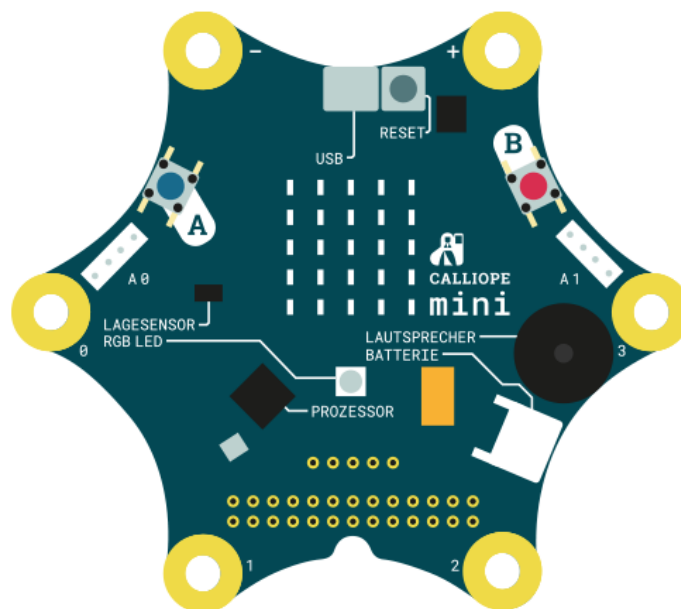


Leitprogramm

Blaulicht mit Sirene, Reizwort-Maschine, Sprite-Pong und Co. – deine ersten Schritte in der Programmierung mit dem *Calliope mini*



Verfasserin: Christina Schramm

Kurz-Info:

Informatischer Inhalt: Grafische Programmierung mit dem Mikrocontroller *Calliope mini*, informatische Konzepte wie Schleifen, Verzweigungen und Variablen, EVA-Prinzip

Jahrgangsstufe: 3 bis 5

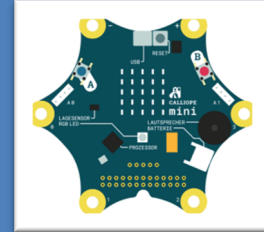
Vorwissen: ausreichende Lesekompetenz, um die Aufgabenstellungen zu verstehen; rudimentäre Fähigkeiten im Umgang mit dem Computer (Tastatur, Maus)

KURZINFORMATION

Titel: Blaulicht mit Sirene, Reizwort-Maschine, Sprite-Pong und Co. – deine ersten Schritte in der Programmierung mit dem *Calliope mini*

Schulstufen: 3 bis 5

Optimale Jahrgangsstufe: 3 bis 5



Themenbereich: grafische Programmierung, Informatiksysteme (Hard- und Software), informatische Konzepte wie Schleifen, Verzweigungen und Variablen, EVA-Prinzip, Fehlererkennung

Außerdem sind fächerübergreifend folgende Themen enthalten: Formulierung von Fragesätzen für ein Quiz (Deutsch); Kreieren eigener Melodien (Musik) und Geschichten (Deutsch); Benennung von Städten, Ländern, Flüssen (Sachunterricht)

Leitideen: Das Leitprogramm kann genutzt werden, um noch recht jungen Kindern einen ersten Einblick in die grafische Programmierung und das Informatiksystem *Mikrocontroller* zu ermöglichen. Hierzu wird auf die einfache visuelle Blockprogrammierung des *MakeCode-Editors* zurückgegriffen. Dieser kann über unterschiedliche Webbrowser kostenlos aufgerufen werden, sodass nicht erst Programme heruntergeladen und installiert werden müssen. Mit Hilfe von *MakeCode* werden die Kinder unter Zuhilfenahme der Anleitungen befähigt, eigene kleine Programme zu gestalten. Getestet werden können diese Programme über den Mikrocontroller *Calliope mini*. Alle Programme sind jedoch bewusst so konzipiert, dass nicht unbedingt ein *Calliope mini* vorhanden sein muss, sondern auch über den in *MakeCode* enthaltenen Simulator getestet werden können.

Vorkenntnisse: ausreichende Lesekompetenz, um die Aufgabenstellungen zu verstehen; rudimentäre Fähigkeiten im Umgang mit dem Computer (Tastatur, Maus, Browser)

Inhaltsbeschreibung: Da das Informatiksystem *Mikrocontroller* den meisten Schülerinnen und Schülern dieses Alters sicherlich in weiten Teilen unbekannt ist, bearbeiten die Kinder zunächst ein einführendes Theoriekapitel, um die Bauteile des konkreten Mikrocontrollers *Calliope mini* kennenzulernen und an diesem Beispiel das EVA-Prinzip nachzuvollziehen. Dieses Kapitel beinhaltet auch eine Kurzeinführung in den Aufbau und die Bedienung des Editors *MakeCode*. Im Rahmen des zweiten Kapitels folgen Basisaufgaben, die die Kinder an den selbstständigen Umgang mit der Programmieroberfläche *MakeCode* herantühren. Nachdem diese Basiskenntnisse im praktischen Umgang erlangt wurden, können die Kinder sich an die Bearbeitung der komplexeren Projekte des dritten Kapitels begeben. Der Schwierigkeitsgrad der Projekte steigt schrittweise an und wird durch die Angabe von Sternen transparent gemacht. Die Bearbeitung des vierten Kapitels, in dem ein erstes simples Spiel programmiert wird, ist fakultativ, weshalb danach auch kein weiterer Test vorgesehen ist.

EINFÜHRUNG

Wenn du dieses Programm bearbeitest, wirst du dich mit ganz allgemeinen Fragen der Informatik beschäftigen.

- ✖ Wie arbeitet ein Computer?
- ✖ Was ist ein Mikrocontroller?
- ✖ Wie funktioniert die Programmierung von Informatiksystemen wie Computer und Mikrocontroller?
- ✖ Und was macht man eigentlich genau beim Programmieren?

Dass du den Antworten auf diese Fragen langsam mehr und mehr auf die Spur kommst, ist wichtig. Denn wie soll man programmieren, wenn man gar nicht so genau weiß, was damit eigentlich gemeint ist? Wie soll man etwas programmieren, dessen Aufbau man nicht kennt?

Aber keine Sorge, du wirst keine trockenen Texte lesen, sondern dir dein neues Wissen auf unterschiedliche Weise zulegen. Manchmal wirst du mit Stift und Papier arbeiten, dann wieder am Computer. Hier wirst du häufig einen Editor nutzen, in dem du die Befehle für dein Programm wie Puzzleteile zusammenstecken kannst. Klingt kinderleicht? Ist es auch. Trotzdem lernst du so aber schon, wie andere Programmiersprachen aufgebaut sind. Dieses Wissen wirst du später immer wieder nutzen und darauf aufbauen können.

Das Leitprogramm mit einem „richtigen“ *Calliope mini* durchzuarbeiten macht natürlich am meisten Spaß. Aber: Du brauchst nicht unbedingt den kleinen Stern. Die Programmierumgebung kann die von dir geschriebenen Programme auch einfach simulieren, also abbilden.



Was du brauchst:

- ✖ Bleistift, Füller oder Kugelschreiber
- ✖ Buntstifte (lila, orange, gelb, grün)
- ✖ Computer / Laptop mit Internetanschluss und Browser
- ✖ *Calliope mini* mit USB-Kabel (nicht unbedingt notwendig)

Arbeite dieses Leitprogramm sorgfältig durch! Überspringe keine Aufgabe, denn diese bauen aufeinander auf. Deshalb ist es auch wichtig, erst weiterzuarbeiten, wenn du die vorherige Aufgabe erfolgreich gelöst hast, oder wenn dein Programm läuft. Arbeite nicht weiter, wenn dein Programm einen Fehler anzeigt. Suche lieber nach diesem Fehler. Auch dies gehört zur Arbeit eines Informatikers oder einer Informatikerin. Manchmal muss man zweimal hinsehen, um das Problem zu finden. Aber lasse dich davon nicht unterkriegen. Du schaffst das!

Viel Spaß und viel Erfolg!

INHALT

Kurzinformation.....	2
Einführung.....	3
Arbeitsanleitung.....	6
Kapitel 1 Kennelernen des <i>Calliope mini</i> und des Editors	8
Lernziel	8
Algorithmus und EVA-Prinzip	9
Mikrocontroller – kleine Computer.....	10
Kapitel 2 Erste Programme selbst erstellen.....	16
Lernziel	16
Endlosschleife.....	16
Download-Ordner ändern	17
Sequenz	18
Neue Projekte anlegen	19
Startblock.....	20
Eingabe über Knopf A und B	21
Kapitel 3 Programmierprojekte	23
Projekt: Quiz-Buzzer ★	23
Lernziel	23
Projekt: Sirene und Blaulicht ★ ★	26
Lernziel	26
Bedingte Verzweigung	27
Pausen	30
Lernziel	31
Computer zählen anders.....	31
Variable	32
Array	36
Kapitel 4: Ein erstes eigenes Spiel – Sprite-Pong ★ ★ ★	41
Lernziel	41
Sprite.....	41
Anhang A: Musterlösungen	45
Musterlösungen zu den Aufgaben aus Kapitel 1:.....	45
Musterlösungen zu den Aufgaben aus Kapitel 2:.....	48
Musterlösungen zu den Aufgaben aus Kapitel 3.....	51
Musterlösungen zu den Aufgaben aus Kapitel 4:	59

Anhang B: Abbildungsverzeichnis	62
Anhang C: Mediothek.....	62
Anhang D: Literaturangaben.....	62

ARBEITSANLEITUNG

Das Leitprogramm kannst du alleine bearbeiten. Die Kapitel bestehen aus unterschiedlichen Teilen, die du an den folgenden Symbolen erkennen kannst:



Lernziel

Was wirst du nach dem Bearbeiten des Kapitels (mehr) können?



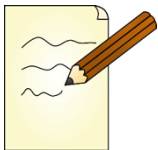
Theorie

Lies diesen Teil aufmerksam. Hier lernst du alles, was du für die Aufgaben und die Tests brauchst.



Übungsaufgaben am Computer

Bearbeite diese Aufgaben am Computer. Arbeite erst weiter, wenn dein Programm läuft. Du darfst diese Aufgaben alleine oder mit einem Partner bearbeiten.



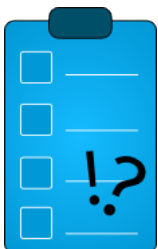
Übungsaufgaben im Heft

Bearbeite diese Übungsaufgaben alleine. Vergleiche sie dann mit der Musterlösung im Anhang.



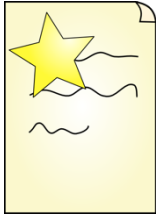
Sicherungsphase

Dieser Teil hilft dir beim Einprägen und der Kontrolle der gelernten Themen!



Test

Teste dich selbst. Hast du alles verstanden, oder musst du dir einzelne Aufgaben noch einmal ansehen? Starte erst mit dem nächsten Kapitel, wenn du den Test erfolgreich gemeistert hast.



Additum

Diese Aufgaben kannst du bearbeiten, wenn du dein Wissen noch vertiefen oder einfach weitertüfteln willst. Du kannst sie aber auch überspringen. Diese Aufgaben sind extra anspruchsvoller.



Calli

Das ist Calli. Er wird dich durch das Leitprogramm begleiten und dir wichtige Tipps und Hinweise geben. Manchmal stellt Calli dir aber auch Fragen.

Übersicht

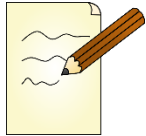
In diesem Kapitel lernst du den *Calliope mini* und den *MakeCode-Editor* kennen. Der *Calliope mini* ist eine Platine, auf der auch ein Mikrocontroller, ein kleiner Computer, ist. Man kann ihn programmieren. Dazu braucht man den Editor.

LERNZIEL



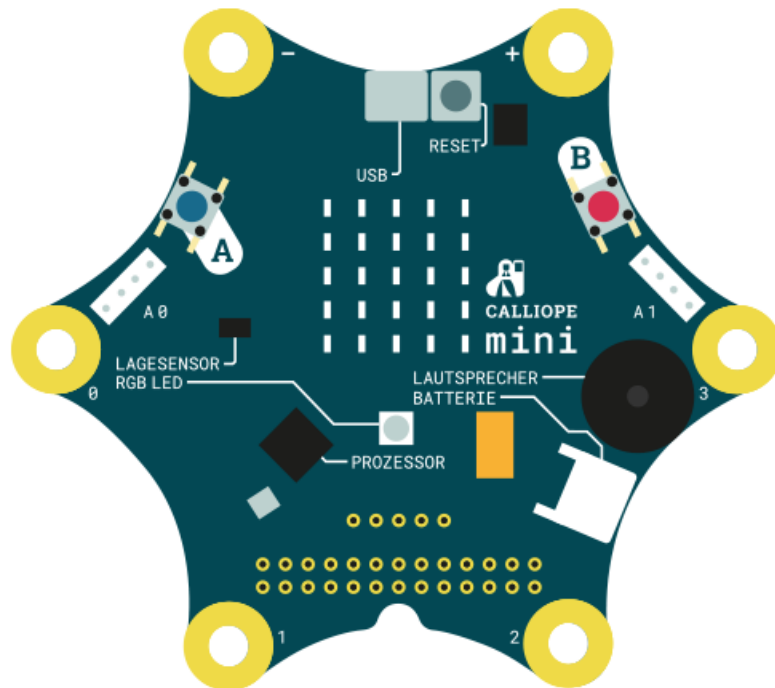
Wenn du dieses Kapitel bearbeitet hast, weißt du, was man unter „programmieren“ versteht. Außerdem wirst du die unterschiedlichen Teile des *Calliope mini* kennen und wissen, wie man den Editor benutzt. Dieses Wissen brauchst du unbedingt, um in den folgenden Kapiteln eigene Programme zu bauen.

Aufgabe 1



Als Erstes lernst du den Mikrocontroller kennen. Denn wie soll man etwas programmieren, das man nicht kennt? Bearbeite der Reihe nach die folgenden Aufgaben:

- × **Schau** dir den *Calliope mini* genau an.



- × Was siehst du?
 - Welche einzelnen **Teile** kannst du erkennen?
 - Was steht neben den einzelnen Teilen **geschrieben**?
 - Kennst du einzelne Teile vielleicht schon?
 - **Wozu** könnten die einzelnen Teile gut sein?
- × Notiere kurz, was dir zu den Fragen einfällt:

ALGORITHMUS UND EVA-PRINZIP



Computer ticken anders als Menschen. Damit ein Computer macht, was der Mensch möchte, muss er programmiert werden. **Programmieren** heißt, dass man dem Computer genaue Befehle gibt. Du kannst dir das in etwa wie bei einem Kochrezept vorstellen. Schritt für Schritt gibst du neue Zutaten in die Schüssel, damit daraus ein Teig entsteht. So machst du das auch mit einem Computer: Du sagst ihm Schritt für Schritt, was er als Nächstes machen soll. Ein solches Rezept für den Computer nennt man **Algorithmus**.



Jedes Programm ist gleich aufgebaut, folgt also demselben **Prinzip**. Den Aufbau des Prinzips kannst du gut an einem **Getränkeautomaten** beobachten:

Verarbeitung

Eingabe

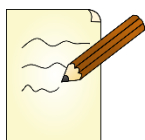
Ausgabe



Eingabe:
Den Knopf des Wunschgetränks drücken.

Verarbeitung:
Der Automat verarbeitet die Bestellung.

Ausgabe:
Der Automat gibt das bestellte Getränk aus.



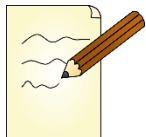
Aufgabe 2

Der Name des Prinzips setzt sich aus den Anfangsbuchstaben zusammen. Wie heißt es also?

