

Herzlich Willkommen!

Weißt du, was eine Simulation ist? Man kann zum Beispiel so tun, als wäre man eine Elfe oder ein Dinosaurier. Das können Forscher auch bei Raketenstarts oder beim Urknall machen: sie tun so, als ob sie wirklich in der Rakete sitzen. Oder sie stellen sich vor, wie der Urknall ausgesehen haben könnte. In Wirklichkeit ist es aber: nur eine Simulation!

Zusammen können wir auch so tun, als hätten wir einen Calliope mini. Damit können wir spannende Sachen programmieren, z.B. Töne und Lichteffekte!

Ich bin schon ganz aufgeregt und möchte am liebsten sofort loslegen. Leider bin ich vor lauter Begeisterung manchmal nicht ganz genau beim Lesen der Anleitungen. Am besten machen wir das gemeinsam, dann kann nichts schiefgehen!

Wenn du Fragen hast, kannst du mir gerne eine E-Mail schreiben. Meine E-Mail Adresse lautet: [pfiffikus@deutsches-museum-bonn.de](mailto:pfiffikus@deutsches-museum-bonn.de)

Und wenn du Spaß am Programmieren hast, kannst du gerne zu einem Workshop mit mir ins Deutsche Museum Bonn kommen!

Los geht's!

Deine Eule Pfiffikus



# 1. Der Calliope mini

Bei dem Calliope mini handelt es sich um einen Mikrocontroller, also eine Art kleiner Computer. Er wurde extra für den Einsatz in Schulen und Lerngruppen entwickelt.

In den Calliope sind bereits einige nützliche Sensoren und Aktoren verbaut. Ein Beispiel für einen Sensor ist ein Temperatursensor. Mit diesem kann die Temperatur gemessen werden. Eine LED ist ein Beispiel für einen Aktor. Dieser kann an und ausgeschaltet werden.

**Aufgabe 1:** Beschrifte die nachfolgende Abbildung des Calliope mini. Nutze dazu die Internetseite: <https://www.roberta-home.de/interaktiver-calliope>

