Aufgabenblatt 4 - Variablen

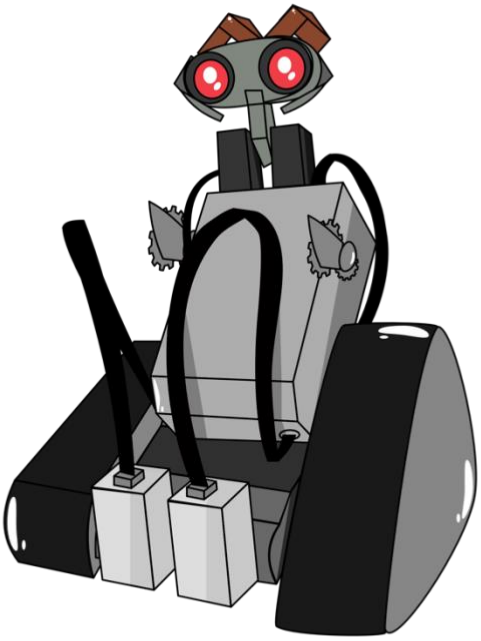
Im vierten Aufgabenblatt geht es darum, Variablen in Open Roberta kennenzulernen. Dazu lernt ihr im ersten Teil die Grundlagen von Variablen und könnt das Wissen dann im zweiten Teil direkt in Open Roberta ausprobieren.

Bei dem Beispiel zu den Binärzahlen hat sich in der ersten Version ein Zahlenfehler eingeschlichen, dieser ist mit dieser Version aktualisiert.

Bei Fragen meldet euch wie immer gerne bei robotikwettbewerb@in.tum.de.

Abgabe für alle Blätter ist der **09.05.2021, 20:00**.

Viel Spaß und Erfolg!



Theorie

In Open Roberta kann man im Startblock mit Klick auf das Pluszeichen eine neue Variable erzeugen.

Eine Variable hat einen **Namen**, einen **Datentyp** und einen **Wert**.

Die Variable in dem Beispiel hat den Namen <*Element*>, den Datentyp Zahl und den Wert 0.
Variablen sollten immer einen aussagekräftigen Namen haben! <*Element*> ist also kein guter Name.

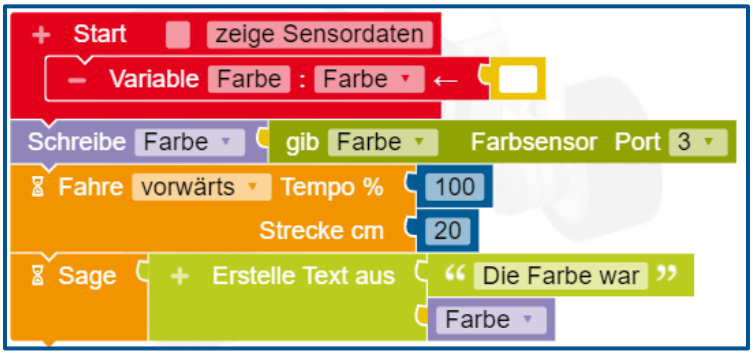
Blöcke in Open Roberta

Für Variablen gibt es in Open Roberta zwei verschiedene Blöcke.

	Mit diesem Block lässt sich der Wert einer Variablen ändern. Da es sich bei der Variablen <*Element*> um eine Zahl handelt, kann man hier nur blaue Mathematik-Blöcke andocken.
	Mit diesem Block lässt sich der Wert einer Variablen ausgeben.

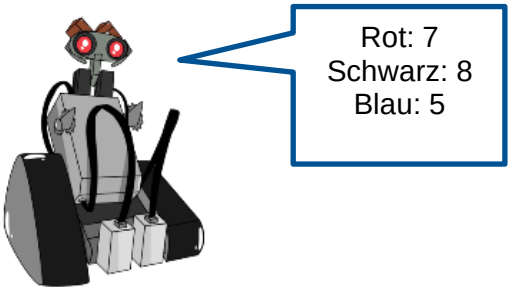
Warum Variablen?

Mithilfe von Variablen lassen sich leicht Werte speichern, die man später noch einmal braucht. In diesem Beispiel-Programm speichert der Roboter eine Farbe, welche sein Farbsensor wahrnimmt. Dann fährt er kleine Strecke und sagt anschließend, welche Farbe er davor gesehen hat, selbst wenn der Roboter jetzt auf einem andersfarbigen Untergrund steht.

