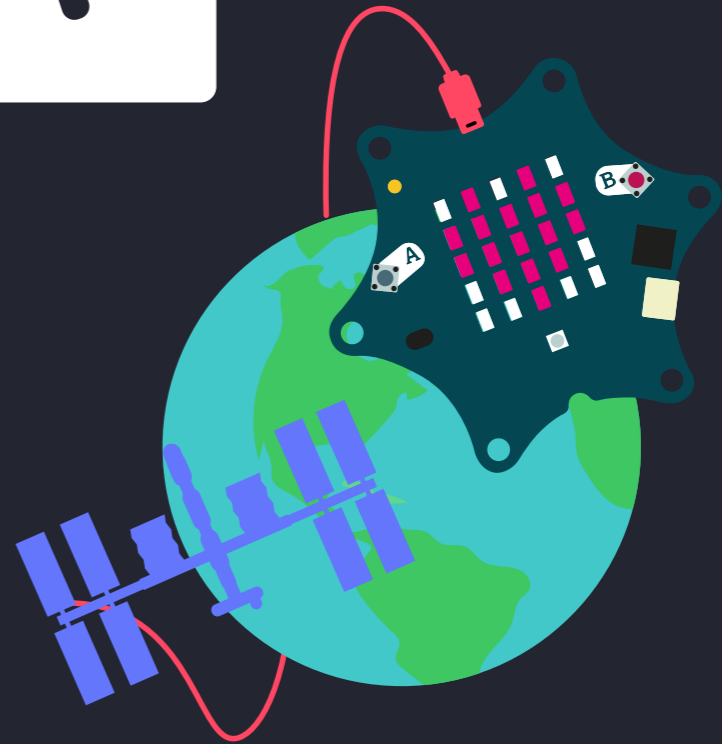


Hallo



Liebe Lehrer:innen, liebe Eltern,

wir wünschen euren Schülern, Schülerinnen und Kindern viel Freude an diesem Mitmachbuch. Es ist so ausgerichtet, dass die Kinder größtenteils eigenständig mit dem Material arbeiten können - ob im Unterricht oder zu Hause.

Um starten zu können, benötigen die Kinder einen **Calliope mini**, einen Computer oder ein Tablet und ihren **Lieblings-Editor**.

Solltet ihr Hilfe benötigen und ihr arbeitet mit Swift, nutzt das Calliope mini-Handbuch für Lehrkräfte. Für andere Editoren könnt ihr die Calliope-Webseite zur Unterstützung heranziehen. Dort gibt es Tipps und Tricks als erste Hilfe und ein Glossar, das die wichtigsten Begriffe aus dem Bereich Coding erklärt.

Weiteres Bastelmaterial, das benötigt wird, ist jeweils aufgelistet.

Viel Spaß beim Eintauchen in die Weiten des Alls...



Alle Informationen
zum Space Special
calliope.cc

DIE CALLIOPE MINI UMGEBUNG



CALLIOPE MINI

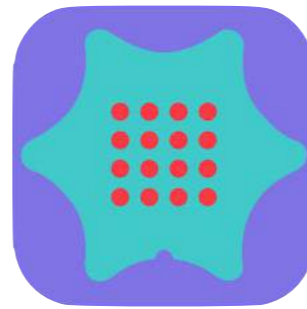
Alles um das Projekt, hilfreiche Tipps, Online-Editoren, Schulmaterial und viele Projekte finden sich auf der Calliope Webseite.

<https://calliope.cc/calliope-mini/uebersicht>



CALLIOPE

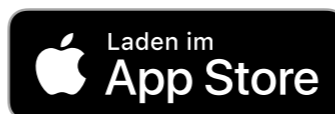
CALLIOPE.CC



CALLIOPE MINI-APP

Die Calliope mini-App kann genutzt werden, um eigene Programme aus den Online-Editoren MakeCode und Open Roberta Lab auf den Calliope mini zu übertragen. Die Geräte werden dabei per Bluetooth verbunden.

<https://calliope.cc/programmieren/mobil>



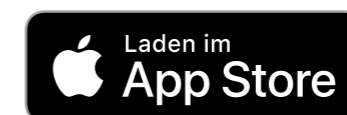
SWIFT PLAYGROUNDS

App: iOS und macOS - Textbasiert

Calliope mini Playground

Abo-URL:

calliope-edu.github.io/playground/feed.json



HANDBUCH

Handbuch für Lehrkräfte

Erste Schritte und Projekte mit Swift



DIE EDITOREN

Programmiert werden kann am Computer, Tablet oder Smartphone.
Hier gibt es eine Übersicht der Editoren, die in diesem Projektbuch unterstützt werden.



MAKECODE

Browser: macOS / Windows
App: iOS / Android
Blockbasiert / Textbasiert
makecode.calliope.cc



OPEN ROBERTA LAB

Browser: macOS / Windows
App: iOS / Android
Blockbasiert / Textbasiert
lab.open-roberta.org



MIND+ (SCRATCH)

App: macOS / Windows
Blockbasiert
mindplus



SWIFT PLAYGROUNDS

App: iOS und macOS
Textbasiert
Abo-URL:
[calliope-edu.github.io/
playground/feed.json](https://calliope-edu.github.io/playground/feed.json)



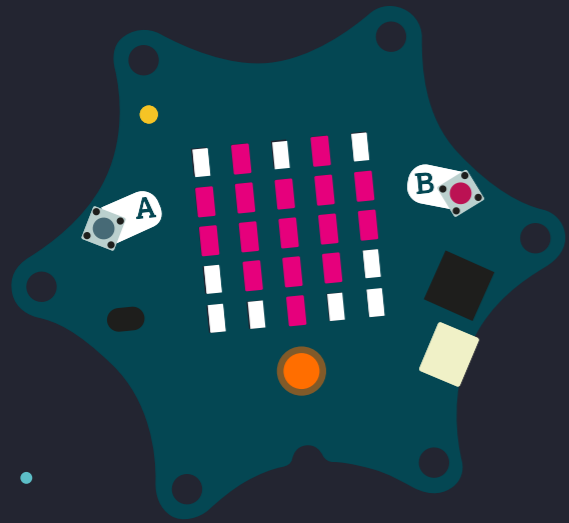
TIGERJYTHON

Browser: macOS / Windows
Textbasiert
tigerjython4kids.ch

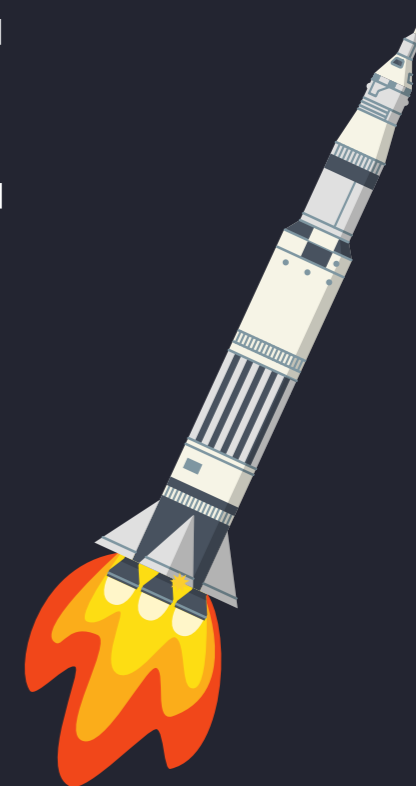


CALLIOPE

[CALLIOPE.CC](https://calliope.cc)



PROJEKTE



PROJEKTE



Raketenstart und
Namen der Planeten

Der große
Ampelstart

Countdown zum
Raketenstart

Weltraum
Taschenlampe

Kommunikation
im All

Licht im Weltraum

Temperaturen im
Weltall

Astronauten
Orakel

Planeten
Quiz



MATERIAL



Neben dem Programmieren lädt der Calliope mini auch zum Basteln von Prototypen und zum individuellen Gestalten von Projekten ein. Dazu eignen sich folgende Materialien:



Calliope mini



Gummiband
(im Calliope mini-
Set enthalten)



Klebeband



Transparentpapier



Schere



Pappe



Stifte



Papier



Acrylkugel



Aluminiumfolie



Aufkleber



Klettband

**CODING-TAGEBUCH
UND WEITERE
MATERIALIEN**





RAKETENSTART UND NAME DER PLANETEN

MATERIAL

Pappe / Tonpapier
Schere
Aluminiumfolie
Klebeband
Deko zum Verzieren

SCHWIERIGKEIT



3... 2... 1... Lift-off!

Der Calliope mini befindet sich an Board einer Rakete und fliegt durch das Weltall. Allerdings hält er nicht an der Internationalen Raumstation, sondern entdeckt die acht Planeten unseres Sonnensystems.

Dazu programmiert ihr einen **animierten Raketenstart** auf der **LED-Matrix** des Calliope mini und hebt ab zu den Sternen. Danach könnt ihr die **Namen** und **Farben** der Planeten anzeigen lassen.

Bastelt euch eine Rakete und befestigt den Calliope mini daran.

CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:



PROGRAMMABLAUFPLAN

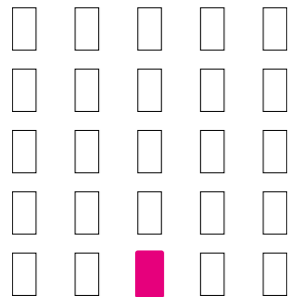
Start

Raketenanimation

1



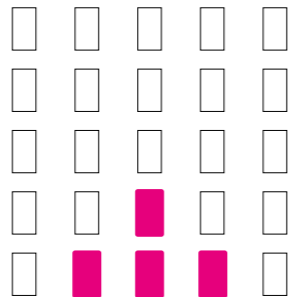
Pause



2



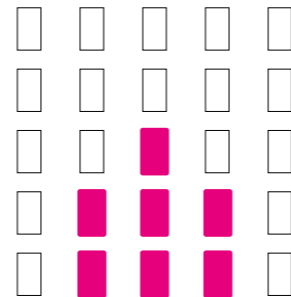
Pause



3



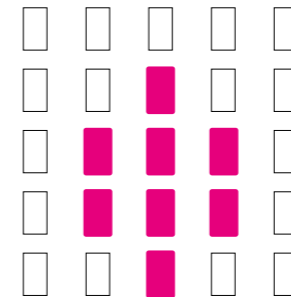
Pause



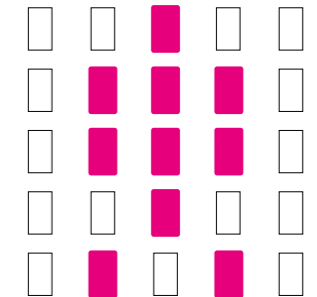
4



Pause



5



Ausgabe: Name des Planeten

„Mars“

Ausgabe: Farbe des Planeten





PROGRAMMIEREN

01 |

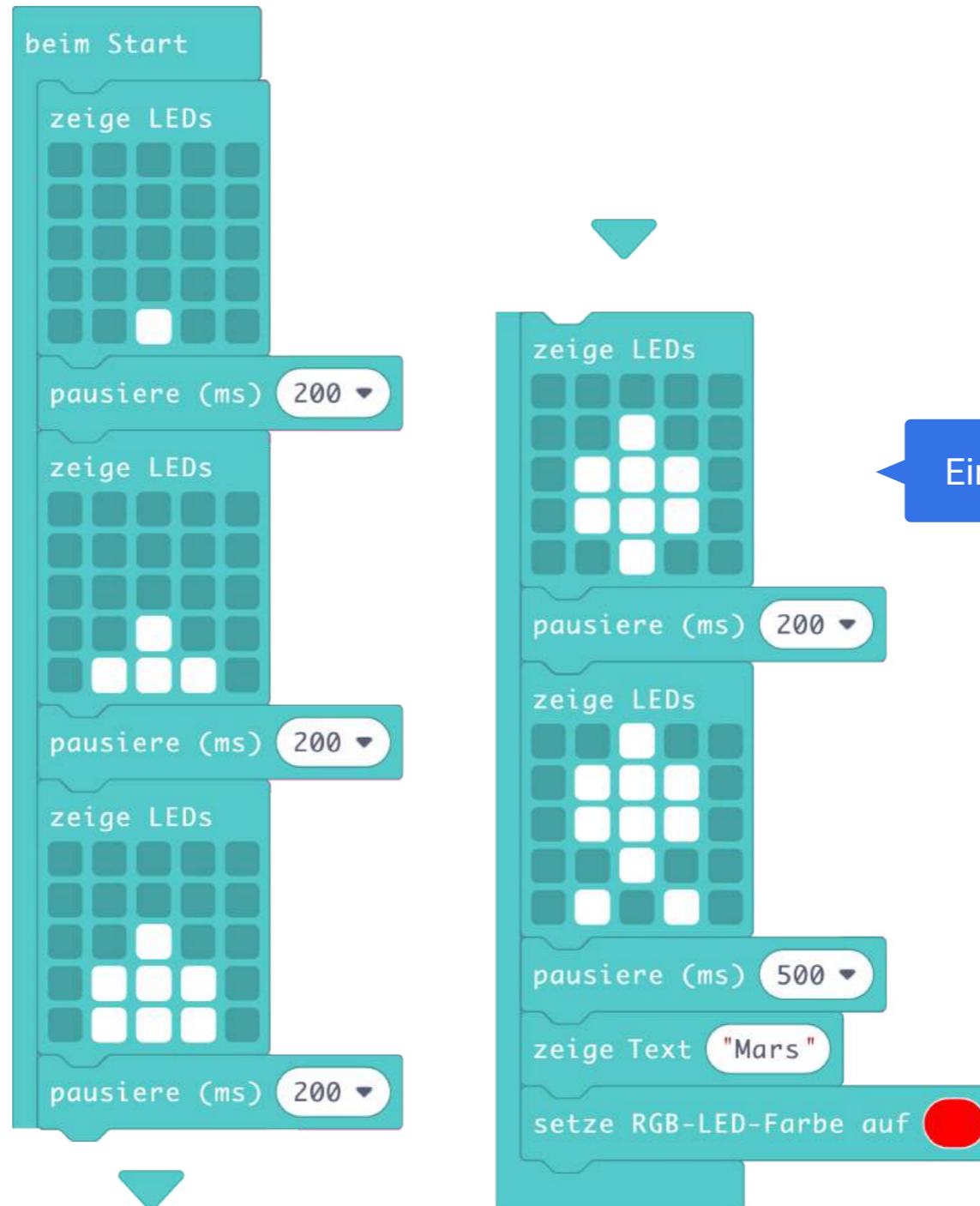
TIPP:

Der *pausiere (ms)* Block ermöglicht euch, die Geschwindigkeit eurer Animation anzupassen. Ändert dafür einfach die Zeit (1000 Millisekunden = 1 Sekunde).

Ihr könnt auch noch weitere Bilder hinzufügen und zum Beispiel die Rakete weiter hinauf fliegen lassen.

PROGRAMM:

mini-Raketenstart.hex



Einzelbilder der Animation

Name und Farbe des Planeten



PROGRAMMIEREN

TIPP:

Der *Warte ms* Block ermöglicht euch, die Geschwindigkeit eurer Animation anzupassen. Ändert dafür einfach die Zeit (1000 Millisekunden = 1 Sekunde).

Ihr könnt auch noch weitere Bilder hinzufügen und zum Beispiel die Rakete weiter hinauf fliegen lassen.

PROGRAMM:

mini-Raketenstart.xml





PROGRAMMIEREN

TIPP:

Der *warte () Sekunden* Block ermöglicht euch, die Geschwindigkeit eurer Animation anzupassen. Ändert dafür einfach die Zeit.

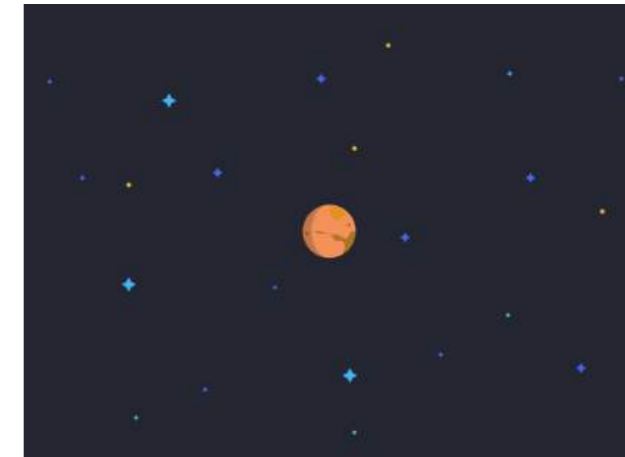
Benutzt im Mind+ Editor die eingebaute Bühne, um eure Programme noch interaktiver zu gestalten.

PROGRAMM:

mini-Raketenstart.sb3

```
Wenn  angeklickt wird
  wechsele zu Bühnenbild Weltraum
  Muster anzeigen 
  warte 0.2 Sekunden
  Muster anzeigen 
  warte 0.2 Sekunden
  Muster anzeigen 
  warte 0.2 Sekunden
  Muster anzeigen 
  warte 0.2 Sekunden
  Muster anzeigen 
  warte 0.5 Sekunden
  wechsele zu Kostüm Mars
  Anzeige Mars
  RGB-LED auf  setzen

Wenn diese Figur angeklickt wird
  sage Mars
```



Bühnenansicht im Mind+ Editor

Einzelbilder der Animation

Name und Farbe des Planeten

Interagiere mit Elementen auf der Bühne



PROGRAMMIEREN

01



TIPP:

Der *mini.sleep* Befehl ermöglicht euch, die Geschwindigkeit eurer Animation anzupassen. Ändert dafür einfach die Zeit (1000 Millisekunden = 1 Sekunde).

Ihr könnt auch noch weitere Bilder hinzufügen und zum Beispiel die Rakete weiter hinauf fliegen lassen.

PLAYGROUND

Ausgabe:

Display, RGB LED

Funktionen:

Start Funktion
Pause

PROGRAMM:

Raketenstart

```
func start() {
    display.show(grid:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])
    mini.sleep(200)
    display.show(grid:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,1,0])
    mini.sleep(200)
    display.show(grid:
[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,1,0,0,1,1,1,0])
    mini.sleep(200)
    display.show(grid:
[0,0,0,0,0,0,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,0,0])
    mini.sleep(200)
    display.show(grid:
[0,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,0,0,0,1,0,1,0])
    mini.sleep(500)
    display.show(text:"Mars")
    rgb.on(color:.red)
}
```

Raketen-
animation

Name und Farbe des Planeten



PROGRAMMIEREN

01 |

TIPP:

Der `sleep()` Befehl ermöglicht euch, die Geschwindigkeit eurer Animation anzupassen. Ändert dafür einfach die Zeit (1000 Millisekunden = 1 Sekunde).

Bilder werden mithilfe von einem Zahlencode programmiert, eine „0“ steht für eine ausgeschaltete LED und eine „9“ lässt die LED auf der Matrix leuchten.

Ihr könnt auch noch weitere Bilder hinzufügen und zum Beispiel die Rakete weiter hinauf fliegen lassen.

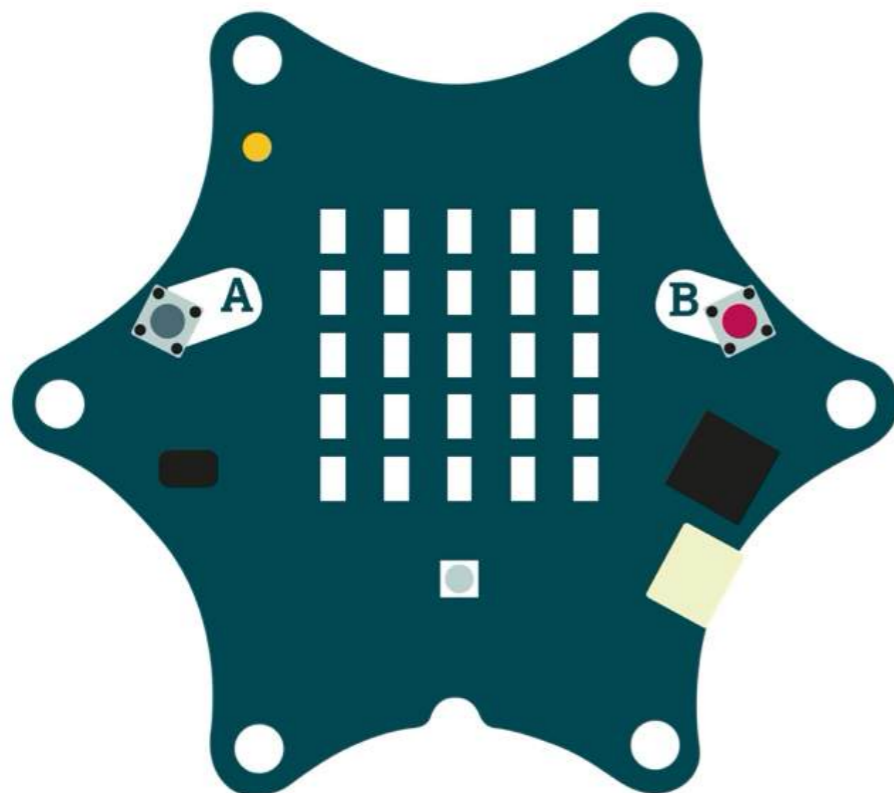
PROGRAMM:

mini-Raketenstart.py

```
from calliope_mini import *  
  
im = Image('00000:00000:00000:00000:00900:')  
display.show(im)  
sleep(200)  
im = Image('00000:00000:00000:00900:09990:')  
display.show(im)  
sleep(200)  
im = Image('00000:00000:00900:09990:09990:')  
display.show(im)  
sleep(200)  
im = Image('00000:00900:09990:09990:00900:')  
display.show(im)  
sleep(200)  
im = Image('00900:09990:09990:00900:09090:')  
display.show(im)  
sleep(500)  
display.scroll("Mars")  
led.set_colors(255, 0, 0)
```

Raketen-
animation

Name und Farbe des Planeten



Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Sieht ihr alle Einzelbilder der Animation?
- Könnt ihr den Namen des Planeten lesen?
- Leuchtet die RGB-LED?
- Habt ihr noch weitere Ideen, die Animation zu erweitern?
- Fallen euch noch weitere Planeten unseres Sonnensystems ein und wisst ihr, welche Farbe sie haben?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Aus Pappe und Aluminiumfolie lässt sich super leicht eine Rakete bauen. Um die Animation auf dem Calliope mini zu sehen, könnt ihr ein kleines Fenster in die Rakete schneiden.