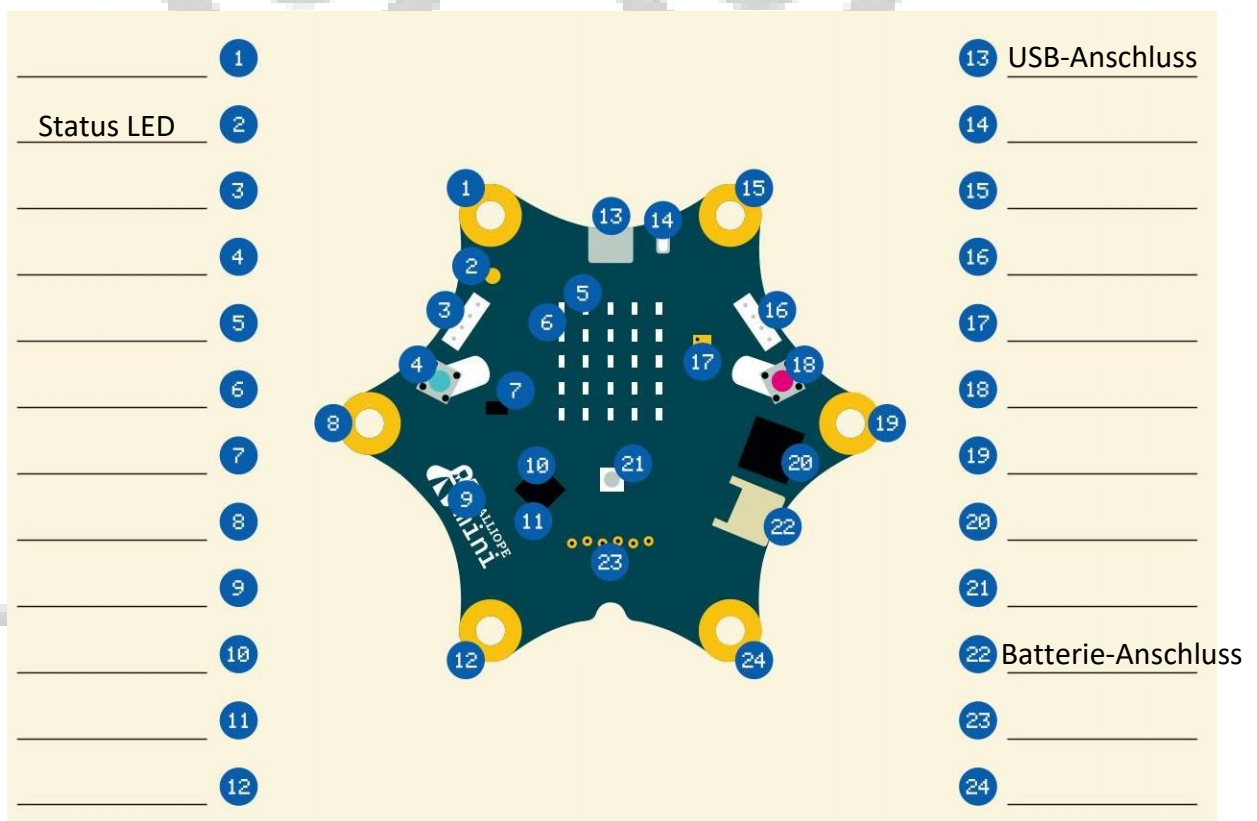




Abbildung 1: Sensoren und Aktoren des Calliope mini

## 2. Editor

Für die Programmierung des Calliope mini benötigst du eine Entwicklungsumgebung. Damit kannst du ein Programm für den Calliope erstellen. Wir nutzen den Editor Open Roberta Lab mit der Programmiersprache NEPO. Dieser ist für jeden kostenlos über den Browser zugänglich.

**Aufgabe 2:** Starte einen Browser deiner Wahl (Google Chrome, Firefox, etc.) und öffne die folgende Webseite: <https://lab.open-roberta.org/> und wähle als System den Calliope mini aus.

Nun siehst du die Programmierumgebung Open Roberta Lab. Dies ist alles, was du für die folgenden Aufgaben benötigst.

Auf der linken Seite befinden sich die Programmierblöcke, die in unterschiedlich farbigen Kategorien sortiert sind. Diese Farben kennzeichnen, zu welcher Kategorie ein Block gehört. Zu Beginn ist der Anfänger-Modus zu sehen. Dies erkennst du an dem Symbol . Wenn du auf das Symbol daneben klickst, gelangst du in den Experten-Modus ().

In der Mitte der Programmierumgebung befindet sich der Startblock. Dort befindet sich später dein Programm.

Am rechten Rand der Programmierumgebung befinden sich fünf verschiedene Symbole. Wenn du auf das zweite Symbol von oben klickst, werden dir Hilfen zu allen Befehlen angezeigt. Das unterste Symbol „SIM“ ist für uns das zweitwichtigste Element dieses Editors. Hier kannst du das von dir entwickelte Programm an einem virtuellen Calliope mini testen.

### 3. Hallo Welt

#### Der Start-Block



Abbildung 2: Start-Block

Der Start-Block ist der Beginn jedes Programms. An ihn werden alle weiteren Blöcke angehängen. Dadurch entsteht dein eigenes Programm. Alle Blöcke werden durch die dreiecksförmige Anschlussstelle miteinander verbunden. Zum Verbinden werden die Blöcke mit der Maus unter den letzten Block gezogen. Wenn die Blöcke blass sind, wurde die Verbindung nicht richtig hergestellt. Die Blöcke müssen näher aneinander gebracht werden.

Dein Programm wird bei der Ausführung von oben nach unten abgearbeitet. Der Calliope führt also zuerst den obersten Block und zum Schluss den untersten Block aus.

Wenn du ein Programm nicht mehr benötigst, dann kannst du es einfach vom Start-Block lösen und in den Papierkorb unten rechts in der Ecke der Programmierumgebung ziehen.

#### Der Block „Zeige Text“



Abbildung 3: Block "Zeige Text"

Über den Block „Zeige Text“ kannst du Texte oder Zeichen auf der LED-Matrix anzeigen lassen. Du kannst durch Klicken auf das Wort „Halo“ einen eigenen Text einfügen, der angezeigt werden soll.

