an Vernies Hub. Nun wird Vernie im Einstellungsfenster angezeigt und du kannst ihn auswählen. Warte, bis die Verbindung zustande gekommen ist.

VERNIE PROGRAMMIEREN

Der Roboter Vernie führt die **Sequenzen** aus, die du programmierst. Hier ist zu beachten, dass jede Sequenz (hier manchmal auch **Arbeitsanweisung** genannt) mit einem Start-Befehl  beginnt und einem Stopp-Befehl  endet. In der Sequenz rechts fährt Vernie beispielsweise **für eine Sekunde geradeaus**.



PARAMETER

Viele der Befehle benötigen Parametereingaben, deren **Datentypen** beispielsweise Zahlen, Farben oder booleschen Werten sind. Du erkennst sie an den Einkerbungen am unteren Rand der Befehlsblöcke. In der Sequenz rechts wartet Vernie bis sein Abstandssensor den **Abstand „5“** wahrnimmt und dreht dann seinen Kopf mit der **Beschleunigung „30“** um die **Gradzahl „20“**.



ZIEL

Mache dich mit der Programmierumgebung des LEGO: Boost vertraut. Befolge dazu die einzelnen Arbeitsaufträge und Hinweise.



30 Minuten

Einführung

Grundlagen

Stationen

Hilfe



Station 0

Station 1

Station 2




Station 3


ARBEITSAUFRÄGE

1. Wähle in der Befehlszeile den Befehl „Start“  und ziehe ihn in die Arbeitsfläche.
2. Suche in der Befehlszeile in der App den Befehl „Fahren“ , ziehe ihn in die Arbeitsfläche und verbinde ihn mit dem Startbefehl.

Hinweis: Eine Übersicht zu Vernies Befehlen findest du ab Seite 13


3. Lass Vernie mit der **Geschwindigkeit „20“** und der **Lenkung „0“** für **„2“ Zeiteinheiten** geradeaus fahren.

Hinweis: Um eine Sequenz zu starten, kannst du einfach auf den „Start“-Befehl  tippen. Um alle Sequenzen zu starten, wähle den grünen „Start-Button“  oben rechts. Wenn ein Befehl schnell abgebrochen werden soll, benutze den großen roten „Stopp-Button“  oben rechts.

4. Bringe Vernie dazu, **rückwärts** zu fahren.
5. Bringe Vernie dazu **einen Kreis zu fahren**. (Hilfe dazu findest du in der Motorenübersicht auf Seite 4)
6. Entferne nun den „Fahren“-Befehl  vom Startbefehl und lösche ihn.

Hinweis: Du kannst Befehle löschen, indem du sie einfach zurück in die Befehlszeile ziehst.

7. Suche in der Befehlszeile den Befehl „Drehen-A“  und verbinde ihn mit dem „Startbefehl“.
8. Bringe Vernie dazu **sich um 360° im Kreis zu drehen**.

9. Ziehe einen Befehl „Klangausgabe“  in die Arbeitsfläche. Ziehe einen weiteren „Start“-Befehl in die Arbeitsfläche und verbinde ihn mit dem Klangbefehl.





10. Lass Vernie einen Laut von sich geben.

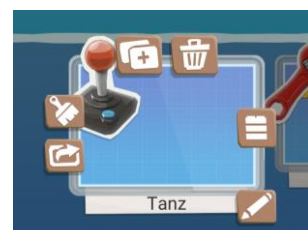
Hinweis: Vernie selbst hat keine Tonausgabe. Die Geräusche, die Vernie machen soll kommen immer aus dem Tablet!

11. Programmiere Vernie so, dass er sich bewegt, und sich darüber freut (Ton- oder Lichtausgabe) dass er sich bewegen kann.
12. Was für eine Bewegung führt Vernie mit der unten abgebildeten Sequenz aus? Habt ihr eine Idee? Dann baut die Sequenz nach und probiert sie aus.



PROJEKT ABSPEICHERN

- Verlasse das Projekt, indem du auf den weißen Pfeil  oben links klickst.
- Wähle zum Öffnen des Bearbeitungsfensters den Schraubenschlüssel  am oberen rechten Projekttrand.
- Mit  kannst du dein Projekt benennen, und mit  ein passendes Motiv wählen.
- Benenne das Projekt um zu **„Tanz“** und bestätige deine Eingabe mit „OK“.
- Zum Schließen des Bearbeitungsfensters klicke in die Mitte des Projekts.



STATION 1

ZIEL

Vernie durchfährt einen Parkour und tritt am Ende in diesem Parkour gegen andere Vernies in einem Wettkampf an. (Das Ziel ist nur erfüllt, wenn Vernie mit **einem** Startbefehl den **vollständigen** Parkour durchfährt!)



100 Minuten

WETTKAMPF

Um am Ende in einem Wettkampf antreten zu können, müsst ihr euch **genau** an die Abstandsvorgaben aus den Parkourplänen halten. Enthalten eure programmierten Sequenzen anspruchsvollere Befehle, bekommt ihr dafür **Zeitboni**. (siehe rechts) Zeitboni gibt es natürlich nur, für **sinnvoll** verwendete Befehle.



Geht es noch besser?

Immer wenn ihr eine Parkourstufe erledigt habt, solltet ihr überlegen, ob man die Sequenz noch besser programmieren oder Befehle einbauen kann, die Zeitboni bringen. Macht ein **Bildschirmfoto** von fertigen Stufen, damit euch keine Informationen verloren gehen.