___ ___ ___ -Prinzip

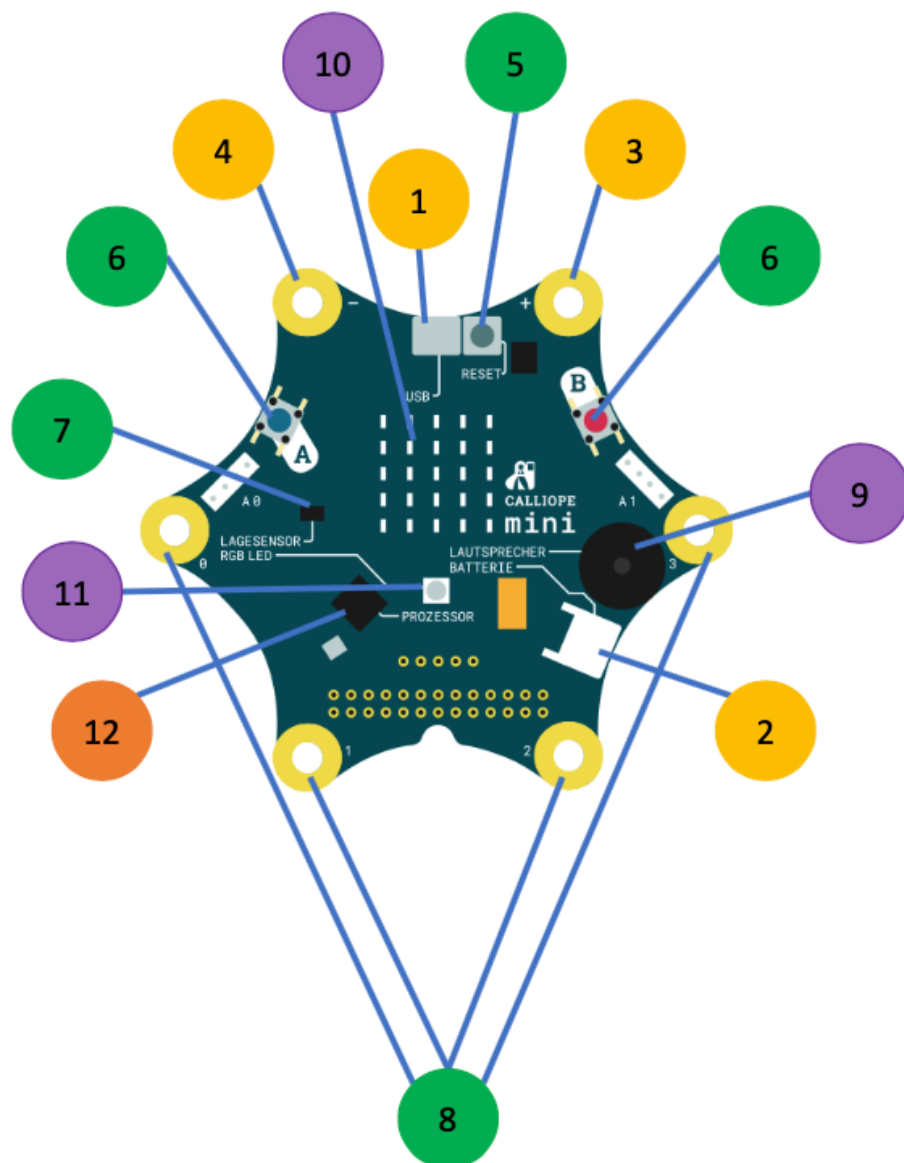


Ein Mikrocontroller wie der *Calliope mini* ist auch ein Computer. Das heißt, er funktioniert auch nach dem EVA-Prinzip.



Aufgabe 3

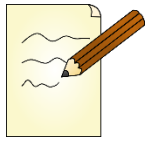
- × Guck dir noch einmal genauer die **einzelnen Teile** des *Calliope mini* an.
- × Ordne dann jedem einzelnen Bauteil die richtige **Erklärung** zu. Trage dazu die entsprechende **Zahl** in die Tabelle (nächste Seite) ein, und male die Felder in der passenden Farbe aus.



Stromversorgung	1	Mirko-USB
	2	Batteriebuchse
	3	Pin +
	4	Pin –
_____	5	Reset-Knopf
	6	Knopf A und Knopf B
	7	Lagesensor
	8	Pin 0 bis 3
_____	9	Lautsprecher
	10	LED-Bildschirm
	11	RGB-LED
_____	12	Prozessor

- × Trage die passende **Zahl** zur Erklärung ein, und **male** sie in der richtigen Farbe aus.

	Der Knopf startet das Programm neu.
	Das ist das Hirn des <i>Calliope mini</i> . Hier werden alle Infos verarbeitet.
	Er kann Töne abspielen.
	Seine 25 LEDs (kleine Lampen) können leuchten.
	Wenn du den <i>Calliope mini</i> wie eine Batterie verwendest, ist das der Minuspol.
1	Hier kommt ein USB-Kabel rein. Es versorgt den <i>Calliope mini</i> mit Strom und verbindet ihn mit dem Computer.
	Wenn du die Knöpfe drückst, machst du eine Eingabe. Du kannst die Knöpfe programmieren. Dann werden beim Drücken die programmierten Befehle ausgeführt.
	Er merkt, in welche Richtung du den <i>Calliope mini</i> bewegst.
	Wenn du den <i>Calliope mini</i> wie eine Batterie verwendest, ist dies der Pluspol.
	Die Rot-Grün-Blau-Leuchte kann in allen Farben leuchten.
	Über diese Pins kannst du auch eine Eingabe machen, damit der <i>Calliope mini</i> Befehle ausführt.
	Hier kannst du die Batterie anschließen.



Aufgabe 4

Jetzt kannst du noch das **EVA-Prinzip** auf die einzelnen Teile des *Calliope mini* anwenden.



- × Schau dir noch einmal an, wie der Getränkeautomat funktioniert.
- × Lies dann noch einmal nach, welche Teile des *Calliope mini* was machen.
- × Ordne schließlich den **grünen**, **orangenen** und **violetten** Gruppen die Begriffe **Eingabe**, **Verarbeitung** und **Ausgabe** zu. Trage dazu die Begriffe in die linke Spalte der Tabelle ein.



Aufgabe 5

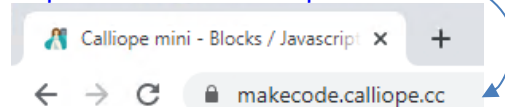
Super! Die Bestandteile des Mikrocontrollers kennst du nun schon. Fehlt noch der Editor, mit dem du Programme für den *Calliope mini* „schreiben“ kannst. Um den kennenzulernen, brauchst du nun einen Computer.

- × Schalte deinen Computer an.
- × Öffne einen **Browser**. Das ist ein Programm, mit dem man ins Internet kommt.

Es gibt zum Beispiel Chrome , Firefox  oder Safari . Du kannst den Browser öffnen, indem du einen Doppelklick auf das Symbol machst.

- × Gib diese Adresse in die Adressleiste (oben) ein:

<https://makecode.calliope.cc/>



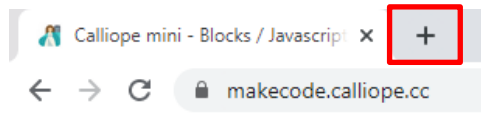
- × Jetzt nur noch mit „Enter“ bestätigen, dann solltest du den *MakeCode-Editor* sehen. Was siehst du? Notiere Schlagworte zu den folgenden Fragen:
 - Was siehst du ganz links im Editor?
 - Was steht in den bunten Kästchen geschrieben? Wozu könnte dies gut sein?
 - Was erkennst du in den Zeilen unten und oben? Kommen dir manche Dinge vielleicht bekannt vor?
 - Schau nach rechts: Woran erinnern dich die Teile in Türkis?



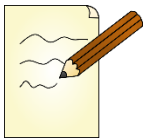
Aufgabe 6

Nachdem du dir nun ein erstes Bild des *Editors* gemacht hast, wäre eine Erklärung, wie dieser funktioniert nicht schlecht, oder?

- × Öffne in deinem Browser ein neues Fenster, indem du **+** anklickst:



- × Gib diese Adresse in die Adressleiste des **neuen Fensters** ein: <https://schuelerlabor.informatik.rwth-aachen.de/module/calliope>
- × Klicke dann das **erste Video** (Video 1: Der Editor) an, um es anzusehen. Vergiss nicht, die Lautsprecher einzuschalten, damit du die Erklärungen auch hören kannst.




Aufgabe 7

Ganz schön viele Infos, aber hast du dir auch alles Wichtige gemerkt? Ordne den **Bestandteilen** (aus dem blauen Kasten unten) die passende **Funktion** zu. Falls du dir unsicher bist, kannst du dir Ausschnitte des Videos noch einmal ansehen.



Name	Funktion
	Beim Anklicken wird das Programm auf den <i>Calliope mini</i> geladen.
	Zeigt dir an, was das Programm auf der Arbeitsfläche macht, noch bevor du es auf den <i>Calliope mini</i> geladen hast.
	Verbindet deinen <i>Calliope mini</i> mit dem Computer und somit mit dem Editor.
	Speichert beim Anklicken dein Programm.
	Darin sind die verschiedenen Blöcke eingeordnet, je nachdem, wie sie zusammengehören.
	Hier werden die Blöcke zu einem Programm zusammengesetzt.
	Wenn du die Schaltfläche anklickst, werden alle Programme angezeigt, die du bisher gespeichert hast.
	Sie sehen aus wie Puzzleteile und bilden die einzelnen Bestandteile deines Programms.

 (speichern)	farbige Schubladen	Simulator	Arbeitsfläche	Blöcke
Projekte	Herunterladen	USB-Kabel		



Aufgabe 8

Schau dir nun das **zweite Video** (Video 2: Mit Blöcken arbeiten) an, um zu lernen, wie man ein Programm mit Blöcken zusammensetzen kann.

Im Video hast du gesehen, dass der Text „Hallo“ ausgegeben wird, wenn man die **Taste B** drückt.

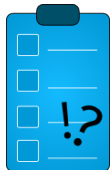
Denk nun noch einmal an das EVA-Prinzip zurück.

- × Welchem Teil des EVA-Prinzips entspricht das Drücken der **Taste B**?

- × Welchem Teil des EVA-Prinzips entspricht das Erscheinen des Textes „Hallo“ auf dem LED-Bildschirm?



Du bist nun am Ende des ersten Kapitels angekommen. Vergleiche deine Antworten mit den **Musterlösungen** im Anhang, bevor du weiterarbeitest!



Test

Alles verstanden bisher? Dann bist du sicher fit und bereit, dein Wissen zu testen.

- × Lies den folgenden Text, der noch einmal das Wichtigste aus Kapitel 1 zusammenfasst, und trage die Wörter aus der **blauen Box** in die passende Lücke ein.

USB	Mikrocontroller	Ausgabe	Simulator	programmieren
Algorithmus	Blöcke	LEDs	Reset	Batterie
Lagesensor	Eingabe	Arbeitsfläche	Verarbeitung	Platine
				RGB

Der *Calliope mini* ist eine _____, auf der mehrere Bauteile verbaut sind. Das wichtigste dieser Bauteile ist der _____, ein kleiner Computer. Daneben findet man viele weitere Bauteile: Strom kann der *Calliope mini* über einen _____-Anschluss oder eine _____ erhalten. Das Quadrat in der Mitte des Sterns besteht aus vielen kleinen, roten _____, die sich einzeln steuern lassen. Eine andere LED, die _____-LED, ist besonders, weil sie durch das Mischen der Farben Rot, Grün und Blau alle möglichen Farben herstellen kann. Neben den Knöpfen A und B findet man einen weiteren Knopf: Wenn man den _____-Knopf drückt, startet das Programm auf dem *Calliope mini* neu. Außerdem verfügt der *Calliope mini* über Sensoren wie den Licht- oder den Bewegungssensor, über die er etwas von außen

wahrnehmen kann. Über einen _____ merkt der *Calliope mini*, wie herum er gehalten wird.

Computer machen nur, was der Mensch ihnen sagt. Wenn man also möchte, dass der *Calliope mini* bestimmte Befehle ausführt, dann muss man ihn _____. Man muss ihm ein Rezept schreiben, das man in der Informatik _____ nennt. Da man dies nicht einfach wie für einen Freund auf einem Stück Papier machen kann, benutzt man einen *Editor*. Hier kann man Programme wie Puzzle über _____ zusammensetzen. Die Programmierblöcke findet man in den bunten Schubladen, aus denen man sie herausziehen kann. Zusammenpuzzeln kann man sie dann in der _____. Der _____ rechts führt dann vor, was das Programm auf dem *Calliope mini* macht.

Jedes Programm folgt demselben Prinzip: dem EVA-Prinzip. Dabei steht E für _____, V für _____ und A für _____.

- ✱ Hast du alle Lücken füllen können? Falls nein, dann suche noch einmal die passende Information in deinen Antworten zu den bisherigen Aufgaben. Falls ja, dann vergleiche deine Antworten mit der Musterlösung im Anhang.

KAPITEL 2 ERSTE PROGRAMME SELBST ERSTELLEN

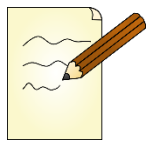
Übersicht

In diesem Kapitel baust du deine ersten kleineren Programme und lädst sie auf den Mikrocontroller. Wenn du keinen eigenen *Calliope mini* hast, kannst du die Programme über den Simulator testen.

LERNZIEL



Wenn du dieses Kapitel bearbeitet hast, wirst du dich selbstständig im Editor zurechtfinden und wissen, wo du die wichtigsten Programmbausteine findest. Außerdem wirst du die Programmblöcke zu ersten, noch recht einfachen, Programmen zusammensetzen können. Diese wirst du auf dem *Calliope mini* ausführen können. Das alles ist wichtig, damit du anschließend größere, spannende Programme entwickeln kannst.



Aufgabe 1

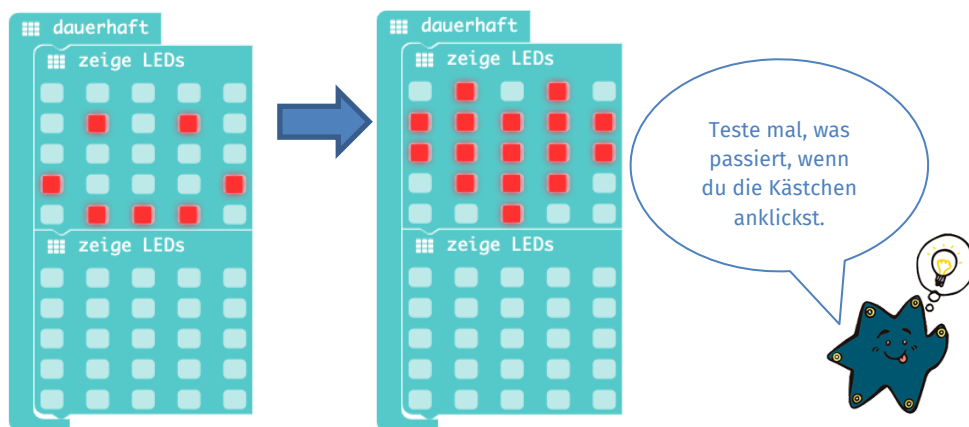
Als Erstes sollst du ein **Herz** programmieren. Folge dazu Schritt für Schritt den Anleitungen in den Aufgaben.

- × Öffne zunächst wieder den Editor: <https://makecode.calliope.cc/>
- × Schau dir dann das Programm an, das in der Arbeitsfläche bereits angezeigt wird, und überlege, was du machen musst, damit aus dem **Gesicht** ein **Herz** wird.




Aufgabe 2

Los geht's: Mache aus dem Gesicht ein Herz!



ENDLOSSCHLEIFE




Alles, was innerhalb des **Dauerhaft**-Blocks  steht, wird vom *Calliope mini* die ganze Zeit, also „dauerhaft“, ausgeführt. Informatiker nennen dies eine **Schleife**. Es gibt verschiedene Arten von Schleifen. Das ist eine Dauerhaft-Schleife, die Informatiker **Endlosschleife** nennen. Sie heißt so, weil die Befehle, die in der Schleife stehen (hier das Herz und das leere Bild) unendlich oft – also endlos – wiederholt werden. Es gibt aber auch andere Schleifen, die nur für eine bestimmte Anzahl an Durchläufen wiederholt werden. Mit ihnen könntest du z. B. programmieren, dass das Herz genau 5-mal blinkt.



Aufgabe 3

Jetzt musst du nur noch auf **Herunterladen** klicken, um das Programm auf dem *Calliope mini* zu speichern. Der kleine Stern führt das Programm dann dauerhaft aus.