DER GROSSE AMPELSTART

02



MATERIAL

Pappe / Tonpapier
Schere
Klebeband
Transparentpapier
Deko zum Verzieren

In diesem Projekt startet ihr die Rakete mit einer Eingabe: eurer Stimme. Der Calliope mini hat ein Mikrofon verbaut und kann auf bestimmte Lautstärken reagieren. Nachdem der Calliope mini eure Stimme erkennt, startet eine Ampel und die Rakete ist startklar.

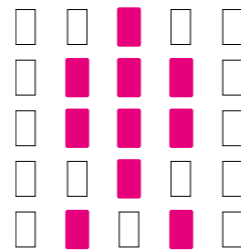
SCHWIERIGKEIT



CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:



Start



Abfrage des Mikrofons

Abgleich der gemessene Lautstärke mit einem Schwellenwert



RGB-LED Ampel

Ausgabe: Note

c'



PROGRAMMIEREN

TIPP:

Das Mikrofon misst die Lautstärke und gibt diese in Werten von 0 bis 255 aus. Mit einem Schwellenwert können die Werte interpretiert und somit die Rakete gestartet werden.

PROGRAMM:

mini-Ampelstart.hex



```
dauerhaft
  zeige LEDs
  wenn Lautstärke > 20 dann
    setze RGB-LED-Farbe auf [Red]
    pausiere (ms) 2000
    setze RGB-LED-Farbe auf [Yellow]
    pausiere (ms) 2000
    setze RGB-LED-Farbe auf [Green]
    spiele Note Mittleres C für 1 Schlag
  eingebaute RGB-LED ausschalten
```

Abfrage des Mikrofons und Abgleich mit einem Schwellenwert

RGB-LED Ampel



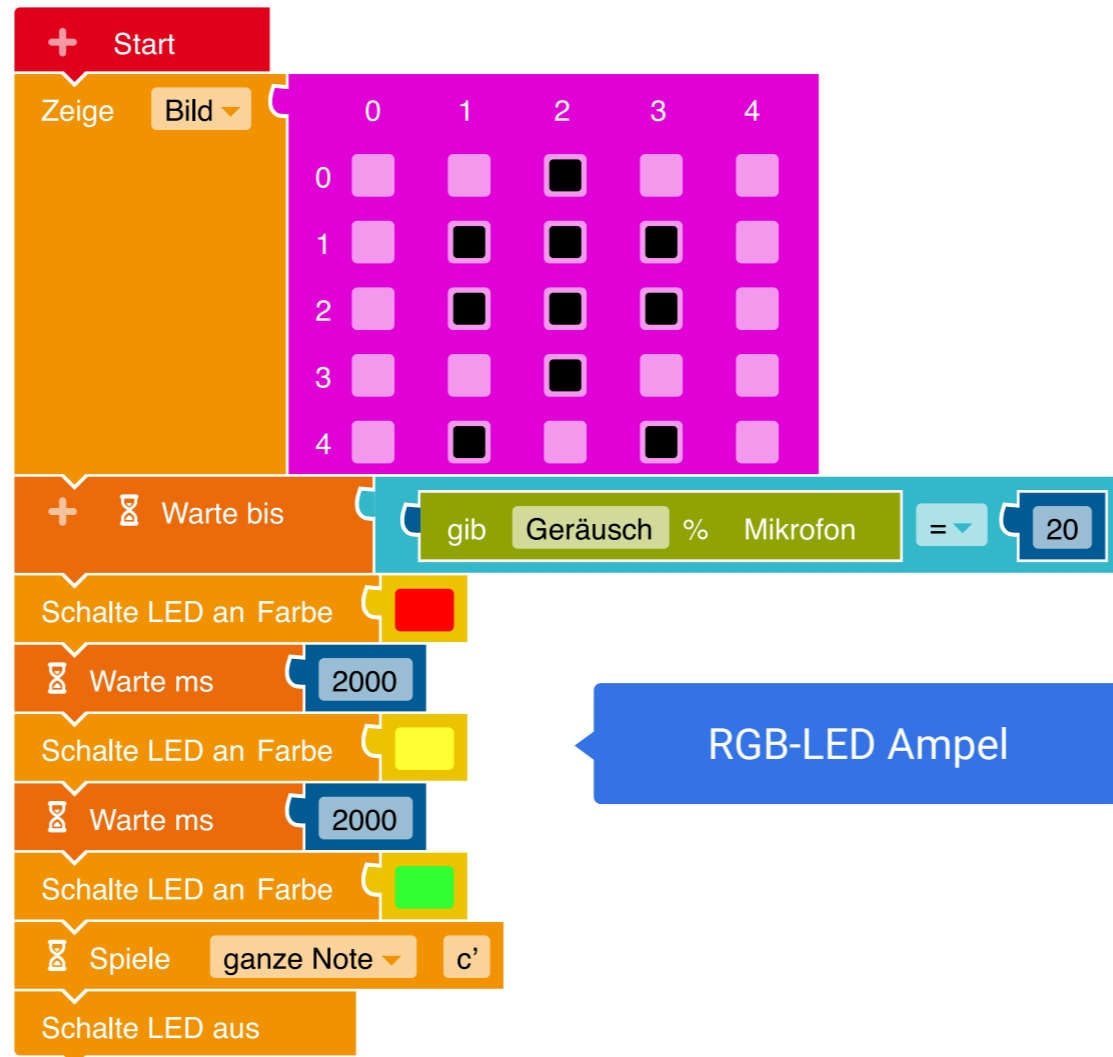
PROGRAMMIEREN

TIPP:

Das Mikrofon misst die Lautstärke und gibt diese in Werten von 0 bis 255 aus. Mit einem Schwellenwert können die Werte interpretiert und somit die Rakete gestartet werden.

PROGRAMM:

mini-Ampelstart.xml



Abfrage des Mikrofons und Abgleich mit einem Schwellenwert

RGB-LED Ampel



PROGRAMMIEREN

02 |

TIPP:

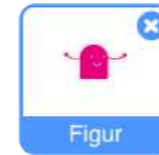
Das Mikrophon misst die Lautstärke und gibt diese in Werten von 0 bis 255 aus. Mit einem Schwellenwert können die Werte interpretiert werden und somit die Rakete gestartet werden.

Die Ampel Figur hat drei verschiedene Kostüme und wechselt anhand der „Ampelstart“ Variable die jeweilige Farbe.

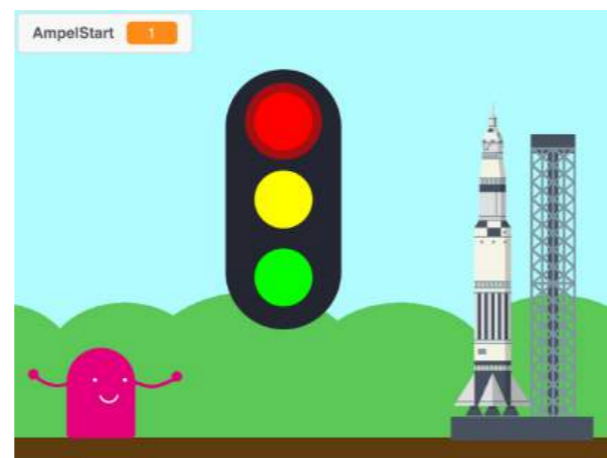
PROGRAMM:

mini-Ampelstart.sb3

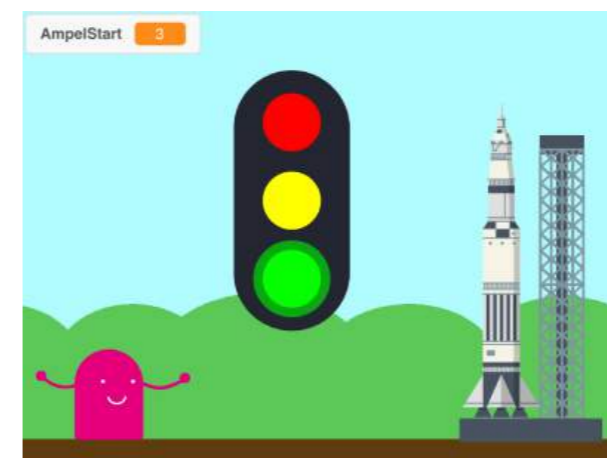
```
Wenn angeklickt wird
setze AmpelStart auf 1
Muster anzeigen
warte bis Lautstärke lesen > 40
RGB-LED auf rot setzen
warte 2 Sekunden
setze AmpelStart auf 2
RGB-LED auf gelb setzen
warte 2 Sekunden
setze AmpelStart auf 3
RGB-LED auf grün setzen
Eingebauter Buzzer Lautsprecher Ton abspielen Mittleres C/C4 für 1 Takt
RGB-LED ausschalten
```



```
Wenn angeklickt wird
wiederhole fortlaufend
falls AmpelStart = 1, dann
wechsle zu Kostüm Ampel_rot
falls AmpelStart = 2, dann
wechsle zu Kostüm Ampel_gelb
falls AmpelStart = 3, dann
wechsle zu Kostüm Ampel_gruen
```



Bühnenansicht im Mind+ Editor | Kostüm: Ampel_rot



Bühnenansicht im Mind+ Editor | Kostüm: Ampel_gruen



TIPP:

Das Mikrofon misst die Lautstärke und gibt diese in Werten von 0 bis 255 aus. Mit einem Schwellenwert können die Werte interpretiert werden und somit die Rakete gestartet werden.

PLAYGROUND

Ausgabe:

Display, RGB LED, Lautsprecher

Funktionen:

Eigene Funktion
Variable
Pause
Lautstärke (noise)

PROGRAMM:

Ampelstart

```
func AmpelStart(){
    let start:UInt16 = 20

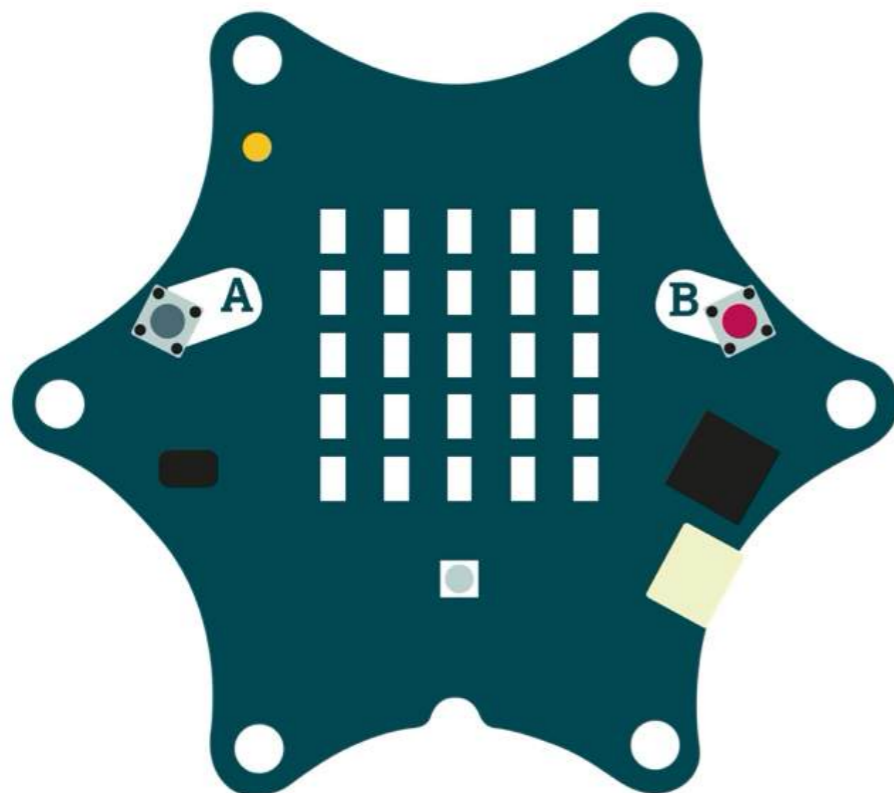
    while io.noise < start {
        mini.sleep(10)
    }

    while io.noise >= start {
        mini.sleep(10)
    }
}

func start(){
    display.show(grid:
[0,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,0,0,0,1,0,1,0])
    AmpelStart()
    rgb.on(color: .red)
    mini.sleep(2000)
    rgb.on(color: .yellow)
    mini.sleep(2000)
    rgb.on(color: .green)
    sound.on(note: .c)
    mini.sleep(400)
    sound.off()
    rgb.off()
}
```

AmpelStart Funktion:
Definieren der **start** Variable
und des Schwellenwertes

RGB-LED Ampel

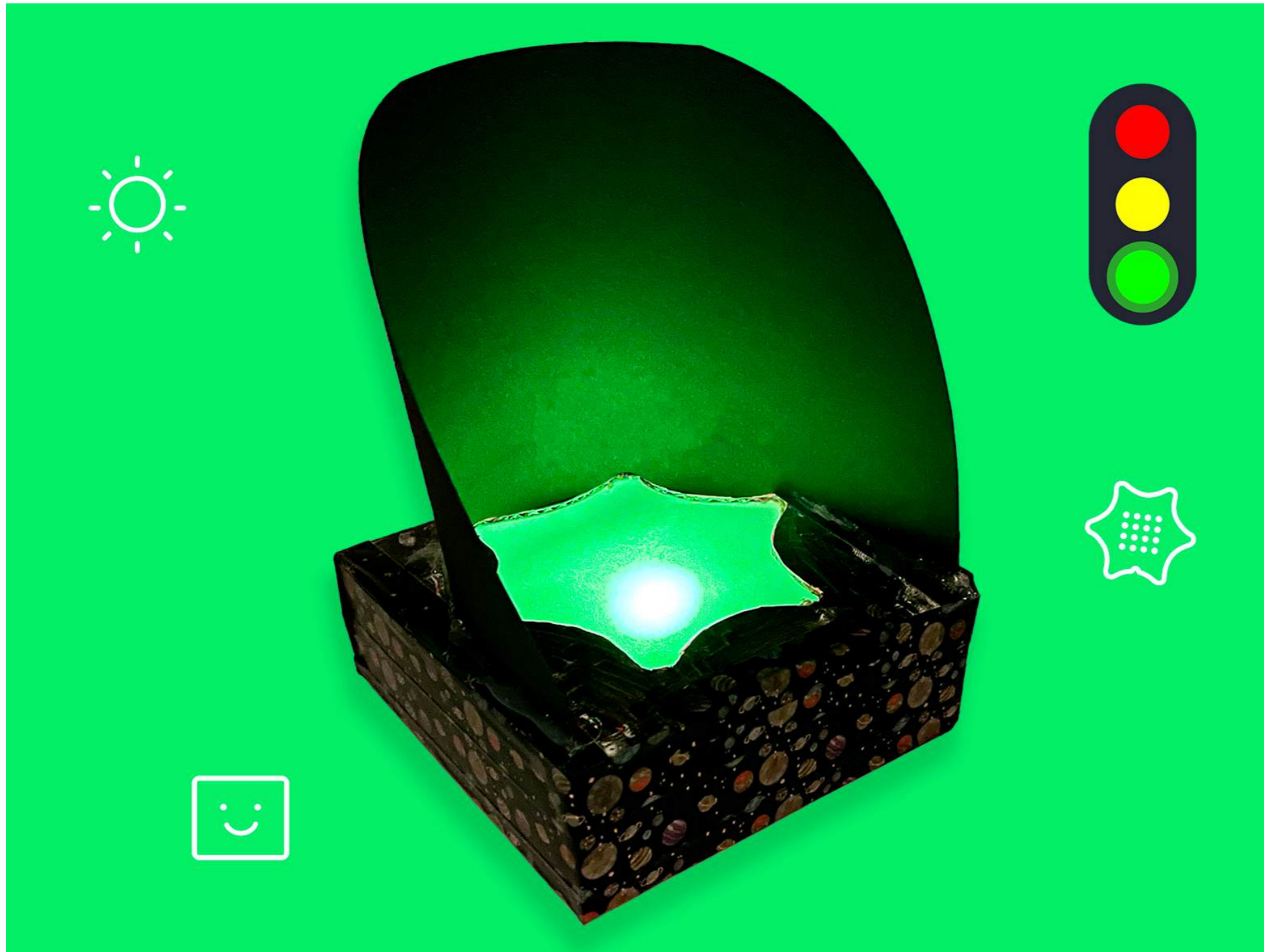


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert das Mikrofon vom Calliope mini?
- Ist der Schwellenwert gut gewählt?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Die Calliope mini Box ist optimal für kleine Bastelprojekte.

Mit dunklem Tonpapier wird die Box blitzschnell zu einer kleinen Calliope mini Ampel. Außerdem könnt ihr mit Transparentpapier das Licht der RGB-LED streuen und tolle Leuchteffekte erzielen.

COUNTDOWN

03



MATERIAL

Pappe / Tonpapier
Schere
Kleber
Gummiband
Deko zum Verzieren

SCHWIERIGKEIT

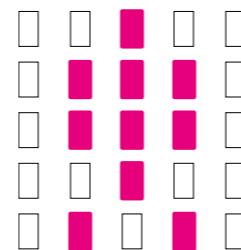
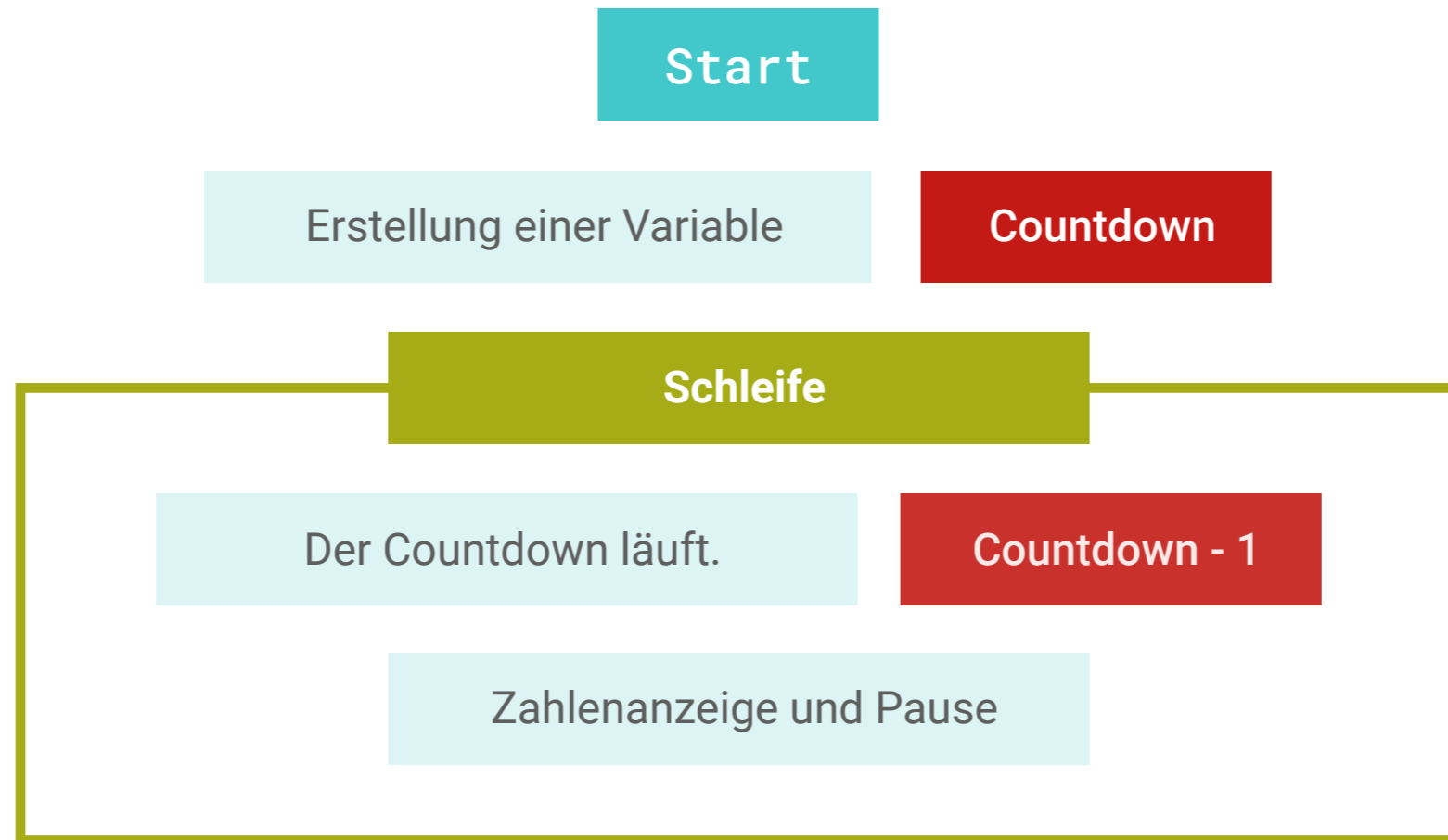


Einer der spannendsten und gefährlichsten Momenten in der Raumfahrt ist der Start der Raketen. Die Rakete steht bereit, der Countdown läuft und die Triebwerke laufen heiß.

