

# REAKTIONSSPIEL

Entdeckt den Punkt so schnell wie möglich, wenn er links oder rechts erscheint.

## Spielablauf:

- Per Zufall leuchtet links oder rechts eine LED auf der LED-Matrix des Calliope mini auf.
- Mit den Knöpfen A und B wird die Seite markiert, auf der der Punkt entdeckt wurde
- Wurden 3 Versuche verspielt, ist das Spiel zu Ende.

**Eingabe:** Knöpfe A und B.

**Verarbeitung:** Zufällige Anzeige und Auswertung.

**Ausgabe:** Die LED-Matrix, die RGB-LED und der Lautsprecher.

## Entwickle das Programm in thematischer Reihenfolge:

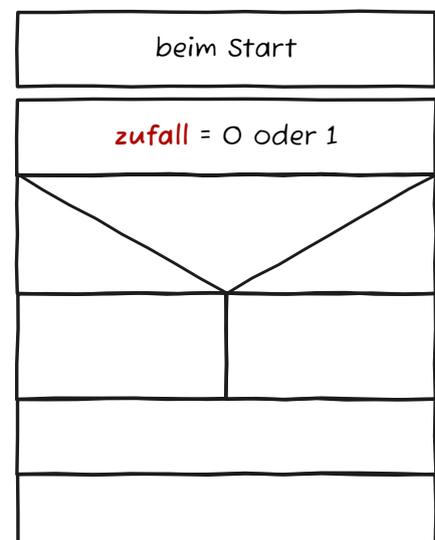
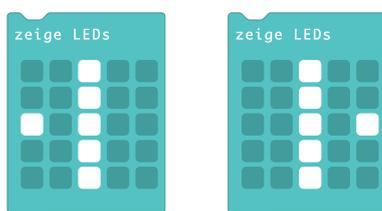
- 1 Auftauchen des Punkts
- 2 Steuerung der Knöpfe
- 3 Auswertung

### 1 Auftauchen des Punkts

- a) Lasse eine Mittellinie auf dem LED-Display leuchten.
- b) Jetzt kommt der Zufall dazu! Lasse dir eine zufällige Zahl aus einem 2er Zahlenraum ausgeben.
- c) Über eine Verzweigung mit folgenden Bedingungen wird die Position des Punkts festgelegt.

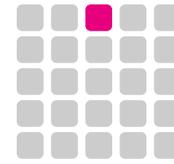
Programmiere anhand des Struktogramms:

Wenn die zufällige Zahl eine 0 ist, lasse den Punkt links erscheinen.  
Wenn die zufällige Zahl eine 1 ist, lasse den Punkt rechts erscheinen.



# 1 Regentropfen Steuerung

a) Erzeuge ein Sprite, positioniere es mittig ganz oben in der ersten Reihe der LED-Matrix und weise es einer Variablen mit dem Namen *wasser* zu.



b) Vervollständige den Code und teste im Simulator.

```
beim Start
  setze wasser auf erzeuge Sprite an Position x: y:
  wasser ändere y um
```

c) Beschreibe was passiert.

---

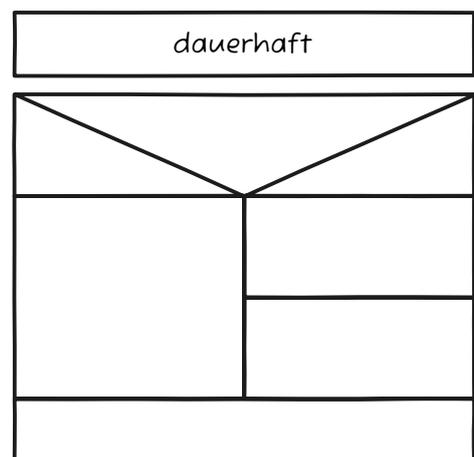
---

---

d) Vervollständige das Struktogramm anhand des Pseudocodes:

Prüfe dauerhaft:  
Wenn die aktuelle y-Position kleiner als 4 ist,  
dann wird der Regentropfen 1 Position nach  
unten bewegt.  
Ansonsten wird die y-Position wieder auf 0  
gesetzt und eine zufällige Position auf der  
x-Achse definiert.

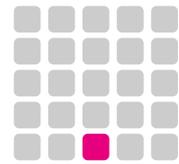
Füge eine Pause hinzu, um das Tempo  
einzustellen.



e) Programmiere anhand des Struktogramms.

## 2 Eimer Steuerung

- Erstelle einen Sprite mit dem Namen Eimer und positioniere ihn mittig ganz unten in der untersten Reihe der LED-Matrix.
- Programmiere den Knopf A so, dass der Eimer sich nach links bewegt.
- Programmiere den Knopf B so, dass der Eimer sich nach rechts bewegt.



## 3 Starteinstellungen

Vervollständige den Spielstart:



- Setze den Spielstand auf 0.
- Setze die Anzahl Leben auf 5.

## 4 Kollisionsabfrage und Auswertung

Erweitere dein Programm um folgende Logik:

Wenn sich die Sprites Eimer und Wasser berühren, wird der Spielstand um einen Punkt erhöht und die RGB-LED leuchten grün, ansonsten werden die Leben um einen Punkt reduziert und die RGB-LED leuchten rot.

### Extra

Was würde das Spiel einfacher oder schwieriger machen?

An einer Stelle im Programmcode kann das Spiel über eine Zahl in der Schwierigkeit verändert werden. Weißt du, an welcher Stelle?

