



Startvideo

3.1 Maulwurfspiel: Das Spiel



Hinweis

Zwischen den "Zeige Bild"-Blöcken muss eine Pause von 500 Millisekunden eingefügt werden.

Info

Das Maulwurfspiel heißt eigentlich **Whac-A-Mole** und wurde 1976 in den USA erfunden. Es wurde sogar zum Patent angemeldet.

Start

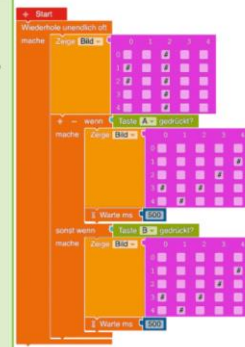
- Gib lab.open-roberta.org in deinen Browser ein. → Dort kannst du programmieren.
- Wähle für das System Calliope mini aus.

Aufgabe

Wir programmieren heute das Maulwurfspiel. Es geht darum, den Maulwurf so schnell wie möglich zu fangen, wenn er aus dem Loch schaut.

Auf dem Calliope wird das Auftauchen durch die LEDs dargestellt. Mit den Tasten A und B fangen wir den Maulwurf. Die ersten Schritte:

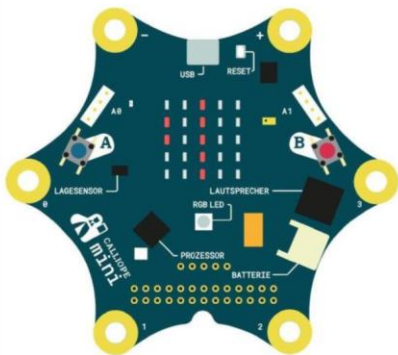
- Lass zwei LEDs auf der linken Seite des LED-Feldes und die Mittellinie leuchten (s. Abbildung rechts). Benutze dafür **Zeige Bild** aus Aktion und markiere die Felder, um die LEDs anzuschalten.
- Wenn **Taste A gedrückt** wird, soll mit **Zeige Bild** ein Haken angezeigt werden. Wenn **Taste B gedrückt** wird, soll ebenfalls ein Haken angezeigt werden.



Probier's aus!

Noch ist der Maulwurf immer links. Egal welchen Knopf man drückt: man bekommt immer das Häkchen angezeigt. Das ist noch nicht so spannend. Aber es ist wichtig, seine Arbeit in kleine Schritte aufzuteilen und immer zu testen, ob das Programm schon funktioniert. Klappt alles? Super, dann geht es weiter mit der nächsten Karte.

3.2 Maulwurfspiel: Zufall



Hinweis

Den Block **ganzzahliger Zufallswert** findest du bei Mathematik unter den **erweiterten Funktionen**. Die erweiterten Funktionen erreichst du, wenn du auf den kleinen Stern klickst.

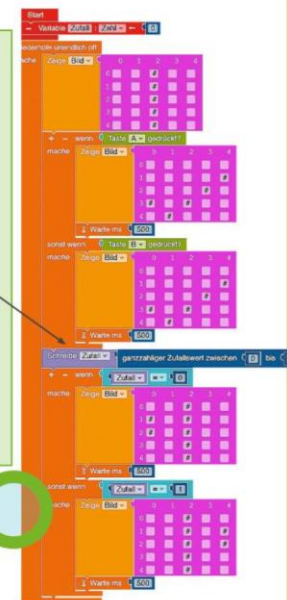
Info

Der Calliope mini kann nur zwei zufällige **Zahlen** auswählen, keine zufälligen Bilder. Mit dem **wenn-mache** Block bringen wir ihm bei, die **Zufallszahl** auf die **Bilder** zu übertragen. Jede Zahl steht für ein Bild.

Aufgabe

Behalte die Blöcke der vorherigen Lernkarte.