Im ersten Kapitel habt ihr bereits eine Animation zum Raketenstart programmiert, nun beschäftigen wir uns mit dem Countdown. Der Calliope mini zählt von 10 herunter und startet zum Schluss die Rakete.

CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:





Ausgabe: RGB-LED



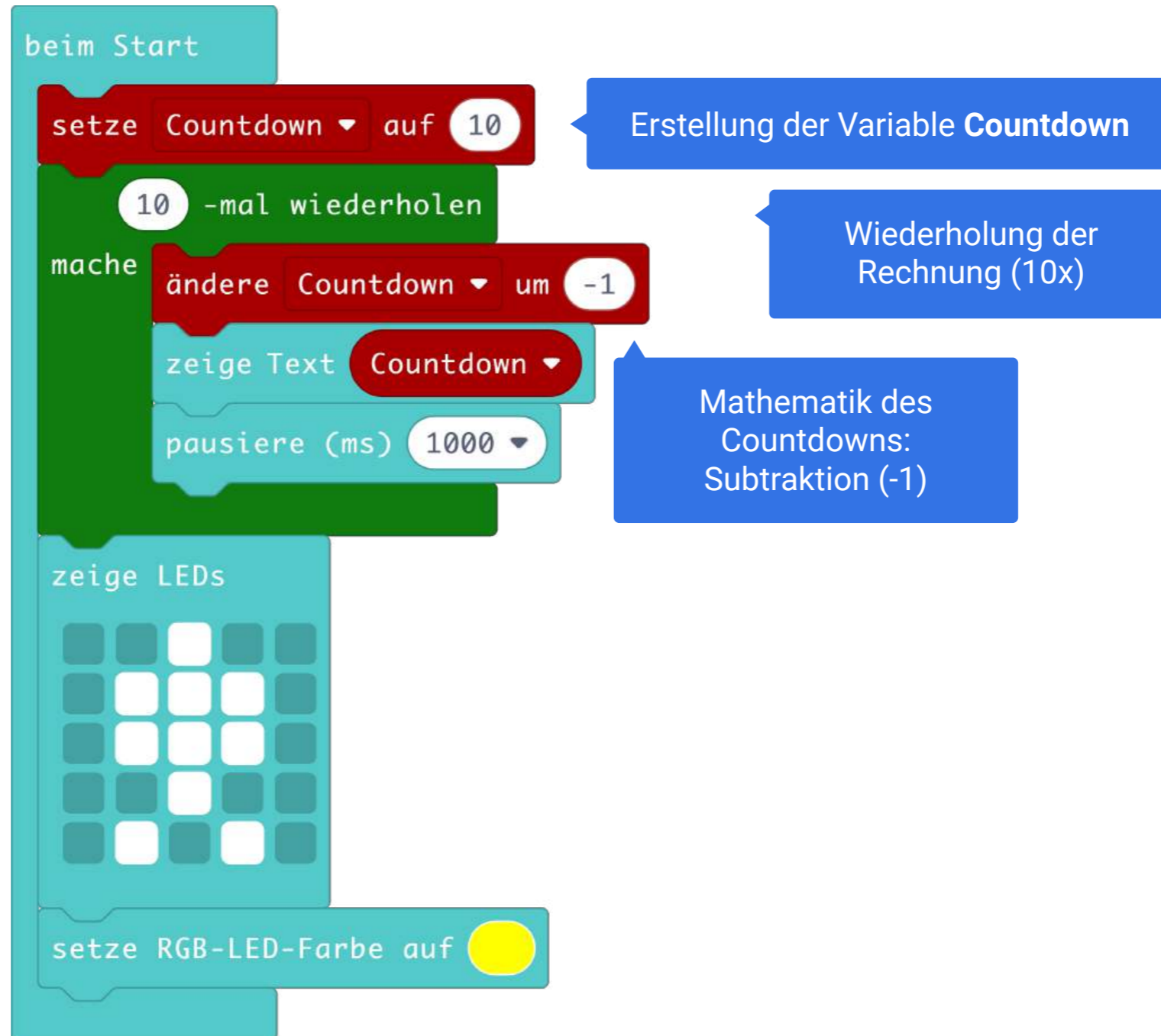


TIPP:

Schleifen ermöglichen eine Wiederholung von Anweisungen und können eure Programme vereinfachen. Manche Schleifen laufen für immer - andere für eine bestimmte Anzahl von Wiederholungen.

PROGRAMM:

mini-Countdown.hex



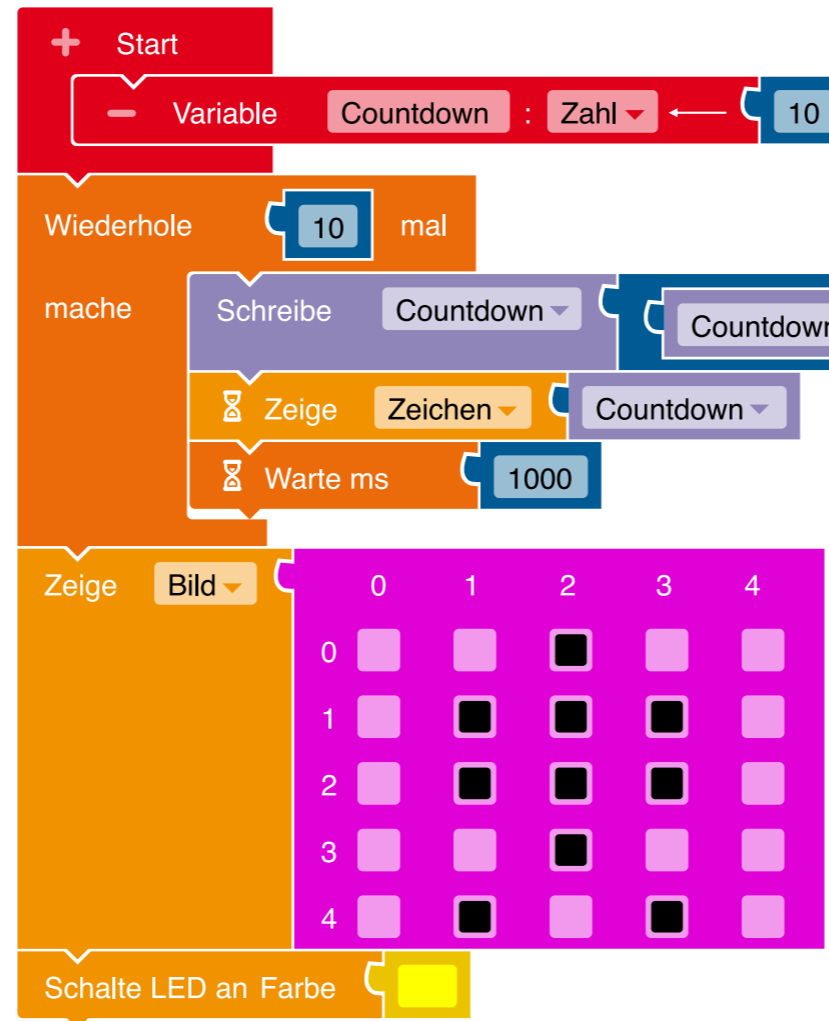


TIPP:

Schleifen ermöglichen eine Wiederholung von Anweisungen und können eure Programme vereinfachen. Manche Schleifen laufen für immer - andere für eine bestimmte Anzahl von Wiederholungen.

PROGRAMM:

mini-Countdown.xml



Erstellung der Variable Countdown

Wiederholung der Rechnung (10x)

Mathematik des Countdowns: Subtraktion (-1)



PROGRAMMIEREN

03



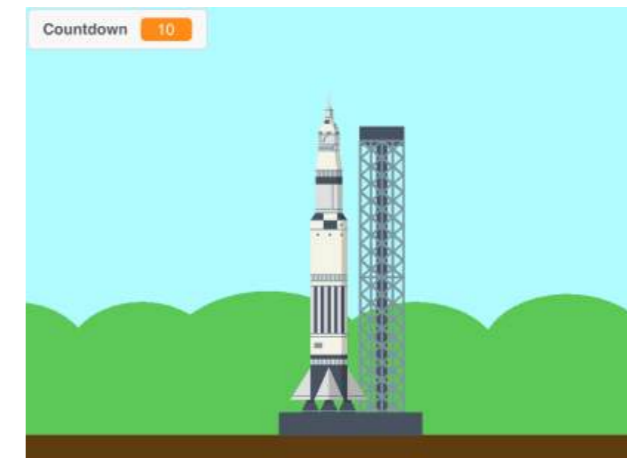
TIPP:

Schleifen ermöglichen eine Wiederholung von Anweisungen und können eure Programme vereinfachen. Manche Schleifen laufen für immer - andere für eine bestimmte Anzahl von Wiederholungen.

PROGRAMM:

mini-Countdown.sb3

```
Wenn  angeklickt wird
  wechsele zu Kostüm Rakete
  setze x auf 0
  setze y auf -54
  setze Countdown auf 10
  wiederhole 10 mal
    ändere Countdown um -1
    Anzeige Countdown
    warte 1 Sekunden
  Muster anzeigen
  RGB-LED auf  setzen
  wechsele zu Kostüm Rakete_Flammen
  warte 0.4 Sekunden
  gleite in 1 Sek. zu x: 0 y: 249
```



Bühnenansicht im Mind+ Editor | Kostüm: Rakete



Bühnenansicht im Mind+ Editor | Kostüm: Rakete_Flammen



TIPP:

Schleifen ermöglichen eine Wiederholung von Anweisungen und können eure Programme vereinfachen. Manche Schleifen laufen für immer - andere für eine bestimmte Anzahl von Wiederholungen.

PLAYGROUND

Ausgabe:

Display, RGB LED

Befehle:

Start Funktion
Pause

PROGRAMM:

Countdown

```
func start (){  
    for i in 1 ... 10 {  
        display.show(number: 10-UInt16(i))  
        mini.sleep(2000)  
    }  
  
    display.show(grid:  
[0,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,0,0,0,1,0,1,0])  
    rgb.on(color: .yellow)  
}
```

Schleife mit 10 Wiederholungen (*i*) und Ausgabe der Wiederholung (*10 - i*)

Raketenstart



PROGRAMMIEREN

03 |

TIPP:

Schleifen ermöglichen eine Wiederholung von Anweisungen und können eure Programme vereinfachen. Manche Schleifen laufen für immer - andere für eine bestimmte Anzahl von Wiederholungen.

PROGRAMM:

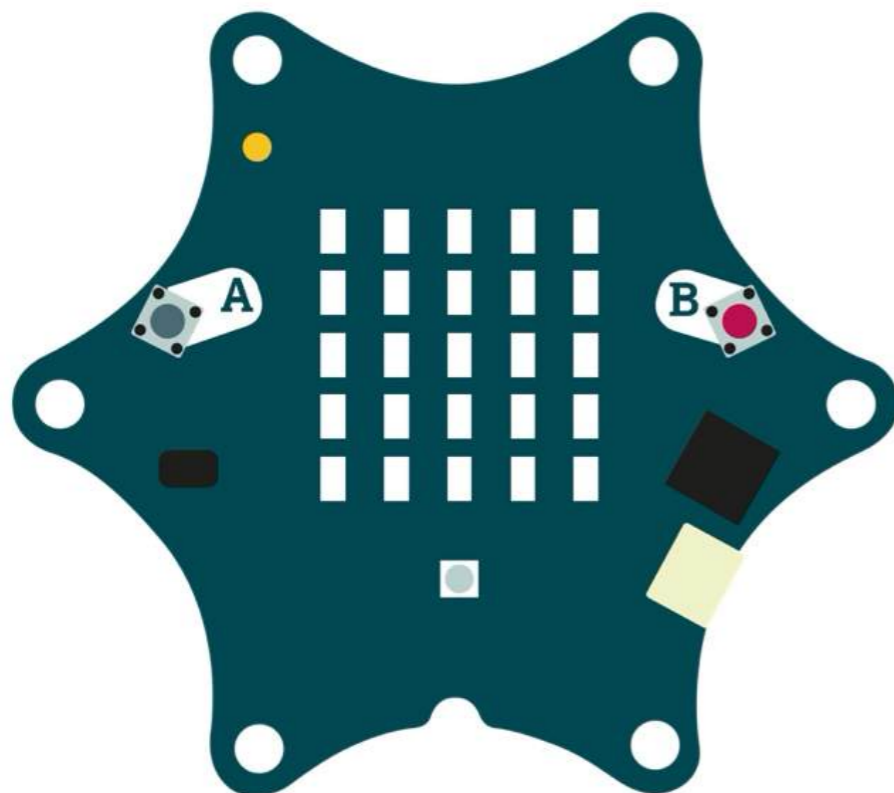
mini-Countdown.py

```
from calliope_mini import *  
  
repeat 1:  
    for x in range(1, 10):  
        display.show(str(10-x))  
        sleep(1000)
```

Schleife mit 10 Wiederholungen (**i**) und Ausgabe der Wiederholung (**10 - i**)

```
im = Image('00900:09990:09990:00900:09090:')  
display.show(im)  
led.set_colors(255, 255, 0)
```

Raketenstart

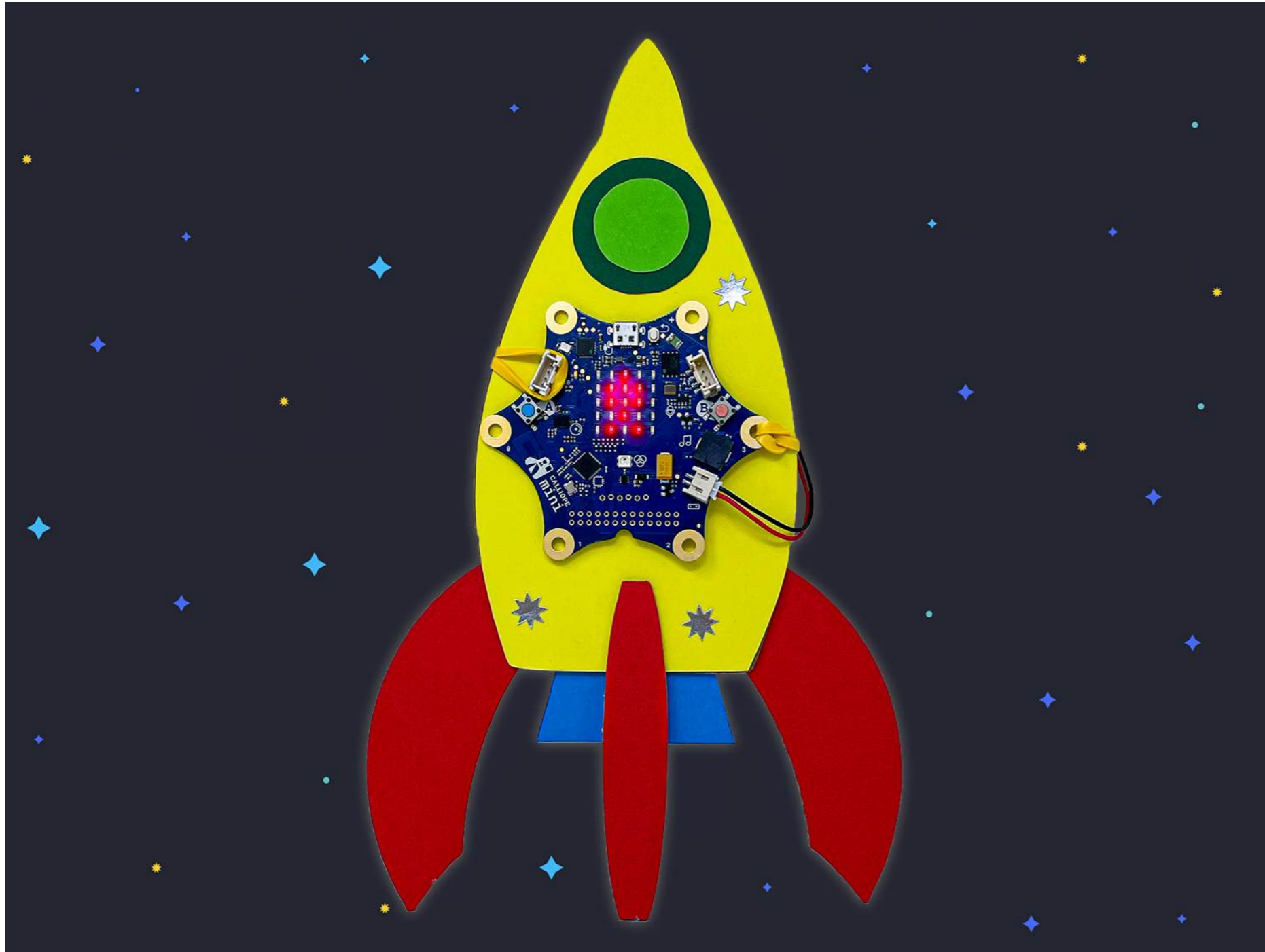


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert der Calliope mini auf eure Eingaben?
- Sind die Bilder lange genug sichtbar?
- Fallen dir noch andere Wege der Kommunikation ohne Sprache ein?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Eine weitere Calliope mini Rakete - diesmal fliegt der Calliope mini nicht in der Rakete mit, sondern wird an der Außenseite befestigt.

WELTRAUM TASCHENLAMPE

04



MATERIAL

Pappe / Tonpapier

Schere

Kleber

Aluminiumfolie

Gummiband

SCHWIERIGKEIT

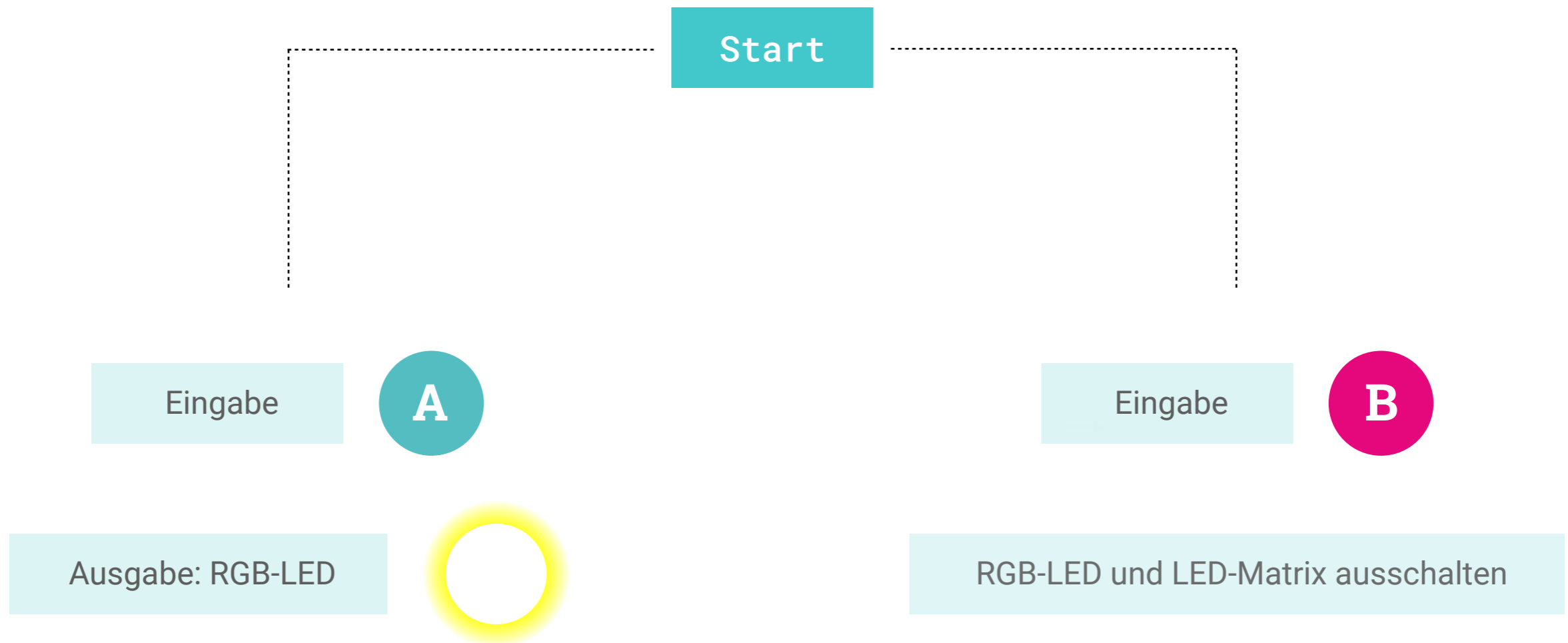


Im Weltraum kann es sehr dunkel sein! Damit die Astronaut:innen trotzdem bei ihrer Arbeit etwas sehen können, benötigen sie Lampen und Lichter. Die RGB-LED des Calliope mini kann sehr hell leuchten und super zu einer Taschenlampe programmiert werden.

Bringt auf Knopfdruck Licht in die Dunkelheit - gerne auch in vielen verschiedenen Farben.