Damit dein Programm auch dann nicht verloren geht, wenn du ein neues baust und

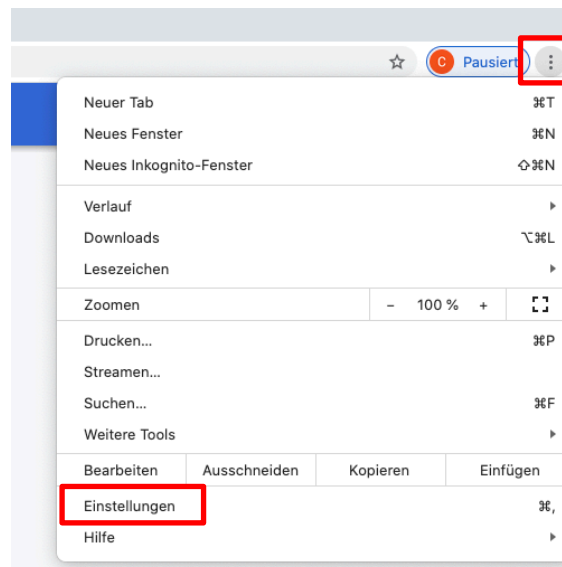
auf den *Calliope mini* überträgst, solltest du es noch **speichern** . Falls du dich nicht mehr erinnerst, wie das geht, schau noch einmal in Kapitel 1 nach.

DOWNLOAD-ORDNER ÄNDERN

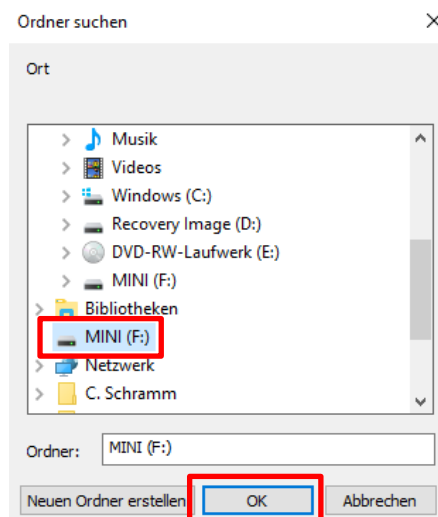


Falls das Programm nicht auf den *Calliope mini* übertragen wird, kann das daran liegen, dass als **Download-Ordner** nicht der *Calliope mini* ausgewählt ist. Lass dir in diesem Fall am besten von jemandem helfen. Der Download-Ordner lässt sich über die Browser-Einstellungen ändern. Die folgenden Schritte können sich je nach Betriebssystem und Browser etwas unterscheiden.

- ✖ In die Browsereinstellungen gelangt man z. B. bei Chrome über die drei Punkte oben rechts. Dann „Einstellungen“ anklicken.



- ✖ **Erweitert** anklicken.
- ✖ Dann unter „Downloads“ den „Ort“ ändern. Hier muss der *Calliope mini* ausgewählt werden.



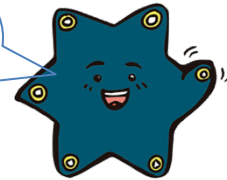


Aufgabe 4

Bisher zeigt der *Calliope mini* die ganze Zeit ein schlagendes Herz. Jetzt soll er nach jedem Herzschlag auch noch einen **Text** anzeigen.

Dazu brauchst du einen **Text-Block**.

Hm, wo kann man
den Text-Block wohl
finden?



- × Suche , und zieh ihn in die Arbeitsfläche.

SEQUENZ



Man muss dem Computer die Reihenfolge vorgeben, in der er die Befehle ausführen soll.

Aktion 1

Aktion 2

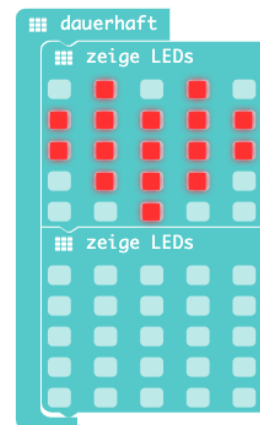
...

Der Computer macht zuerst **Aktion 1**, dann **Aktion 2** ... Das nennt man in der Fachsprache **Sequenz** oder Ablauf.

Wegen des **Dauerhaft**-Blocks, also der Endlosschleife, fängt das Programm nach der letzten Aktion wieder bei **Aktion 1** an.



- × Zieh den **Text-Block** jetzt an die Stelle des Programms, an der der Text erscheinen soll.





- × Ändere den Text „Hello“ jetzt noch in eine **eigene Botschaft**. Du kannst z. B. deinen Namen schreiben.
- × Vergiss nicht, das Programm unter einem sinnvollen Namen zu **speichern** und auf den *Calliope mini* zu übertragen.

NEUE PROJEKTE ANLEGEN



Du wirst dein Programm jetzt weiter verändern. Damit du deine Programme nachher noch alle aufrufen kannst, solltest du für jedes Programm ein **neues Projekt** anlegen.

- ✗ Dazu gehst du oben auf  und klickst dann auf .
- ✗ Es öffnet sich dann wieder das Anfangsprogramm.
- ✗ Du kannst mit den Blöcken arbeiten oder sie löschen, indem du sie in den **Papierkorb** ziehst. Dazu einfach die Blöcke, die du löschen möchtest, über die Schubladen ziehen, dann erscheint der Papierkorb.
- ✗ Fertig mit deinem neuen Programm? Dann einfach wieder einen passenden Namen eingeben und speichern.

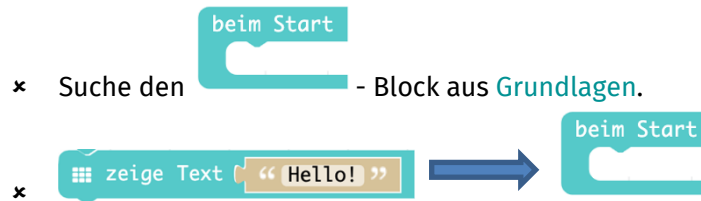


Denke daran, vor jedem neuen Programm ein neues Projekt anzulegen!



Aufgabe 5

In deinem Programm hat der *Calliope mini* bisher **bei jedem** Herzschlag die Nachricht angezeigt. Allerdings soll die Nachricht jetzt nur noch am **Anfang** angezeigt werden.



STARTBLOCK



Der *Calliope mini* startet mit dem  - Block.

Alles, was in diesem Block steht, wird nur ein einziges Mal – beim Start – ausgeführt und danach nicht mehr wiederholt.



Aufgabe 6

Suche **auf** dem *Calliope mini* den **Reset-Knopf**, und drück ihn.

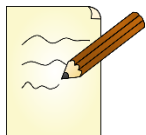
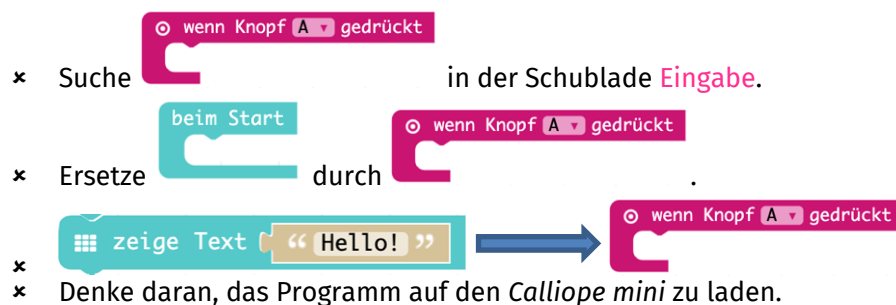
Was passiert?



Aufgabe 7

Es gibt auch andere Möglichkeiten, ein Programm zu starten. Der *Calliope mini* kann z. B. auf die **Knöpfe** reagieren oder wenn man ihn **schüttelt**.

Programmiere den *Calliope mini* so, dass er startet, wenn man **Knopf A** drückt.



Aufgabe 8

Was musst du nun machen, damit dein Text auf dem *Calliope mini* erscheint?

EINGABE ÜBER KNOPF A UND B



An dieser Aufgabe kannst du gut das **EVA-Prinzip** beobachten: Du hast ein Programm erstellt, das reagiert, wenn du **Knopf A** drückst. Dein Programm erkennt, was du mit dem *Calliope mini* machst, wann du also eine **Eingabe** machst. In deinem Fall ist die Eingabe das Drücken von **Knopf A**.

Die **Ausgabe** ist in diesem Fall das Anzeigen des Textes, z. B. „Hello!“, auf dem LED-Bildschirm.

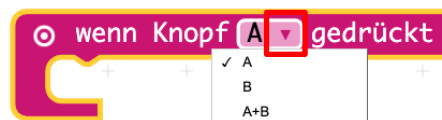


Aufgabe 9

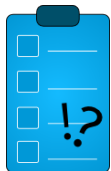
Es wäre doch super, wenn man in einem Programm gleich mehrere Eingaben machen könnte, oder?

Programmiere den *Calliope mini* so, dass unterschiedliche Nachrichten angezeigt werden, wenn du verschiedene Knöpfe drückst und ihn schüttelst.

Tipp: Wenn du den kleinen **Pfeil** neben dem Buchstaben anklickst, kannst du zwischen unterschiedlichen Buchstaben auswählen.



Super! Du hast deine ersten kleineren Programme erstellt und bist nun am **Ende von Kapitel 2** angekommen. Im Anhang findest du die Lösungen zu den unterschiedlichen Aufgaben. Mit ihrer Hilfe solltest du noch einmal deine eigenen Ergebnisse kontrollieren. Arbeit erst danach weiter.



Alles verstanden? Dann bist du bereit für den nächsten **Test**.

Beantworte die folgenden Fragen, indem du die richtige Antwort ankreuzt.

- * Welchen Block verwendest du, damit etwas nur am **Anfang** ausgeführt wird?

- ☐
- ☐
- ☐
- ☐
- ☐

- * Ein Computer führt alle Befehle in einer bestimmten Reihenfolge aus. Wie nennt man diesen **Ablauf** in der Informatik?

- ☐ Sequenz
- ☐ Frequenz
- ☐ Endlosschleife
- ☐ Bedingung
- ☐ Simulator

- × Betrachte das folgende Programm. Was musst du machen, damit die Nachricht „Gut“ angezeigt wird?



- ☐ Knopf A drücken
 ☐ Knopf B drücken
 ☐ Reset-Knopf drücken
 ☐ *Calliope mini* schütteln
 ☐ Gar nichts
- × In ein Programm kann man Schleifen einbauen, damit Programmteile wiederholt werden. Wie nennen Informatiker die **Dauerhaft-Schleife**?
 - ☐ Reset-Schleife
 - ☐ Zählschleife
 - ☐ Endlosschleife
 - ☐ For-Schleife
 - ☐ While-Do-Schleife
- × Wenn du Knopf A drückst, dann machst du eine
 - ☐ Ausgabe
 - ☐ Aufgabe
 - ☐ Aufforderung
 - ☐ Verarbeitung
 - ☐ Eingabe

Konntest du alle Fragen beantworten? Falls nein, dann suche noch einmal in Kapitel 2 nach den passenden Informationen. Falls ja, dann vergleiche deine Antworten mit der Musterlösung im Anhang.

KAPITEL 3 PROGRAMMIERPROJEKTE

Kurze Erinnerung: In diesem Kapitel findest du mehrere Projekte, die unterschiedlich schwer sind: je mehr Sterne, desto schwieriger die Aufgabe. Bearbeite die Projekte in der vorgegebenen Reihenfolge. So steigert du langsam den Schwierigkeitsgrad.

Das sind die Projekte:

Quiz-Buzzer



Sirene und Blaulicht



Reizwort-Maschine



PROJEKT: QUIZ-BUZZER

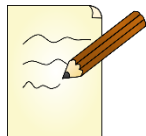


Übersicht Wenn du schon immer einmal an einem Quiz teilnehmen wolltest, ist dieses Projekt genau der richtige Anfang dafür. Du programmierst hier nämlich einen Quiz-Buzzer, mit dem du feststellen kannst, ob Spieler A oder B zuerst gedrückt hat. Nur wer zuerst gedrückt hat, darf antworten.

LERNZIEL





Wenn du dieses Projekt bearbeitet hast, wirst du ein Programm für einen Quiz-Buzzer entwickelt haben, indem du die beiden Knöpfe A und B programmierst.

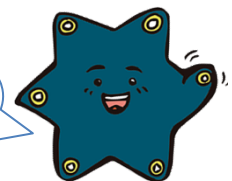


Aufgabe 1

Füll zunächst diese Tabelle aus. Sie wird dir bei der Programmierung helfen.

	Eingabe	Ausgabe
1	Knopf A  wird gedrückt.	Calliope mini zeigt „A“ an.
2	Knopf B  wird gedrückt.	
3		

Warum ist bei 3 eine ganz leere Zeile? Was kann denn noch passieren, um eine Eingabe zu machen?

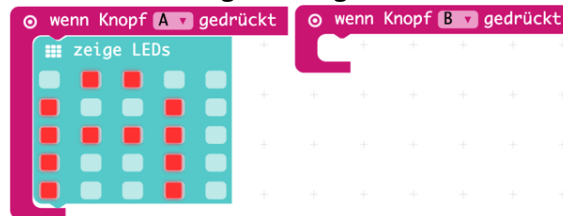




Aufgabe 2

Tabelle komplett ausgefüllt? Super! Dann kannst du nun mit der Programmierung des Quiz-Buzzers beginnen:

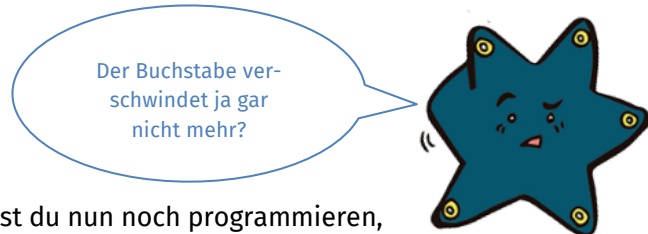
- × Starte dazu zunächst ein **neues Projekt**, und gib ihm einen sinnvollen Namen.
- × **Lösche** dann alle Blöcke.
- × Hier ist der Anfang des Programms:



- × Erweitere das Programm zu einem funktionierenden Quiz-Buzzer.
- × Speicher dein Programm, und lade es auf den *Calliope mini*.
- × Was passiert?



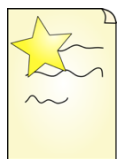
Aufgabe 3



Da hat Calli recht. Deshalb sollst du nun noch programmieren, dass nichts mehr angezeigt wird, wenn der *Calliope mini* **geschüttelt** wird.



- × Ziehe aus der Schublade **Eingabe**.
- × Ergänze dann den passenden Block aus **Grundlagen**, damit die LEDs nichts mehr anzeigen.

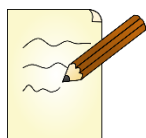


Aufgabe 4

Du kannst dein Programm noch erweitern. Eine blinkende Animation oder eine Begrüßung beim Spielstart wäre doch sehr schön.



Prima! Du hast einen eigenen Quiz-Buzzer programmiert. Falls dein Quiz-Buzzer noch nicht so funktioniert wie er soll, kannst du einen Blick in die Musterlösungen im Anhang werfen. Wenn dein Quiz-Buzzer funktioniert, kannst du weiterarbeiten.



Aufgabe 5

Damit du deinen Quiz-Buzzer auch in einem richtigen Quiz ausprobieren kannst, brauchst du noch Fragen.

- × Formuliere mindestens **zehn Quizfragen** und deren **Antworten**. Benutze dazu mindestens fünf verschiedene **Fragewörter**.



- 



Übersicht

Gelegentlich hörst und siehst du sie, wenn du durch die Straßen läufst: die Sirenen und Blaulichter von Polizei, Feuerwehr und Rettungsdienst. In diesem Projekt baust du deine eigene Sirene samt Blaulicht.

LERNZIEL



Wenn du dieses Projekt bearbeitet hast, kannst mit deinem *Calliope mini* eine Sirene und ein Blaulicht simulieren. Dazu wirst du die **bedingte Verzweigung** kennenlernen.