- Ändere den ersten **Zeige Bild** Block in einen Balken (ohne Maulwurf) um.
- Lege eine neue **Variable** vom Typ Zahl an und nenne sie **Zufall**.
- Speichere zufällig 0 oder 1 in der neuen **Variable Zufall**. Das geht so:
- Benutze **wenn-mache** und **=** und prüfe, ob **Zufall = 0** ist.
- Wenn Zufall = 0** ist, dann ist der Maulwurf auf der linken Seite, **sonst wenn Zufall = 1** ist, dann ist der Maulwurf auf der rechten Seite. Rechts siehst du, wie das aussieht.
- Pausiere NACH** dem "wenn-mache"-Block für eine halbe Sekunde (= 500 ms), bevor ein neuer Maulwurf auftaucht.



Probier's aus!

Bist du schnell genug, um den Maulwurf zu fangen? Wenn alles klappt, dann geht es weiter mit der nächsten Lernkarte.



3.3 Maulwurfspiel: Variablen lesen APP CAMPS

Aufgabe

Behalte alle Blöcke der vorherigen Lernkarte. Jetzt prüfen wir, ob der Spieler den richtigen Knopf gedrückt hat. Wie war das nochmal?

Wenn Zufall = 0 ist, dann taucht der Maulwurf links auf und wir müssen ihn mit Taste A fangen.

Wenn Zufall = 1 ist, dann taucht der Maulwurf rechts auf und wir müssen ihn mit Taste B fangen.

Zum Prüfen, ob der Maulwurf mit dem Klick gefangen wurde, ändern wir die Knopf-Blöcke so:

Info

Variablen sind toll.

Wir **speichern** einen Wert und können ihn später immer wieder auslesen.

Probier's aus!

Klasse. Jetzt kannst du schon richtig spielen! Lass uns das Spiel noch erweitern und die gefangenen Maulwürfe mitzählen. Wie das geht, siehst du auf der nächsten Lernkarte.

< Basiskurs NEPO />
www.appcamps.de

3.4a Maulwurfspiel: Spielstand APP CAMPS

Aufgabe

Wir wollen zählen, wie oft wir den Maulwurf erwischt haben. Das können wir mit einer eigenen Variable machen.

- Füge eine weitere Variable ein und nenne sie *Spielstand*.
- Wenn der Maulwurf links erscheint und dabei Taste A und **nicht** Taste B gedrückt wird, soll der Spielstand um 1 erhöht werden und es soll ein Sound ertönen (Erweiterte Funktionen > Klang > Spiele Frequenz; Frequenz = 440). Ansonsten soll das Bild mit dem Kreuz angezeigt werden. Dasselbe gilt für Taste B, wenn der Maulwurf rechts erscheint (Taste B gedrückt und **nicht** Taste A gedrückt).

Dazu führen wir zwei **Funktionen** **pruefeLinks** und **pruefeRechts** ein. Wir lagern alle Aufgaben, die ausgeführt werden sollen, wenn der Maulwurf **links** erscheint, in die Funktion **pruefeLinks** aus. Wir lagern alle Aufgaben, die ausgeführt werden sollen, wenn der Maulwurf **rechts** erscheint, in die Funktion **pruefeRechts** aus.

- Zeige den **Spielstand** an, wenn der Calliope geschüttelt wird für 500 ms.

Die Lösungen findest du auf der nächsten Seite.

Info

Mit **Funktionen** kannst du dein Programm übersichtlicher gestalten.

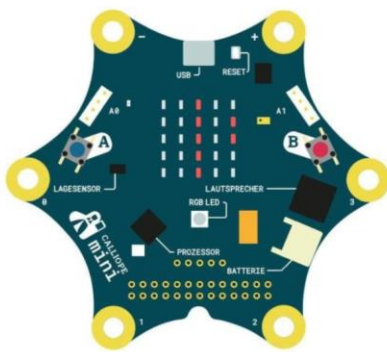
Jede programmierte Funktion steht als eigener Programmtext zur Verfügung. Auf Funktionen kann man vom Hauptprogramm zugreifen, indem man die Funktionen **aufruft**, d.h. man erwähnt den Namen der Funktion im Hauptprogramm.

Mit Funktionen kann man also das Hauptprogramm in logische "Einzelprogramme" gliedern.

< Basiskurs NEPO />
www.appcamps.de



3.4b Maulwurfspiel: Spielstand



Probier's aus!