CODING TAGEBUCH
FÜR EUER ZUM
HERUNTERLADEN:







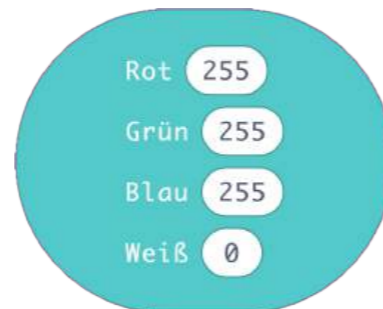
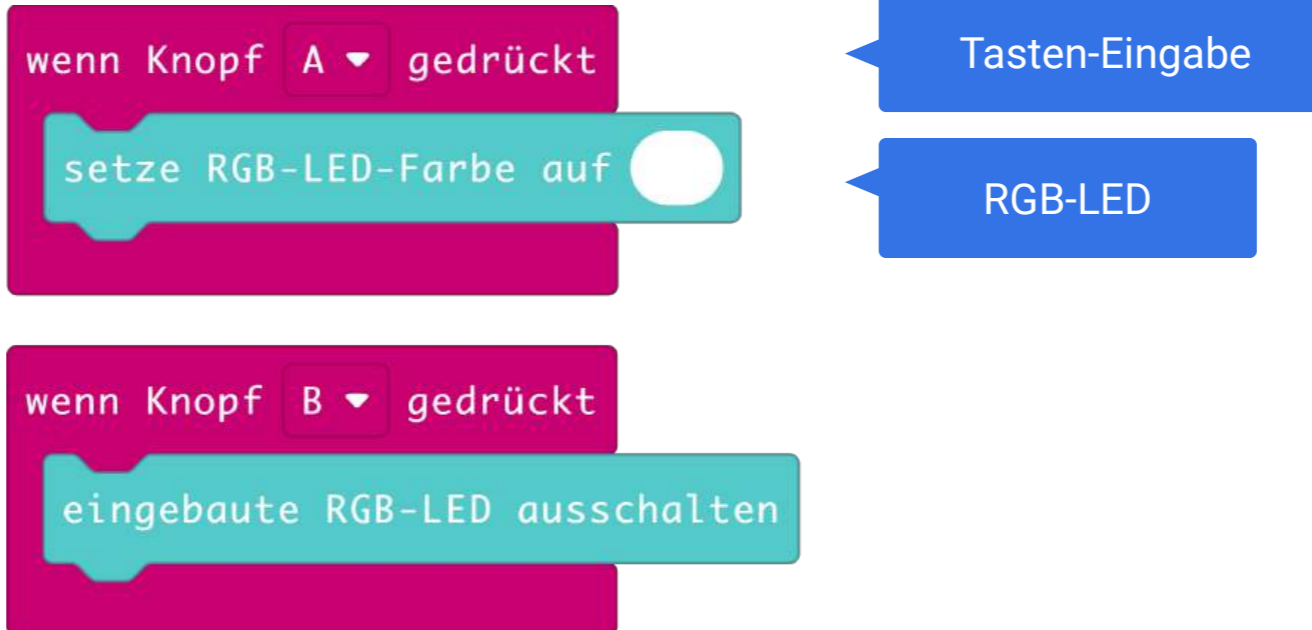
PROGRAMMIEREN

TIPP:

Die RGB-LED des Calliope mini mischt die Farben rot, grün und blau zusammen und kann so in ganz vielen verschiedenen Farben leuchten. Es gibt bereits eine Reihe vorgefertigter Farben, allerdings könnt ihr mit dem RGB-Farblock auch eigene Farben erzeugen.

PROGRAMM:

mini-Taschenlampe.hex



Mischt eigene Farben mit dem RGB-Farblock aus der Grundlagen-Kategorie.



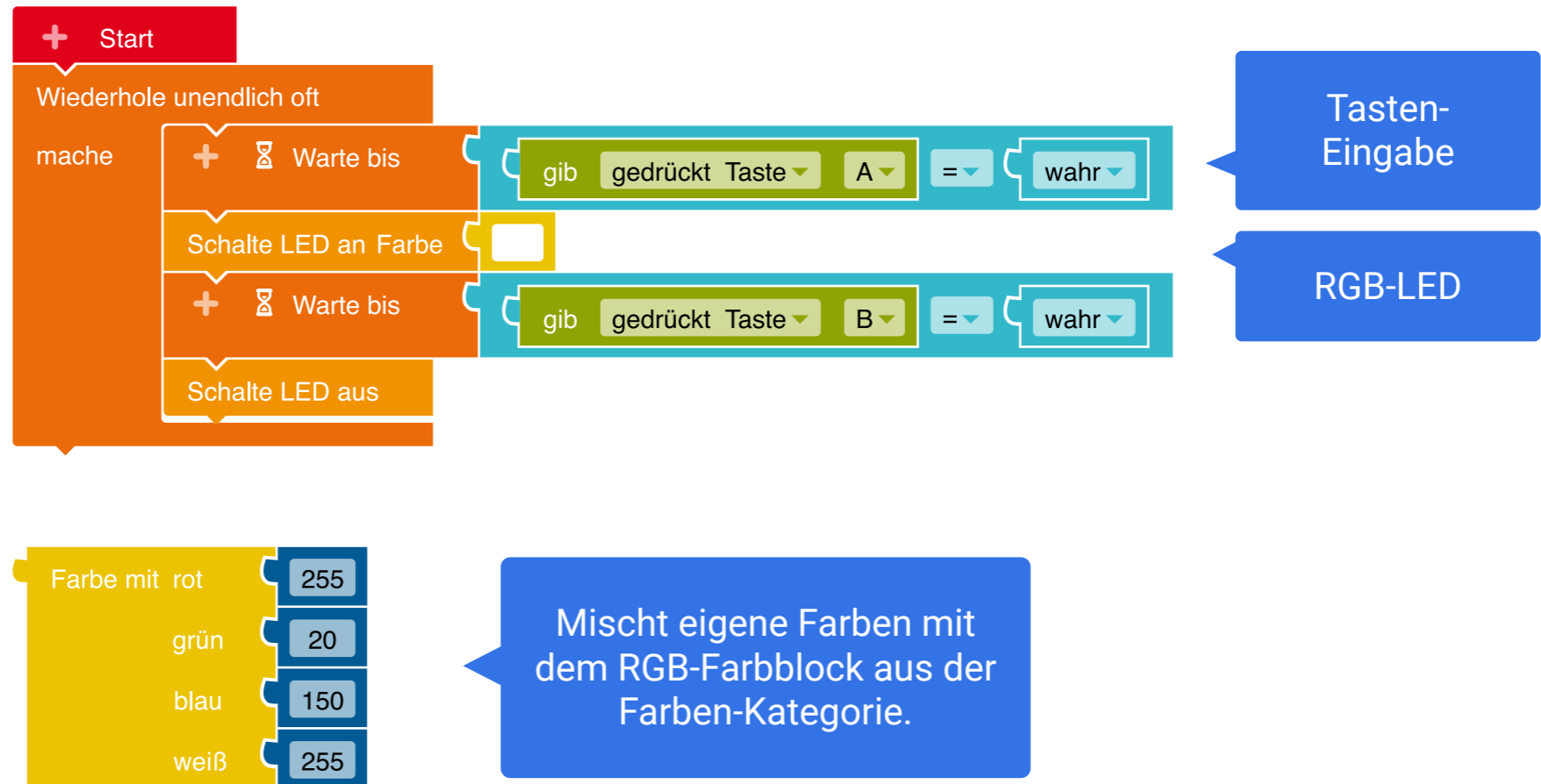
PROGRAMMIEREN

TIPP:

Die RGB-LED des Calliope mini mischt die Farben rot, grün und blau zusammen und kann so in ganz vielen verschiedenen Farben leuchten. Es gibt bereits eine Reihe vorgefertigter Farben, allerdings könnt ihr mit dem RGB-Farblock auch eigene Farben erzeugen.

PROGRAMM:

mini-Taschenlampe.xml





PROGRAMMIEREN

TIPP:

Die RGB-LED des Calliope mini mischt die Farben rot, grün und blau zusammen und kann so in ganz vielen verschiedenen Farben leuchten. Es gibt bereits eine Reihe vorgefertigter Farben, allerdings könnt ihr mit dem RGB-Farblock auch eigene Farben erzeugen.

PROGRAMM:

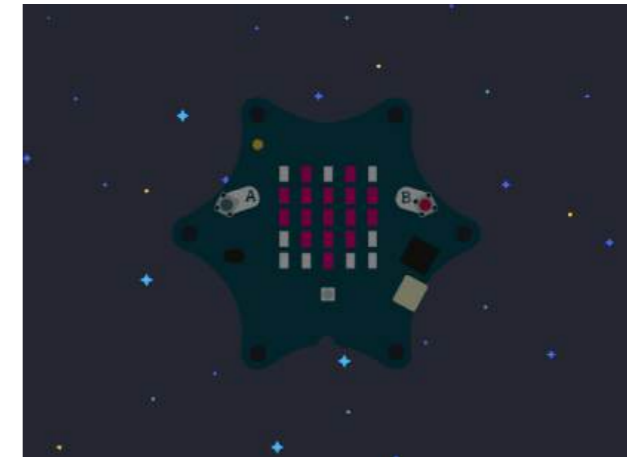
mini-Taschenlampe.sb3

```
wenn Taste A gedrückt
  RGB-LED auf [ ] setzen
  wechsele zu Kostüm Licht_an
```

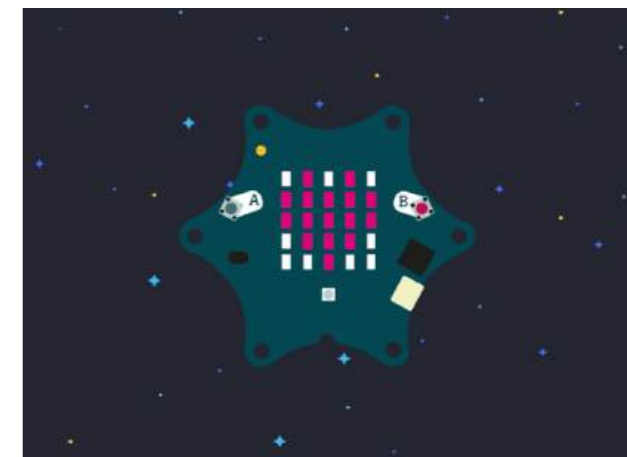
```
wenn Taste B gedrückt
  RGB-LED ausschalten
  wechsele zu Kostüm Licht_aus
```

```
rot 255 grün 255 blau 255 weiß 255
```

Mischt eigene Farben mit dem RGB-Farblock aus der Grundlagen-Kategorie.



Bühnenansicht im Mind+ Editor | Kostüm: Licht_aus



Bühnenansicht im Mind+ Editor | Kostüm: Licht_an



PROGRAMMIEREN

04 | 

TIPP:

Die RGB-LED des Calliope mini mischt die Farben rot, grün und blau zusammen und kann so in ganz vielen verschiedenen Farben leuchten. Es gibt bereits eine Reihe vorgefertigter Farben, allerdings könnt ihr mit dem RGB-Befehl auch eigene Farben erzeugen.

PLAYGROUND

Ausgabe:

RGB LED

Eingabe:

Button A, Button B

PROGRAMM:

Taschenlampe

```
func onButtonA() {  
    rgb.on(color: .white)  
}
```

Tasten-Eingabe

```
func onButtonB() {  
    rgb.off()  
}
```

RGB-LED

```
rgb.on(r: 255, g: 255, b: 0)
```

Mischt eigene Farben mit dem RGB-Befehl.



PROGRAMMIEREN

TIPP:

Die RGB-LED des Calliope mini mischt die Farben rot, grün und blau zusammen und kann so in ganz vielen verschiedenen Farben leuchten. Es gibt bereits eine Reihe vorgefertigter Farben, allerdings könnt ihr mit dem RGB-Farblock auch eigene Farben erzeugen.

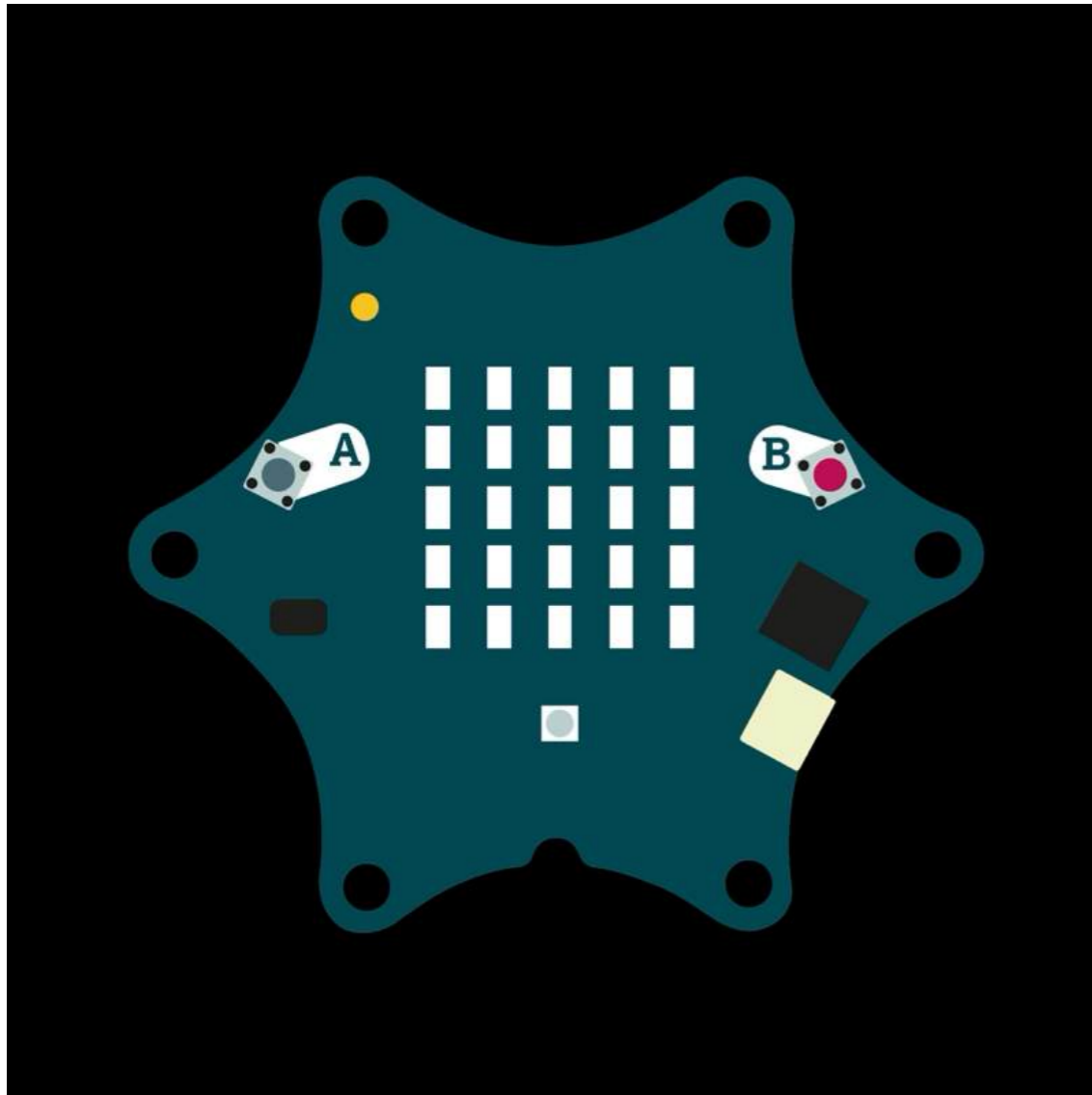
PROGRAMM:

mini-Taschenlampe.py

```
from calliope_mini import *  
  
while True:  
    if button_a.was_pressed():  
        led.set_colors(255, 255, 255)  
  
    if button_b.was_pressed():  
        led.set_colors(0, 0, 0)
```

Tasten-Eingabe

RGB-LED

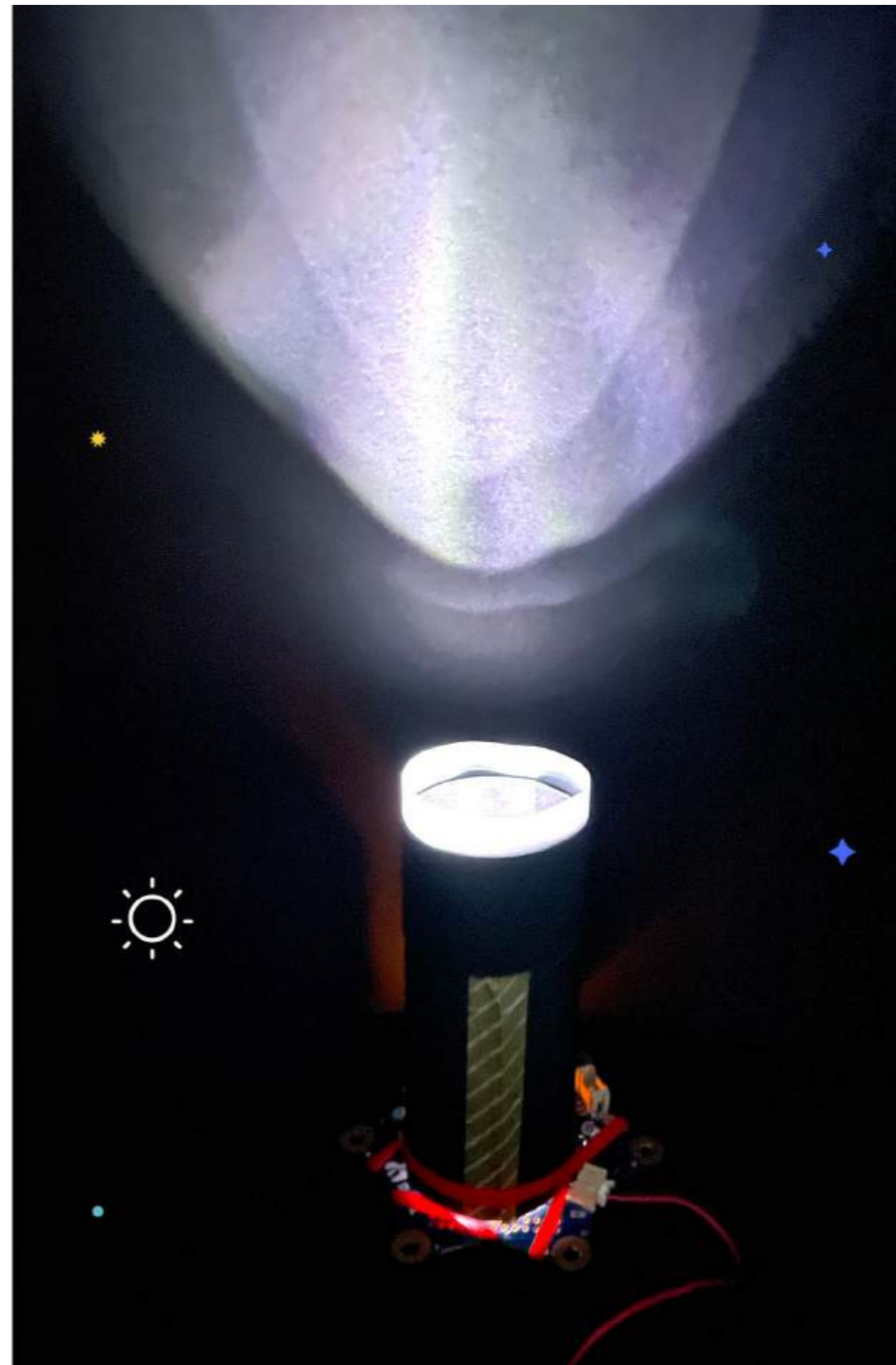


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert der Calliope mini auf eure Eingabe?
- Leuchtet die RGB-LED?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Die RGB-LED des Calliope mini kann super hell leuchten. Mit einem Zylinder aus Tonpapier wird der Lichtstrahl gebündelt und der Calliope mini wird zu einer richtigen Taschenlampe.

KOMMUNIKATION IM ALL

05



MATERIAL

Pappe / Tonpapier

Schere

Kleber

Stifte

Deko zum Verzieren

SCHWIERIGKEIT



Auf der Internationalen Raumstation befinden sich viele Astronauten:innen unterschiedlicher Länder, weswegen nicht alle die gleiche Sprache sprechen.

Sprache ist allerdings nicht der einzige Weg der Kommunikation. Mit dem Calliope mini könnt ihr kleine Bilder und verschiedene Farben anzeigen und so zum Beispiel ausdrücken, wie es euch geht oder was ihr gerade macht.

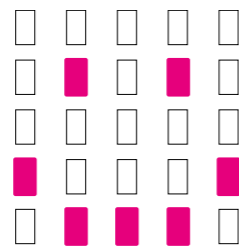
**CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:**



Start

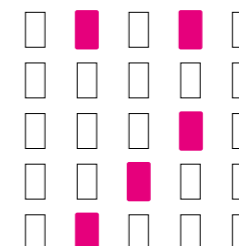
Eingabe

A



Eingabe

B



Ausgabe: RGB-LED



RGB-LED und LED-Matrix ausschalten

Ausgabe: RGB-LED



RGB-LED und LED-Matrix ausschalten



PROGRAMMIEREN

TIPP:

Benutzt bei der Kommunikation ohne Sprache einfache und leicht verständliche Symbole.