**Aufgabe 3:** Schreibe ein Programm, mit dem „Hallo Welt“ auf der LED-Matrix angezeigt wird. Du darfst dafür alle Blöcke verwenden, die du bisher kennengelernt hast. Probiere aus, was der Unterschied zwischen „Zeige Text“ und „Zeige Zeichen“ ist.

**Aufgabe 4:** Führe nun dein Programm in der Simulation aus. Dazu gehst du in die Simulationsansicht und klickst auf den Play-Button (siehe Abbildung 4). Nun wird dein Programm ausgeführt.

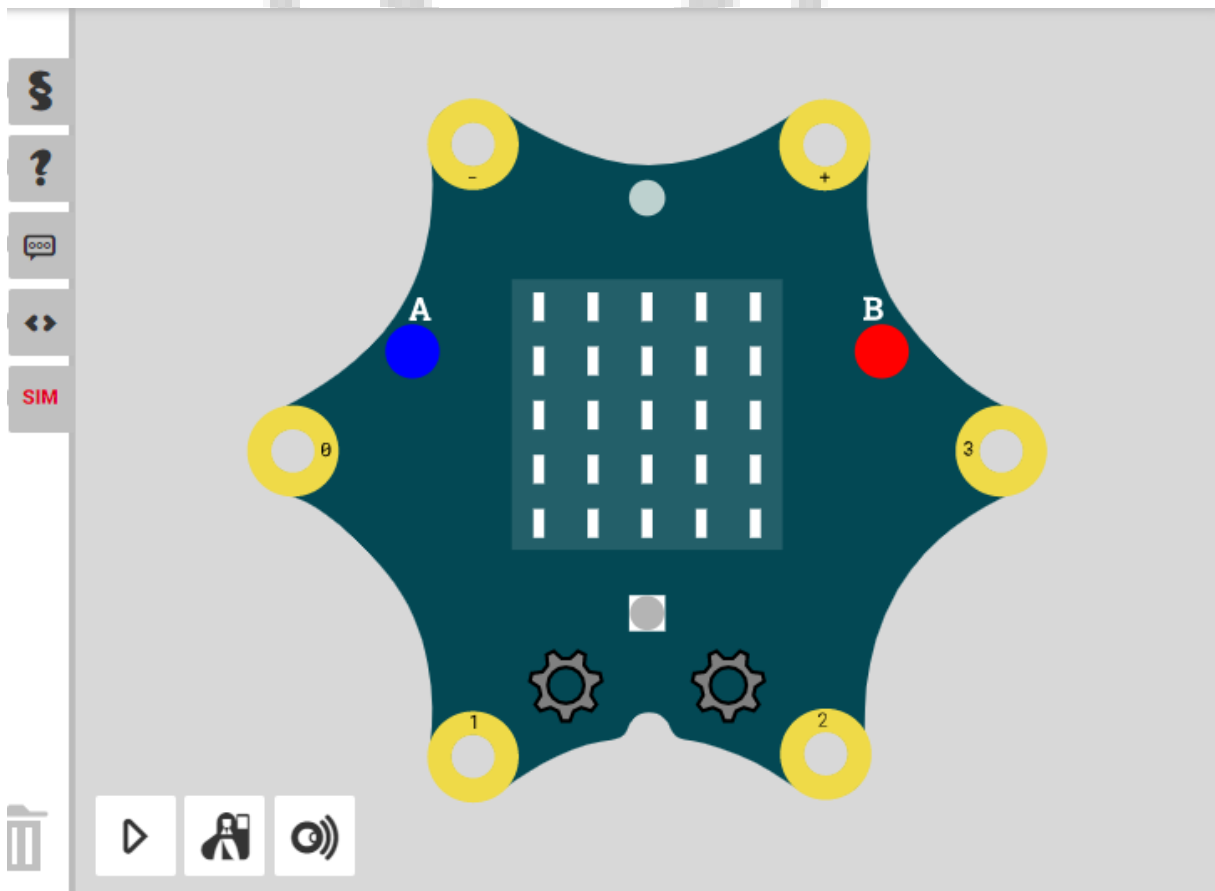


Abbildung 4: Simulationsansicht

## 4. Bilderwechsel

### Der Block „Zeige Bild“



Abbildung 5: Block "Zeige Bild"

Mit diesem Block wird ein Bild auf der LED-Matrix angezeigt. Über den Pfeil neben dem Bild kannst du dir auch ein anderes Bild aussuchen, das angezeigt werden soll.