Aufgabe 1

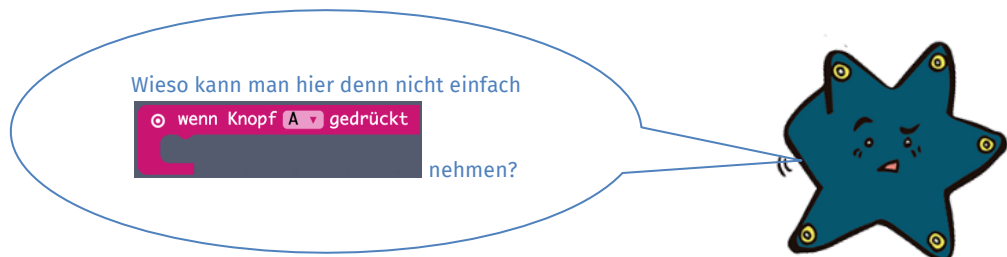
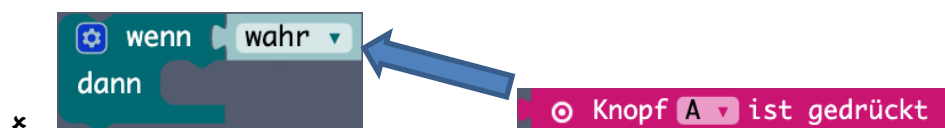
Du beginnst mit der **Sirene**. Sie soll ertönen, wenn du **Knopf A** gedrückt hältst.

- × Du startest auch dieses Mal, indem du ein **neues Projekt** anlegst und es unter einem passenden Namen **speicherst**. Vergiss nicht, deine Fortschritte auch zwischendurch immer wieder zu speichern.
- × Damit die Sirene nur dann ertönt, wenn du **Knopf A** gedrückt hältst, brauchst



du aus **Logik** diesen Block:
Bau ihn in den **Dauerhaft-Block**.

- × Wenn **Knopf A** gedrückt wird, **dann** soll die Sirene erklingen. Was muss also in die Lücke, in der „**wahr**“ steht?






Würdest du den **wenn Knopf A gedrückt**-Block nehmen, dann würde deine Sirene starten, wenn du **Knopf A** drückst, und dann auch angeschaltet

bleiben. Mit dem **wenn wahr dann**-Block kannst du aber erreichen, dass deine Sirene **nur** erklingt, solange **Knopf A gedrückt bleibt**. Wenn du **Knopf A** also loslässt, ist der *Calliope mini* wieder stumm.

- × Um eine Sirene zu bauen, brauchst du nun noch **Töne**.



In welcher Schublade kannst du wohl Töne finden? Was meinst du?

- × Suche diesen Block   für , und setze ihn in die Lücke hinter „dann“.
- × Lade das Programm auf den *Calliope mini*, und teste es. Was passiert?

BEDINGTE VERZWEIGUNG



Der **Wenn-dann**-Block wird dir in der Programmierung häufiger begegnen.

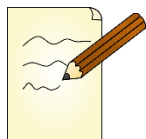
Wenn Knopf A gedrückt wird,
dann wird ein Ton abgespielt.

Oder allgemeiner:

Wenn eine **Bedingung** wahr ist,
dann wird ein **Befehl** ausgeführt.

Deshalb spricht man in der Informatik auch von einer **bedingten Verzweigung**. Mit dem **Wenn** wird erst einmal abgefragt, ob eine bestimmte Bedingung erfüllt ist (hier: das Drücken von **Knopf A**).

Wenn dies der Fall ist, die Bedingung also **wahr** ist, dann wird der Befehl hinter **dann** ausgeführt. Falls nicht, geht das Programm zum nächsten Befehl über.



Aufgabe 2

Wenn du **Knopf A** drückst, ertönt ein **Ton**. Aber das hört sich noch nicht wirklich nach einer Sirene an, oder?

- × Sirenen, wie wir sie kennen, bestehen immer aus mehreren Tönen. Wie kannst du nun dein Programm erweitern, damit es sich nach einer richtigen Sirene anhört?

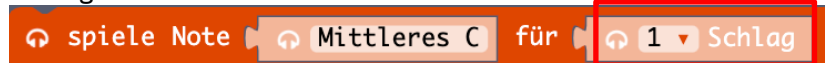


- × Baue nun **weitere Töne** in dein Programm ein, damit eine richtige Sirene entsteht. Teste dann dein Programm.



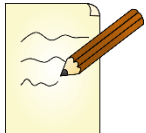
Polizei und Feuerwehr benutzen ein so genanntes Folgetonhorn. Du kannst es in etwa mit den **Tönen a und d** nachbilden. Du kannst diese Töne testen oder aber deine ganz eigene Sirene mit anderen Tönen kreieren.

- × Wenn du dein Programm jetzt abspielst, wechseln die Töne sehr schnell nacheinander. Das liegt daran, dass die einzelnen Töne nur je für einen **Schlag** erklingen.



Ändern kannst du die Tonlänge, indem du die Anzahl der Schläge änderst. Dazu einfach auf den **kleinen Pfeil** neben der Eins klicken.

Experimentiere ein wenig mit den Tönen und den Längen, bis du eine Sirene komponiert hast, die dir gefällt.



Aufgabe 3

Fehlt noch das Blaulicht. Welches Bauteil musst du nutzen, um ein Blaulicht zu programmieren?



Aufgabe 4

Du brauchst also die **RGB-LED**. Um diese zu programmieren, nutzt du diesen Block:



„Set led to“ ist Englisch und heißt „setze LED auf“. Die Farbe, auf die die LED gesetzt werden soll, kannst du noch ändern, indem du den **kleinen Pfeil** anklickst und anschließend die gewünschte Farbe auswählst.

Wenn du **Knopf B** gedrückt hältst, soll das Blaulicht leuchten. Du brauchst zusätzlich zu deinem **set-led-to**-Block also noch die Blöcke, die du schon von der Sirene kennst. Baue sie in der richtigen Reihenfolge in dein Programm ein, und teste es.



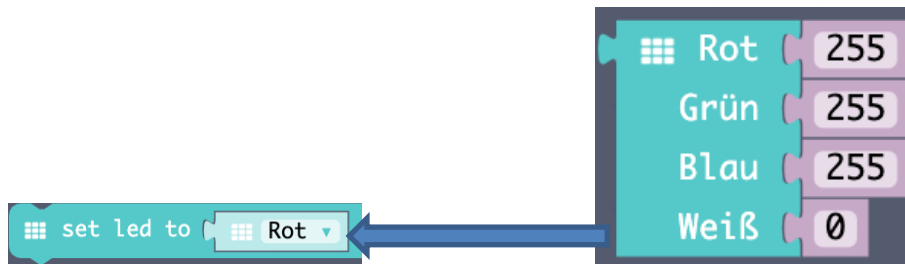
Was passiert, wenn du nun **Knopf B** drückst?



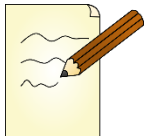
Aufgabe 5

Die RGB-LED soll nun nicht nur Blau leuchten, sondern **an- und ausgehen**.

- × Leider hat der Editor aber keinen Block, mit dem man einfach angeben kann, dass die RGB-LED ausgeht. Deshalb brauchst du einen kleinen Trick. Ergänze unter deinem Blau-Block diese beiden Blöcke:



Die **RGB-LED** setzt sich aus drei kleinen LEDs zusammen. Diese haben die Farben Rot, Grün und Blau. Die **Zahl** hinter der Farbe gibt an, wie hell die LED in der entsprechenden Farbe leuchten soll. Je größer die Zahl ist, desto heller leuchtet die LED.



- × Was musst du also tun, damit **keine** der LEDs leuchtet?

- × Wenn du die beiden Blöcke oben eingebaut und die Werte verändert hast, kannst du dein Programm erneut **testen**. (Wenn du nicht sicher bist, ob du die Aufgabe bisher richtig gemacht hast, gleiche deine Lösung kurz mit der Musterlösung im Anhang ab.)



Aufgabe 6

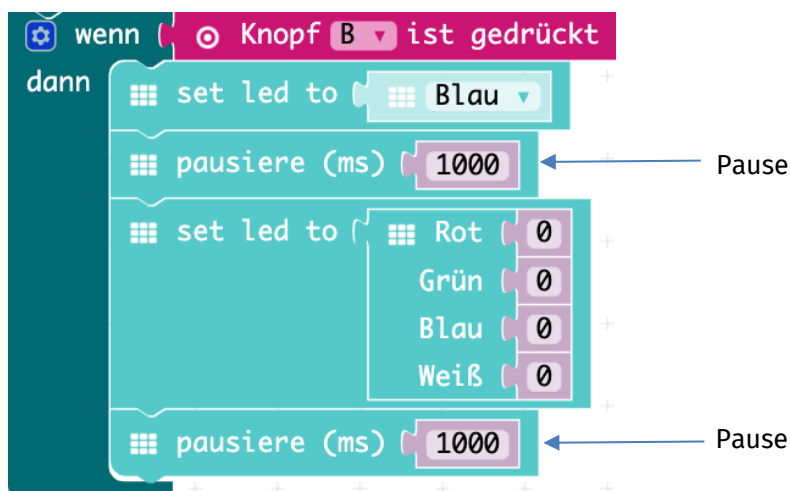
Wahrscheinlich ist dir aufgefallen, dass die RGB-LED jetzt nur ein bisschen flackert, wenn du **Knopf B** gedrückt hältst. Das ist so, weil die beiden Aktionen

setze LED auf Blau

und schalte LED Rot, Grün, Blau, Weiß auf 0

ganz schnell hintereinander ausgeführt werden – so schnell, dass das menschliche Auge nicht mitkommt.

Damit dein Auge den Unterschied zwischen „LED blau“ und „LED aus“ wahrnehmen kann, musst du den Wechsel zwischen „blau“ und „aus“ etwas verlangsamen. Das kannst du machen, indem du **Pausen** einbaust.



PAUSEN



Das „ms“ im **Pausen-Block** steht für **Millisekunden**. 1000 Millisekunden sind eine Sekunde.



Teste nun noch einmal Sirene und Blaulicht zusammen. Klappt alles? Super, dann hast du dieses Projekt erfolgreich gemeistert. Sollte es noch irgendwo ein Problem geben, schaust du am besten in die Musterlösung im Anhang. Arbeite erst weiter, wenn du mit deinem Blaulicht und deiner Sirene zufrieden bist.



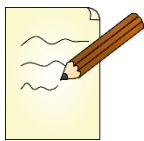
Übersicht

Geschichten selber zu schreiben macht Spaß. Jedenfalls ist das so, wenn man genügend Ideen hat. Damit dir nie die Einfälle für neue Geschichten ausgehen oder du auch deinen Freunden zu mehr Kreativität verhelfen kannst, programmierst du in diesem Projekt eine Reizwort-Maschine.

LERNZIEL



Wenn du dieses Projekt bearbeitet hast, zeigt der *Calliope mini* dir auf Knopfdruck ein zufälliges Wort aus einer Wortsammlung an. In diesem Zusammenhang wiederholst du noch einmal die **bedingte Verzweigungen**. Außerdem lernst du, was man in der Informatik unter **Variablen** versteht.



Aufgabe 1

Wenn du **Knopf A** drückst, soll der *Calliope mini* dir ein zufälliges Wort aus einer Liste von Wörtern anzeigen, die du vorher programmiert hast.

Was liegt also näher, als zunächst eine **Liste** mit **Wortideen** zu füllen. Finde mindestens acht weitere Wörter, die als kreative Idee für eine Geschichte dienen können:

0.	Sommerferien
1..	
2.	
3.	
4.	
5.	
6.	
7.	
8.	

COMPUTER ZÄHLEN ANDERS



Die (Papier-)Liste beginnt mit einer 0, weil Computer mit der 0 anfangen zu zählen. Damit der *Calliope mini* die einzelnen Wörter später finden kann, musst du jedem Wort eine Nummer (hier zwischen 0 und 8) zuweisen.

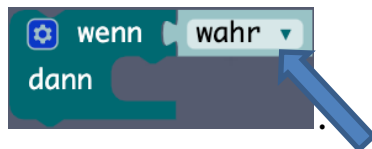


Aufgabe 2

Du startest wie immer, indem du ein **neues Projekt** anlegst und es unter einem passenden Namen **speicherst**. Vergiss nicht, deine Fortschritte auch zwischendurch immer wieder zu speichern.

Lösche dann alle Blöcke des Beispielsprogramm außer den **Dauerhaft-Block**.

Wenn der **Knopf A** gedrückt wird, **dann** soll der *Calliope mini* eines der Wörter aus der Liste anzeigen. Dazu musst du zunächst diesen Block aus Logik in den **Dauerhaft-Block** einfügen:



Setze dann den **Knopf A ist gedrückt**-Block dort ein, wo „**wahr**“ steht.



Aufgabe 3

Damit nicht immer dasselbe Wort, sondern ein zufälliges Wort aus der Liste angezeigt wird, musst du nun eine **Variable** anlegen.

- × Öffne dazu die Schublade **Variablen**.
- × Klicke auf **Neue Variable anlegen**.
- × Gib dann einen passenden Namen für deine Variable ein, und klicke auf OK:

New variable name:

OK ✓
Abbrechen ✕

VARIABLE

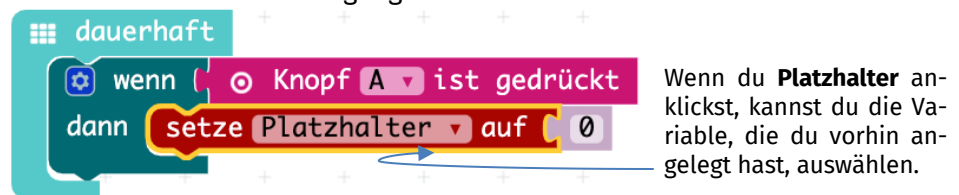


Im Programm benutzt du eine sogenannte **Variable**. In dieser Variable speichert der *Calliope mini* die Zahlen zu den Wörtern. Eine Variable kannst du dir also in etwa wie eine große Kiste vorstellen, in der etwas aufbewahrt wird.



Aufgabe 4

- × Ziehe aus **Variablen** diesen Block **setze Platzhalter auf 0** in den **Dann**-Teil deiner Bedingung:



- × Nun musst du dem *Calliope mini* noch sagen, dass er eine **zufällige Zahl** zwischen **0 und 8** aus deiner Liste aufruft, wenn du **Knopf A** drückst. Dazu musst

du diesen Block suchen: **pick random 0 to 4**.

- × „Pick random 0 to 4“ ist Englisch und heißt soviel wie „wähle eine zufällige Zahl zwischen 0 und 4“.

Welche Zahl musst du nun also statt der 4 einsetzen? _____



- × Nun musst du die **Zahlen** 0 bis 8, die in deiner **Variablen** „zufallszahl“ gespeichert sind, noch mit den **Wörtern** in deiner Liste verbinden. Dazu brauchst du die folgenden Blöcke:



Versuche, die Blöcke richtig zusammenzupuzzeln.

- × Wenn du Probleme beim letzten Programmierschritt hattest, ist das gar nicht schlimm. Hier noch ein **Tipp**: Würde man den Programmierschritt beschreiben, könnte dies für das Wort „Sommerferien“, dem die Zahl 0 zugeordnet ist, so lauten:
Wenn die Zufallszahl gleich Null ist,
dann wird der Text Sommerferien angezeigt.
- × Teste dein Programm.

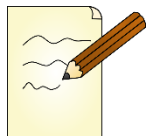


Aufgabe 5

Bis jetzt wird nur das Wort „Sommerferien“ angezeigt, wenn du **Knopf A** drückst. Ergänze deshalb dein Programm, sodass auch alle anderen Wörter deiner Liste zufällig angezeigt werden können.



Teste nun noch einmal dein Programm. Funktioniert deine **Reizwort-Maschine**? Falls ja, dann ist das super, und du kannst direkt weiterarbeiten. Falls nein, dann zieh die Musterlösung des Programms im Anhang zu Rate.



Aufgabe 6

Nun ist es an der Zeit, den *Calliope mini* auch wirklich als Reizwort-Maschine einzusetzen: Drücke **Knopf A** und **schreibe eine Geschichte** rund um das Wort, das der *Calliope mini* dir anzeigt. Deine Geschichte sollte mindestens zehn Sätze lang sein. Also, ran an die Stifte!

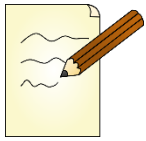
[illegible]

Aufgabe 7

Stadt	Land	Fluss	Name	Tier	Beruf	Punkte
Stuttgart	Schweden	Spree	Sandra	Specht	Sportler	
...						