PROGRAMM:

mini-Kommunikation.hex



wenn Knopf A gedrückt

zeige Symbol

setze RGB-LED-Farbe auf

pausiere (ms) 1000

Bildschirminhalt löschen

eingebaute RGB-LED ausschalten

Tasten-Eingabe

wenn Knopf B gedrückt

zeige Symbol

setze RGB-LED-Farbe auf

pausiere (ms) 1000

Bildschirminhalt löschen

eingebaute RGB-LED ausschalten

Kommunikation durch Bilder und Farben

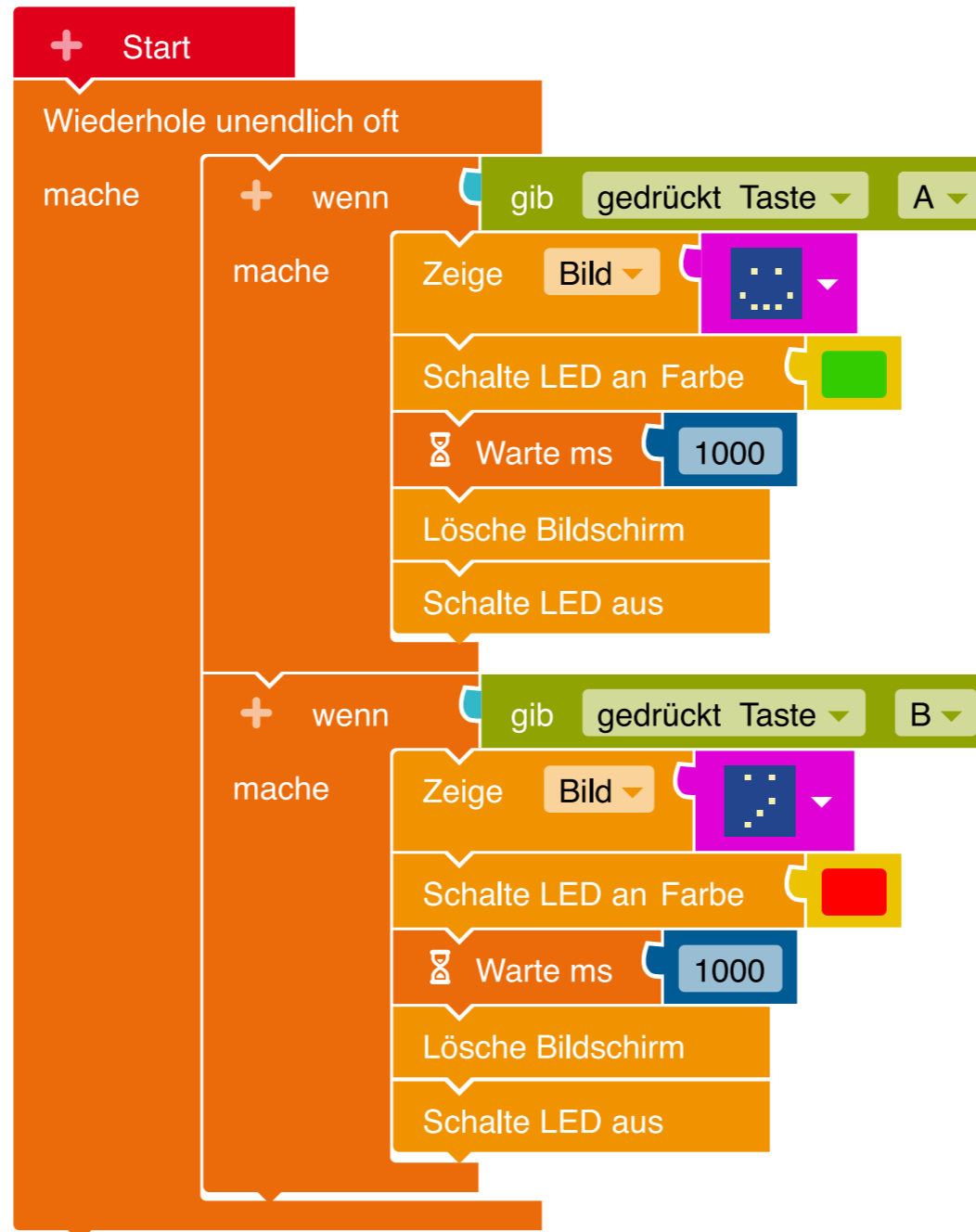


TIPP:

Benutzt bei der Kommunikation ohne Sprache einfache und leicht verständliche Symbole.

PROGRAMM:

mini-Kommunikation.xml



Tasten-Eingabe

Kommunikation durch Bilder und Farben



PROGRAMMIEREN

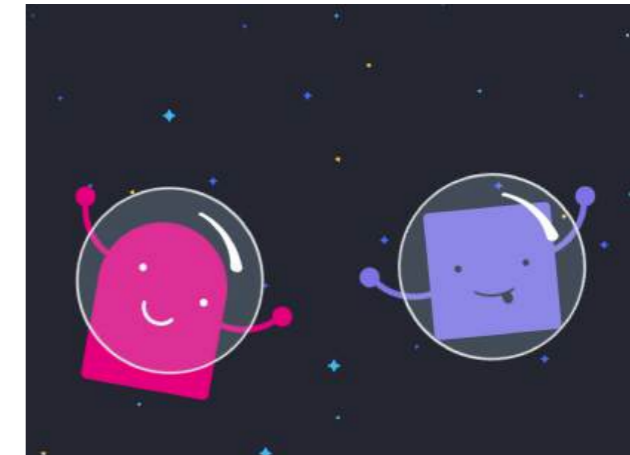
TIPP:

Benutzt bei der Kommunikation ohne Sprache einfache und leicht verständliche Symbole.

PROGRAMM:

mini-Kommunikation.sb3

```
wenn Taste A gedrückt
Muster anzeigen
RGB-LED auf Grün setzen
sage 😊 für 2 Sekunden
warte 1 Sekunden
Bildschirm löschen
RGB-LED ausschalten
```



Bühnenansicht im Mind+ Editor

```
wenn Taste B gedrückt
Muster anzeigen
RGB-LED auf Rot setzen
sage 😞 für 2 Sekunden
warte 1 Sekunden
Bildschirm löschen
RGB-LED ausschalten
```





PROGRAMMIEREN

05 |

TIPP:

Benutzt bei der Kommunikation ohne Sprache einfache und leicht verständliche Symbole.

PLAYGROUND

Ausgabe:

Display, RGB LED

Eingabe:

Button A, Button B

Befehle:

Pause
Löschen

PROGRAMM:

Kommunikation

```
func onButtonA() {  
    display.show(image: .smiley)  
    rgb.on(color: .green)  
    mini.sleep(1000)  
    display.clear()  
    rgb.off()  
}
```

Tasten-Eingabe

```
func onButtonB() {  
    display.show(image: .sad)  
    rgb.on(color: .red)  
    mini.sleep(1000)  
    display.clear()  
    rgb.off()  
}
```

Kommunikation
durch Bilder und
Farben



PROGRAMMIEREN

05 | 

TIPP:

Benutzt bei der Kommunikation ohne Sprache einfache und leicht verständliche Symbole.

PROGRAMM:

mini-Kommunikation.py

```
from calliope_mini import *
```

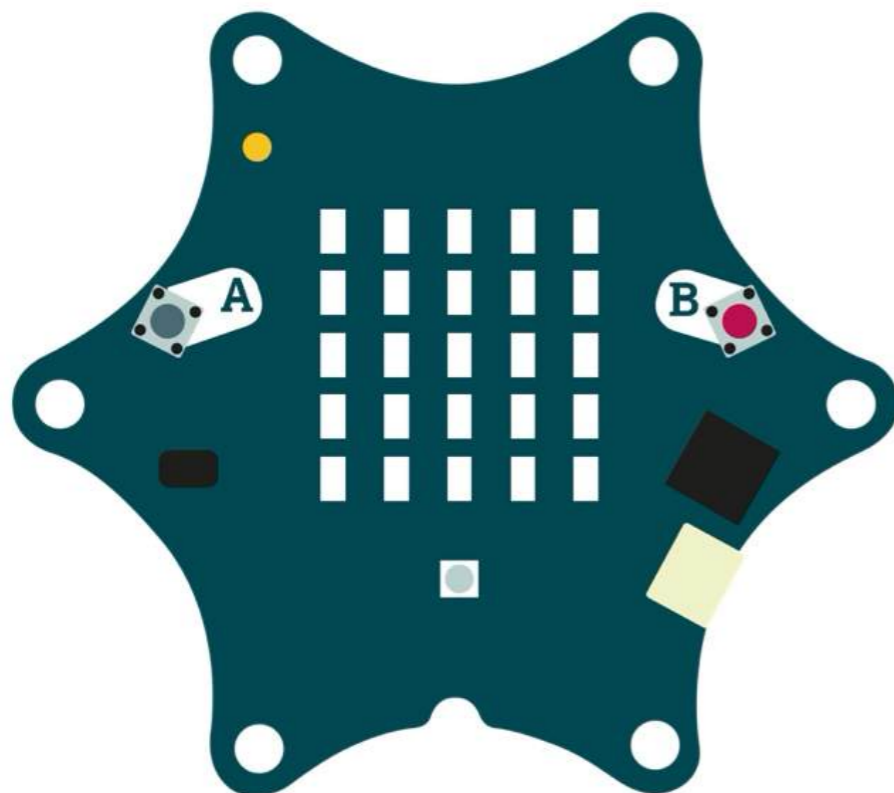
```
while True:
```

```
    if button_a.was_pressed():  
        display.show(Image.HAPPY)  
        led.set_colors(0, 255, 0)  
        sleep(1000)  
        display.clear()  
        led.set_colors(0, 0, 0)
```

```
    if button_b.was_pressed():  
        display.show(Image.SAD)  
        led.set_colors(255, 0, 0)  
        sleep(1000)  
        display.clear()  
        led.set_colors(0, 0, 0)
```

Tasten-Eingabe

Kommunikation
durch Bilder und
Farben

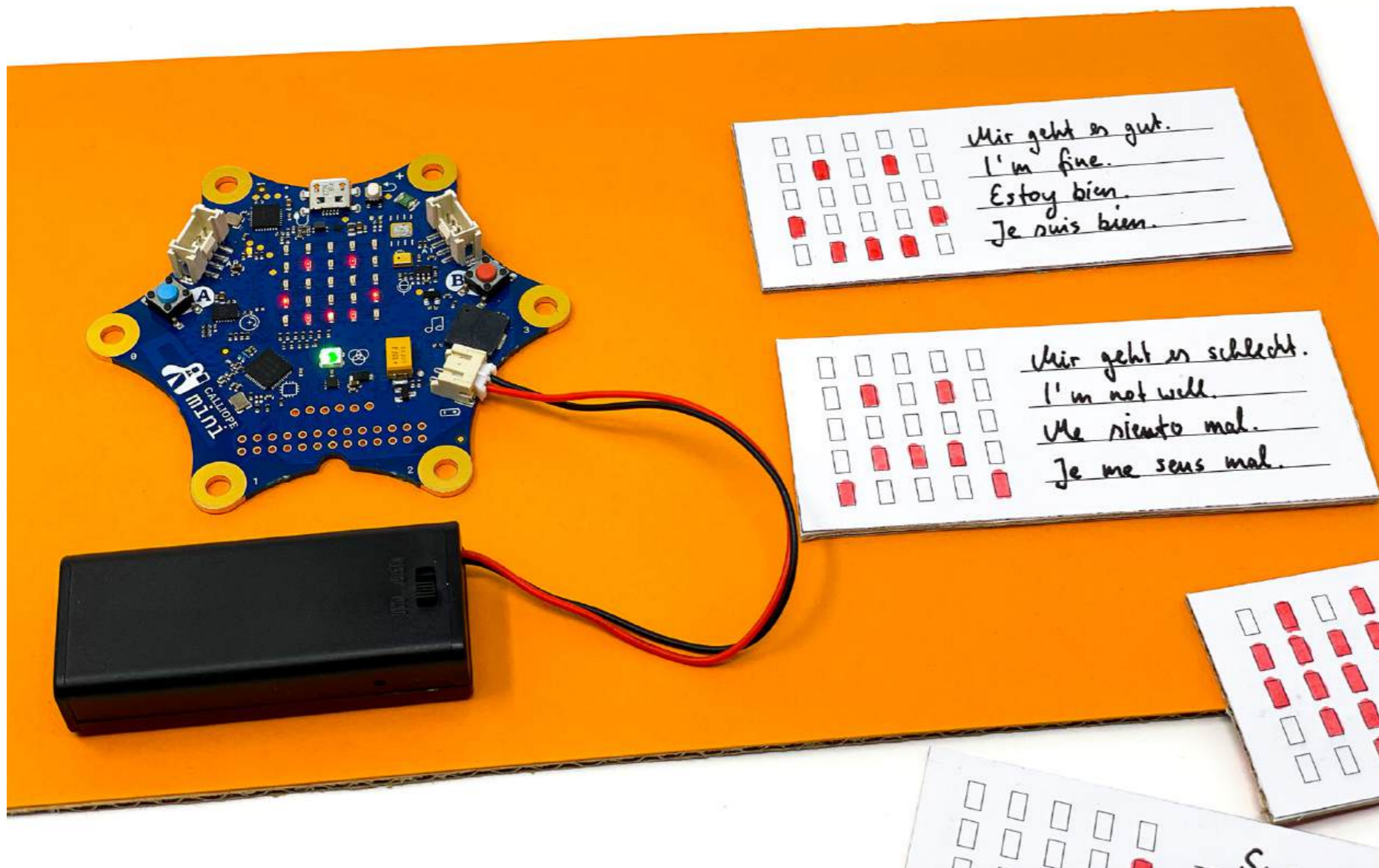


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert der Calliope mini auf eure Eingabe?
- Sind die Bilder lange genug sichtbar?
- Fallen euch noch andere Wege der Kommunikation ohne Sprache ein?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Kommunikation ohne Sprache kann sehr schwierig sein. Bastle kleine Erklärungskarten, um eure Bilder zu beschreiben. Kennt ihr noch andere Sprachen?

LICHT IM WELTALL

06



MATERIAL

Pappe / Tonpapier
Schere
Acrylkugel
Gummiband
Deko zum Verzieren

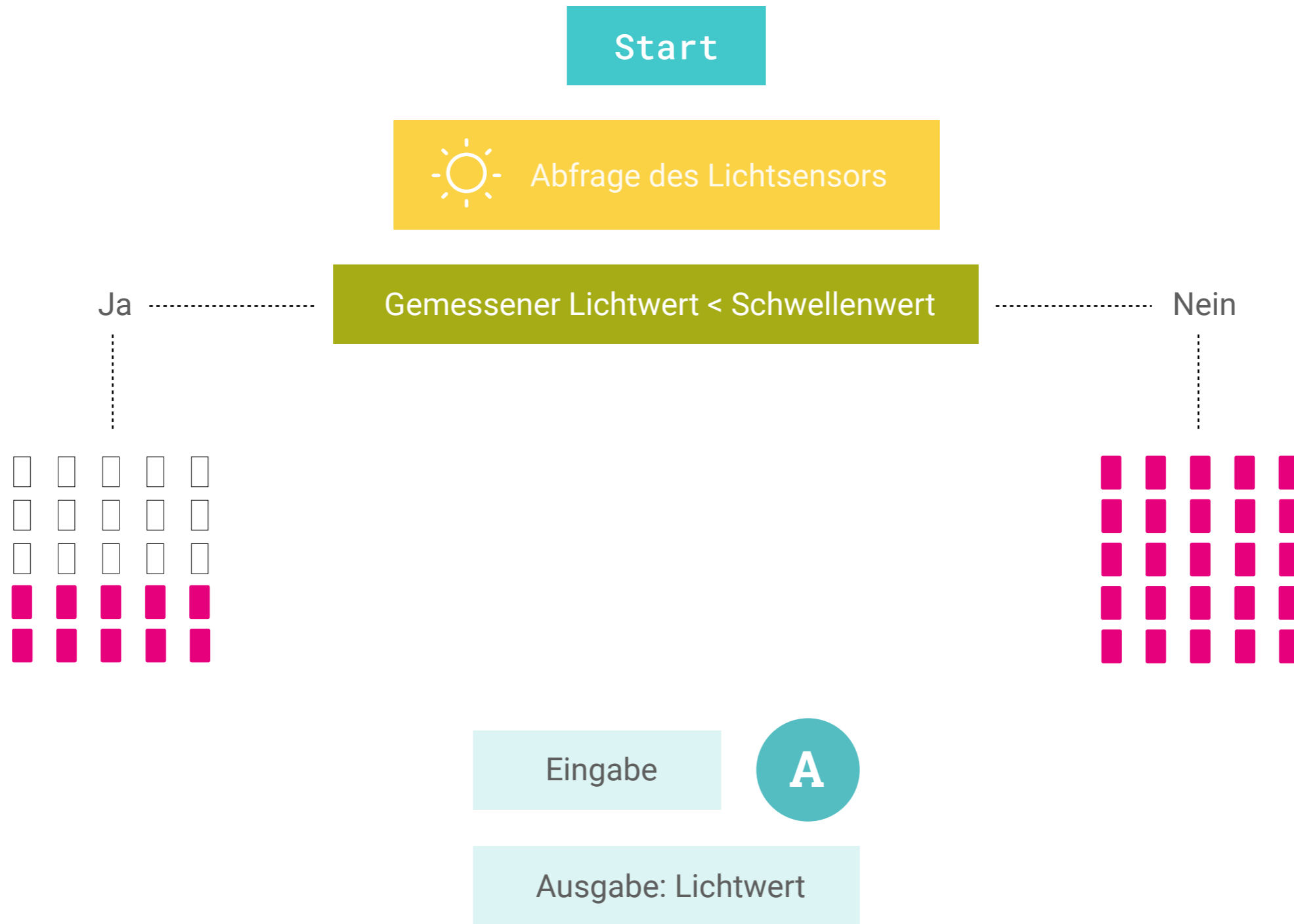
SCHWIERIGKEIT



In den unendlichen Weites des Weltraums gibt es keine Lampen oder andere elektrischen Lichter wie bei uns auf der Erde. Lichtquellen sind Sonnen und andere leuchtende Himmelskörper. Der Calliope mini kann mit Hilfe der 25 LED Matrix die Helligkeit der Umgebung messen und als Wert anzeigen - und das ist ganz einfach programmiert.

**CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:**







PROGRAMMIEREN

06



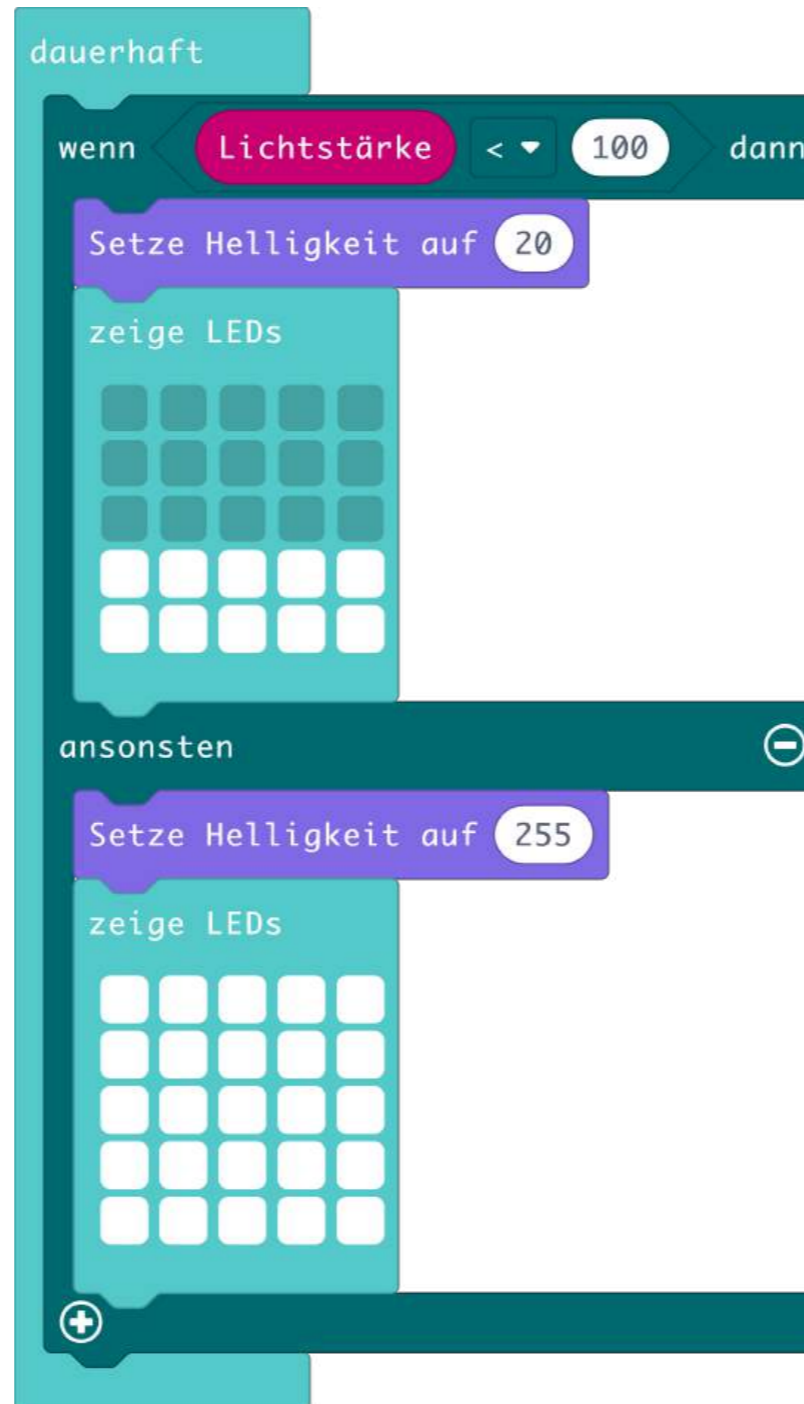
TIPP:

Der Lichtsensor misst die Helligkeit der Umgebung und gibt diese in Werten zwischen 0 und 255 aus.