Ebenso wie dieser Block haben viele weitere Blöcke eine Einkerbung. In diese Einkerbung können Blöcke mit Kontaktstellen angeschlossen werden. Die Farbe der Einkerbung zeigt dir, aus welcher Kategorie der Block stammen darf (siehe Abbildung 6). Wenn keine Farbe zu sehen ist, können unterschiedliche Blöcke angeschlossen werden. An den Block „Zeige Bild“ können also auch andere Blöcke aus der Kategorie „Bilder“ angehängt werden (siehe Abbildung 7).



Abbildung 6: Block "Zeige Bild"

Es gibt auch die Möglichkeit, ein eigenes Bild zu erstellen. Dazu wird ebenfalls der Block „Zeige Bild“ verwendet, an den nun aber ein Abbild der LED-Matrix angehängt ist (siehe Abbildung 7). Jedes Kästchen stellt eine LED der 5x5-LED-Matrix dar. Durch Klicken in das Kästchen wird ein „#“ eingefügt. Dies bedeutet, dass die LED leuchten soll.

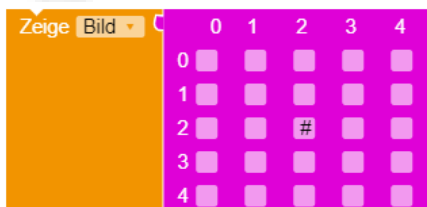


Abbildung 7: Block "Zeige Bild"

**Aufgabe 5:** Schreibe ein Programm, in dem zwei Bilder nacheinander angezeigt werden und führe es in der Simulation aus.