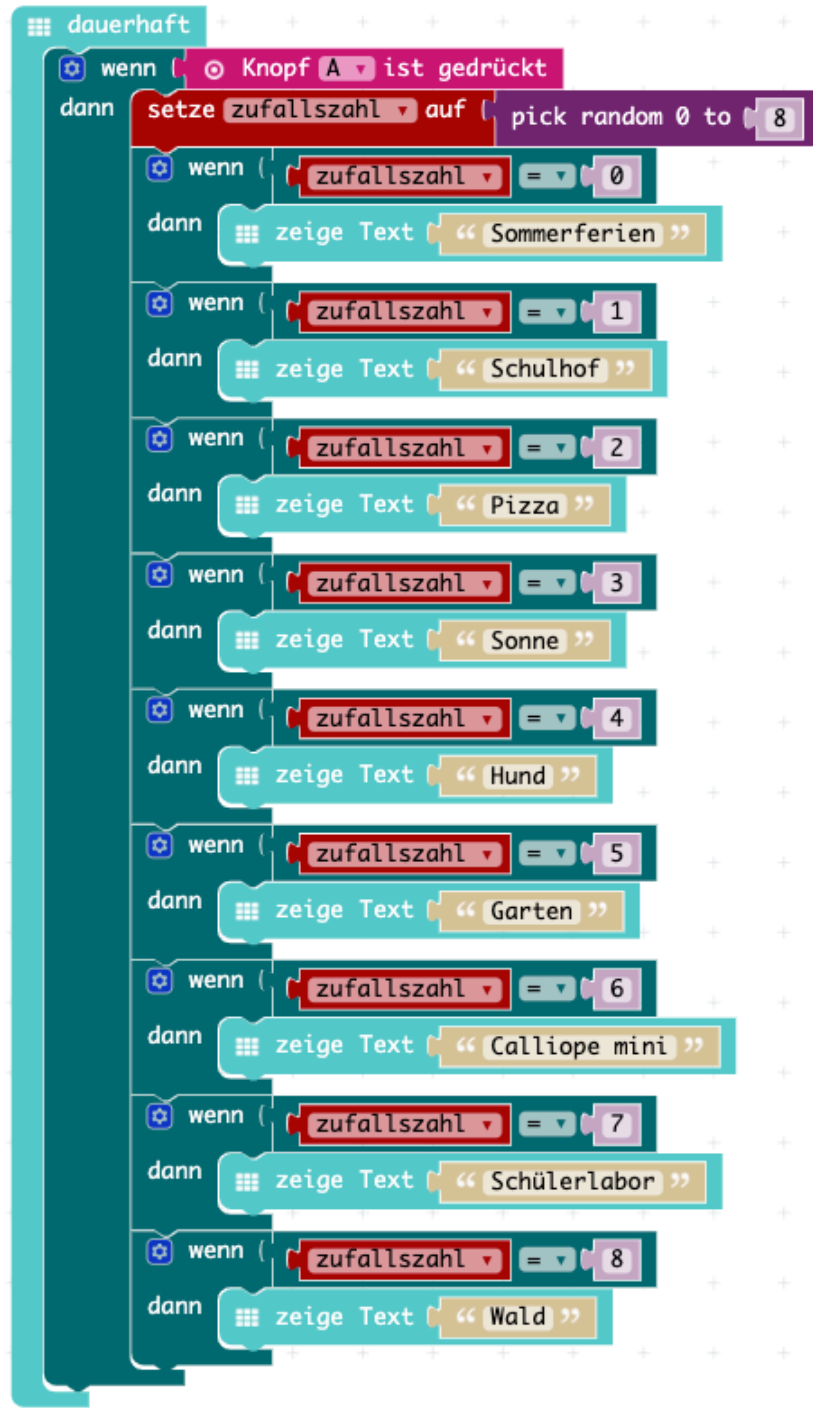
- ✖ 20 Punkte, wenn kein anderer Spieler einen Begriff in der Kategorie hat.
- ✖ 10 Punkte, wenn andere Spieler auch einen Begriff in der Kategorie haben, diese aber unterschiedlich sind.
- ✖ 5 Punkte, wenn zwei Spieler denselben Begriff in einer Kategorie haben.

Mit ein paar kleinen Änderungen kannst du deine **Reizwort-Maschine** auch für die **Buchstabensuche** beim Spiel *Stadt, Land, Fluss* einsetzen.



Wenn du **Knopf A** drückst, dann soll der *Calliope mini* einen zufälligen Buchstaben anzeigen.

Kreise nun in dem Programm ein, wo du etwas ändern musst, damit ein zufälliger Buchstabe angezeigt wird.





Schau einmal in der Musterlösung nach, ob du alle Programmteile gefunden und eingekreist hast, die für das Programm zur Buchstabensuche geändert werden müssen.

ARRAY



Sicherlich ist dir aufgefallen, dass das Programm für den **Buchstaben-Zufallsgenerator** mit allen 26 Buchstaben seeeeeehr lang werden würde, wenn du das Programmgerüst der Reizwort-Maschine einfach so übernehmen würdest.

Damit Programme nicht endlos lang werden, haben Informatikerinnen und Informatiker sich jedoch den ein oder anderen Trick einfallen lassen.

Einen dieser Tricks kannst du auch für deinen Buchstaben-Zufallsgenerator nutzen: ein **Array**.

Ein Array ist eine Aneinanderreihung von Elementen desselben Typs; in diesem Fall sind das Buchstaben. Du kannst dir ein Array (wie schon die Variable) als Kiste vorstellen. Allerdings hat diese Kiste weitere Unterteilungen, also quasi so etwas wie Schubladen. Die Schubladen sind nummeriert und in ihnen sind die einzelnen Elemente, hier die Buchstaben, eingeordnet.

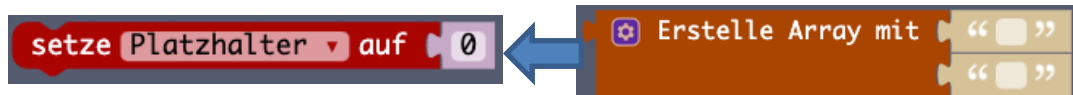
Aufgepasst: Auch hier beginnt der Computer wieder ab 0 zu zählen!




Baue nun das Programm für den Buchstaben-Zufallsgenerator, indem du Schritt für Schritt den Anweisungen folgst:

- × Lege ein neues Projekt an, und lösche alle Blöcke außer den **Dauerhaft**-Block.
- × Ziehe dann **setze Platzhalter** auf **0** in deinen **Dauerhaft**-Block. Klicke dann auf **Platzhalter** und wähle „**Liste**“ aus.

- × Nun kommt das **Array** ins Spiel.

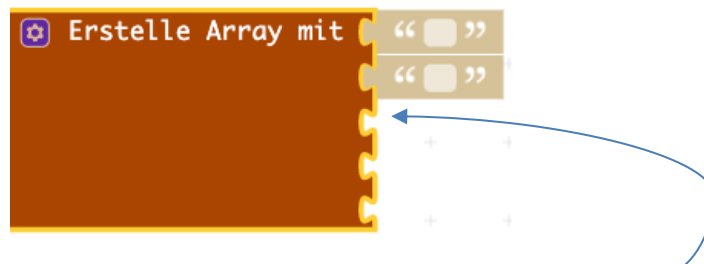


Setze den **Erstelle-Array-mit**-Block in den **Liste**-Block. Den **Array**-Block findest du in der Schublade **Fortgeschritten** unter **Arrays**.

- × Bis jetzt hat das Array nur zwei „Schubladen“. Das reicht nicht für 26 Buchstaben. Aber das ist kein Problem, denn man kann ganz einfach weitere Schubladen hinzufügen, indem man  anklickt und weitere **Werte** an die schon vorhandenen anpinnt:



- × Jetzt hast du zwar ein **Array**, das Platz für 26 Buchstaben hat, aber du kannst die Buchstaben noch gar nicht eintragen.



Dazu musst du zunächst weitere **Textblöcke**  in die **Lücken** einfügen. Du findest sie unter **Fortgeschritten** → **Text**.

Fülle die Lücken mit den noch fehlenden **Textblöcken**. Schreibe dann in jeden **Textblock** je einen Buchstaben, sodass du schließlich das gesamte Alphabet in deinem Array stehen hast.

- × Das Array ist schon einmal gefüllt. Aber nun musst du noch programmieren, wann und wie die einzelnen Buchstaben aufgerufen werden sollen:

Wenn du Knopf A drückst,
dann soll ein Buchstabe abgerufen werden,
der sich im Array an einer zufälligen Stelle zwischen 0 und 25 befindet.

Dazu brauchst die folgenden Blöcke. Baue sie so zusammen, dass dein Programm funktioniert.



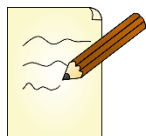
Achtung! Der Computer zählt ab 0.
Bei 26 Buchstaben kann die Zufallszahl also zwischen 0 und 25 liegen.



- × Teste dein Programm für den Buchstaben-Zufallsgenerator.



Dein **Buchstaben-Zufallsgenerator** funktioniert noch nicht wie er soll? Dann durchsuche dein Programm noch einmal nach möglichen Fehlern. Wenn auch das nicht hilft, kannst du einen Blick in die Musterlösung werfen.



Der **Buchstaben-Zufallsgenerator** funktioniert? Genial! Dann kannst du nun loslegen und deine Eltern und Geschwister zu einem Spiel herausfordern. Mal sehen, wer von euch die meisten Städte, Länder und Flüsse zu den Buchstaben findet.



Wenn du mit deinen Projekten zufrieden bist und deine Programme laufen, dann bist du bereit für einen letzten **Test**.

Beantworte die folgenden Fragen:

- * Du hast einen Begrüßungstext programmiert, der erscheint, wenn du **Knopf A** drückst. Außerdem erscheint ein Verabschiedungstext, wenn du **Knopf B** drückst. Nun möchtest du noch programmieren, dass die Nachrichten wieder verschwinden, wenn du den *Calliope mini* **schüttelst**. Welche Blöcke brauchst du dazu?

☐ **Knopf A** ist gedrückt

☐ dauerhaft

☐ wenn geschüttelt

☐ zeige Text "Hello!"

☐ zeige LEDs

☐ wenn Knopf A gedrückt

- * Bei einem Programm wird zunächst einmal abgefragt, ob eine **Bedingung** wahr ist, bevor ein Befehl ausgeführt wird. Wie nennt man dies in der Informatik?

- ☐ Bedingte Verzweigung
- ☐ Eingabe
- ☐ Verarbeitung
- ☐ Ausgabe
- ☐ Variable

- * Du möchtest ein **Diskolicht** programmieren, das in **Rot**, **Grün** und **Blau** leuchtet, wenn du **Knopf A** drückst. Dazu brauchst du folgende Blöcke. Bringe sie in die richtige Reihenfolge:

☐ set led to Blau

☐ wenn wahr dann

☐ set led to Grün

☐ dauerhaft



- * Vervollständige den Satz: Eine **Variable** kann man sich vorstellen wie ...
 - ☐ eine Bedingung, die erfüllt sein muss, damit der *Calliope mini* etwas macht.
 - ☐ eine große Kiste, in der etwas aufbewahrt wird.
 - ☐ einen Knopf, den man drücken kann.
 - ☐ ein Bildschirm, auf dem etwas angezeigt wird.



- * Was macht dieser Block?
 - ☐ Die Zahlen 0 bis 4 werden angezeigt.
 - ☐ Es wird rückwärtsgezählt von 4 bis 0.
 - ☐ Eine zufällige Nummer zwischen 0 und 4 wird aufgerufen.
 - ☐ Man sieht auf jeden Fall eine 4 auf dem Display.



Übersicht

In diesem Kapitel programmierst du dein erstes Spiel namens Sprite-Pong. Bei diesem Spiel bewegt sich ein Licht über den LED-Bildschirm. Wenn das Licht an bestimmten Stellen des Bildschirms ankommt, müssen die Spieler durch das Drücken der Tasten A und B Punkte sammeln.

LERNZIEL

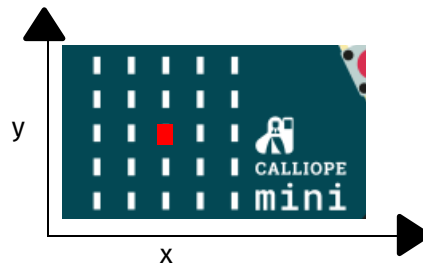


Wenn du dieses Kapitel durchgearbeitet hast, wirst du nicht nur dein erstes Spiel programmiert haben, sondern auch erklären können, was man in der Informatik unter einem **Sprite** versteht. Außerdem wirst du dieses Sprite bewegen können.

SPRITE

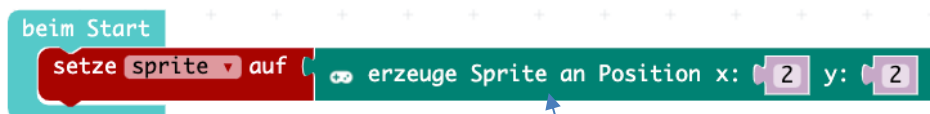


Ein **Sprite** ist ein Bild, das sich auf einem Bildschirm bewegen kann. Ein solches Bild kann auch einfach eine einzelne leuchtende LED sein. Ein Sprite hat immer eine Position auf dem Bildschirm. Diese Position wird mit einer **X-** und einer **Y-Koordinate** angegeben.



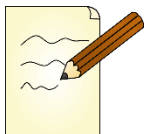
Aufgabe 1

Erzeuge als Erstes ein **Sprite**, indem du eine **Variable** mit dem Namen „sprite“ anlegst und anschließend festlegst, wo das Sprite sich zeigen soll. Das Sprite soll erzeugt werden, wenn das Programm **startet**.



Den Block findest du unter **Fortgeschritten**
→ [Spiel](#).

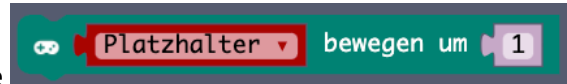
Lade das Programm herunter. Was passiert?



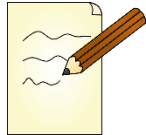


Aufgabe 2

Damit das Sprite sich bewegt, muss es den Befehl dazu erhalten. Ziehe dazu den **Dauerhaft**-Block in die Arbeitsfläche. Füge in den **Dauerhaft**-Block ein. Die **Variable** „Platzhalter“ musst du noch in „sprite“ umändern.



Teste erneut, was passiert:





Aufgabe 3

Das Sprite soll natürlich nicht stehenbleiben, sobald es am Rand angekommen ist.

Hierfür gibt es diesen Block:



Baue auch ihn in den **Dauerhaft**-Block ein, und ändere „Platzhalter“ in „sprite“ um. Dann kannst du dein Programm erneut testen.



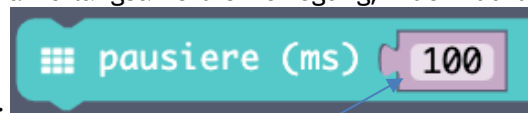
Aufgabe 4



Ui, da wird mir ja ganz schwindelig.

Da hat Calli recht. Verlangsame die Bewegung, indem du diesen Block in dein Pro-

gramm einbaust:



Experimentiere ein wenig mit der **Länge** des Pausen-Blocks, um eine gute Geschwindigkeit zu finden.