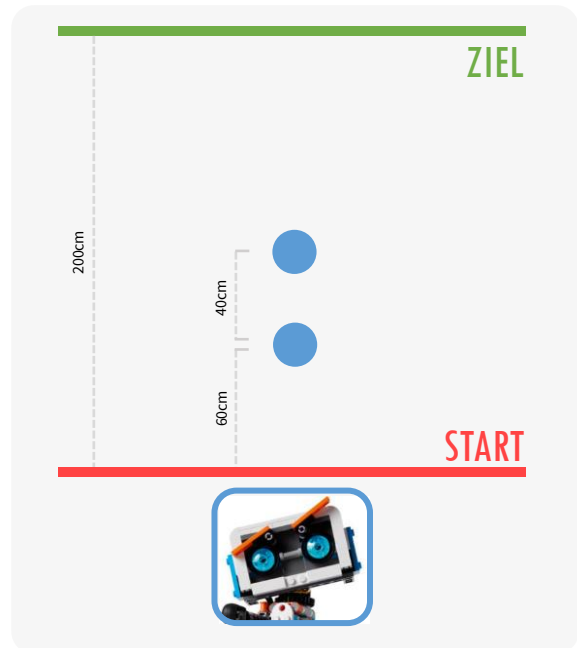
Bild	Bezeichnung	Bonus
	Abstandswahrnehmung	1 sec
	Start (Farbwahrnehmung)	1 sec
	Farbwahrnehmung	1 sec
	Start (Geräuschwahrnehmung)	1 sec
	Geräuschwahrnehmung	1 sec
	Warten auf „wahr“	2 sec
	Verzweigung	2 sec

PARKOUR: STUFE 1

- Baue den Parkour gemäß dem Plan rechts auf.
- Bringe Vernie dazu hinter der Startlinie loszufahren, beide Hindernisse **im Slalom** zu umkreisen und dann die Ziellinie zu passieren.



vgl Abb. (3)



NÜTZLICHE BEFEHLE



Fahren (Geschwindigkeit, Lenkung, Zeit)



Fahren (Geschwindigkeit einzeln)



Warten (auf Zeit)



Linker Fuß (Grad)



Rechter Fuß (Grad)



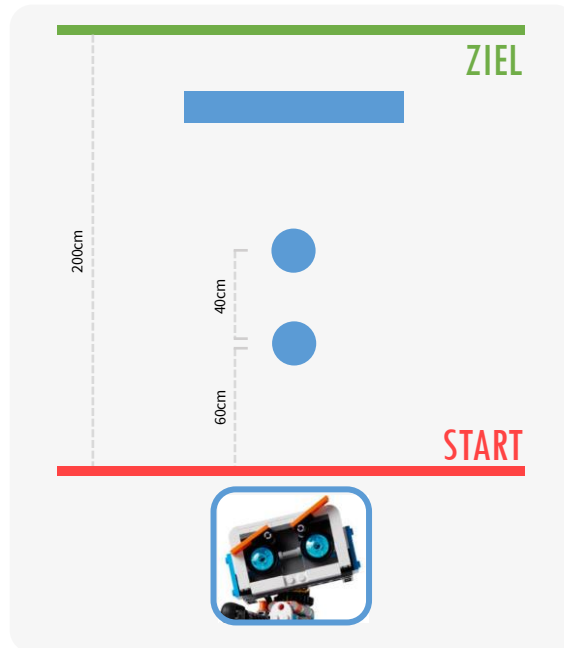
Stopp (Fahren)

PARKOUR: STUFE 2

- Programmier Vernie so, dass er ein großes Hindernis **wahrnimmt** und es **umfährt**.
- Positioniere die Wand gemäß dem Plan rechts.
- Programmier Vernie so, dass er die ersten beiden Hindernisse im Slalom umfährt, dann die Wand wahrnimmt, sie umfährt und dann die Ziellinie erreicht.



Abb. (10)



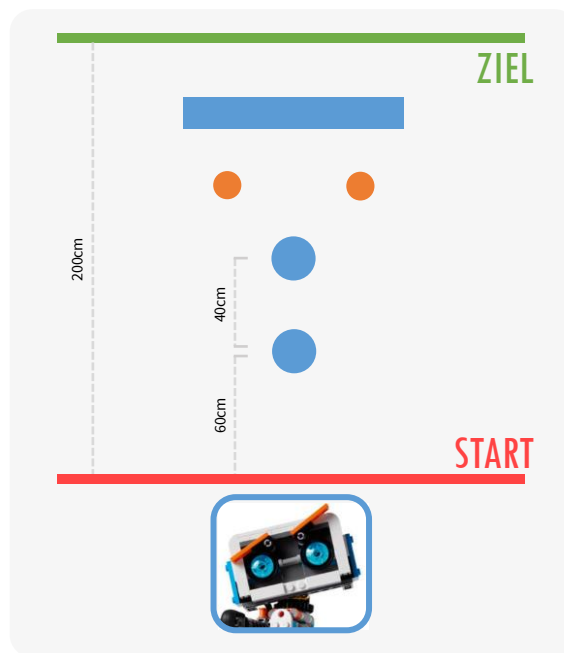
PARKOUR: STUFE 3

- Positioniert die orangenen Hindernisse gemäß dem Plan rechts
- Modifiziert den Programmablauf so, dass Vernie erst die Wand erkennt, die beiden neuen Hindernisse umwirft und dann die Ziellinie erreicht.