Der Block **Setze Helligkeit auf** bestimmt die Leuchtstärke der roten LED-Matrix. Ist es dunkler, leuchten die LEDs auch dunkler und umgekehrt.

PROGRAMM:

mini-Helligkeit.hex



Abfrage des Lichtsensors und Abgleich mit einem Schwellenwert

wenn Knopf A gedrückt
zeige Zahl Lichtstärke



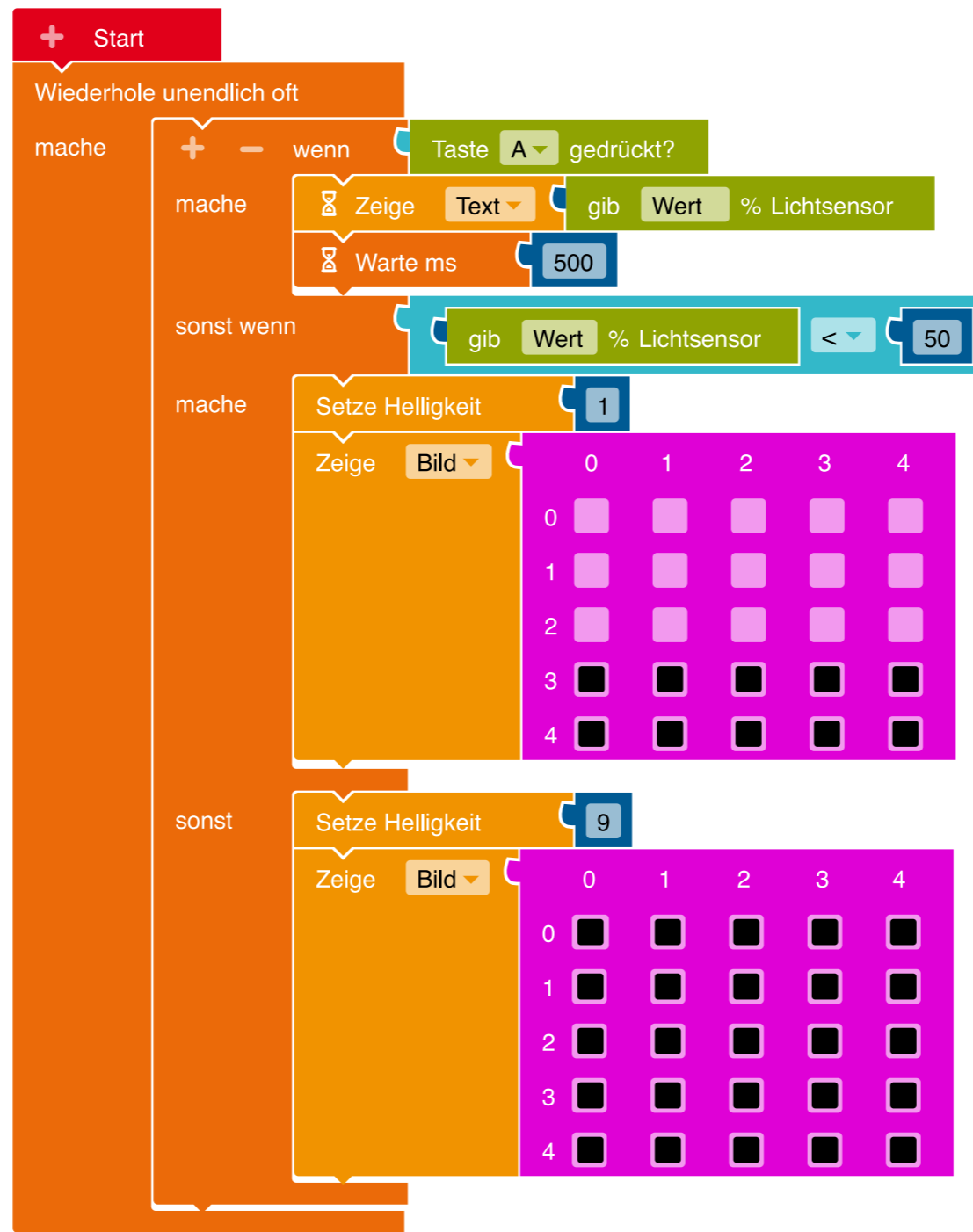
TIPP:

Der Lichtsensor misst die Helligkeit der Umgebung und gibt diese in Prozentwerten an.

Der Block **Setze Helligkeit** bestimmt die Leuchtstärke der roten LED-Matrix. Ist es dunkler, leuchten die LEDs auch dunkler und umgekehrt.

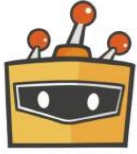
PROGRAMM:

mini-Helligkeit.xml



Abfrage des Helligkeit-/ Lichtsensors

Abgleich mit einem Schwellenwert



PROGRAMMIEREN

06



TIPP:

Der Lichtsensor misst die Helligkeit der Umgebung und gibt diese in Werten zwischen 0 und 255 aus.

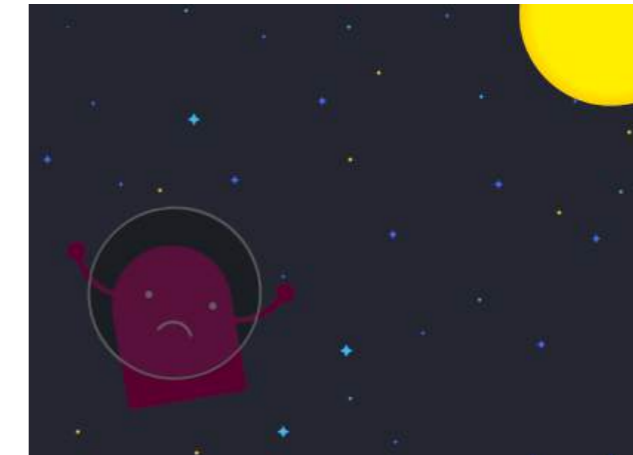
Der Block **Helligkeit setzen** bestimmt die Leuchtstärke der roten LED-Matrix. Ist es dunkler, leuchten die LEDs auch dunkler und umgekehrt.

PROGRAMM:

mini-Helligkeit.sb3

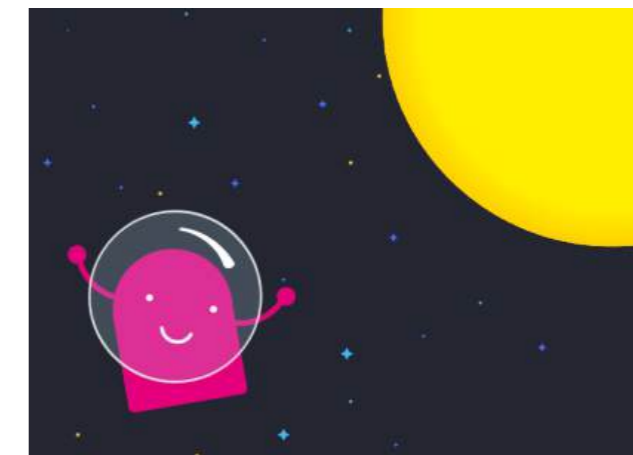
```
Wenn [ ] angeklickt wird
wiederhole fortlaufend
  falls [Umgebungslicht-Helligkeit lesen < 50], dann
    wechsele zu Kostüm Astronaut_Dunkel
    Helligkeit setzen 1
    Muster anzeigen
  sonst
    wechsele zu Kostüm Astronaut_Hell
    Helligkeit setzen 9
    Muster anzeigen
```

```
wenn Taste A gedrückt
  Anzeige Umgebungslicht-Helligkeit lesen
```



Bühnenansicht im Mind+ Editor | Kostüm: Dunkel

```
Wenn [ ] angeklickt wird
wiederhole fortlaufend
  falls [Umgebungslicht-Helligkeit lesen < 50], dann
    setze Größe auf 40
  sonst
    setze Größe auf 100
```



Bühnenansicht im Mind+ Editor | Kostüm: Hell



TIPP:

Der Lichtsensor misst die Helligkeit der Umgebung und gibt diese in Werten zwischen 0 und 255 aus. Ein Schwellenwert ermöglicht euch, die gemessenen Werte zu vergleichen und zu differenzieren.

PLAYGROUND

Ausgabe:

Display

Eingabe:

Button A

Befehle:

Unendlich Funktion
Variable
Licht

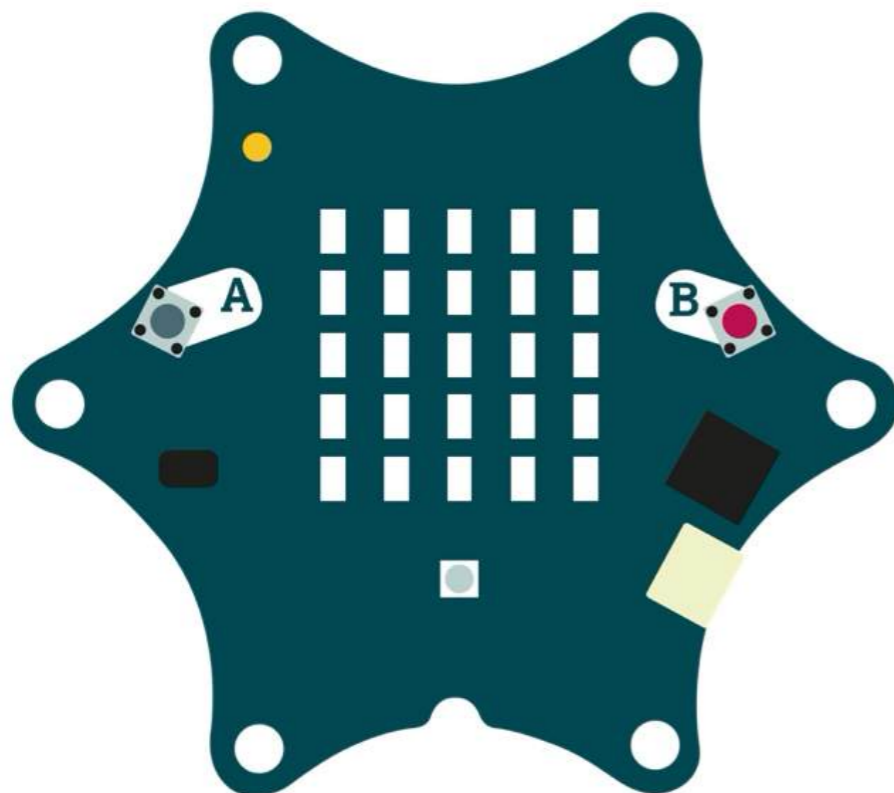
PROGRAMM:

Helligkeit

```
func forever (){
    let Helligkeit:UInt16 = UInt16(io.brightness)
    if Helligkeit < 60 {
        display.show(grid:
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1])
    } else {
        display.show(grid:
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1])
    }
}

func onButtonA (){
    let Helligkeit:UInt16 = UInt16(io.brightness)
    display.show(number: Helligkeit)
    mini.sleep(2000)
}
```

Lichtsensor schreibt Werte in die Variable **Helligkeit** und gleicht diese mit einem Schwellenwert ab



Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert der Calliope mini auf Lichtveränderungen?
- Ist der Schwellenwert gut gewählt?
- An welcher Stelle ist es am hellsten in eurem Zimmer?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Der Calliope mini stellt unsere Erde in einer Acrylkugel dar. Der Lichtsensor erkennt, wann die eine Seite der Erde zur Sonne gedreht ist.

TEMPERATUR IM WELTALL

07



MATERIAL

Pappe / Tonpapier
Schere
Kleber
Gummiband
Deko zum Verzieren

SCHWIERIGKEIT



Das Universum hat eine Temperatur von -270 Grad Celsius und auch viele Planeten unseres Sonnensystems weisen eisige Temperaturen auf. Zum Glück ist es bei uns auf der Erde nicht so kalt - außer vielleicht das leckere Eis an heißen Sommertagen.

Der Calliope mini hat einen Temperatursensor verbaut und kann so die Temperatur messen. Mit einem programmierten Eis-Thermometer könnt ihr die Temperatur anzeigen lassen und durch verschiedene Ausgabemöglichkeiten des Calliope mini visuell abbilden.

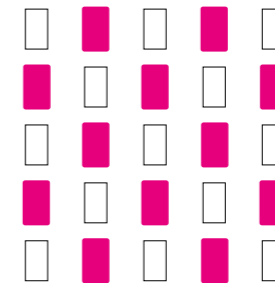
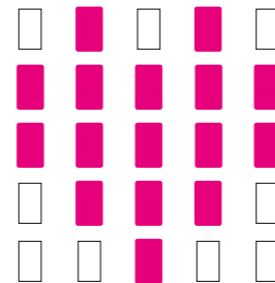
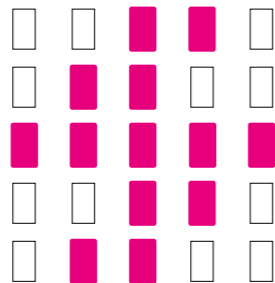
CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:



Start

 Abfrage des Temperatursensors

Abgleich der gemessene Temperatur mit verschiedenen Schwellenwerten



Eingabe

P0

Ausgabe: Temperatur



PROGRAMMIEREN

07

TIPP:

Der Temperatursensor misst die Temperatur in Grad Celsius. Durch Schwellenwerte könnt ihr die gemessenen Werte miteinander vergleichen. Um die Temperatur anzuzeigen, könnt ihr den golden Touch-Pin P0 des Calliope mini berühren. Vergesst dabei nicht, immer den Minus-Pin (-) mit zu drücken.

PROGRAMM:

mini-Temperatur.hex



```
dauerhaft
  setze Temperatur auf Temperatur (°C)
  wenn Temperatur < 5 dann
    setze RGB-LED-Farbe auf [Cyan]
    zeige LEDs
  sonst wenn Temperatur < 15 dann
    zeige Symbol [Grid]
    setze RGB-LED-Farbe auf [Orange]
  ansonsten
    zeige Symbol [Checkered]
    setze RGB-LED-Farbe auf [Red]
```

Abfrage des Temperatursensors

Abgleich mit einem Schwellenwert

Eingabe: Pin P0

wenn Pin P0 gedrückt

zeige Zahl Temperatur



PROGRAMMIEREN

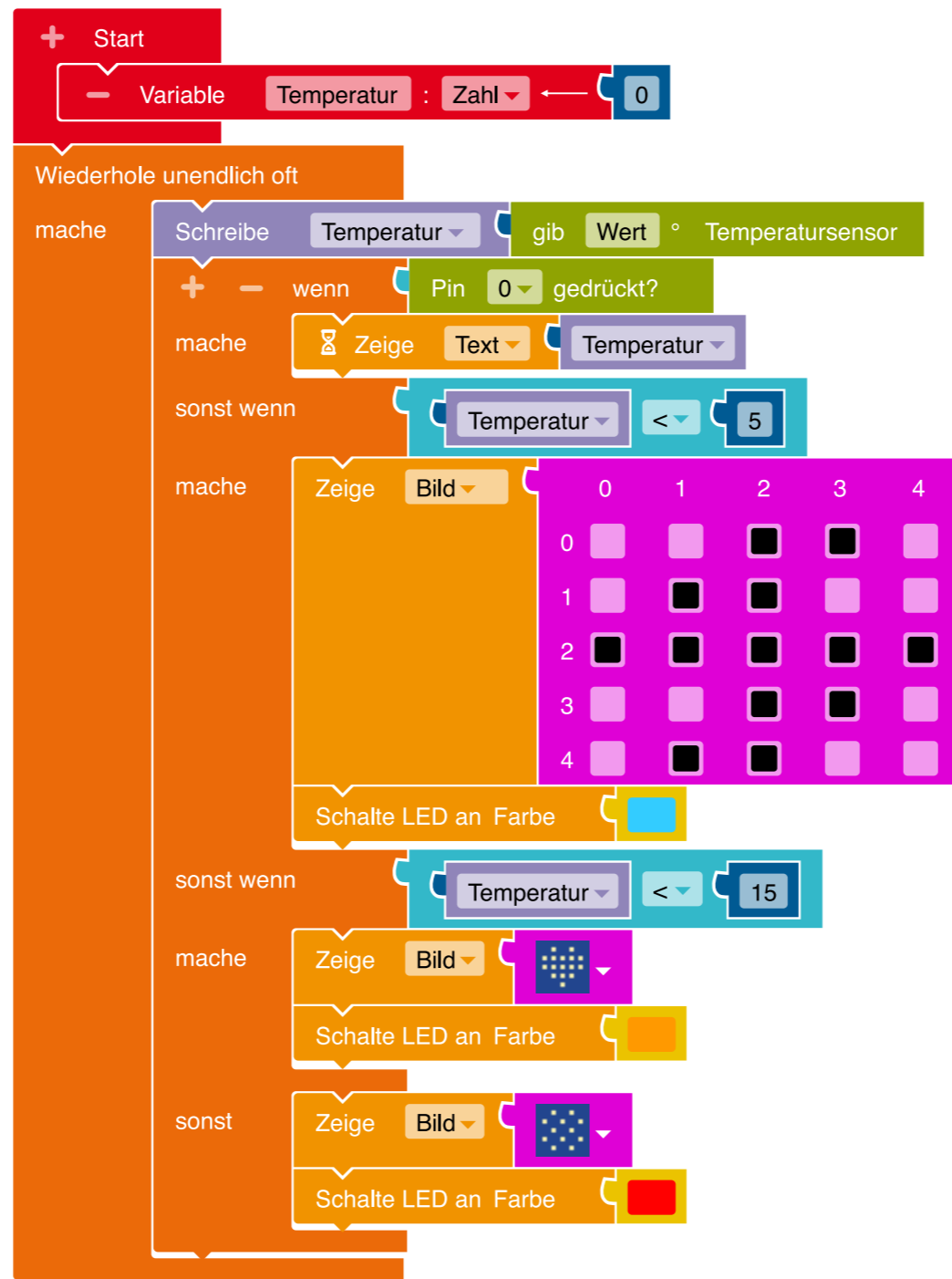
07 | 

TIPP:

Der Temperatursensor misst die Temperatur in Grad Celsius. Durch Schwellenwerte könnt ihr die gemessenen Werte miteinander vergleichen. Um die Temperatur anzuzeigen, könnt ihr den golden Touch-Pin P0 des Calliope mini berühren. Vergesst dabei nicht, immer den Minus-Pin (-) mit zu drücken.

PROGRAMM:

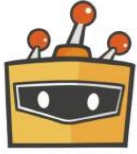
mini-Temperatur.xml



Abfrage des
Temperatursensors

Eingabe: Pin P0

Abgleich mit einem
Schwellenwert



PROGRAMMIEREN

TIPP:

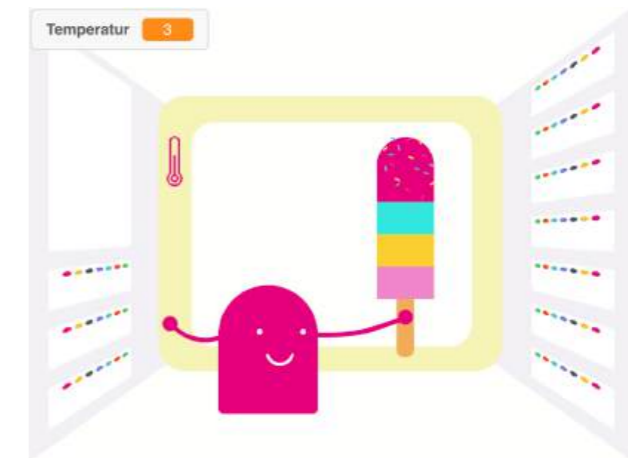
Der Temperatursensor misst die Temperatur in Grad Celsius. Durch Schwellenwerte könnt ihr die gemessenen Werte miteinander vergleichen. Um die Temperatur anzuzeigen, könnt ihr den golden Touch-Pin P0 des Calliope mini berühren. Vergesst dabei nicht, immer den Minus-Pin (-) mit zu drücken.

PROGRAMM:

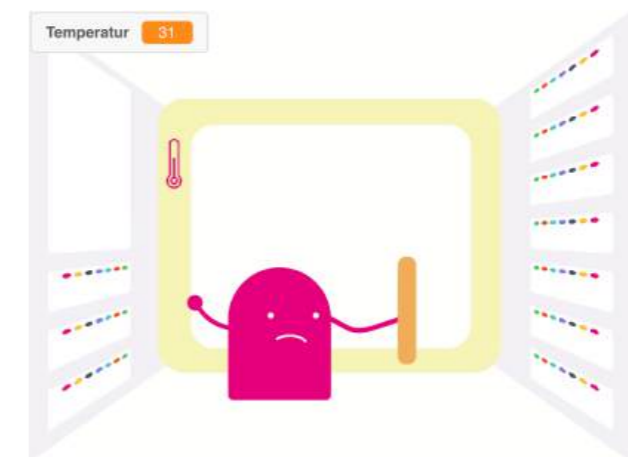
mini-Temperatur.sb3

```
Wenn  angeklickt wird
wiederhole fortlaufend
  setze Temperatur auf  Temperatur lesen
  falls Temperatur < 5, dann
    wechsle zu Kostüm Eis_1
     Muster anzeigen
     RGB-LED auf  setzen
  sonst falls Temperatur < 15, dann
    wechsle zu Kostüm Eis_2
     Muster anzeigen
     RGB-LED auf  setzen
  sonst
    wechsle zu Kostüm Eis_3
     Muster anzeigen
     RGB-LED auf  setzen
```

```
 wenn Pin P0 angeschlossen
 Anzeige Temperatur
```



Bühnenansicht im Mind+ Editor | Kostüm: Eis_1



Bühnenansicht im Mind+ Editor | Kostüm: Eis_3



PROGRAMMIEREN

07 |

TIPP:

Der Temperatursensor misst die Temperatur. Durch Schwellenwerte könnt ihr die gemessenen Werte miteinander vergleichen. Um die Temperatur anzuzeigen, könnt ihr die golden Touch-Pins des Calliope mini berühren. Vergesst dabei nicht, immer den Minus-Pin (-) mit zu drücken.