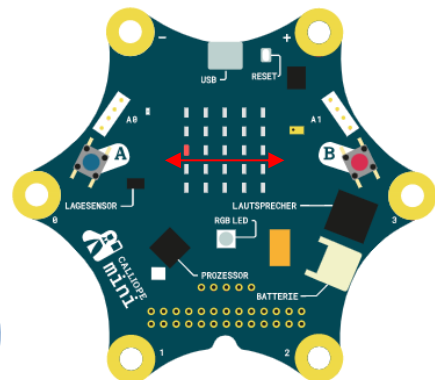


Aufgabe 5

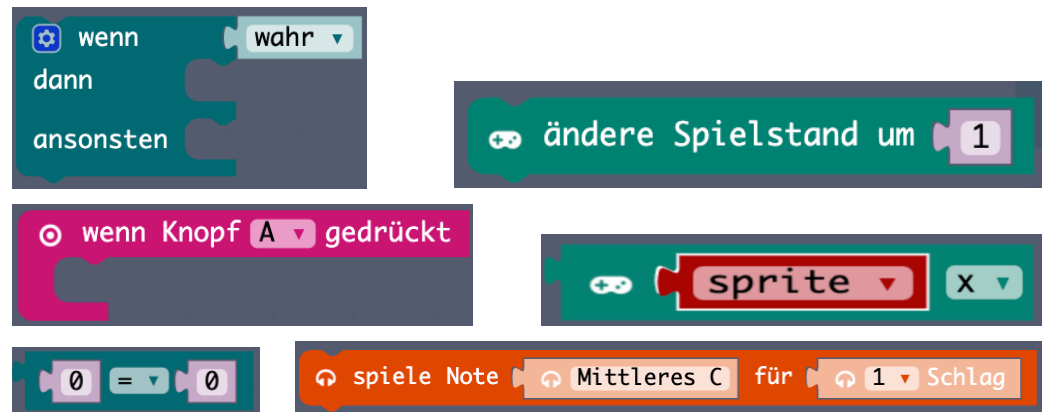
Nun zum eigentlichen Spiel: Spieler A soll einen Punkt erhalten, wenn er **Knopf A** genau dann drückt, wenn das Sprite am **linken Rand** angekommen ist. Gelingt ihm das, soll außerdem ein **Ton** abgespielt werden.



Das Sprite ändert nur auf der X-Achse seine Position. Wenn es am linken Rand angekommen ist, ist **X = 0**.



Kombiniere dazu diese Blöcke:



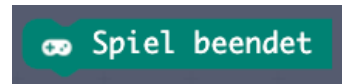
Bevor du nun weiterarbeitest, solltest du mit Hilfe der Musterlösung überprüfen, ob dein Programm bisher stimmig ist.



Aufgabe 6

Bis jetzt erhält Spieler A einen Punkt, und es ertönt ein Ton, wenn Spieler A genau dann Knopf A drückt, wenn das Sprite am linken Rand ($x = 0$) ist. Die Punkte werden aber nicht angezeigt. Außerdem passiert auch gar nichts, wenn Spieler A Knopf A drückt und das Sprite nicht am Rand ist.

Das Spiel braucht also noch ein Ende. Es soll enden, wenn Spieler A **Knopf A** drückt und das Sprite dann **nicht** am Rand ist. Dann soll auch der **Punktestand** angezeigt werden. Der *MakeCode-Editor* hat hierfür einen fertigen Block:



Füge diesen in den **Sonst**-Teil deines **Wenn-dann-sonst**-Blocks ein, lade das Programm auf den *Calliope mini*, und teste es.

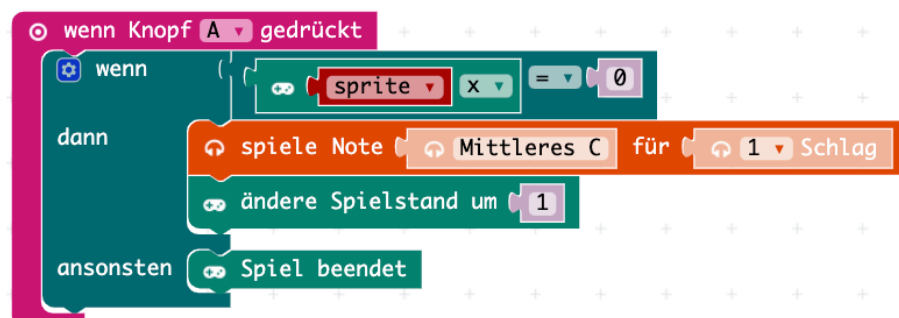


Aufgabe 7

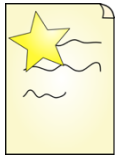
Supertoll! Dein erstes eigenes Spiel ist nun eigentlich schon fertig. Wenn du es aber mit einem weiteren Spieler als **Team** spielen möchtest, lässt es sich leicht noch ein wenig erweitern.

Spieler B soll für das Team einen Punkt erhalten, wenn er **Knopf B** genau dann drückt, wenn das Sprite am rechten Rand ($x = 4$) ist.

Du kannst das Programm von **Knopf A** für **Knopf B** nachbauen und dann ein paar kleine Änderungen vornehmen. Kreise die Stellen des Programms ein, die du verändern musst:



Ergänze dann dein Programm, sodass auch Spieler B mitspielen kann.



Aufgabe 8

Du kannst noch einen Überraschungsfaktor einbauen, um auf diese Weise den Schwierigkeitsgrad deines Spiels zu erhöhen.

Bisher bewegt das Sprite sich regelmäßig um je ein Feld weiter. Aber es wäre doch noch spannender, wenn man die Bewegung des Sprites nicht so leicht voraussagen könnte, weil es seine Position auch schon einmal um zwei oder drei Felder ändert. Alle Blöcke, die du hierfür brauchst, kennst du bereits aus den anderen Projekten.

- ✖ Überlege, welche Blöcke du brauchst, um dein Programm zu verändern.

- ✖ Setze deine Überlegungen nun in deinem Programm um.



Funktioniert dein Spiel so, wie du es gerne möchtest? Falls ja, dann ran an den *Calliope mini*. Mal sehen, wie viele Punkte du bzw. ihr holen könnt. Falls nein, dann suche in deinem Programm noch einmal nach Fehlern. Falls auch das nicht hilft, kannst du die Musterlösung zu Rate ziehen.

ANHANG A: MUSTERLÖSUNGEN

MUSTERLÖSUNGEN ZU DEN AUFGABEN AUS KAPITEL 1:

Aufgabe 1

Beispiel-Antworten (andere Antworten möglich): USB-Anschlussstelle, Batterieanschlussstelle, +, -, Schriftzug *Calliope mini*, Buchstabe A neben einem blauen Knopf, Buchstabe B neben einem roten Knopf, Reset-Knopf, die Zahlen 0 bis 3 neben goldenen Ringen, Lautsprecher (kann Töne abspielen), Lagesensor (bemerkt Bewegungen), Prozessor (rechnet), golden eingerahmte Löcher, A0 und A1 (Anschlüsse), RGB-LED (Lampe), Quadrat (LED-Bildschirm)

Aufgabe 2

EVA-Prinzip

Aufgabe 3

5	Der Knopf startet das Programm neu.
12	Das ist das Hirn des <i>Calliope mini</i> . Hier werden alle Infos verarbeitet.
9	Er kann Töne abspielen.
10	Seine 25 LEDs (kleine Lampen) können leuchten.
4	Wenn du den <i>Calliope mini</i> wie eine Batterie verwendest, ist das der Minuspol.
1	Hier kommt ein USB-Kabel rein. Es versorgt den <i>Calliope mini</i> mit Strom und verbindet ihn mit dem Computer.
6	Wenn du die Knöpfe drückst, machst du eine Eingabe. Du kannst die Knöpfe programmieren. Dann werden beim Drücken die programmierten Befehle ausgeführt.
7	Er merkt, in welche Richtung du den <i>Calliope mini</i> bewegst.
3	Wenn du den <i>Calliope mini</i> wie eine Batterie verwendest, ist dies der Pluspol.
11	Die Rot-Grün-Blau-Leuchte kann in allen Farben leuchten.
8	Über diese Pins kannst du auch eine Eingabe machen, damit der <i>Calliope mini</i> Befehle ausführt.
2	Hier kannst du die Batterie anschließen.

Aufgabe 4

Stromversorgung	1	Mirko-USB
	2	Batteriebuchse
	3	Pin +

	4	Pin –
<u>Eingabe</u>	5	Reset-Knopf
	6	Knopf A und Knopf B
	7	Lagesensor
	8	Pin 0 bis 3
<u>Ausgabe</u>	9	Lautsprecher
	10	LED-Bildschirm
	11	RGB-LED
<u>Verarbeitung</u>	12	Prozessor

Aufgabe 5


Beispiel-Antworten (andere Antworten möglich):

- ✗ Ganz links im Editor: die Simulation
- ✗ In den bunten Kästchen stehen unterschiedliche Kategorien, z. B. Töne. Rechts stehen zwei Blöcke in Türkis. Diese sind – ihrer Farbe entsprechend – unter Grundlagen zu finden.
- ✗ In der oberen Zeile: das Calliope- und auch das Microsoft-Logo; ein Ordner-Symbol, neben dem Projekte geschrieben steht
- ✗ In der unteren Zeile: ein Button / Knopf zum Herunterladen; eine Zeile, in die man etwas schreiben kann; daneben ein Speicher-Symbol, weitere Symbole wie + und –
- ✗ Die Teile in Türkis sehen aus wie kleine Lampen.

Aufgabe 6

Hier solltest du dir *Video 1: Der Editor* ansehen.

Aufgabe 7

Name	Funktion
Herunterladen	Beim Anklicken wird das Programm auf den <i>Calliope mini</i> geladen.
Simulator	Zeigt dir an, was das Programm auf der Arbeitsfläche macht, noch bevor du es auf den <i>Calliope mini</i> geladen hast.
USB-Kabel	Verbindet deinen <i>Calliope mini</i> mit dem Computer und somit mit dem Editor.
	Speichert beim Anklicken dein Programm.
Farbige Schubladen	Darin sind die verschiedenen Blöcke eingeordnet, je nachdem, wie sie zusammengehören.

Arbeitsfläche	Hier werden die Blöcke zu einem Programm zusammengesetzt.
Projekte	Wenn du die Schaltfläche anklickst, werden alle Programme angezeigt, die du bisher gespeichert hast.
Blöcke	Sehen aus wie Puzzle-Teile und bilden die einzelnen Bestandteile deines Programms.

Aufgabe 8

- ✖ Das Drücken der Taste B entspricht einer Eingabe.
- ✖ Das Erscheinen des Textes „Hallo“ ist eine Ausgabe.

Test

Der *Calliope mini* ist eine Platine, auf der mehrere Bauteile verbaut sind. Das wichtigste dieser Bauteile ist der Mikrocontroller, ein kleiner Computer. Daneben findet man viele weitere Bauteile: Strom kann der *Calliope mini* über einen USB-Anschluss oder eine Batterie erhalten. Das Quadrat in der Mitte des Sterns besteht aus vielen kleinen, roten LEDs, die sich einzeln steuern lassen. Eine andere LED, die RGB-LED, ist besonders, weil sie durch das Mischen der Farben Rot, Grün und Blau alle möglichen Farben herstellen kann. Neben den Knöpfen A und B, findet man einen weiteren Knopf: Wenn man den Reset-Knopf drückt, startet das Programm auf dem *Calliope mini* neu. Außerdem verfügt der *Calliope mini* über Sensoren wie z. B. den Licht- oder den Bewegungssensor, über die er etwas von außen wahrnehmen kann. Über einen Lagesensor merkt der *Calliope mini*, wie herum er gehalten wird.

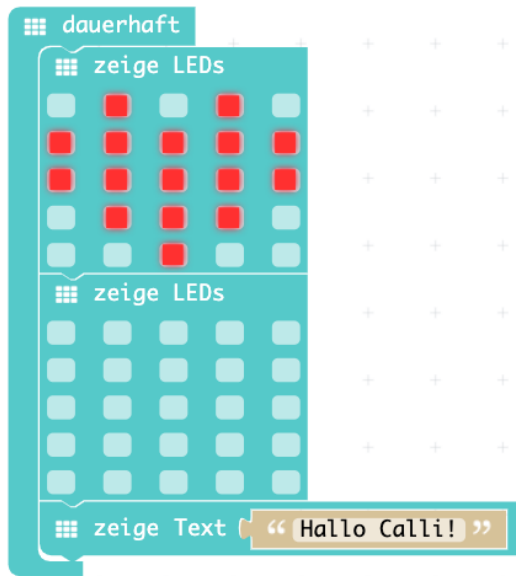
Computer machen nur, was der Mensch ihnen sagt. Wenn man also möchte, dass der *Calliope mini* bestimmte Befehle ausführt, dann muss man ihn programmieren. Man muss ihm ein Rezept schreiben, das man in der Informatik Algorithmus nennt. Da man dies nicht einfach wie für einen Freund auf einem Stück Papier machen kann, benutzt man einen *Editor*. Hier kann man Programme wie Puzzle über Blöcke zusammensetzen. Die Programmierblöcke findet man in den bunten Schubladen, aus denen man sie herausziehen kann. Zusammenpuzzeln kann man sie dann in der Arbeitsfläche. Der Simulator rechts führt dann vor, was das Programm auf dem *Calliope mini* macht.

Jedes Programm folgt demselben Prinzip: dem EVA-Prinzip. Dabei steht E für Eingabe, V für Verarbeitung und A für Ausgabe.

Aufgabe 1 und 2

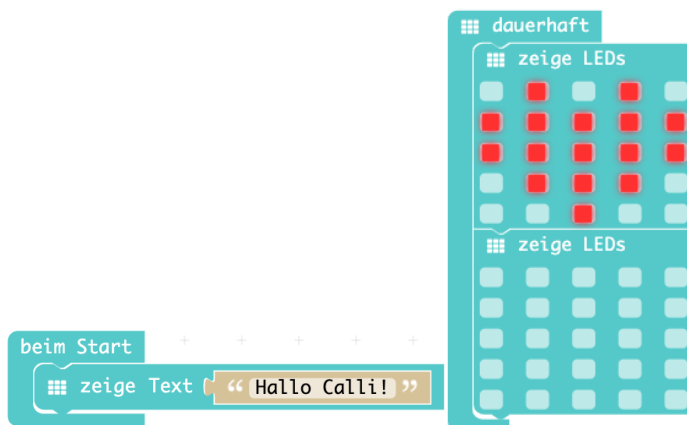
Um aus dem Gesicht ein Herz zu machen, müssen manche der roten LEDs aus-, andere angeschaltet werden. Die LEDs kannst du an- und ausschalten, indem du sie einfach anklickst.

Aufgabe 4 – Beispielprogramm



Den Text änderst du, indem du ihn anklickst, löschst und dann deinen neuen Text in das Kästchen schreibst.

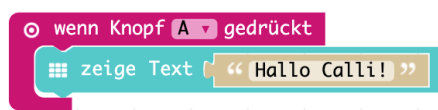
Aufgabe 5 – Beispielprogramm



Aufgabe 6

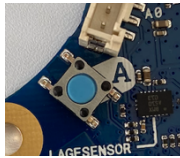
Wenn du den **Rest-Knopf** drückst, dann wird das ganze Programm neugestartet. Es beginnt also von vorne. Alles, was in dem **Start-Block** steht, wird dann auch noch einmal beim Start ausgeführt.

Aufgabe 7 - Beispielprogramm



Aufgabe 8

Knopf A drücken



Aufgabe 9 – Beispielprogramm



Test

- × Welchen Block verwendest du, damit etwas nur am **Anfang** ausgeführt wird?

☐ wenn Knopf A gedrückt

☐ beim Start

☒ dauerhaft

☐ zeige Text: Hallo!

☐ wenn geschüttelt

- × Ein Computer führt alle Befehle in einer bestimmten Reihenfolge aus. Wie nennt man diesen **Ablauf** in der Informatik?

☒ Sequenz

☐ Frequenz

☐ Endlosschleife

☐ Bedingung

☐ Simulator

- × Betrachte das folgende Programm. Was musst du machen, damit die Nachricht „Gut“ angezeigt wird?



☐ Knopf A drücken

☒ Knopf B drücken

- ☐ Reset-Knopf drücken
- ☐ *Calliope mini* schütteln
- ☐ Gar nichts

✖ In ein Programm kann man Schleifen einbauen, damit Programmteile wiederholt werden.
Wie nennen Informatiker die **Dauerhaft-Schleife**?