Hinweis: Vernie wird nicht immer an genau derselben Stellen versuchen, das Hindernis umzuwerfen. Die Hindernisse können aufgestellt werden, während Vernie schon den Parkour durchfährt.

ABSCHLUSS

- Testet, ob Vernie den Parkour wirklich zuverlässig durchfährt. Kann man einzelne Abschnitte noch besser machen und Befehle einbauen, die **Zeitboni** liefern?
- Macht ein **Video** von eurem Parkourdurchlauf und **stoppt die Zeit**.



NÜTZLICHE BEFEHLE



Verzweigung (Bedingung)



Schleife (Bedingung)



Abstands-Variable



Größer als



Lautstärke



Farbe

STATION 2

ZIEL

In dieser Station könnt ihr **Roboterregisseure** sein. Inszeniert mit euren Vernies kurze Sketche und produziert ein Video in dem die Vernies als **Hauptdarsteller-Duo** auftreten.



100 Minuten

EMOTIONEN FÜR DEN ROBOTER

Programmiert für Vernie folgende ausdrückbare Emotionen. Lasst euch dafür von einem Betreuer erklären, wie Vernie seine **Augenbrauen** bewegen kann. Beginnt mit einer **Tonausgabe** (siehe nützliche Befehle), die zur jeweiligen Emotion passt und überlegt danach, welche Bewegung Vernie dazu machen könnte, um die Emotion zu verstärken.



Freude



Neugier



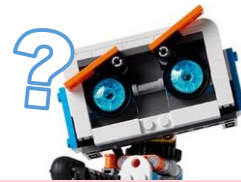
Trauer



Zorn



Scham



Falsch oder echt?

Während Menschen ihre Emotionen auch unbewusst zeigen, ist ein Roboter oder eine Künstliche Intelligenz ein **reiner Schauspieler**. Je vielfältiger ihr Ausdruck, desto natürlicher und menschlicher wirken sie.



Woher die Emotionen nehmen?

Überlegt euch gut mit welchen Mitteln Menschen ihre Emotionen ausdrücken. Vernie ist humanoid, aber nicht zur Gänze Mensch – Seine Mimik ist eingeschränkt, und seine Arme bewegungsunfähig. Daher müsst ihr mit **wenigen Mitteln** besonders **viel Ausdruck** schaffen! Sound, Bewegung, Kopfeigung, Licht, all das sind Kommunikationsarten, die Vernie besitzt. Nehmt euch Zeit, um einzelne Emotionen so genau wie möglich nachzubilden, und macht euch die Perspektive eurer Kamera zu nutze.

EINEN SCHAUPIELER PROGRAMMIEREN

Um das ganze Potential des Roboters zu nutzen, lohnt es sich die folgenden Befehlsblöcke genauer anzuschauen.

Tonaufnahme

Euer Hauptwerkzeug zum Erstellen von Sprachdialogen und Soundeffekten. Unter „Klangausgabe“ auf Seite X findet ihr zusätzliche Filter.



Neigungsmotor

Über Neigung des Kopfes kann Vernie seine Augenbrauen justieren. Die Neigungsgeschwindigkeit reicht von -100 (links) bis 100 (rechts).



Bewegungsmotoren

Mit Neigungen könnt ihr seinen Blick zur Kamera kontrollieren. Subtile und kurze Bewegungen drücken manchmal mehr aus!



Licht und Farbe

Menschen können das zwar nicht, aber jeder weiß was ein Roboter sagt, wenn er rot leuchtet. Nutzt die Farben der beiden LEDs und lasst sie blinken.



Diese Befehle sind nur die Spitze des Eisberges. Eine detaillierte Übersicht findet ihr ab **Seite 13**. Seid kreativ!

NÜTZLICHE BEFEHLE



Verzweigung



Kopf bewegen (Grad)



Kopf bewegen (absolut)



Linker Fuß (Grad)



Rechter Fuß (Grad)



Fahren (Geschwindigkeit, Lenkung, Zeit)

SITUATIONSDRAMA

- Wählt nun eines der folgenden Szenarien aus, und programmiert eure Schauspieler entsprechend! Wenn ihr wollt, könnt ihr natürlich auch eine **eigene Idee** nutzen.
- Schreibt dazu zunächst euer Drehbuch mit einem kurzen Handlungsverlauf, Dialogen und Kameraeinstellungen. Falls ihr möchtet kann auch ein menschlicher Teilnehmer mit Vernie interagieren.
- Die Szene muss nicht lange dauern. Versucht jedoch eine **Pointe** unterzubringen.
- Dreht anschließend das **Video** zu eurer Inszenierung!

Begegnung

1

Rahmenhandlung

Zwei Vernies lernen sich auf einer Party kennen und freunden sich an.

Liebeserklärung

2

Rahmenhandlung

Ein Vernie gesteht einem anderen Vernie seine Liebe. Dieser erwidert sie (oder auch nicht...)

Diebstahl

3

Rahmenhandlung

Ein Vernie wird beschuldigt etwas gestohlen zu haben. Dieser leugnet es und es entsteht eine Prügelei.