## Der Block „Warte ms“

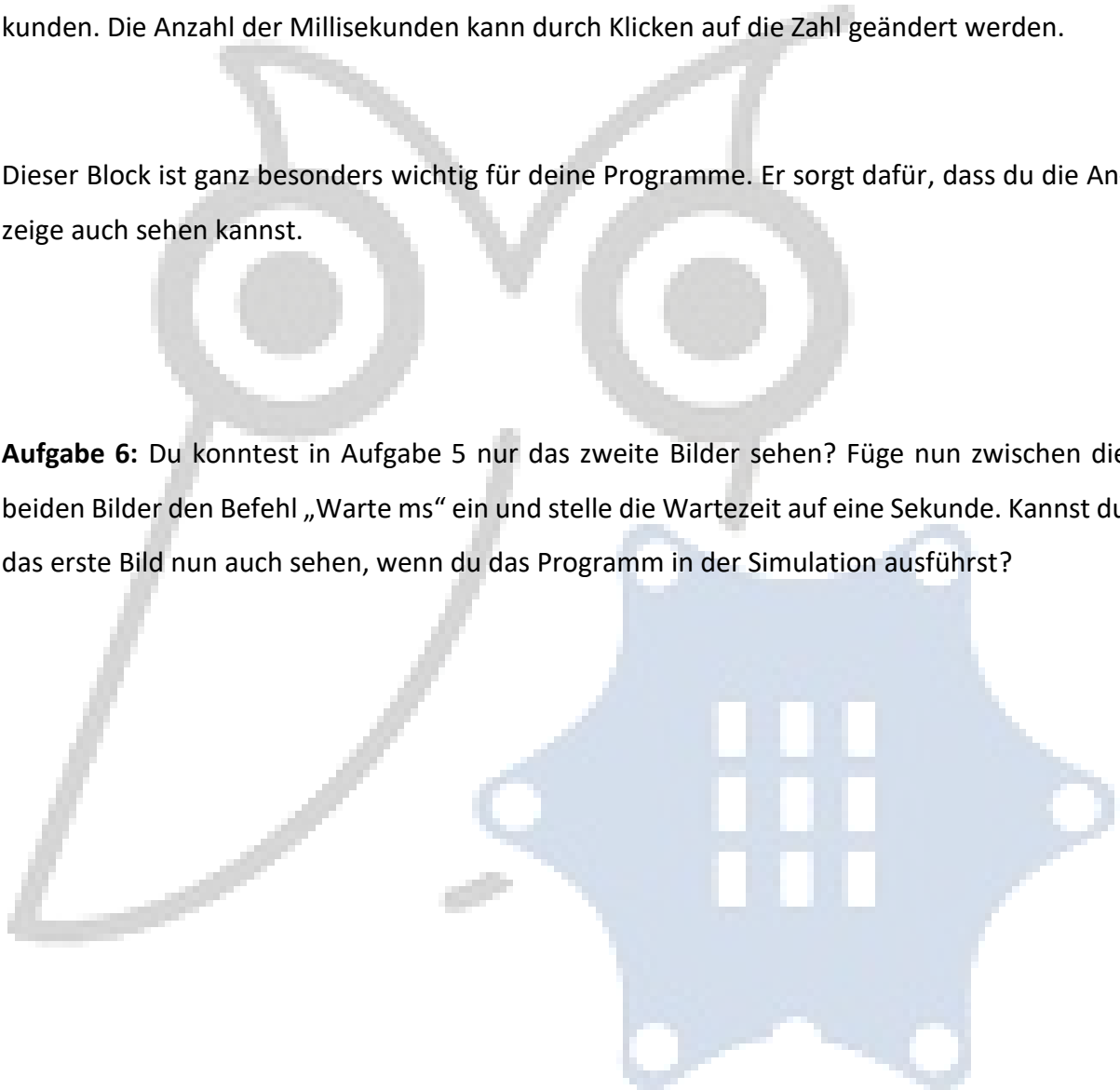


Abbildung 8: Block "Warte ms"

Durch den Block „Warte ms“ wird das Programm für eine bestimmte Anzahl an Millisekunden unterbrochen und gewartet. „ms“ steht für Millisekunden. 1 Sekunde entspricht 1000 Millisekunden. Die Anzahl der Millisekunden kann durch Klicken auf die Zahl geändert werden.

Dieser Block ist ganz besonders wichtig für deine Programme. Er sorgt dafür, dass du die Anzeige auch sehen kannst.

**Aufgabe 6:** Du konntest in Aufgabe 5 nur das zweite Bilder sehen? Füge nun zwischen die beiden Bilder den Befehl „Warte ms“ ein und stelle die Wartezeit auf eine Sekunde. Kannst du das erste Bild nun auch sehen, wenn du das Programm in der Simulation ausführst?



## 5. Trickfilm

### Der Block „Wiederhole ... mal, mache ...“



Abbildung 9: Schleife mit bestimmter Anzahl

Bei dem Block „Wiederhole ... mal, mache ...“ handelt es sich um eine Schleife. Sie führt die darin enthaltenen Befehle für eine angegebene Anzahl an Wiederholungen aus. Durch Klicken auf die Zahl kann diese verändert werden.

**Aufgabe 7:** Erstelle ein Programm, das 5-mal hintereinander zwischen einem lachenden und einem weinenden Smiley wechselt. Danach soll ein Herz angezeigt werden. Führe es anschließend in der Simulation aus.





## 6. Musik

### Der Block „Wiederhole unendlich oft“



Abbildung 10: Block "Wiederhole unendlich oft"

Auch bei diesem Block handelt es sich um eine Schleife. Sie hört aber nie auf. Aus diesem Grund wird sie auch Endlosschleife genannt.

### Der Block „Spiele ganze Note“

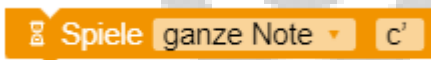


Abbildung 11: Block "Spiele ganze Note"

Durch diesen Block kann der Calliope Töne ausgeben. Viele Töne zusammengesetzt ergeben eine Melodie. Durch die Pfeile kann sowohl die Länge der Töne als auch die Höhe des Tons verändert werden.