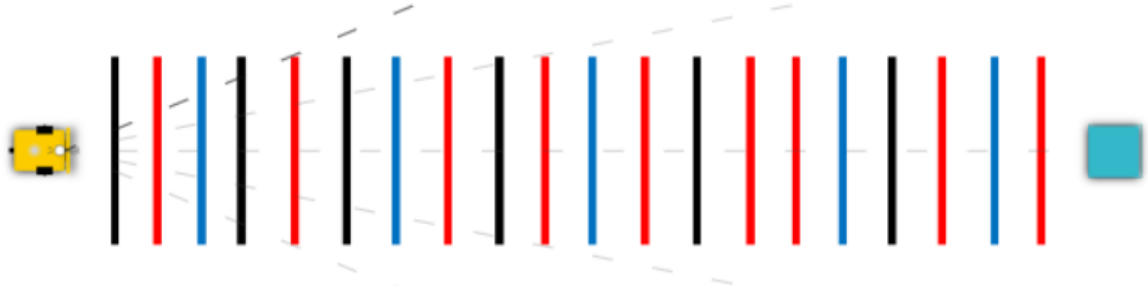


Aufgabe 4.1: Striche zählen

Programmiert einen Roboter, der Striche auf dem Boden mithilfe seines Farbsensors erkennt und zählt. Der Roboter soll in gerader Linie über verschiedenfarbige Striche fahren und zählen, wie viele Striche von jeder Farbe ihm begegnen.
Am Ende soll der Roboter mithilfe seines Berührungssensors vor dem blauen Block anhalten und anschließend wiedergeben, wie viele Striche von jeder Farbe er gezählt hat.



1. Ladet zunächst den Hintergrund *Aufgabe4.1.png* für die Aufgabe 4 in die Simulationsansicht von Open Roberta und verschiebt den blauen Block und den Roboter in die vorgegebenen Felder.



2. Überlegt euch nun, wie ihr mithilfe von 3 Variablen die schwarzen, roten und blauen Striche zählen könnt.

Nutzt den Block ***Schreibe*** zusammen mit einem **Mathematik-Block,** um den Wert von Variablen zu verändern:

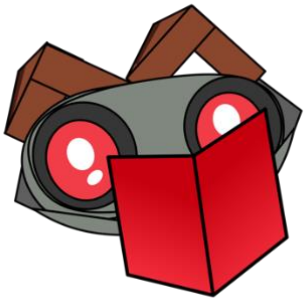
In diesem Beispiel werden von der Zahl in der Variablen <*Rot*> 2 abgezogen.

3. Überlegt euch, wie ihr verhindern könnt, dass der Roboter bei jedem Strich zu viel dazu zählt.

Nutzt dazu den Block ***Warte,*** um das Programm zu unterbrechen, bis etwas bestimmtes passiert. Das kann zum Beispiel so aussehen:

Wie müsst ihr dieses Beispiel ändern, damit es für euch funktioniert?

4. Überlegt euch schließlich, wie der Roboter nun sagen kann, wie viele Striche er von jeder Farbe gezählt hat.



Wenn man eine Variable an den Block **Sage** hängt, dann gibt der Roboter den Wert dieser Variable wieder.

In diesem Beispiel gibt der Roboter den Wert der Variablen <Rot> wieder. Wenn ihr zum Beispiel in <Rot> die Zahl zwei gespeichert habt, würde der Roboter sagen: „Zwei“

Wenn wir euer Programm testen, werden wir die Farben der Striche vertauschen und eventuell die Anzahl der Striche ändern. Stellt also sicher, dass euer Programm auch dann noch funktioniert!

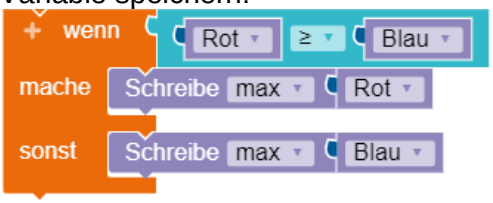
Aufgabe 4.2: Variablen vergleichen

Der Roboter soll nun zusätzlich ermitteln, welche Farbe er am häufigsten gezählt hat.
Dies soll der dann auch wiedergeben.