Anti-Raucher-Kampagne

4

Rahmenhandlung

Zwei Vernies agieren in einem Werbespot für eine Anti-Raucher-Kampagne.

NÜTZLICHE BEFEHLE

Zufall

Warten (auf wahr)

Verzweigung (Bedingung)

Klang

Lautstärke

Schleife (Bedingung)

Einführung

Grundlagen

Stationen

Hilfe

Station 0

Station 1

Station 2

Station 3

★★★ STATION 3

ZIEL

Programmiere Vernie so, dass er zu einem **nützlichen** Freund und Helfer im Haushalt wird, wobei er verschiedene kleine Aufgaben auf **Kommando** übernehmen können soll.



100 Minuten

'MITFÜHLENDE' ROBOTER

Statt der Sinnesorgane, die wir Menschen verwenden, müssen Maschinen auf Sensoren vertrauen, um ihre Umgebung wahrzunehmen. Nicht nur Vernie, sondern alle möglichen Maschinen nutzen solche Sensoren – Darunter zum Beispiel Staubsaugerroboter, Autos, Messinstrumente und nicht zuletzt das Smartphone.

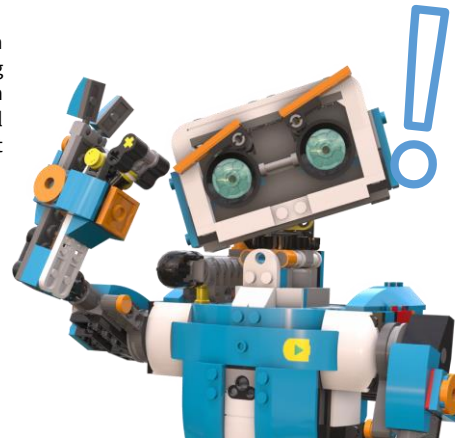


Abb. (11)



Die Technik dahinter

Einem Sensor liegt stets ein physikalisches Wirkungsprinzip zu Grunde. Manche Bauteile geben unter äußerem Einfluss elektrische Signale ab, wie etwa bei einer Solarzelle. Bei anderen Bauteilen verändern sich lediglich ihre Parameter, wie etwa bei Dehnungsmessstreifen einer Waage.

Messgegenstand	Wirkprinzip	Sensor-Beispiel
Abstand	Thermoelektrisch	Bewegungsmelder (Pyroelek. IR-Sensor)
Farbe	Optisch	RGB-Farbsensor (mittels Photodioden)
Neigung	Induktiv	Neigungssensor, Gyroskop-Sensor
Lautstärke	Akustisch	Mikrofon
Druck	Kapazitiv	Druck- und Regensensor
Temperatur	Thermoelektrisch	Thermoelement
Ausrichtung	Magnetisch	Kompassensensoren

VERNIES SENSOREN

Auch Vernie bedient sich dieser Technologie, um seine Umgebung erkennen zu können. Seine Sensoren sind zwar nicht so vielfältig, wie es bei einem Menschen der Fall ist, arbeiten dafür aber weitaus exakter. So haben auch Menschen einen Neigungssensor in Form ihres Gleichgewichtssinns, können jedoch nur selten ihren exakten Neigungswinkel mit Nachkommastellen angeben.

Eine Erklärung zu Vernies Sensoren findest du in der Einführung auf Seite 4.

Optisch	Abstandssensor
Optisch	Farbsensor
Induktiv	Neigungssensor
Induktiv	Motorsensoren
Andere	Mikrofon (via Smartphone od. Tablet!)

NÜTZLICHE BEFEHLE



Farbe



Abstand



Lautstärke



Hub-Ausrichtung



Tablet-Neigung X



Fahr-Variable (Position)

WHAT IS MY PURPOSE?

- Programmiert nun Vernie mit Hilfe seiner **Sensoren** so, dass er kleine Aufgaben übernehmen kann.
- Sucht euch zunächst **einen Arbeitsauftrag** aus und erstellt einen **Plan**, mit welchen Motoren und Sensoren Vernie diese Aufgabe erfüllen könnte.
- Achte darauf, keine programmierten Sequenzen zu löschen! Man kann sie später vielleicht noch gebrauchen. Ihr könnt für jeden Arbeitsauftrag ein **neues Projekt** anlegen.
- Nehmt jedes fertiggestellte Programm auf Video auf.

ARBEITSAUFTRÄGE

- 1. Alarmanlage**
 Programmiere Vernie so, dass er reagiert, wenn er jemanden mit seinem Abstandssensor wahrnimmt.
 Programmiere Vernie so, dass er reagiert, wenn er ein Geräusch wahrnimmt.
- 2. Diebstahlschutz**
 Programmiere Vernie so, dass er reagiert, wenn er bewegt wird.
- 3. Flucht**
 Programmiere Vernie so, dass er eine Wand absucht, bis er die Tür gefunden hat.
- 4. Buttler**
 Programmiere Vernie so, dass er aus verschieden-farbigen Gegenständen den richtigen auswählt.
- 5. Labyrinth**
 Programmier Vernie so, dass er einen Weg fährt und diesen dann von alleine wieder zurück findet.