**Aufgabe 8:** Komponiere deine eigene Melodie und lasse sie in Endlosschleife ertönen.

**Zusatzaufgabe 1:** Wenn du möchtest, kannst du auch eine einfache Melodie, wie zum Beispiel „Happy Birthday“ oder „Alle meine Entchen“, endlos ertönen lassen.

## 7. Interne LED

### Der Block „Schalte LED an“



Abbildung 12: Block "Schalte LED an"

Durch diesen Befehl kannst du die Farbe der internen RGB-LED am Calliope programmieren. Durch Klicken auf die rote Fläche kann eine andere Farbe ausgewählt werden.

Eine weitere Möglichkeit für die Auswahl einer Farbe ist der Kontaktblock „Farbe mit ...“ (siehe Abbildung 13). Jeweils für rot, grün, blau und weiß kann ein Wert zwischen 0 und 255 eingetragen werden. Dadurch ergeben sich weitere Farben, in der die RGB-LED leuchten kann.

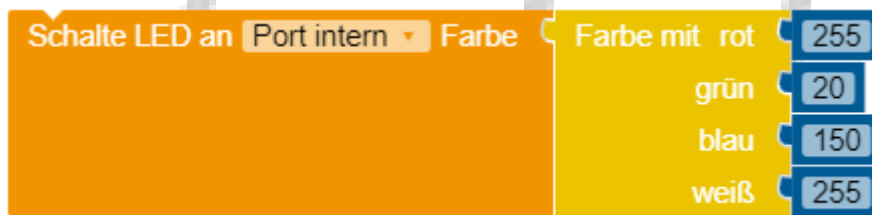


Abbildung 13: Block "Farbe mit ..."

**Aufgabe 9:** Lass die interne LED des Calliope abwechselnd rot und grün leuchten.

**Zusatzaufgabe 2:** Lass die interne LED abwechselnd in zwei von dir gemischten Farben leuchten.

## 8. Farben Touch-Pins

### Der Block „Wenn ..., mache ...“

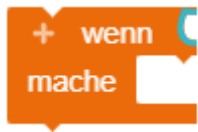


Abbildung 14: Block "Wenn ..., mache ..."

Dieser Block hängt von einer Bedingung ab, die an der Kontaktstelle festgelegt wird. Ist die Bedingung erfüllt, wird der Block ausgeführt. Ist die Bedingung nicht erfüllt, wird der Block ignoriert und das Programm nach dem Block weiter ausgeführt. Meistens möchte man, dass eine Bedingung nicht nur einmal, sondern mehrmals hintereinander geprüft wird. Aus diesem Grund muss dieser Befehl immer am besten in eine Endlosschleife gesetzt werden. Mehrere Tests auf unterschiedliche Bedingungen können hintereinander in die gleiche Schleife gesetzt werden.

### Der Block „Pin 0 gedrückt“



Abbildung 15: Block "Pin 0 gedrückt"

Dieser Block kann eine mögliche Bedingung für den „Wenn ..., mache ...“-Block sein. Wenn der Touch-Pin 0 gedrückt wird, sollen die Blöcke innerhalb der Bedingung ausgeführt werden. Durch das Klicken auf den Pfeil können die Touch-Pins 0 bis 3 ausgewählt werden. Dieser Block gehört zu der Kategorie „Sensoren“ dazu, da gemessen wird, ob der Pin gedrückt wird oder nicht. In der Simulation kannst du einfach auf die Touch-Pins klicken, damit sie gedrückt werden.

**Aufgabe 10:** Schreibe ein Programm, das für jeden Touch-Pins 0 bis 3 testet, ob er gedrückt wurde. Wenn dies der Fall ist, so soll die interne LED für jeden Touch-Pin in einer anderen Farbe leuchten. Du kannst dir auch eine eigene Farbe mischen.