PLAYGROUND

Ausgabe:

Display, RGB LED

Eingabe:

Touchpins

Befehle:

Unendlich Funktion,
Wenn... dann, Variable
Thermometer

PROGRAMM:

Temperatur

```
func forever (){  
    let Temperatur:UInt16 = UInt16(io.temperature)  
    if Temperatur < 5 {  
        display.show(image: .flash)  
        rgb.on(color: .cyan)  
    } else if Temperatur < 15 {  
        display.show(image: .heart)  
        rgb.on(color: .orange)  
    } else {  
        display.show(image: .full)  
        rgb.on(color: .red)  
    }  
}
```

```
func onPin(pin:UInt16){  
    let Temperatur:UInt16 = UInt16(io.temperature)  
    display.show(number: Temperatur)  
    mini.sleep(2000)  
}
```

Temperatursensor schreibt Werte in die Variable **Temperatur** und gleicht diese mit einem Schwellenwert ab

Eingabe: Die Touch-Pins geben die gemessenen Temperatur wieder.



TIPP:

Der Temperatursensor misst die Temperatur in Grad Celsius. Durch Schwellenwerte könnt ihr die gemessenen Werte miteinander vergleichen. Um die Temperatur anzuzeigen, könnt ihr die golden Touch-Pins des Calliope mini berühren. Vergesst dabei nicht, immer den Minus-Pin (-) mit zu drücken.

PROGRAMM:

mini-Temperatur.py

```
from calliope_mini import *

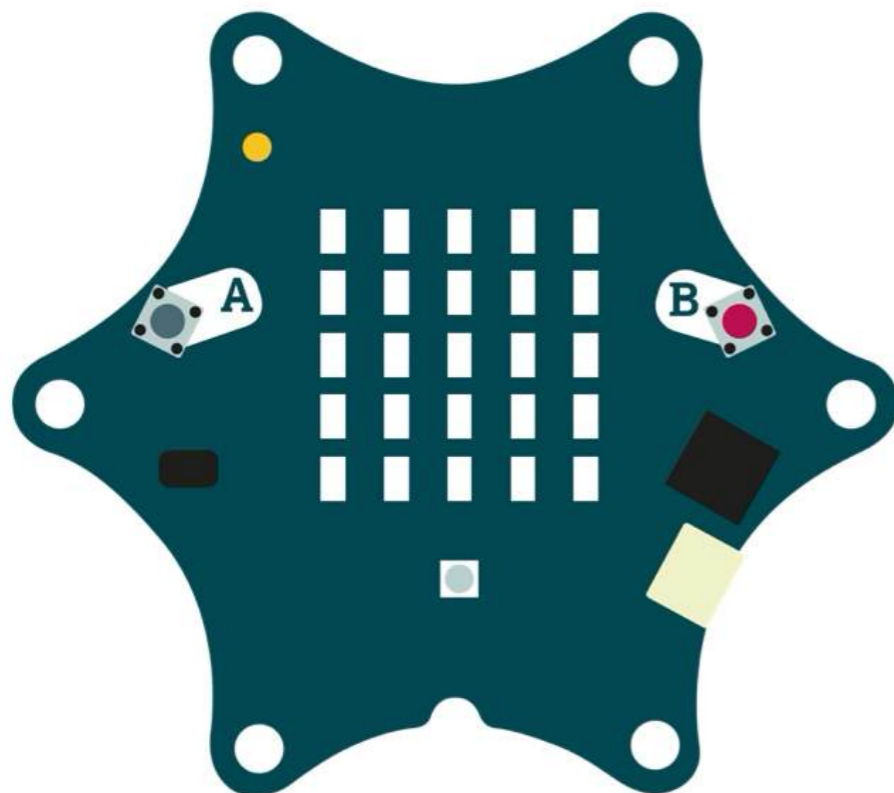
Temperatur = temperature()

def pin():
    if pin1.read_analog() > 400:
        display.scroll(str(Temperatur))
        sleep(2000)
    else:
        sleep(10)

while True:
    pin()
    if Temperatur < 5:
        im = Image('00990:09900:99999:00990:09900:')
        display.show(im)
        led.set_colors(0, 255, 255)
    elif Temperatur < 15:
        display.show(IMAGE.HEART)
        led.set_colors(255, 255, 0)
    else:
        im = Image('09090:90909:09090:90909:09090:')
        display.show(im)
        led.set_colors(255, 0, 0)
```

Temperatursensor schreibt Werte in die Variable **Temperatur**

Eingabe: Die Touch-Pins geben die gemessenen Temperatur wieder.



Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Reagiert der Calliope mini auf Temperaturveränderungen?
- Ist der Schwellenwert gut gewählt?
- Funktioniert die Pin-Eingabe?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Ein kleines und einfaches Bastelprojekt: das Eis-Thermometer.

Stanze oder schneide ein kleines Loch in die Pappe, damit die LED-Matrix des Calliope mini zu sehen ist.

ASTRONAUTEN-ORAKEL

08



MATERIAL

Pappe / Tonpapier
Schere
Acrylkugel
Gummiband
Deko zum Verzieren

Astronaut:in ist ein sehr cooler Beruf, den nur wenige Menschen auf der Welt ausüben dürfen. Ihr müsst sehr schlau sein und sportlich fit. Ob ihr das Zeug zum Astronauten bzw. zur Astronautin habt, beantwortet euch das Calliope mini Orakel.

SCHWIERIGKEIT



CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:





Start

Ausgabe: Orakel - Frage als Text

Werde ich später Astronaut:in?

Erstellung einer Variable

Zufall



Eingabe: Schütteln

Zufallszahl zwischen 0 und 3

Zufall → 3

Ausgabe: Zugeordneter Text der Variable **Zufall**

"Ja, du bist sportlich und intelligent."





TIPP:

Der Variable *Zufall* wird bei jeder Schüttel-Eingabe eine neue Zahl zugeordnet und danach eine Antwort zugeteilt.

PROGRAMM:

mini-Orakel.hex



```
beim Start
  zeige Text "Werde ich spaeter Astronaut:in?"

wenn geschüttelt
  setze Zufall auf wähle eine zufällige Zahl von 0 bis 3
  wenn Zufall = 0 dann
    zeige Text "Ja sicher, du bist sehr schlau."
  +
  wenn Zufall = 1 dann
    zeige Text "Vielleicht, aber du musst dich anstrengen."
  +
  wenn Zufall = 2 dann
    zeige Text "Nein, du wirst eher Polizist:in."
  +
  wenn Zufall = 3 dann
    zeige Text "Ja, du bist sportlich und intelligent."
  +
```

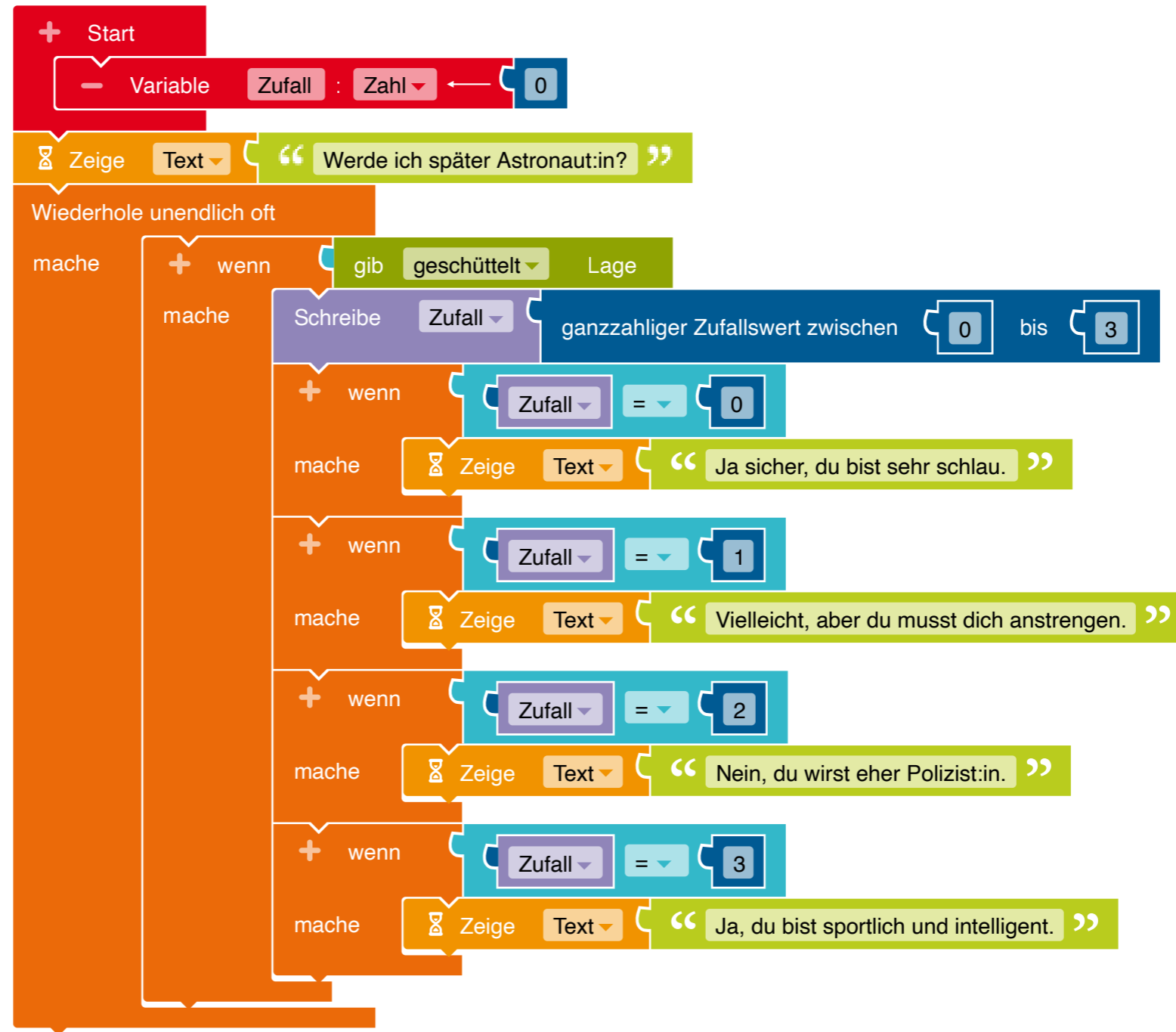


TIPP:

Der Variable *Zufall* wird bei jeder Schüttel-Eingabe eine neue Zahl zugeordnet und danach eine Antwort zugeteilt.

PROGRAMM:

mini-Orakel.xml





PROGRAMMIEREN

08

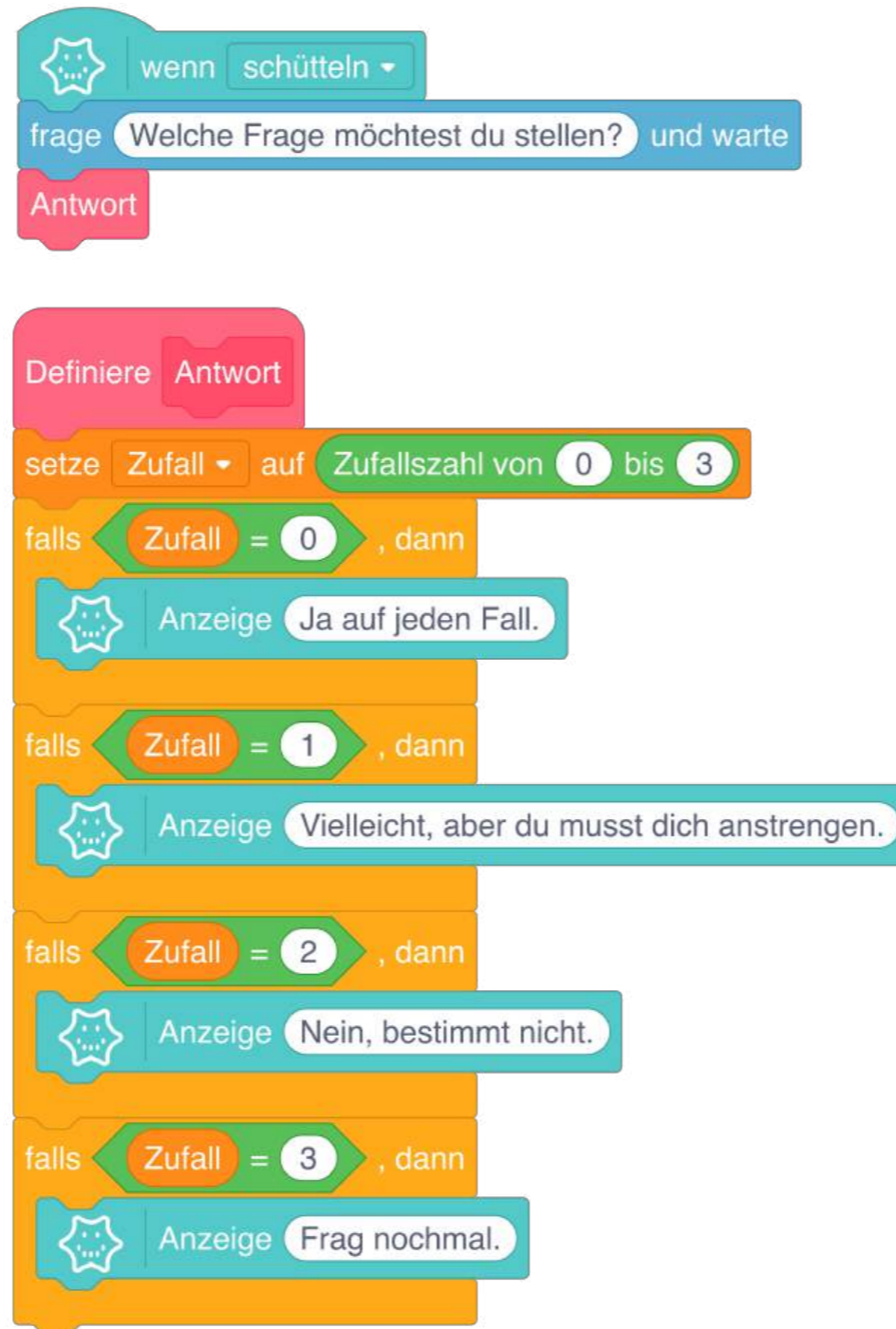


TIPP:

Mit dem Block *frage ... und warte* könnt ihr direkt eigene Fragen in der Bühne eingeben. Der Calliope mini gibt euch danach eine zufällige Antwort. Definiert dazu den Block *Antwort* in der Kategorie „Meine Blöcke“.

PROGRAMM:

mini-Orakel.sb3



Bühnenansicht im Mind+ Editor



TIPP:

Damit der Text vollständig angezeigt wird, benutzt erneut den *mini.sleep* Befehl.

Der Variable *Zufall* wird bei jeder Schüttel-Eingabe eine neue Zahl zugeordnet und danach eine Antwort zugeteilt.

PLAYGROUND

Ausgabe:
LED-Matrix

Eingabe:
Lagesensor

Befehle:
Wenn...dann
Zufall

PROGRAMM:

Orakel

```
func start(){
    display.show(text: "Werde ich spaeter Astronaut:in?")
    mini.sleep(3000)
}

func onShake() {
    let Zufall:UInt16 = random(0...3)
    if Zufall == 0 {
        display.show(text:"Ja sicher, du bist sehr schlau.")
        mini.sleep(3000)
    }
    if Zufall == 1 {
        display.show(text:"Vielleicht, aber du musst dich anstrengen.")
        mini.sleep(3000)
    }
    if Zufall == 2 {
        display.show(text:"Nein, du wirst eher Polizist:in.")
        mini.sleep(3000)
    }
    if Zufall == 3 {
        display.show(text:"Ja, du bist sportlich und intelligent.")
        mini.sleep(3000)
    }
}
```



TIPP:

Der Variable *Zufall* wird bei jeder Schüttel-Eingabe eine neue Zahl zugeordnet und danach eine Antwort zugeteilt.

PROGRAMM:

mini-Orakel.py

```
from calliope_mini import *  
import random
```

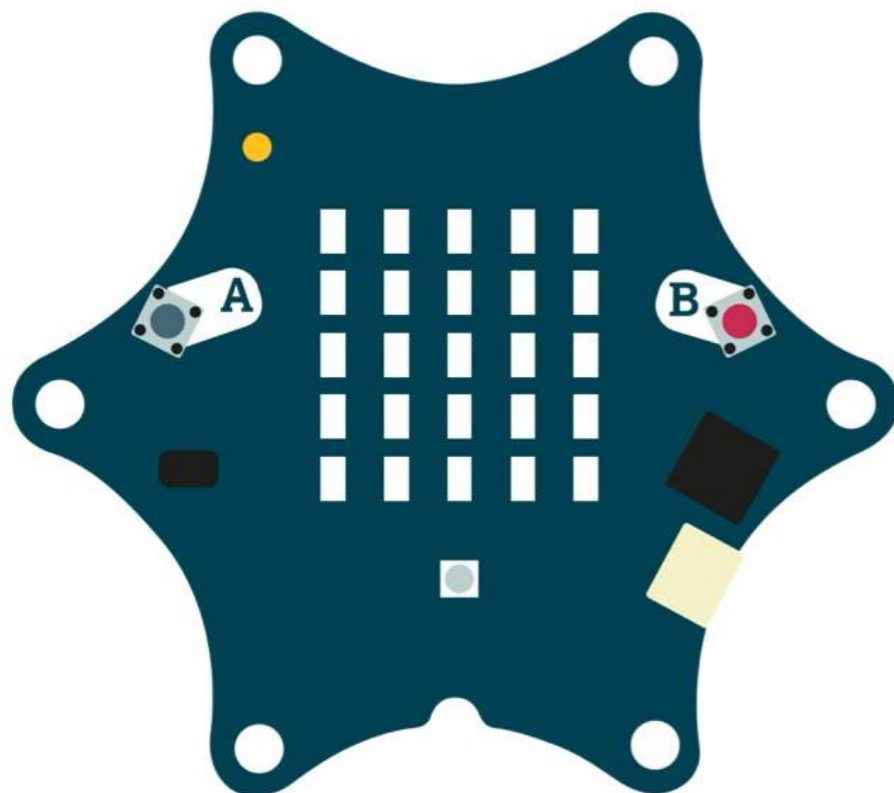
```
display.scroll("Werde ich spaeter Astronaut:in?")
```

Start

```
def orakel():  
    Zufall = random.randint(0,3)  
    if Zufall == 0:  
        display.scroll("Ja sicher, du bist sehr schlau.")  
    elif Zufall == 1:  
        display.scroll("Vielleicht, aber du musst dich anstrengen.")  
    elif Zufall == 2:  
        display.scroll("Nein, du wirst eher Polizist:in.")  
    elif Zufall == 3:  
        display.scroll("Ja, du bist sportlich und intelligent.")
```

```
while True:  
    x = accelerometer.get_x()  
    if x > 20:  
        orakel()  
    else:  
        sleep(10)
```

Eingabe: Schütteln

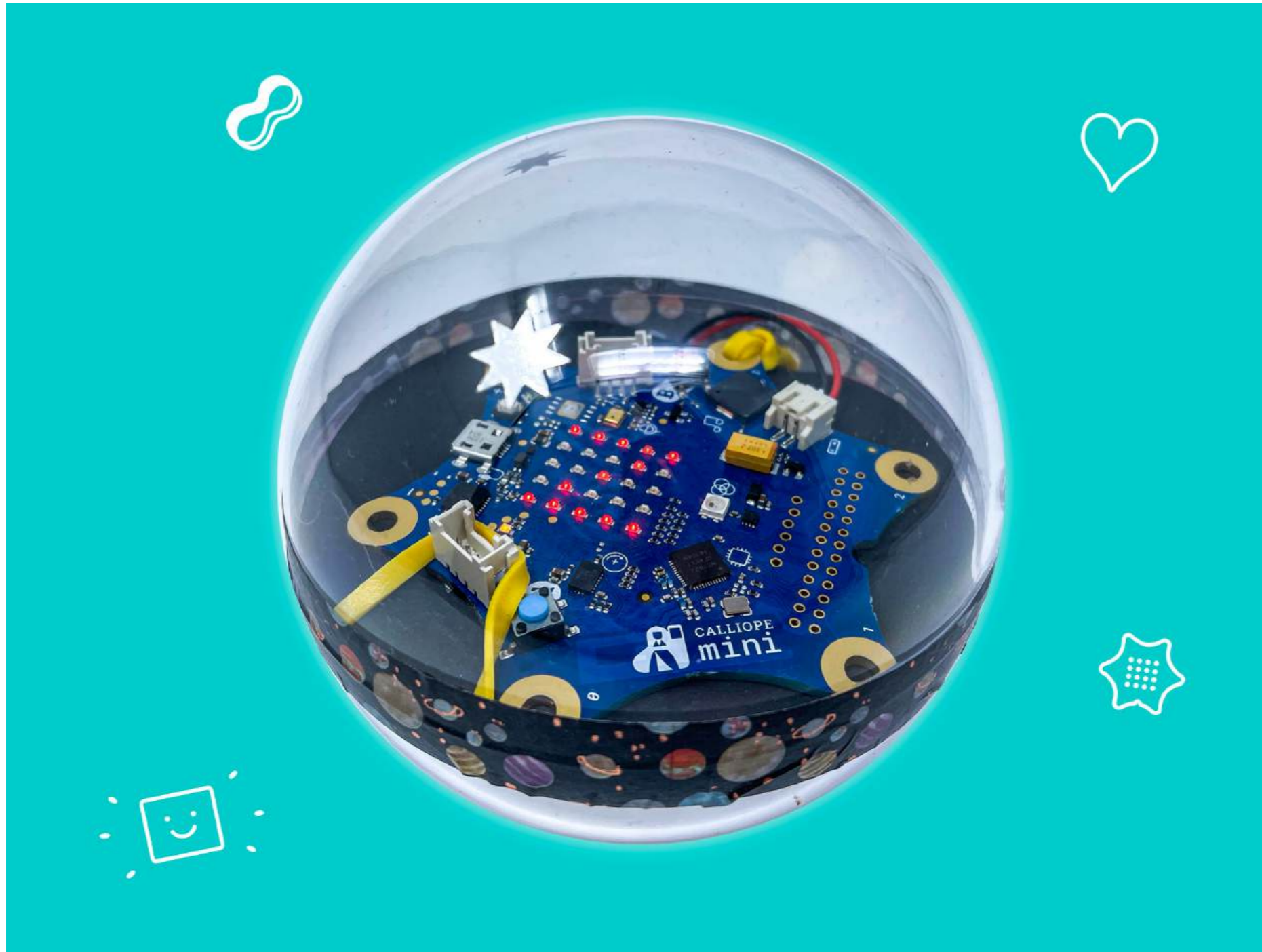


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Habt ihr die Variable richtig benannt und konfiguriert?
- Sind die möglichen Antworten sinnvoll gewählt?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Der Calliope mini wird zu einer kleinen Wahrsagekugel und sagt euch die Zukunft voraus.