Wow! Jetzt hast du ein richtiges Spiel programmiert - sogar mit Spielstand. Um das Spiel noch etwas komplizierter zu machen, hier eine Idee: Erweitere das Spiel so, dass auch zwei Maulwürfe gleichzeitig auftauchen können, die man mit A und B gleichzeitig fangen muss. Hast du eine Idee, wie das geht? Auf der nächsten Lernkarte findest du eine Lösung.

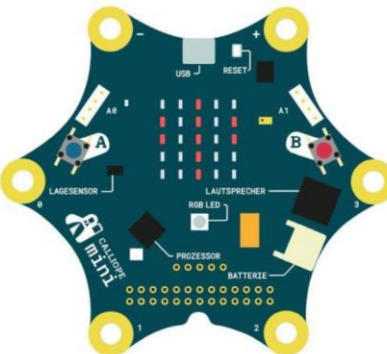
Hier findest du die Lösung:

```

+ Start
- Variable Zufall: Zahl = 0
- Variable Spielstand: Zahl = 0

Wiederhole unendlich oft
mache
+ wenn gib geschüttelt Lage
mache
+ Zeige Zeichen = Spielstand
+ Warte ms = 500
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
3 f f f f f
+ Warte ms = 500
Schreibe Zufall = ganzzahliger Zufallswert zwischen 0 bis 1
+ wenn Zufall = 0
mache
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
+ wenn Zufall = 1
mache
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
+ prüfeLinks
+ prüfeRechts
    
```

3.5 Maulwurfspiel: Erweiterung



Aufgabe

Wir erweitern das Spiel so, dass auch zwei Maulwürfe gleichzeitig auftauchen können, die man mit **A** und **B** fangen muss.

Du musst ein paar Stellen anpassen und einige Blöcke ergänzen.

Versuch es erstmal selbst. Unten findest du die Lösung, mit den angepassten Stellen und ergänzten Blöcken.

```

+ Start
- Variable Zufall: Zahl = 0
- Variable Spielstand: Zahl = 0

Wiederhole unendlich oft
mache
+ wenn gib geschüttelt Lage
mache
+ Zeige Zeichen = Spielstand
+ Warte ms = 500
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
Schreibe Zufall = ganzzahliger Zufallswert zwischen 0 bis 1
+ wenn Zufall = 0
mache
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
+ wenn Zufall = 1
mache
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
+ prüfeLinks
+ prüfeRechts
+ prüfeBeide
+ wenn Taste A gedrückt? und Taste B gedrückt?
mache
Schreibe Spielstand = 
Spielstand + 1
+ Spiele Frequenz Hz = 440
Dauer ms = 500
sonst
Zeige Bild = 
0 0 1 2 3 4
1 f f f f f
2 f f f f f
+ Warte ms = 500
    
```

Info

Nun gibt es drei Möglichkeiten:

1. Maulwurf ist links (Zufall = 0)
2. Maulwurf ist rechts (Zufall = 1)
3. Zwei Maulwürfe sind da (Zufall = 2)

Daher musst du eine zufällige Zahl zwischen 0 und 2 wählen.

Glückwunsch!

Dein Spiel ist fertig! Toll. Stoppe doch mal die Zeit und schau, wie oft du den Maulwurf in einer Minute fangen kannst. Hast du Ideen, wie du das Spiel noch erweitern kannst? Viel Spaß dabei!



3.6 Maulwurfspiel: Bonuskarte

Schon alles erledigt?

Du hast heute Schritt für Schritt dein erstes richtiges Spiel entwickelt. Probiere nun selbst aus, was möglich ist. Am Ende der Stunde kannst du den anderen in deiner Klasse zeigen, was du gemacht hast. Du könntest noch weitere Regeln einbauen, zum Beispiel:

- Das Spiel endet nach einer bestimmten Zeit.
- Das Spiel endet, wenn du zehn Punkte hast.
- Das Spiel endet, nachdem 10x ein Maulwurf erschienen ist.
- ...

wiederhole 10 mal
mache

Du kannst diesen Block ausprobieren.