- ☐ Reset-Schleife
- ☐ Zählschleife
- ☒ Endlosschleife
- ☐ For-Schleife
- ☐ While-Do-Schleife

✖ Wenn du Knopf A drückst, dann machst du eine

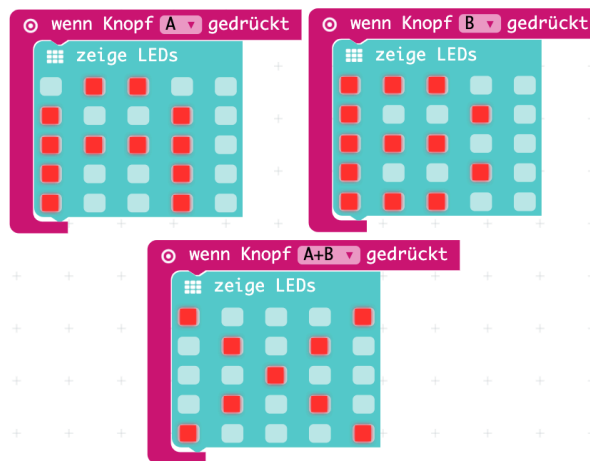
- ☐ Ausgabe
- ☐ Aufgabe
- ☐ Aufforderung
- ☐ Verarbeitung
- ☒ Eingabe

Projekt: Quiz-Buzzer

Aufgabe 1

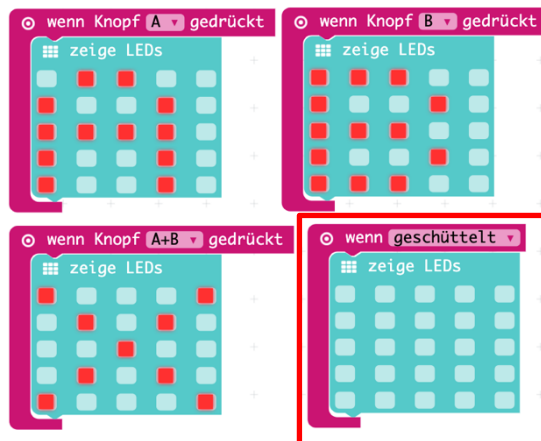
	Eingabe	Ausgabe
1	Knopf A  wird gedrückt.	Calliope mini zeigt „A“ an.
2	Knopf B  wird gedrückt.	Calliope mini zeigt „B“ an.
3	Knopf A und B werden gleichzeitig gedrückt.	Calliope mini zeigt „X“ an.

Aufgabe 2 – Beispielprogramm




Wenn **Knopf A** gedrückt wird, erscheint ein „A“. Wenn **Knopf B** gedrückt wird, erscheint ein „B“. Wenn **A und B** gleichzeitig gedrückt werden, erscheint ein „X“. Allerdings gehen die Buchstaben nicht mehr weg.

Aufgabe 3 – Beispielprogramm





Aufgabe 4

beim Start

Für dein Programm brauchst du in jedem Fall den  – Block. In diesen kannst du dann deine Animation mit dem **zeige-LEDs**-Block oder dem **zeige-Text**-Block einbauen.

Projekt: Sirene und Blaulicht

Aufgabe 1

- ✖ In die Lücke, in der „wahr“ steht, muss ein Block für **Knopf A** eingesetzt werden. Du findest ihn unter **Eingabe**: 
- ✖ Die Blöcke für die **Töne** findest du unter .



Bei diesem Programm erklingt ganz schnell hintereinander ein Ton. Es handelt sich um den Ton C, der für einen Schlag, also eine ganze Note lang, ertönt. Der Dauerton ist eine Täuschung, weil so lange immer wieder das C für einen Schlag gespielt wird, wie du **Knopf A** drückst.

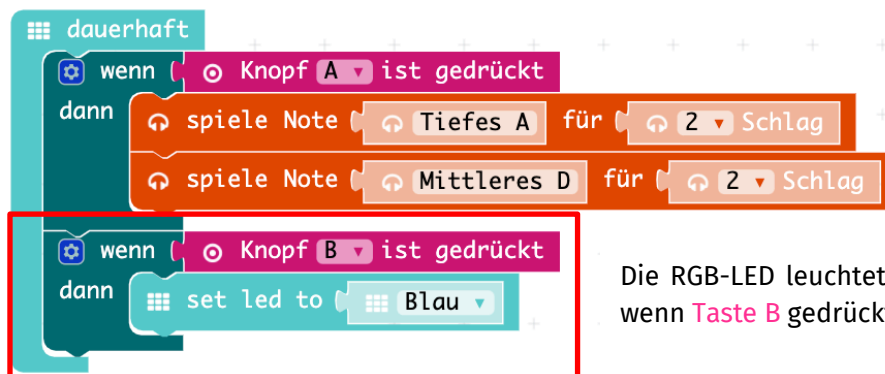
Aufgabe 2

Du kannst unter dem C-Block noch weitere Ton-Blöcke einbauen.

Aufgabe 3

Wenn der *Calliope mini* farbig leuchten soll, dann musst du die **RGB-LED** programmieren. Das größere LED-Display kann nämlich nur in Rot leuchten.

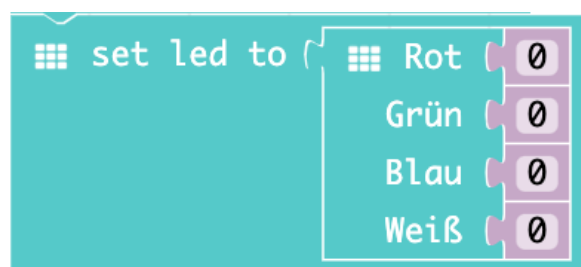
Aufgabe 4



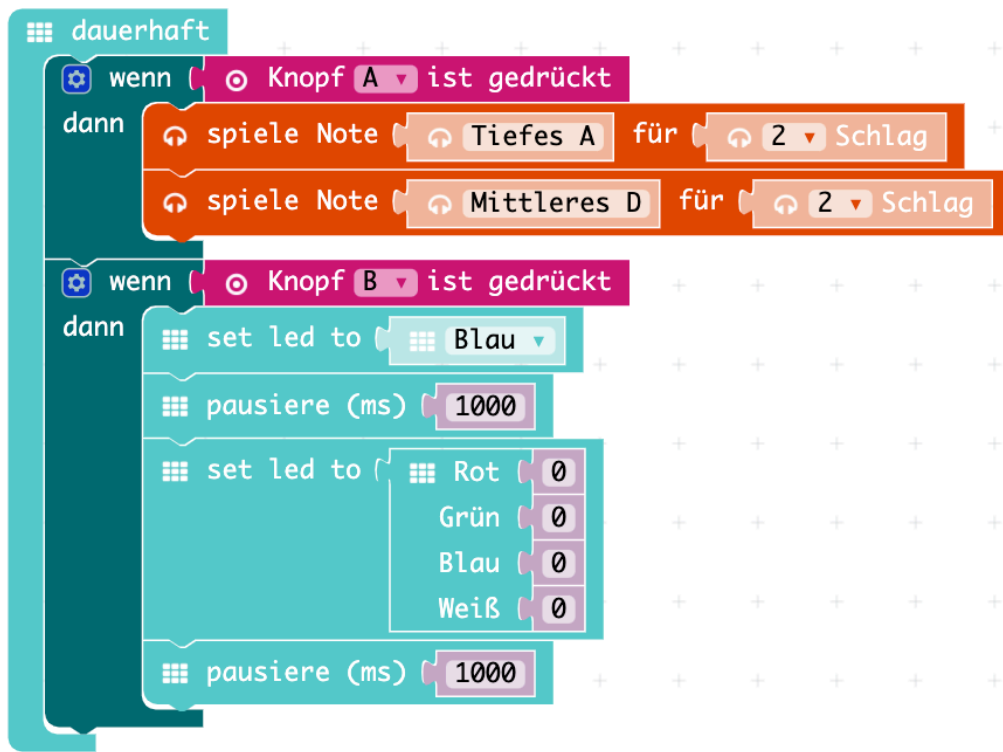
Die RGB-LED leuchtet dauerhaft blau, wenn **Taste B** gedrückt wird.

Aufgabe 5

Damit keine der drei LEDs leuchtet, müssen alle Werte auf 0 gesetzt werden:



Aufgabe 6 – Beispielprogramm



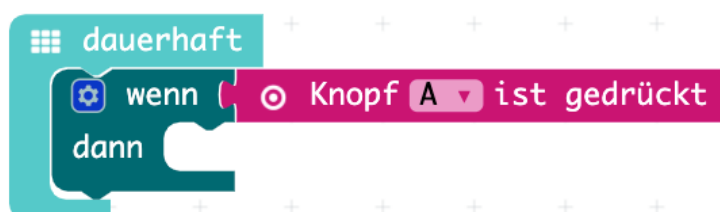
Projekt: Reizwort-Maschine

Aufgabe 1

Hier wird jeder eine andere Liste anlegen. Ein Beispiel:

0.	Sommerferien
1..	Schulhof
2.	Pizza
3.	Sonne
4.	Hund
5.	Garten
6.	<i>Calliope mini</i>
7.	Schülerlabor
8.	Wald

Aufgabe 2



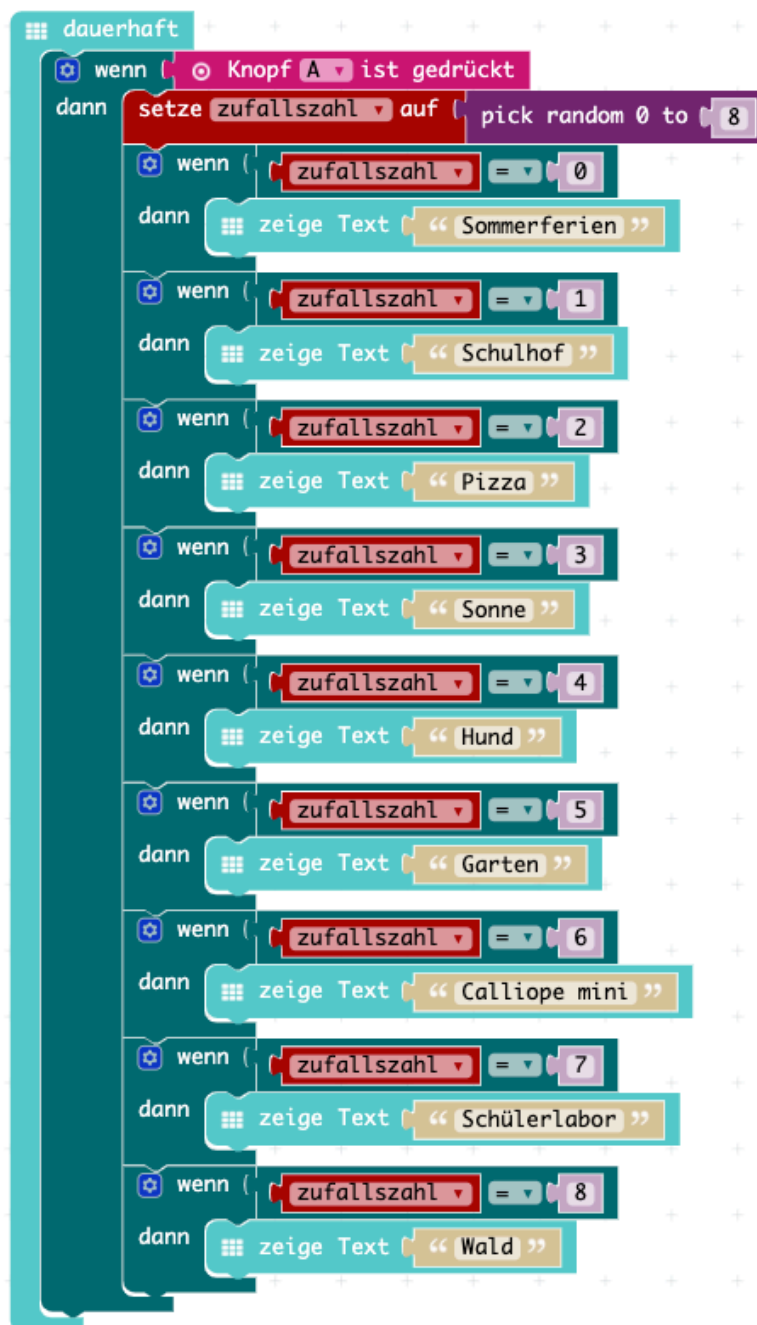
Aufgabe 3

Wenn du den Namen deiner neuen Variablen eingetippt hast und dann **OK** anklickst, findest du deine neue Variable in der Schublade **Variablen** wieder.

Aufgabe 4



Aufgabe 5 – Beispielprogramm



Aufgabe 6

Für deine Geschichte gibt es keine Musterlösung, da sie deiner Phantasie entstammt. Vielleicht hast du aber Lust, deinen Eltern oder Geschwistern deine superkreative Geschichte vorzulesen.

Aufgabe 7

Programmbausteine, die geändert werden müssen:

Da es 26 Buchstaben gibt, müssen Zufallszahlen zwischen 0 und 25 möglich sein.

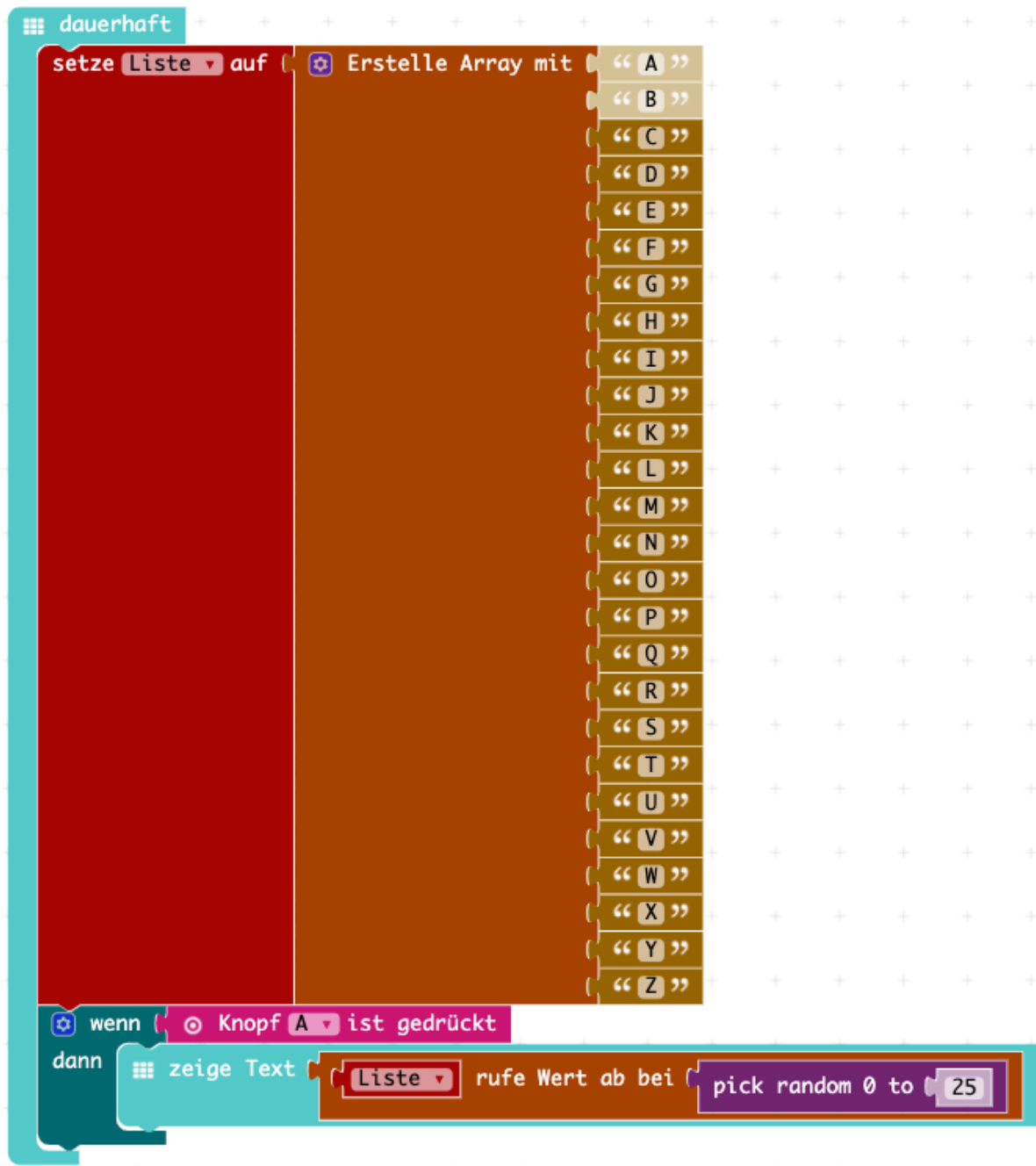
In den Textblöcken müssen die Wörter durch Buchstaben ersetzt werden.

Hier müssen 17 weitere Wenn-dann-Blöcke eingebaut werden, da das Alphabet 26 Buchstaben hat.