So könnte euer Orakel aussehen.



PLANETEN-QUIZ

MATERIAL

Pappe / Tonpapier

Schere

Klettband

Stifte

Deko zum Verzieren

SCHWIERIGKEIT



Ihr kennt die acht Planeten unseres Sonnensystems. Diese Planeten kreisen auf unterschiedlichen Umlaufbahnen um die Sonne herum und werden anhand ihres Abstandes zur Sonne entsprechend sortiert. Der Merkur ist dabei der erste Planet mit einer Distanz von ca. 60 Millionen Kilometern zur Sonne.

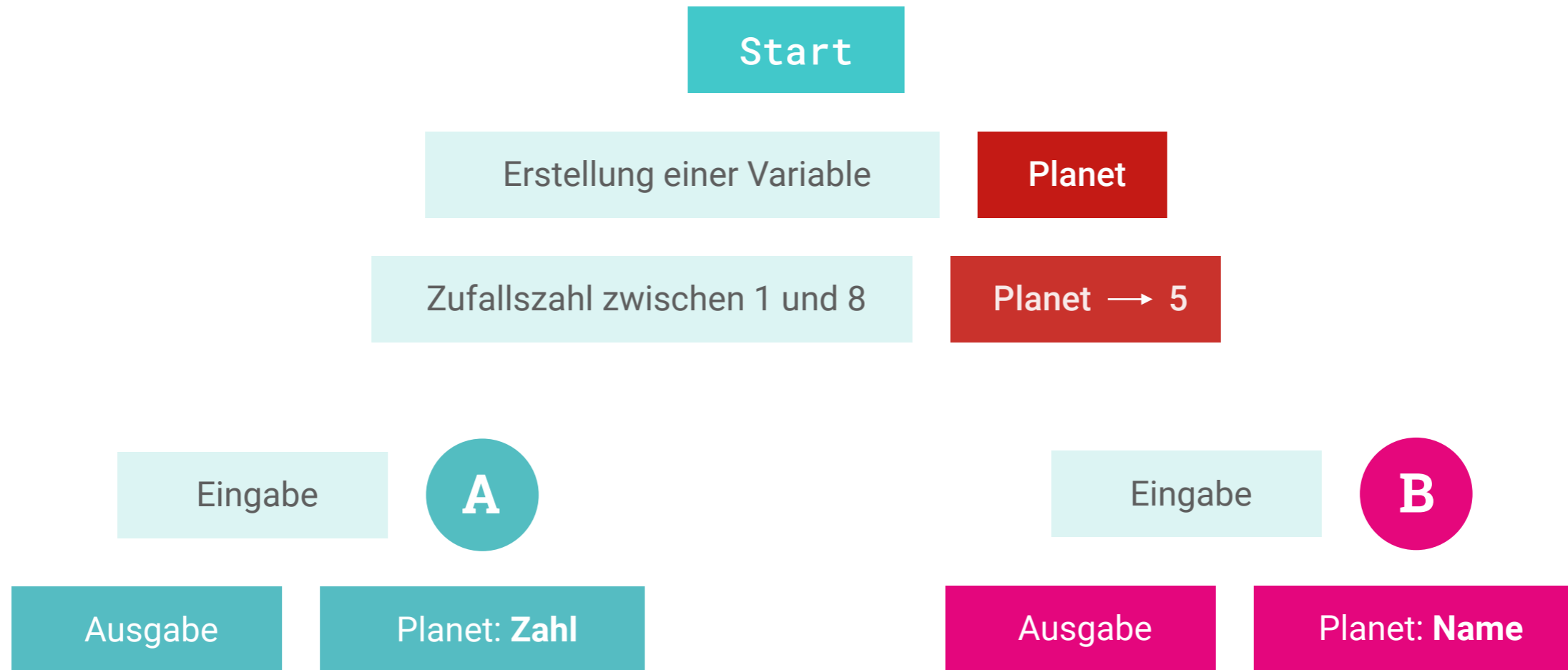
Um sich die Reihenfolge der Planeten besser merken zu können, gibt es folgenden Merksatz:

Mein Vater erklärt mir jeden Samstag unseren Nachthimmel.

Dabei stehen die Anfangsbuchstaben immer für einen Planeten. Mit dem Calliope mini kannst du ein Würfelspiel programmieren, um die Reihenfolge der Planeten besser kennenzulernen.

CODING TAGEBUCH
FÜR EUCH ZUM
HERUNTERLADEN:





1



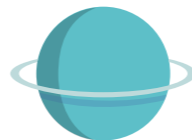
2



3



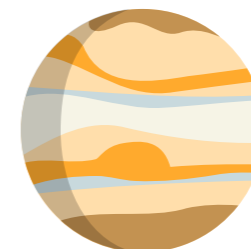
4



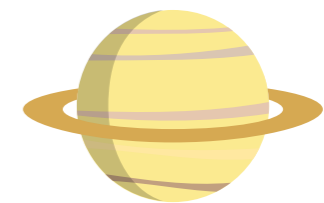
5



6



7



8



TIPP:

Der Variable **Planet** wird bei jeder Tasten-Eingabe eine neue Zahl und somit ein neuer Planet zugeordnet. Je nach Schwierigkeitsgrad könnt ihr Antwortmöglichkeiten hinzufügen oder entfernen.

PROGRAMM:

mini-Planetenquiz.hex



```
wenn Knopf A gedrückt
  setze Planet auf wähle eine zufällige Zahl von 1 bis 8
  zeige Zahl Planet

wenn Knopf B gedrückt
  wenn Planet = 1 dann
    zeige Text "Merkur"
  sonst wenn Planet = 2 dann
    zeige Text "Venus"
  sonst wenn Planet = 3 dann
    zeige Text "Erde"
  sonst wenn Planet = 4 dann
    zeige Text "Mars"
  sonst wenn Planet = 5 dann
    zeige Text "Jupiter"
  sonst wenn Planet = 6 dann
    zeige Text "Saturn"
  sonst wenn Planet = 7 dann
    zeige Text "Uranus"
  ansonsten
    zeige Text "Neptun"
```

Das Herzstück des Programms: der Zufallsgenerator.

Abgleich und Zuordnung zu den jeweiligen Planeten



TIPP:

Der Variable **Planet** wird bei jeder Tasten-Eingabe eine neue Zahl und somit ein neuer Planet zugeordnet. Je nach Schwierigkeitsgrad könnt ihr Antwortmöglichkeiten hinzufügen oder entfernen.

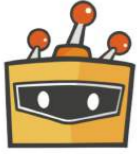
PROGRAMM:

mini-Planetenquiz.xml

```
Start
Variable Planet: Zahl ← 0
Wiederhole unendlich oft
  mache
    Warte bis
      gib gedrückt Taste A = wahr
    Schreibe Planet
      ganzzahliger Zufallswert zwischen 1 bis 8
    Zeige Zeichen
      Planet
    Warte bis
      gib gedrückt Taste B = wahr
    wenn
      Planet = 1
        mache
          Zeige Text
            "Merkur"
      sonst wenn
        Planet = 2
        mache
          Zeige Text
            "Venus"
      sonst wenn
        Planet = 3
        mache
          Zeige Text
            "Erde"
      sonst wenn
        Planet = 4
        mache
          Zeige Text
            "Mars"
      sonst wenn
        Planet = 5
        mache
          Zeige Text
            "Jupiter"
      sonst wenn
        Planet = 6
        mache
          Zeige Text
            "Saturn"
      sonst wenn
        Planet = 7
        mache
          Zeige Text
            "Uranus"
      mache
        Zeige Text
          "Neptun"
```

Das Herzstück des Programms: der Zufallsgenerator.

Abgleich und Zuordnung zu den jeweiligen Planeten



PROGRAMMIEREN

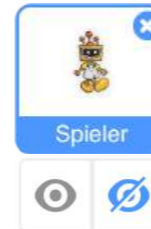
TIPP:

Das Programm ist auf insgesamt 9 Kostüme aufgeteilt: den Spieler und die acht Planeten. Das Programm für die Planeten ändert sich nur in der Abfrage der Variable (Planet = 1; Planet = 2 ...).

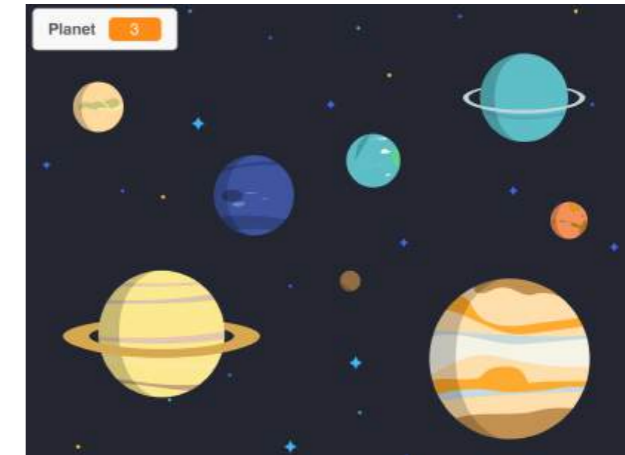
PROGRAMM:

mini-Planetenquiz.sb3

```
wenn Taste A gedrückt
  setze Planet auf Zufallszahl von 1 bis 8
  Anzeige Planet
```



```
wenn Taste B gedrückt
  falls Planet = 1, dann
    Anzeige Merkur
  sonst falls Planet = 2, dann
    Anzeige Venus
  sonst falls Planet = 3, dann
    Anzeige Erde
  sonst falls Planet = 4, dann
    Anzeige Mars
  sonst falls Planet = 5, dann
    Anzeige Jupiter
  sonst falls Planet = 6, dann
    Anzeige Saturn
  sonst falls Planet = 7, dann
    Anzeige Uranus
  sonst
    Anzeige Neptun
```



Bühnenansicht im Mind+ Editor

```
Wenn diese Figur angeklickt wird
  falls Planet = 1, dann
    RGB-LED auf Grün setzen
    sage 😊 für 2 Sekunden
  sonst
    RGB-LED auf Rot setzen
    sage 😊 für 2 Sekunden
  warte 3 Sekunden
  RGB-LED ausschalten
```





TIPP:

Der Variable *Planet* wird beim Programmstart eine neue Zahl und somit ein neuer Planet zugeordnet. Je nach Schwierigkeitsgrad könnt ihr Antwortmöglichkeiten hinzufügen oder entfernen.

PLAYGROUND

Ausgabe:

LED-Matrix, Text

Funktionen:

Tasten Funktion
Pause

PROGRAMM:

Planetenquiz

```
let Planet:UInt16 = random(1...8)
```

Erstellung einer Variable

```
func onButtonA(){  
    display.show(number: Planet)  
}
```

Eingabe: Taste A

```
func onButtonB(){  
    if Planet == 1 {  
        display.show(text:"Merkur")  
    }  
    if Planet == 2 {  
        display.show(text:"Venus")  
    }  
    if Planet == 3 {  
        display.show(text:"Erde")  
    }  
    if Planet == 4 {  
        display.show(text:"Mars")  
    }  
    if Planet == 5 {  
        display.show(text:"Jupiter")  
    }  
    if Planet == 6 {  
        display.show(text:"Saturn")  
    }  
    if Planet == 7 {  
        display.show(text:"Uranus")  
    }  
    if Planet == 8 {  
        display.show(text:"Neptun")  
    }  
}
```

Abgleich und Zuordnung zu den jeweiligen Planeten



TIPP:

Der Variable *Planet* wird bei jeder Tasten-Eingabe eine neue Zahl und somit ein neuer Planet zugeordnet. Je nach Schwierigkeitsgrad könnt ihr Antwortmöglichkeiten hinzufügen oder entfernen.

PROGRAMM:

mini-Planetenquiz.py

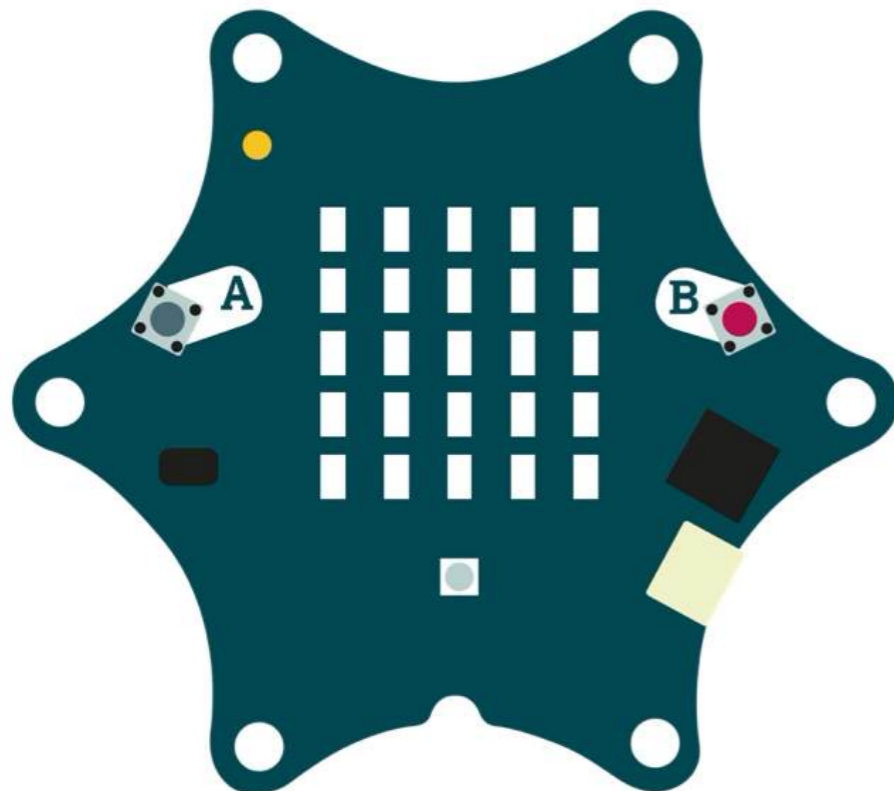
```
from calliope_mini import *
import random

while True:
    if button_a.was_pressed():
        Planet = random.randint(1,8)
        display.show(str(Planet))
    if button_b.was_pressed():
        if Planet == 1:
            display.scroll("Merkur")
        elif Planet == 2:
            display.scroll("Venus")
        elif Planet == 3:
            display.scroll("Erde")
        elif Planet == 4:
            display.scroll("Mars")
        elif Planet == 5:
            display.scroll("Jupiter")
        elif Planet == 6:
            display.scroll("Saturn")
        elif Planet == 7:
            display.scroll("Uranus")
        elif Planet == 8:
            display.scroll("Neptun")
```

Erstellung einer Variable mit Zufallswert

Eingabe: Tasten

Abgleich und Zuordnung zu den jeweiligen Planeten

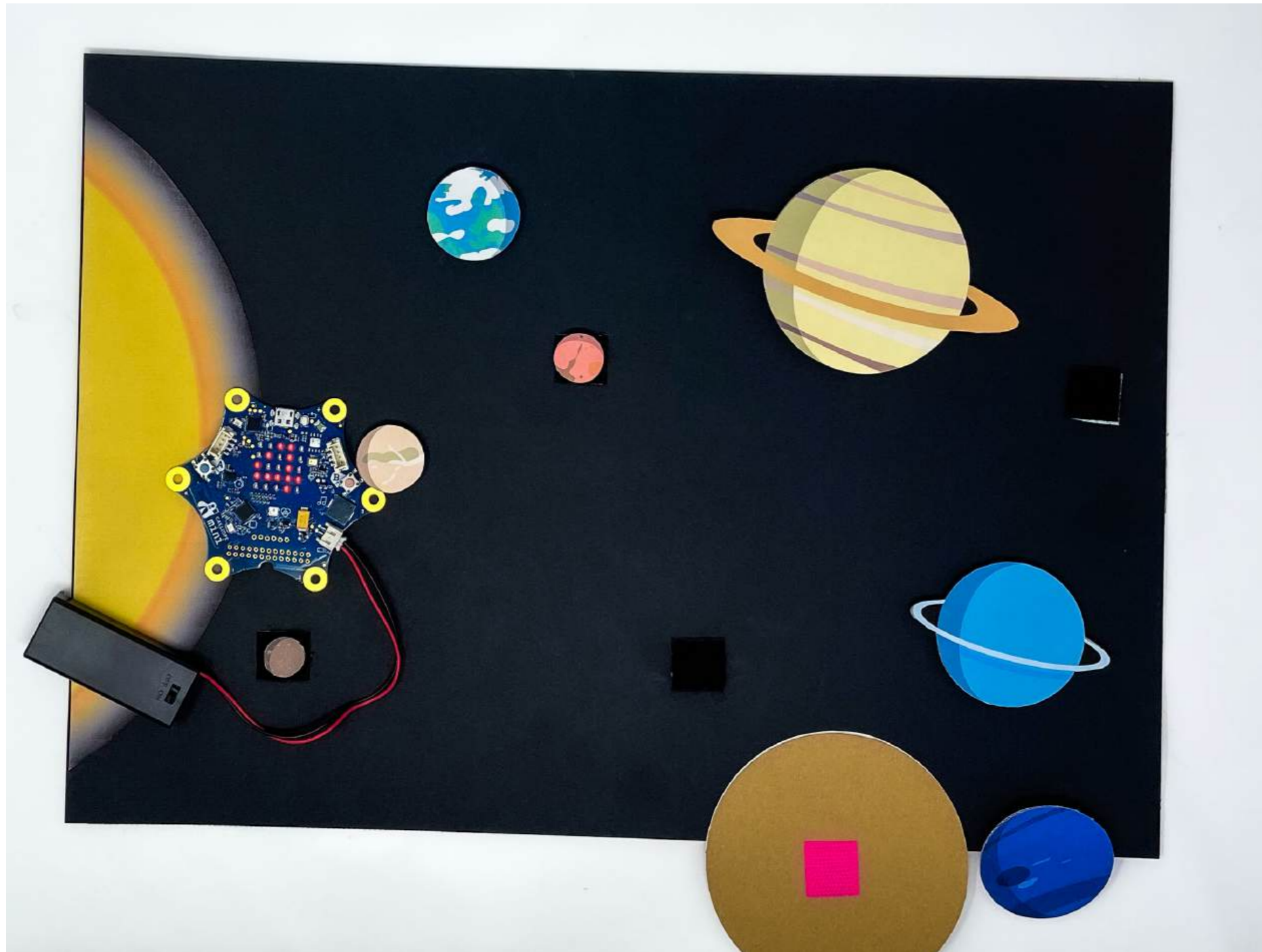


Jetzt könnt ihr euer Programm ausprobieren und testen, ob alles so funktioniert wie gewünscht und bei Bedarf noch anpassen und weitere Ideen entwickeln.

Folgende Fragestellungen helfen beim Testen:

- Habt ihr die Variable richtig benannt und konfiguriert?
- Stimmt die zugeordnete Reihenfolge der Planeten?

Haltet eure Ergebnisse in eurem Calliope Coding-Tagebuch fest.



Zur Übung und gleichzeitig zur Übersicht hilft eine Zeichnung unseres Sonnensystems.

Die Planeten werden mit Klettband auf verschiedenen Positionen angebracht und anschließend zusammen mit dem Calliope mini kontrolliert.



CALLIOPE

Calliope gGmbH, Berlin

Verantwortlich:

Franka Futterlieb und Amando Pascotto

Anregungen und Anmerkungen können an hilfe@calliope.cc gesendet werden.