Die Farbe mit den meisten Strichen ist <*Farbe*> mit <*Anzahl der Striche*> Strichen.

1. Überlegt euch zuerst, wie ihr **Logik-Blöcke** nutzen könnt, um den Wert von Variablen zu vergleichen.
2. Benutzt zusätzliche Variablen zum Speichern der häufigsten Farbe und der Anzahl der Striche.

Mithilfe der **wenn…mache…sonst** Kontrollblöcke kann man zwei Variablen vergleichen, und die größere der beiden in einer neuen Variable speichern:

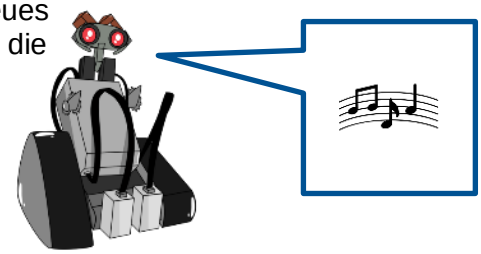


In diesem Beispiel werden die Werte der Variablen <Rot> und <Blau> verglichen. Die größere der beiden Werte wird dann in der Variablen <max> gespeichert.

3. Ergänzt nun euer Programm aus Aufgabe 4 entsprechend.

Bonusaufgabe 4.3: Musik spielen

Lasst den Roboter ein Lied für euch spielen. Erstellt dazu ein neues Programm und ladet den Hintergrund für die Bonusaufgabe 4.3 in die Simulationsansicht von Open Roberta. Platziert anschließend wieder den blauen Block und den Roboter auf den vorgegebenen Feldern.
Der Roboter soll nun jedes Mal eine Note spielen, wenn er über einen Strich fährt und schließlich wieder vor dem blauen Block anhalten. Abhängig von der Farbe des Strichs, soll der Roboter verschiedene Noten spielen.



Nutzt zum Spielen einer Note den folgenden Aktionsblock und stellt ihn auf „Viertelnote“.

Die folgende Tabelle verrät, bei welcher Farbe welche Note gespielt werden soll:

FARBE	NOTE
gelb	c‘
rot	d‘
schwarz	e‘
blau	f‘
grün	g‘

Stellt das Tempo eures Roboters auf 50%, um ein optimales Ergebnis zu erzielen!

Stellt bei dieser Aufgabe unbedingt die Lautstärke von eurem Computer auf sehr leise! Open Roberta spielt die Töne in einer hohen Lautstärke.

Bonusaufgabe 4.4: Binärsystem

Menschen verwenden meistens das Dezimalsystem zum Rechnen. Ein Computer hingegen verwendet das sogenannte Binärsystem, welches anders als im Dezimalsystem nur die beiden Zeichen 0 und 1 kennt. Aus diesen zwei Zahlen ergibt sich damit die Basiszahl 2. Beim Dezimalsystem ist die Basiszahl 10.
Computer rechnen im Binärsystem, weil sie prinzipiell nur erkennen können, ob Strom fließt (1) oder ob kein Strom fließt (0).

Umwandlung vom Binär- in Dezimalsystem:

Um zu erklären, wie man eine Zahl vom Binär- ins Dezimalsystem umrechnet, schauen wir uns ein Beispiel an. Nehmen wir also die binäre Zahl 1011.
Bevor wir die Zahl umwandeln können müssen wir die Basiszahl, in unserem Fall also die 2, exponentiell fortführen. Exponentiell bedeutet, dass man eine Zahl immer wieder mit sich selbst multipliziert. Die exponentielle Reihe von 2 lautet demnach wie folgt:

$2^0 = 1$ $2^1 = 2$ 2 $2^2 = 2*2=4$ $2^3 = 2*2*2=8$ $2^4 = 2*2*2*2=16$ usw.

Nun trägt man die Binärzahl entsprechend ihrer Wertigkeit ein. Dabei entspricht die von rechts gesehen erste Zahl (bei uns die 1) der niedrigsten Wertigkeit, also $2^0=1$. Damit ergibt sich folgendes Ergebnis:

1*8 + 0*4 + 1*2 + 1*1 = 11

Umwandlung vom Dezimal- ins Binärsystem:

Um eine Zahl von dezimal in binär umzurechnen gibt es zwei verschiedene Möglichkeiten. Diese möchten wir wieder anhand eines Beispiels erklären. Wir möchten nun die dezimale Zahl 200 in binär umwandeln.