An die zwei schon vorhandenen Werte müssen noch 24 weitere.


Buchstaben-Zufallsgenerator

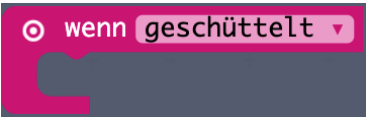



Test


- × Du hast einen Begrüßungstext programmiert, der erscheint, wenn du Knopf A drückst. Außerdem erscheint ein Verabschiedungstext, wenn du Knopf B drückst. Nun möchtest du noch programmieren, dass die Nachrichten wieder verschwinden, wenn du den *Calliope mini* schüttelst. Welche Blöcke brauchst du dazu?

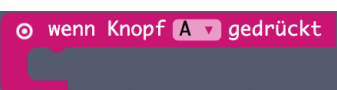
☐ 

☐ 

☐ 

☒ 




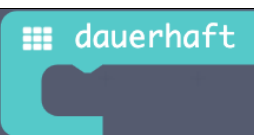

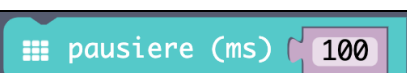
☐ 

☒ 

- ✖ Bei einem Programm wird zunächst einmal abgefragt, ob eine **Bedingung** wahr ist, bevor ein Befehl ausgeführt wird. Wie nennt man dies in der Informatik?

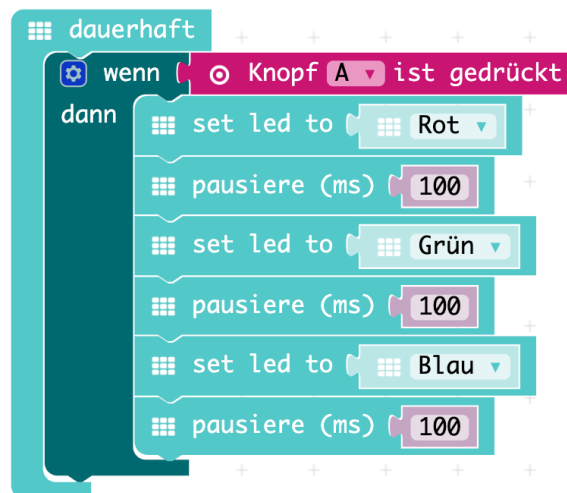
- ☒ Bedingte Verzweigung
- ☐ Eingabe
- ☐ Verarbeitung
- ☐ Ausgabe
- ☐ Variable

- ✖ Du möchtest ein **Diskolicht** programmieren, das in **Rot**, **Grün** und **Blau** leuchtet, wenn du Knopf A drückst. Dazu brauchst du folgende Blöcke. Bringe sie in die richtige Reihenfolge:

8	
2	
6	
1	
4	
5	

7	
9	
3	

Das Programm sieht dann so aus:



- ✖ Vervollständige den Satz: Eine **Variable** kann man sich vorstellen wie ...
 - ☐ eine Bedingung, die erfüllt sein muss, damit der *Calliope mini* etwas macht.
 - ☒ eine große Kiste, in der etwas aufbewahrt wird.
 - ☐ einen Knopf, den man drücken kann.
 - ☐ ein Bildschirm, auf dem etwas angezeigt wird.

- Was macht dieser Block ?
 - ☐ Die Zahlen 0 bis 4 werden angezeigt.
 - ☐ Es wird rückwärtsgezählt von 4 bis 0.
 - ☒ Eine zufällige Nummer zwischen 0 und 4 wird aufgerufen.
 - ☐ Man sieht auf jeden Fall eine 4 auf dem Display.

Aufgabe 1

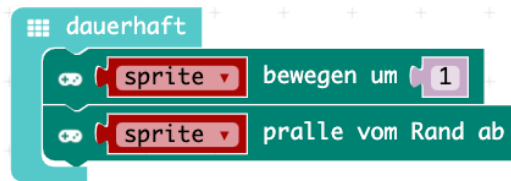
In der Mitte des LED-Bildschirms (an Position x: 2, y: 2) leuchtet eine LED. Diese bewegt sich jedoch nicht.

Aufgabe 2

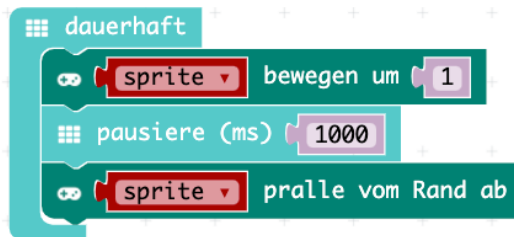


Das Sprite bewegt sich; allerdings nur bis zum Rand. Dann bleibt es stehen.

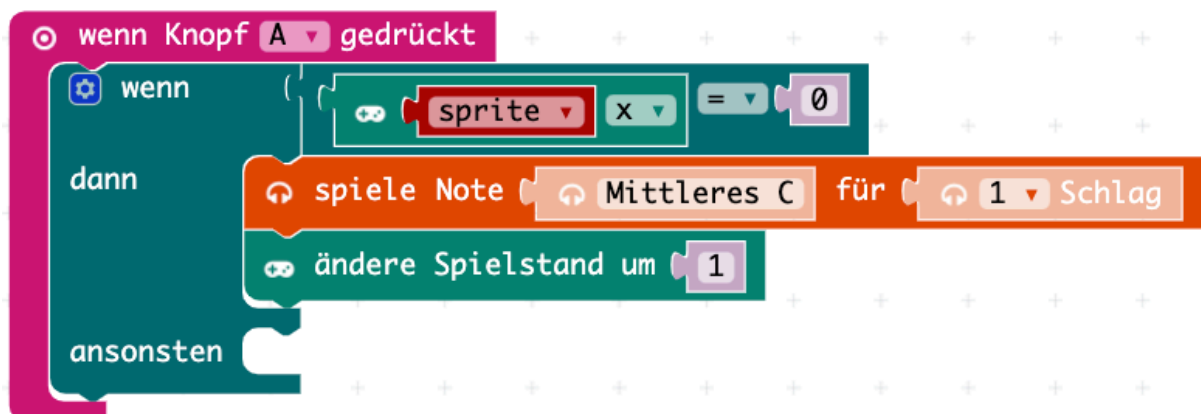
Aufgabe 3



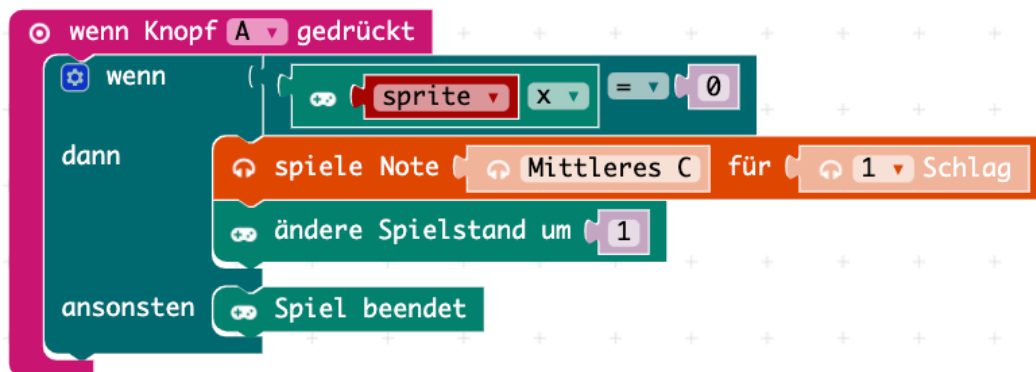
Aufgabe 4



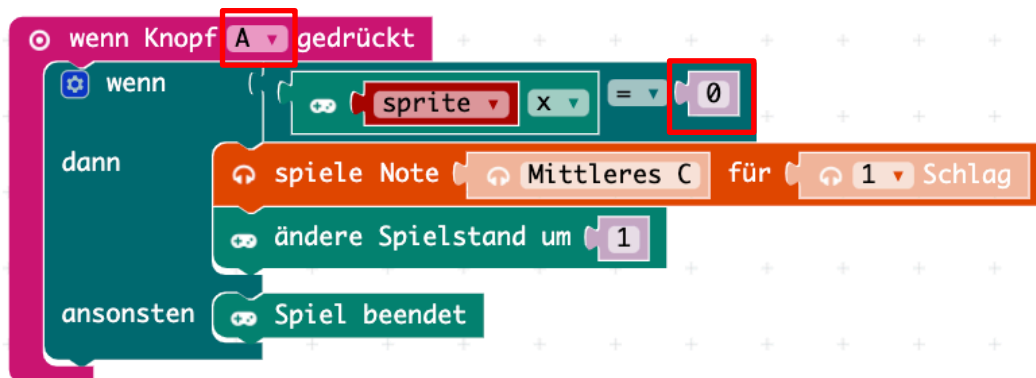
Aufgabe 5



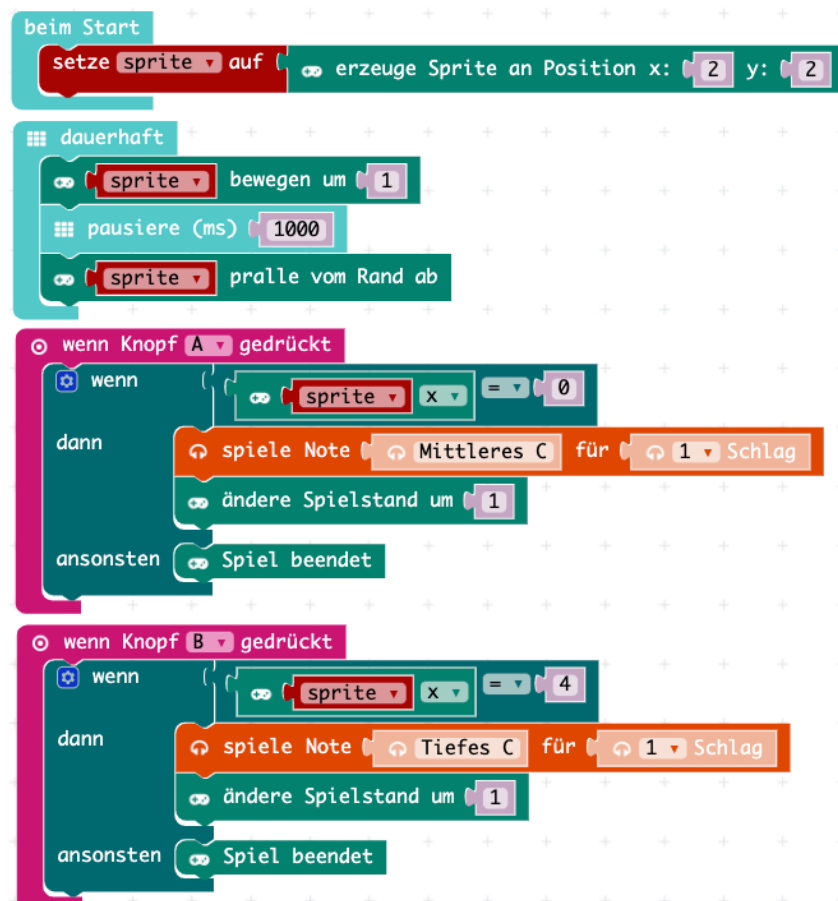
Aufgabe 6



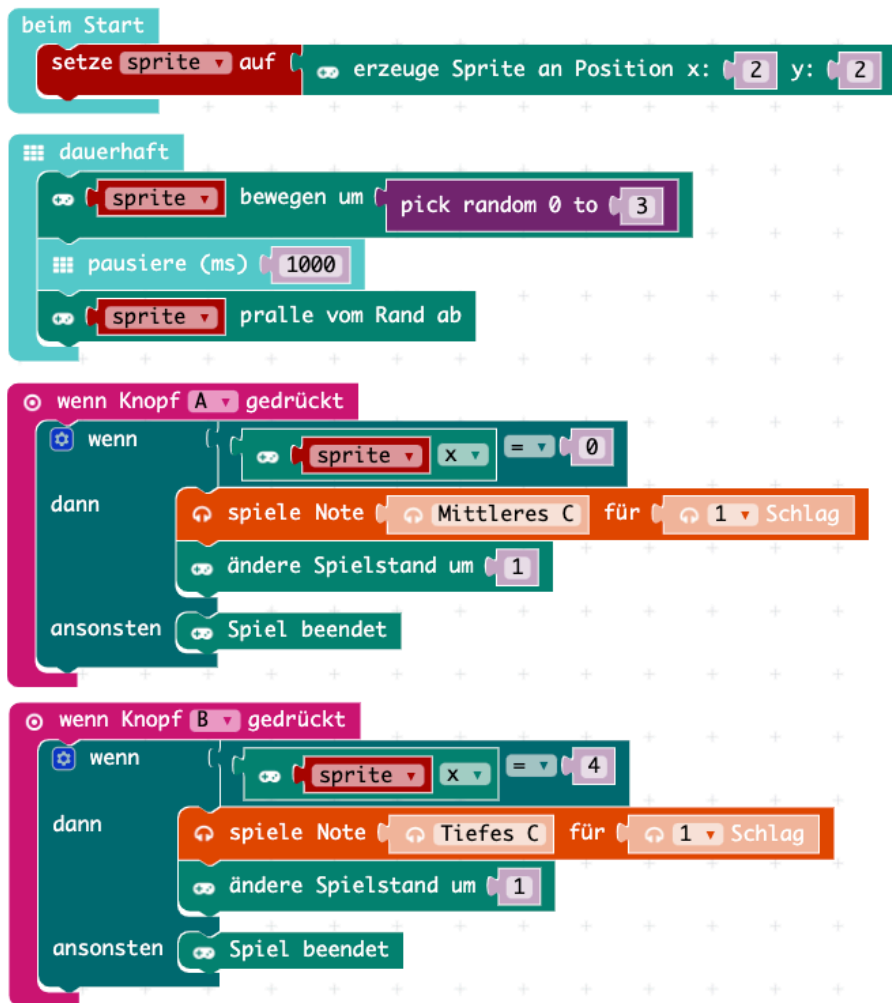
Aufgabe 7



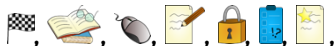






Sprite-Pong-Beispielprogramm (regelmäßige Bewegung des Sprites)



Sprite-Pong-Beispielprogramm (unregelmäßige Bewegung des Sprites)



ANHANG B: ABBILDUNGSVERZEICHNIS

- **Titelseite und Kurzinformation** – Quelle: Screenshots des MakeCode-Editors, <https://makecode.calliope.cc/>
- **Arbeitssymbole**  – Quelle: InfoSphere
- **Calli**  – Quelle: InfoSphere
- **Getränkeautomat**  – Quelle: InfoSphere
- **Chrome-Logo**  – Quelle: wikipedia.org (gemeinfrei / public domain)
- **Firefox-Logo**  – Quelle: wikipedia (Creative-Commons 3.0)
- **Safari-Logo**  – Quelle: commons.wikimedia.com, Autor: Bruce The Deus (CC-BY-SA-4.0)
- **Schrank**  – Quelle: InfoSphere
- **Programmierblöcke** – Quelle: Screenshots des MakeCode-Editors, <https://makecode.calliope.cc/>

ANHANG C: MEDIOTHEK

- InfoSphere – Schülerlabor Informatik (2017): Video 1 – Der Editor, https://www.youtube.com/watch?time_continue=1&v=76ryLoWfYhs&feature=emb_logo, zuletzt abgerufen am: 6.5.2020.
- InfoSphere – Schülerlabor Informatik (2017): Video 2 – Mit Blöcken Arbeiten, https://www.youtube.com/watch?v=Bzt7cnlBZsg&feature=emb_logo, zuletzt abgerufen am: 6.5.2020.
- MakeCode-Editor: <https://makecode.calliope.cc/>, zuletzt abgerufen am: 6.5.2020.

ANHANG D: LITERATURANGABEN

- Abend, Michael; Gramowski, Kirstin; Pelz, Lars und Poloczek, Bernd (2017): *Coden mit dem Calliope mini. Programmieren in der Grundschule. Schülermaterial ab Klasse 3*. Berlin: Cornelsen.
- Bergner, Nadine und Leonhardt, Thiemo (2018): *Programmieren mit dem Calliope mini für Dummies Junior*. Weinheim: WILEY.