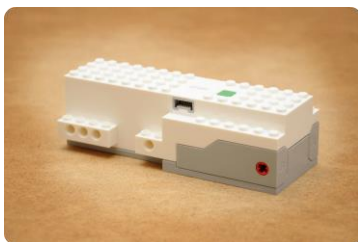
ABSCHLUSS

Kombiniert Arbeitsaufträge, um Programme zu erstellen, die Vernie wie einen tatsächlichen Buttler wirken zu lassen. Nutzt hierfür „**Warten**“-**Befehle**, sowie Start-Befehle, die an die Sensoren gekoppelt sind, sodass Vernie erst auf Kommando reagiert.

Sucht den besten Arbeitsauftrag aus den erledigten Aufgaben aus, um ihn in der Abschlussrunde euren Mitschülern präsentieren zu können.

UNTERSCHIEDE DER BEFEHLSBLÖCKE

Block	Bedeutung
	Führe Befehl aus WENN Farbe = X
	Warte BIS Farbe = X
	Parameter = Farbe, die vom Sensor wahrgenommen wird



Move-Hub

Abb. (12)



Motor-Sensor

Abb. (13)



Farb- & Abstands-Sensor

Abb. (14)

NÜTZLICHE BEFEHLE



Start (Bedingung)



Warten (auf wahr)



Verzweigung (Bedingung)



Addition






Kleiner als



Schleife (Bedingung)

BEFEHLS-GLOSSAR

Datentypen / Anschlüsse		
Bild	Name	Werte
	Zahlenwert	... -2, -1, 0, 1, 2 ...
	Boolescher Wert	wahr, falsch
	Farbwert	rot, grün, blau ...






Parameter		
Symbol	Name	Mögliche Werte
	Zeit	0 bis 9999
	Geschwindigkeit	-100 bis 100
	Lenkung	-100 (links) bis 100 (rechts)
	Gradzahl (relativ)	-9999 (links) bis 9999 (rechts)
	Gradzahl (absolut)	-9999 (links) bis 9999 (rechts)

Bild	Name	Beschreibung
Steuerelemente		
	Start	Startet die verbundene Sequenz
	Start (Bedingung)	Startet die verbundene Sequenz, wenn die anliegende Bedingung wahr ist.
	Start (Bedingung/Grenze)	Startet die Ausführung der verbunden Sequenz, wenn sich die Bedingung von „falsch“ zu „wahr“ ändert. Wird unterbrochen, wenn sie erneut ausgelöst wird.
	Start (Flagge)	Startet die Ausführung der verbunden Sequenz, wenn die Flaggennummer aktiviert wird.
	Aktivierung (Flagge)	Aktivieren der Flaggennummer
	Stopp (alle anderen Sequenzen)	Unterbricht alle anderen Programmsequenzen und führt die verbundene Sequenz aus
	Stopp	Stoppt die verbundene Sequenz
Warten		
	Warten (auf Zeit)	Vernie wartet für eine einzustellende Zeit (ca. in Sekunden)
	Warten (auf wahr)	Vernie wartet, bis die angeschlossene Bedingung wahr ist.
	Timer	Es zählt ein Timer im Hintergrund mit. Die Zeit muss vorher in Sekunden eingestellt werden. Er zählt ab der eingestellten Zeit aufwärts.
	Timer-Variable	Angabe des aktuell eingestellten Timer-Werts
	Timer Stopp	Anhalten des Timers
Schleifen und Verzweigungen		
	Schleife (Anzahl)	Wiederholen der enthaltenen Sequenz, Parameter notwendig
	Schleife (Bedingung)	Wiederholen der enthaltenen Sequenz, solange die Bedingung wahr ist
	Schleife (Endlos)	Wiederholen der enthaltenen Sequenz in Endlosschleife
	Verzweigung (Bedingung)	Wenn die Bedingung wahr ist, Ausführen der obersten Sequenz, sonst Ausführen der untersten Sequenz
	Verzweigung	Es werden beide Sequenzen gleichzeitig ausgeführt