Möglichkeit 1:

Zunächst schreibt man sich die exponentielle Reihe von 2 so weit auf, bis ein Wert größer als die gesuchte Zahl herauskommt und notieren uns die Werte in einer Tabelle.
Bei uns ist die erste Zahl, die größer als 200 ist, die $2^8 = 256$, Damit erhalten wir in diesem Beispiel die folgende Tabelle:

128	64	32	16	8	4	2	1

Nun vergleichen wir, ob der größte Wert (hier 128) in die umzurechnende Zahl (hier 200) hineinpasst. Wenn ja, dann vermerken wir uns an der Stelle in der Tabelle eine 1 und ziehen den Wert vom Ursprungswert ab.
In unserem Fall passt die 128 in die 200 hinein, wir vermerken also eine 1 und rechnen mit dem Rest 200 – 128 = 72 weiter.

128	64	32	16	8	4	2	1
1							

Passt der Wert der Tabelle nicht in unsere Zahl hinein, so vermerken wir in der Tabelle eine 0 und gehen zum nächsten Tabellenwert, ohne unsere Zahl zu verändern. Das wiederholen wir solange, bis wir am Ende der Tabelle angelangt sind. Die untere Zeile der Tabelle ergibt dann von links gelesen unsere gesuchte Binärzahl.

In unserem Beispiel rechnen wir also nun mit der 72 weiter. Die 64 passt in die 72 hinein, wir vermerken also eine 1 und rechnen 72 – 64 = 8.
32 passt nicht in die 8 rein, also vermerken wir eine 0.
16 passt ebenfalls nicht in die 8 rein, wir vermerken also eine weitere 0.
8 passt in 8 rein, es wird also eine 1 eingetragen, und wir rechnen 8 – 8 = 0.

Da in 0 sicher kein Wert mehr reinpasst, werden die restlichen Stellen mit 0 aufgefüllt und wir erhalten folgende Tabelle:

128	64	32	16	8	4	2	1
1	1	0	0	1	0	0	0

Damit ergibt sich die Dezimalzahl 200 zu 11001000 in binär.

Möglichkeit 2:
Die zweite Methode, um Dezimalzahlen in Binärzahlen umzuwandeln, ist sogar noch einfacher.
Man nimmt die Ursprungszahl und dividiert sie so lange durch 2 bis kein Rest mehr übrigbleibt. Dabei ist es wichtig, den Rest zu vermerken, denn wenn man die Zahl durch 2 dividiert, kann der Rest nur entweder 0 oder 1 sein und genau das entspricht dann der binären Zahl, die wir suchen.

Wir erklären es wieder am Beispiel der Dezimalzahl 200:

200	/ 2	=	100	Rest 0
100	/ 2	=	50	Rest 0
50	/ 2	=	25	Rest 0
25	/ 2	=	12	Rest 1
12	/ 2	=	6	Rest 0
6	/ 2	=	3	Rest 0
3	/ 2	=	1	Rest 1
1	/ 2	=	0	Rest 1

Die gesuchte Binärzahl ergibt sich wie gesagt aus den Resten, wobei diese von unten nach oben gelesen werden. Damit ergibt sich, wie auch schon bei Methode 1 die Binärzahl 11001000.

Aufgabe

1. Ladet die Datei *Aufgabe4.4.png* als Hintergrundbild in die Simulationsansicht von Open Roberta.
2. Ladet die Datei *Aufgabe4.4.json* als Simulationseinstellung hoch. Dadurch befindet sich der Roboter dann in einer der beiden angegebenen Startpositionen. Ihr könnt ihn danach gerne in die andere Startposition verschieben. Durch das Hochladen der Simulationseinstellung schaut der Roboter zu Beginn nach links.
3. Programmiert euren Roboter nun so, dass er von rechts nach links über die Farbfelder fährt und die Binärzahl liest.
    a. Ein rotes Feld steht für die 1
    b. Ein blaues Feld steht für die 0
4. Wenn euer Roboter auf dem Zielfeld ankommt, soll er die Zahl mit der `Sage`-Funktion aussprechen oder auf den Bildschirm schreiben.