## 9. Tasten A und B

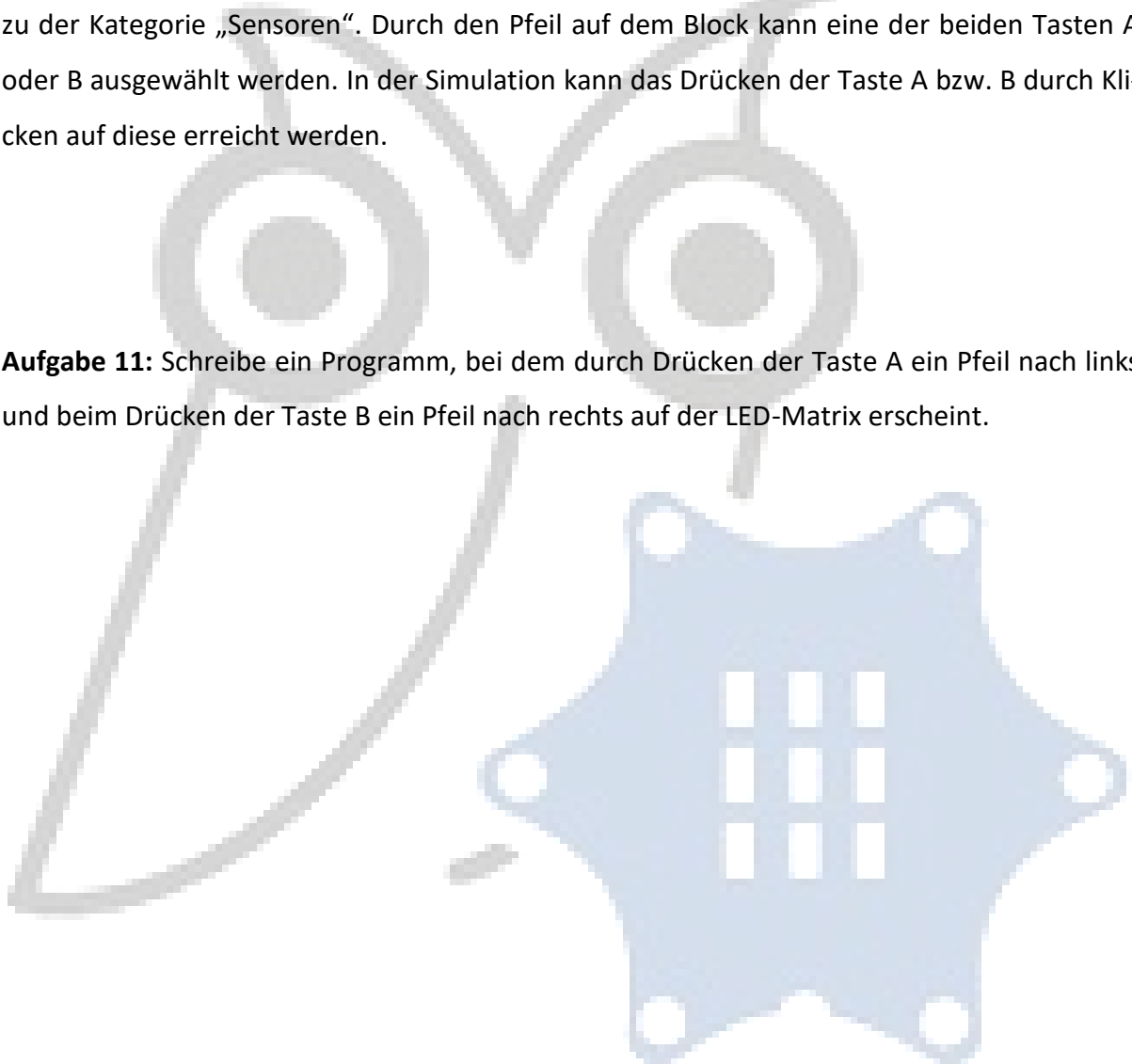
### Der Block „Taste A gedrückt“

Taste **A** gedrückt?

Abbildung 16: Block "Taste A gedrückt"

Dieser Block ist ebenfalls eine Bedingung für den Block „Wenn ..., mache ...“ und gehört auch zu der Kategorie „Sensoren“. Durch den Pfeil auf dem Block kann eine der beiden Tasten A oder B ausgewählt werden. In der Simulation kann das Drücken der Taste A bzw. B durch Klicken auf diese erreicht werden.