Bild	Name	Beschreibung
Motoren: Move-Hub und Kopf		
	Kopf bewegen (Zeit)	Vernie bewegt seinen Kopf nach links oder rechts mit einer Geschwindigkeit um eine Gradzahl. Intervall für Geschwindigkeit: -100 (links) bis 100 (rechts) Intervall für Zeit: 0 bis 9999
	Kopf bewegen (Grad)	Vernie bewegt seinen Kopf nach links oder rechts mit einer Geschwindigkeit um eine Gradzahl. Intervall für Geschwindigkeit: -100 (links) bis 100 (rechts) Intervall für Grad: -9999 (links) bis 9999 (rechts)
	Kopf bewegen (Absolut)	Vernie dreht seinen Kopf in eine vorher einzustellende Position. Dreht allerdings Überdrehungen auch wieder aus.
	Kopf bewegen (Geschwindigkeit)	Vernie bewegt seinen Kopf nach links oder rechts mit einer Geschwindigkeit. Intervall für Geschwindigkeit: -100 (links) bis 100 (rechts)
	Stopp (Kopf bewegen)	Stoppt die Bewegung des Kopfes
	Stopp (Kopf bewegen), Auslauf	Stoppt die Bewegung des Kopfes und lässt die Bewegung auslaufen
	Stopp (Kopf bewegen), Position	Stoppt die Bewegung des Kopfes und behält die Position
	Obergrenze (Kopf bewegen)	Legt die maximale Leistung der Kopfbewegung fest (0 bis 100)
	Beschleunigungsdauer (Kopf bewegen)	Festlegen der Beschleunigungsdauer (in Sekunden) für Kopfbewegung
	Bremsdauer (Kopf bewegen)	Festlegen der Bremsdauer (in Sekunden) für Kopfbewegung
	Fahren (Geschwindigkeit einzeln, Zeit)	Betreiben der Antriebsbasis im Modus „Hebellenkung“ mit einer Geschwindigkeit von Motor A (-100 bis 100) und einer Geschwindigkeit von Motor B (-100 bis 100) für eine bestimmte Zeit (in Sekunden)
	Fahren (Geschwindigkeit einzeln, Grad)	Betreiben der Antriebsbasis im Modus „Hebellenkung“ mit einer Geschwindigkeit von Motor A (-100 bis 100) und einer Geschwindigkeit von Motor B (-100 bis 100) für eine bestimmte Entfernung (in Grad)
	Fahren (Geschwindigkeit, Lenkung, Zeit)	Betreiben der Antriebsbasis im Modus „Steuerradlenkung“ mit einer Antriebsbasis-Geschwindigkeit (-100 bis 100) und Lenkrichtung (-100 bis 100) für eine bestimmte Zeit (in Sekunden)
	Fahren (Geschwindigkeit, Lenkung, Grad)	Betreiben der Antriebsbasis im Modus „Steuerradlenkung“ mit einer Antriebsbasis-Geschwindigkeit (-100 bis 100) und Lenkrichtung (-100 bis 100) für eine bestimmte Entfernung (in Grad)
	Fahren (Geschwindigkeit einzeln)	Starten der Antriebsbasis im Modus „Hebellenkung“ mit einer Geschwindigkeit von Motor A (-100 bis 100) und einer Geschwindigkeit von Motor B (-100 bis 100)
	Fahren (Geschwindigkeit, Lenkung)	Starten der Antriebsbasis im Modus „Steuerradlenkung“ mit einer Antriebsbasis-Geschwindigkeit (-100 bis 100) und Lenkrichtung (-100 bis 100)
	Stopp (Fahren)	Anhalten der Antriebsbasis
	Stopp (Fahren), Auslauf	Anhalten der Antriebsbasis und Motoren auslaufen lassen
	Stopp (Fahren), Position	Anhalten der Antriebsbasis und beibehalten der Positionen der Motoren
	Obergrenze (Fahren)	Festlegen der max. Leistung (0 bis 100) der Antriebsbasis
	Beschleunigungsdauer (Fahren)	Festlegen der Beschleunigungsdauer (in Sekunden) für die Antriebsbasis
	Bremsdauer (Fahren)	Festlegen der Bremsdauer (in Sekunden) für die Antriebsbasis

Einführung

Grundlagen

Stationen

Hilfe

Glossar

Notizen

BEFEHLS-GLOSSAR

Bild	Name	Beschreibung
Motoren: Linker Fuß (Motor A) und Rechter Fuß (Motor B)		
	Linker Fuß (Zeit)	Festlegen der Bremsdauer (in Sekunden) für den Motor A
	Linker Fuß (Grad)	Betreiben von Motor A mit einer Geschwindigkeit (-100 bis 100) für eine bestimmte Dauer (in Sekunden)
	Linker Fuß (Absolut)	Betreiben von Motor A mit einer Geschwindigkeit (-100 bis 100) für eine bestimmte Entfernung (in Grad)
	Linker Fuß (Geschwindigkeit)	Starten von Motor A mit einer bestimmten Geschwindigkeit (-100 bis 100)
	Stopp (Linker Fuß)	Anhalten von Motor A
	Stopp (Linker Fuß), Auslauf	Anhalten von Motor A und Motor auslaufen lassen
	Stopp (Linker Fuß), Position	Anhalten von Motor A und beibehalten der Motorposition
	Obergrenze (Linker Fuß)	Festlegen der max. Leistung (0 bis 100) von Motor A
	Beschleunigungsdauer (Linker Fuß)	Festlegen der max. Leistung (0 bis 100) von Motor A
	Bremsdauer (Linker Fuß)	Festlegen der Beschleunigungsdauer (in Sekunden) für Motor A
	Rechter Fuß (Zeit)	Festlegen der Bremsdauer (in Sekunden) für den Motor B
	Rechter Fuß (Grad)	Betreiben von Motor B mit einer Geschwindigkeit (-100 bis 100) für eine bestimmte Dauer (in Sekunden)
	Rechter Fuß (Absolut)	Betreiben von Motor B mit einer Geschwindigkeit (-100 bis 100) für eine bestimmte Entfernung (in Grad)
	Rechter Fuß (Geschwindigkeit)	Starten von Motor B mit einer bestimmten Geschwindigkeit (-100 bis 100)
	Stopp (Rechter Fuß)	Anhalten von Motor B
	Stopp (Rechter Fuß), Auslauf	Anhalten von Motor B und Motor auslaufen lassen
	Stopp (Rechter Fuß), Position	Anhalten von Motor B und beibehalten der Motorposition
	Obergrenze (Rechter Fuß)	Festlegen der max. Leistung (0 bis 100) von Motor B
	Beschleunigungsdauer (Rechter Fuß)	Festlegen der Beschleunigungsdauer (in Sekunden) für Motor B
	Bremsdauer (Rechter Fuß)	Festlegen der Bremsdauer (in Sekunden) für den Motor B