## 2 Streicheln

Dein Haustier liebt es, gestreichelt zu werden. Berühre den Pin P1, um es zu streicheln. Wird dein Haustier gestreichelt, blinkt ein Herz auf der LED-Matrix. Je mehr es gestreichelt wird, desto schneller ist die Blinkfrequenz. Wurde es genug gestreichelt, erscheint das Herz permanent auf der LED-Matrix. Berührst du es an Pin P2, wird es ärgerlich und die Blinkfrequenz verlangsamt sich.

**Eingabe:** Pin P1 und Pin P2

**Verarbeitung:** Vergleich mit Schwellenwerten und Berechnung einer Pausenlänge

**Ausgabe:** LED-Matrix

Gehe Schritt für Schritt vor:

a) Erstelle eine Variable mit dem Namen **streicheln**.

wenn Pin P1 berührt

b) Verändere den Wert der Variable **streicheln** anhand des Struktogramms.

**streicheln** + 1

c) Überprüfe im Debugger. 

wenn Pin P2 berührt

d) Erstelle eine dauerhaft blinkende Herz-Animation.

**streicheln** - 1

e) Füge Pausen hinzu.

f) Erstelle eine weitere Variable mit dem Namen **tempo** und ersetze die Angaben der Pausen durch die Variable **tempo**.

g) Das Tempo verändert sich. Schau in die Tabelle und stelle eine Gleichung auf:



<b>streicheln</b>	<b>tempo</b>
0	1000
1	800
2	600
3	400
4	200
5	0

h) Entferne das Blinken, sobald das Tempo kleiner oder gleich 0 ist.

## 5 Lichtschalter

**Aufgabe:** Schaltet die LED-Matrix durch ein Klatschen an und durch ein weiteres Klatschen wieder aus.

Wichtige Blöcke:

Lautstärke

**Eingabe:** Mikrofon

**Verarbeitung:** Auswertung gemessener Werte (Lautstärke)

**Ausgabe:** LED-Matrix und RGB-LED

Geht Schritt für Schritt vor:

a) Erstellt ein Struktogramm anhand folgender Logik.

Beim Start:

- Setze die Variable **schalter** auf falsch.
- Licht (LED-Matrix) ist aus.

Prüfe dauerhaft:

- Wenn geklatscht wird, prüft, ob das Licht an oder aus ist.
- Ob das Licht an oder aus ist, wird in der Variablen **schalter** gespeichert.
- Wenn geklatscht wird und das Licht ist aus, schaltet es an.
- Wenn geklatscht wird und das Licht ist an, schaltet es aus.

b) Programmiert anhand des Struktogramms.

**Hinweis:** Geschafft ist die Aufgabe, wenn das Licht an und wieder ausgeschaltet wurde.



## 6 Binärcode

**Aufgabe:** Schreibe die Zahl 36 im Binärcode.

Gib über die Tasten den Binärcode ein: Knopf A = 0 und Knopf B = 1.

Schließe deine Eingabe mit dem gleichzeitigen Drücken der Tasten A + B ab.

a) Vervollständigt die Tabelle.

Dezimal- zahlen	Zweierpotenzen							Binärcode
	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
	64	32	16	8	4	2	1	
12				1	1	0	0	$8 + 4 \rightarrow 1100$
24			1	1	0	0	0	$16 + 8 \rightarrow 11000$
76	1	0	0	1	1	0	0	$64 + 8 + 4 \rightarrow 1001100$
36								

**Wichtige Blöcke:**

" "

verbinde

"Hallo "

"Welt "

- +

vergleiche

"this "

mit

" "

**Eingabe:** Taste A und Taste B

**Verarbeitung:** Speicherung der Tastenkombination, Vergleich Zeichenketten

**Ausgabe:** LED-Matrix und RGB-LED

Gehe Schritt für Schritt vor.

b) Setze den Binärcode der Zahl 36 als Wert der Variable **ziel**.

c) Starte mit einem leeren String in der Variable **code**.

d) Wird die Taste A gedrückt, verbinde den aktuellen Wert der Variable **code** mit einer „0“.

e) Verwende die gleiche Logik für die Taste B.

f) Werden die Tasten A+B gleichzeitig gedrückt, vergleiche die Variable **ziel** mit der Variable **code**.

g) Sind die Werte der Variablen identisch, ist die Aufgabe gelöst.

h) Ist die Bedingung nicht erfüllt, leere die Variable **code**, sodass eine erneute Eingabe möglich ist.

# Fahrbewegungen

Die Fahrbewegung wird bestimmt über die Bewegungsrichtung, die Geschwindigkeit und die Dauer der Bewegung.

- 1 Schau dir die Programmcodes genau an und beschreibe was passiert.

```

beim Start
  Motor beide Richtung vorwärts Tempo 200
  pausiere (ms) 500
  Motor links Richtung vorwärts Tempo 50
  Motor rechts anhalten
  
```

---



---



---



---



---



---



---



---



---



---

```

beim Start
  Motor beide Richtung vorwärts Tempo 200
  pausiere (ms) 500
  Motor links Richtung vorwärts Tempo 50
  Motor rechts Richtung rückwärts Tempo 50
  
```

- 2 Passe die Parameter **Tempo** und **Zeit** so an, dass das MotionKit eine 90° Kurve fährt.
- 3 Erstelle ein Programm, dass den Calliope mini ein Quadrat fahren lässt. Gehe Schritt für Schritt vor:

a) Unterteile die Strecke in einzelne Einheiten und kennzeichne Wiederholungen.