**Aufgabe 11:** Schreibe ein Programm, bei dem durch Drücken der Taste A ein Pfeil nach links und beim Drücken der Taste B ein Pfeil nach rechts auf der LED-Matrix erscheint.



## 10. Würfel

### Der Block „ganzzahliger Zufallswert zwischen ...“

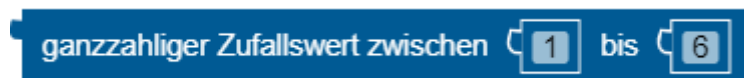


Abbildung 17: Block "ganzzahliger Zufallswert zwischen ..."

Dieser Block liefert einen zufälligen Wert in einem bestimmten Zahlenraum, den du selbst festlegen kannst. Der Block kann ebenfalls an den Block „Zeige Zeichen“ angehängt werden.

**Aufgabe 12:** Programmiere deinen eigenen Würfel. Wenn eine Taste oder ein Touch-Pin gedrückt wird, soll die zufällig gewählte Zahl auf der LED-Matrix angezeigt werden.



## 11. Würfel mit Augenzahlen

### Variable

Eine Variable ist ein Platzhalter für unterschiedliche Werte. Damit du eine Variable nutzen kannst, muss sie vorher erstellen. Dazu klickst du auf das „+“ im Startblock. Durch das „-“ kann die Variable wieder entfernt werden. Du kannst der Variablen einen Namen geben. Hinter dem Doppelpunkt steht der Typ der Variablen. Dies können Zahlen, Zeichenketten, logische Werte (wahr und falsch), Farben oder Bilder sein. Abhängig vom Typ muss der Variablen ein Anfangswert zugewiesen werden.



Abbildung 18: Start-Block mit einer Variable

### Der Block „Schreibe Zahl“

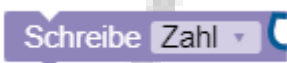


Abbildung 19: Block "Schreibe Zahl"

Durch diesen Block kann der Wert der vorher erstellten Variablen verändert werden. Dazu kann zum Beispiel ein Sensor-Block mit dunkelblauer Kontaktstelle oder auch der Block „ganzzahliger Zufallswert“ verwendet werden.

### Der Block „Zahl“

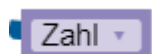


Abbildung 20: Block "Zahl"

Dieser Block dient dazu, den Wert der Variablen im Programm zu nutzen.

## Ein Beispiel für die Verwendung von Variablen



Abbildung 21: Programm "Würfel mit Variablen"

**Aufgabe 13:** Programmiere das Programm „Würfel mit Variablen“ nach und führe es aus.

### Der Block „... = ...“



Abbildung 22: Block "... = ..."

Mit diesem Block können zwei verschiedene Werte miteinander verglichen werden. Ein Beispiel dazu siehst du in Abbildung 23. Hier wird die Variable „Zahl“ mit dem Wert 1 verglichen. Dieser Block kann ebenfalls eine Bedingung für den Block „Wenn ..., mache ...“ darstellen. Außer des Tests auf Gleichheit kann auch auf andere mathematische Eigenschaften getestet werden. Klickst du auf den Pfeil, kannst du zwei Werte auch auf ungleich („≠“), kleiner („<“), kleiner oder gleich („≤“), größer („>“) oder größer oder gleich („≥“) testen.

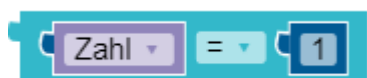


Abbildung 23: Vergleich zweier Werte

**Aufgabe 14:** Programmiere erneut einen Würfel. Durch Drücken der Taste A soll gewürfelt werden. Allerdings soll nicht mehr der Zahlenwert auf der LED-Matrix angezeigt werden, sondern die Anzahl der Augen - wie auf einem richtigen Würfel. Nutze dazu alles, was du bisher über die Programmierung des Calliope gelernt hast.

**Zusatzaufgabe 3:** Überlege dir selbst Programme und teste sie mit dem virtuellen Calliope mini aus. Dazu kannst du alles nutzen, was du bisher über die Programmierung gelernt hast. Ich würde mich freuen, wenn du mir deine Ideen als Bild per Mail zuschickst.