Bild	Name	Beschreibung
Farbsensor		
	Start (Farbe)	Auslösen einer Aktion, wenn der Sensor eine bestimmte Farbe misst
	Farbe	Warten, bis die vom Sensor gemessene Farbe einer bestimmten Farbe entspricht
	Farb-Variable	Informiert über die aktuelle Farbe, gemessen vom Sensor
Abstandssensor		
	Start (Abstand)	Auslösen einer Aktion, wenn die vom Sensor gemessene Entfernung kürzer als eine bestimmte Entfernung ist
	Abstand	Warten, bis die vom Sensor gemessene Entfernung kürzer ist als eine bestimmte Entfernung
	Abstands-Variable	Informiert über die aktuelle Entfernung, gemessen vom Sensor
	Helligkeits-Variable	Informiert über die aktuelle Stärke des Umgebungslichts, gemessen vom Sensor
Mikrofon		
	Start (Lautstärke)	Auslösen einer Aktion, wenn die Lautstärke höher als eine bestimmte Lautstärke ist
	Lautstärke	Warten, bis die Lautstärke höher ist als eine bestimmte Lautstärke
	Lautstärken-Variable	Informiert über die aktuelle Lautstärke
Erschütterungssensor		
	Erschütterungssensor	Festlegen der Empfindlichkeit (0 bis 10) des Erschütterungssensors in der Antriebsbasis
	Erschütterungs-Variable (Boolesch)	Informiert über den aktuellen booleschen Zustand des Erschütterungssensors
	Erschütterungs-Variable (Anzahl)	Zählen der Anzahl der Erschütterungen des Hubs
Ausrichtung		
	Start (Hub-Ausrichtung)	Auslösen einer Aktion, wenn eine bestimmte Hub-Ausrichtung vorliegt
	Hub-Ausrichtung	Warten, bis die Ausrichtung des Hubs einer bestimmten Hub-Ausrichtung entspricht
	Hub-Ausrichtungs-Variable	Informiert über die aktuelle Ausrichtung des Hubs
	Hub-Ausrichtungs-Variable X	Informiert über den aktuellen Neigungswinkel des Hubs auf der X-Achse (-90 bis 90)
	Hub-Ausrichtungs-Variable Y	Informiert über den aktuellen Neigungswinkel des Hubs auf der Y-Achse (-90 bis 90)
	Hub-Ausrichtung (Absolut)	Konfigurieren der Ausrichtung des Hub-Neigungssensors
	Kopf-Drehen-Variable (Geschwindigkeit)	Informiert über die aktuelle Geschwindigkeit des externen Motors
	Kopf-Drehen-Variable (Position)	Informiert über die aktuelle Position des externen Motors
	Kopf Drehen (Absolut)	Voreinstellung der Position des externen Motors auf eine bestimmte Zahl

Einführung

Grundlagen

Stationen

Hilfe

Glossar

Notizen

BEFEHLS-GLOSSAR

Bild	Name	Beschreibung
Ausrichtung		
	Fahr-Variable (Geschwindigkeit)	Informiert über die aktuelle Geschwindigkeit der Antriebsbasis
	Fahr-Variable (Position)	Informiert über die aktuelle Position der Antriebsbasis
	Fahren (Position)	Voreinstellung der Basisantriebsposition auf eine bestimmte Zahl
	Linker-Fuß-Variable (Geschwindigkeit)	Informiert über die aktuelle Geschwindigkeit von Motor A
	Linker-Fuß-Variable (Position)	Informiert über die aktuelle Position von Motor A
	Linker Fuß (Position)	Voreinstellung der Position des Motors A auf eine bestimmte Zahl
	Rechter-Fuß-Variable (Geschwindigkeit)	Informiert über die aktuelle Geschwindigkeit von Motor B
	Rechter-Fuß-Variable (Position)	Informiert über die aktuelle Position von Motor B
	Rechter Fuß (Position)	Voreinstellung der Position des Motors B auf eine bestimmte Zahl
	Tablet-Neigung X	Informiert über den aktuellen Neigungswinkel des Tablets auf der Längsachse
	Tablet-Neigung Y	Informiert über den aktuellen Neigungswinkel des Tablets auf der Querachse
Licht und Farbe		
	Lampe: Hub-LED	Einstellen der Farbe der Hub-LED (lila, blau, hellblau, türkis, grün, gelb, orange, rot, weiß)
	Lampe: Sensor-LED	Einstellen der Farbe der LED des Sensors (blau, grün, rot und weiß)
Tonausgabe		
	Klang	Abspielen des ausgewählten Klangs aus der Klangbibliothek
	Klang (Filter)	Abspielen des ausgewählten Klangs aus der Klangbibliothek mit den angewandten Filtern
	Filter	Einstellen des Tonhöhenfilters (-100 bis 100) für das Abspielen eines Klangs mit dem Filterblock
	Verzerrer	Einstellen des Verzerrungsfilters (0 bis 100) für das Abspielen eines Klangs mit dem Filterblock
	Echo	Einstellen des Echofilters (0 bis 100) für das Abspielen eines Klangs mit dem Filterblock
	Reset (Filter)	Zurücksetzen aller Klangfilter, die beim Abspielen eines Klangs mit dem Filterblock verwendet wurden
	Lautstärke	Festlegen der Lautstärke für alle Blocks auf eine bestimmte Stufe (0 bis 100)
	Tonaufnahme	Aufnehmen eines eigenen Tones oder Geräusches

Bild	Name	Beschreibung
Operatoren		
	Zufall	Ergibt eine Zufallszahl zwischen Zahl 1 und Zahl 2 (beide Zahlen im Bereich)
	Gleich	Erfassen, ob eine Zahl gleich einer anderen Zahl ist
	Kleiner als	Erfassen, ob eine Zahl kleiner als eine andere Zahl ist
	Größer als	Erfassen, ob eine Zahl größer als eine andere Zahl ist
	Ungleich	Erfassen, ob eine Zahl ungleich einer anderen Zahl ist
	UND	Angabe des Werts „wahr“, wenn beide Operanden wahr sind und „falsch“ bei zwei falschen Operanden
	ODER	Angabe des Werts „wahr“, wenn ein Operand oder beide Operanden wahr sind bzw. „falsch“ bei einem oder zwei falschen Operanden
	Addition	Hinzuzählen einer Zahl zu einer anderen Zahl
	Subtraktion	Malnehmen einer Zahl mit einer anderen Zahl
	Multiplikation	Malnehmen einer Zahl mit einer anderen Zahl
	Division	Teilen einer Zahl durch eine andere Zahl
	Andere Mathematische Befehle	Auf nächste Zahl runden, Aufrunden, Abrunden, Absolutwert, Wurzelziehen, Quadrieren, Sinus, Kosinus, Tangens, Arkussinus, Arkuskosinus und Arkustangens
	Potenzieren	Potenz aus der Zahl zu seinem Exponenten
	Maximum	Höchstwert von einer Zahl und einer anderen Zahl
	Minimum	Mindestwert von einer Zahl und einer anderen Zahl
	Modulo	Rest nach einer Division von einer Zahl durch eine andere Zahl
Eigene Sequenzen und Parameter		
	Eigene Sequenz	Damit kann ein eigener Modulblock erstellt werden. Ihm können bis zu 4 Parameter (siehe unten) übergeben werden. Die Anzahl und Art der Parameter wird beim erstellen des Modulblocks festgelegt.
	Parameter 1 (Boolesch)	Erster Modulparameter soll vom Typ Boolean (wahr / falsch) sein. ACHTUNG, man kann den Typ nach dem ersten setzen nicht mehr ändern.
	Parameter 1 (Zahl)	Erster Modulparameter soll vom Typ Zahl sein. ACHTUNG, man kann den Typ nach dem ersten Setzen nicht mehr ändern.
Variablenmanipulation		
	Variable anzeigen (lokal)	Anzeigen der Zahl, die in der lokalen Variable gespeichert ist
	Variable schreiben (lokal)	Aktualisieren der lokalen Variable zum Speichern einer Zahl
	Variable anzeigen (global)	Anzeigen der Zahl, die in der globalen Variable gespeichert ist
	Variable schreiben (global)	Aktualisieren der globalen Variable zum Speichern einer Zahl

Einführung

Grundlagen

Stationen

Hilfe

Glossar

Notizen

NOTIZEN



Einführung

Grundlagen

Stationen

Hilfe

Glossar

Notizen

Bildquellen: (Abrufdatum:01.02.2020)

- (1) <https://pixabay.com/de/photos/verbindung-hand-menschliche-roboter-3308188/>
- (2) https://lan.lego.com/gallery/image/133-lego_boost_logojpg/
- (3) <https://www.lego.com/de-at/product/boost-creative-toolbox-17101>
- (4) <https://www.deviantart.com/alistair-d-borthwick/art/Ren-Man-2010-30129645>
- (5) <https://www.nexxt.com/articles/nao-robots-reach-and-teach-autistic-children-11957-article.html>
- (6) [https://de.wikipedia.org/wiki/Atlas_\(Roboter\)#/media/Datei:Atlas_from_boston_dynamics.jpg](https://de.wikipedia.org/wiki/Atlas_(Roboter)#/media/Datei:Atlas_from_boston_dynamics.jpg)
- (7) http://www.faceteam.it/wp-content/uploads/2010/01/IMG_2289_715x600.jpg
- (8) https://www.itreseller.ch/Artikel/art_bildergalerie.cfm?aid=86474&arts_idx=90665
- (9) https://www.spielemax.de/media/catalog/product/cache/1/image/500x/9df78eab33525d08d6e5fb8d27136e95/5/0/500-17101_01.jpg
- (10) https://www.jing.fm/clipimg/full/42-422109_lego-boost-creative-toolbox-vernier-robot.png
- (11) https://mir-s3-cdn-cf.behance.net/project_modules/max_1200/3dca6d65545483.5af868c132022.png
- (12) <https://www.robocamp.eu/en/blog/lego-boost-review/boost-electronics-hub.png>
- (13) <https://www.robocamp.eu/en/blog/lego-boost-review/boost-electronics-motor.png>
- (14) <https://www.robocamp.eu/en/blog/lego-boost-review/boost-electronics-sensor.png>

Icons und Screenshots wurden übernommen aus der LEGO:Boost App (Android)

Abbildungen und Videoszenen der Broschüre werden nach Anwendung des Urheberrechtsgesetzes, spezifisch §60a,b unter Werke geringem Umfangs für die Erstellung von Lehr- und Lernmaterialien genutzt.