



Hint

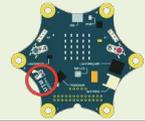
A pause of 500 milliseconds must be inserted between the "Show image" blocks.

Info

The mole game is actually called Whac-A-Mole and was invented in 1976 in the USA. It was even registered as a patent.

Start

- Enter lab.open-roberta.org into a browser. → This is where you can start programming.
- As far as the system goes choose Calliope Calliope and then Calliope 2017.



Task

In this card set we'll program the mole game. It is about catching the mole as fast as possible when he peeks out of the hole. On the Calliope the appearance is represented by the LEDs. With the keys A and B we catch the mole. The first steps:

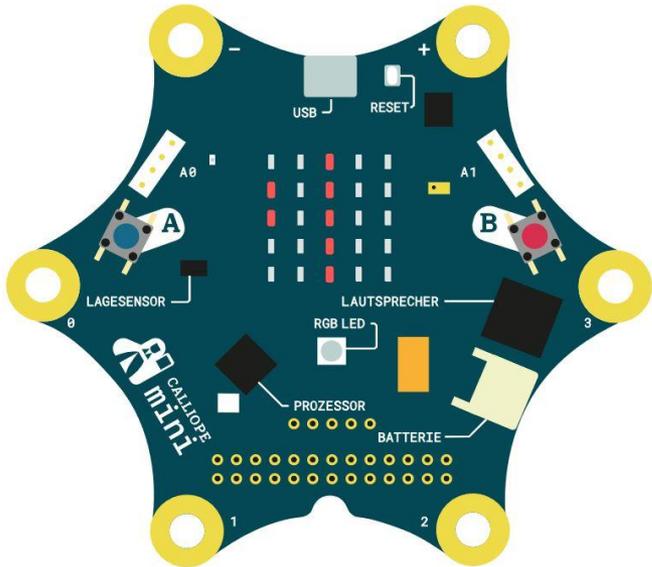
1. Leave two LEDs on the left side of the LED panel and the centerline illuminated (see illustration on the right). Use Show Picture from Action and highlight the fields to turn on the LEDs.
2. If key A is pressed, a check mark is to be displayed with Show image. If button B is pressed, a check mark should also be displayed.

```

+ start
repeat indefinitely
do
  show image [0 1 2 3 4]
  [0] [ ] [ ] [ ] [ ]
  [1] [#] [ ] [ ] [ ] [ ]
  [2] [ ] [ ] [ ] [ ] [ ]
  [3] [ ] [ ] [ ] [ ] [ ]
  [4] [ ] [ ] [ ] [ ] [ ]
  wait ms [500]
  + if button A pressed?
  do
    show image [0 1 2 3 4]
    [0] [ ] [ ] [ ] [ ]
    [1] [ ] [ ] [ ] [ ]
    [2] [ ] [ ] [ ] [ ]
    [3] [#] [ ] [ ] [ ] [ ]
    [4] [ ] [ ] [ ] [ ]
    wait ms [500]
  else if button B pressed?
  do
    show image [0 1 2 3 4]
    [0] [ ] [ ] [ ] [ ]
    [1] [ ] [ ] [ ] [ ]
    [2] [ ] [ ] [ ] [ ]
    [3] [ ] [ ] [ ] [ ]
    [4] [ ] [ ] [ ] [ ]
    wait ms [500]
  
```

Try it out!

The mole is still on the left. No matter which button you press: you will always see the check mark. That's not that exciting yet. But it is important to split up your work into small steps and always test whether it works up to this point. Does everything work? Great, let's continue with the next card.



Hint

The block of **integer random** values can be found in Mathematics under the **extended functions**. The advanced functions can be accessed by clicking on the small star.

Info

The Calliope mini can only select two random **numbers**, no random images. With the **if-do** block we teach it to transfer the random **number** to the **images**. Each number represents one image.

Task

Keep the blocks from the previous flashcard.

1. Change the first **Show Image** block to a bar (without mole).
2. Create a new **variable** of type number and call it **random**.
3. Randomly save 0 or 1 in the new **variable random**. This is how it works:
4. Use if-do and = and check if coincidence = 0.
5. **If random = 0**, then the mole is on the left side, **otherwise if random = 1**, then the mole is on the right side. On the right you can see what this looks like.
6. **Pause AFTER** the "if-do" block for half a second (=500 ms) before a new mole appears.

```

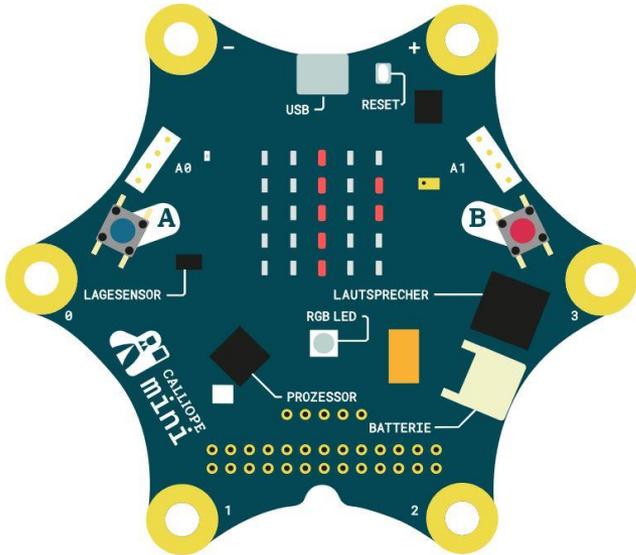
set Random to random integer from 0 to 1
+ if Random = 0
do show image 0 1 2 3 4
wait ms 500
+ if Random = 1
do show image 0 1 2 3 4
wait ms 500
    
```

```

+ start
variable Random Number = 0
repeat indefinitely
do show image 0 1 2 3 4
wait ms 500
do show image 0 1 2 3 4
wait ms 500
else if button A pressed?
do show image 0 1 2 3 4
wait ms 500
else if button B pressed?
do show image 0 1 2 3 4
wait ms 500
do show image 0 1 2 3 4
wait ms 500
set Random to random integer from 0 to 1
do show image 0 1 2 3 4
wait ms 500
else if Random = 1
do show image 0 1 2 3 4
wait ms 500
    
```

Try it out!

Are you fast enough to catch the mole?
If everything works out, then you can continue with the next flashcard.



Task

Keep all blocks from the previous flashcard. Now we'll check if the player pressed the right button. How did that work again?

If **coincidence = 0**, then the mole appears on the **left** and we have to catch it with **A**.

If **coincidence = 1**, then the mole appears on the **right** and we have to catch it with **B**.

To check if the mole was caught with the click, we change the button blocks like this:

```

+ - if button A pressed?
do
+ if Random = 0
do
show image 0 1 2 3 4
0 1 2 3 4
1 # # # #
2 # # # #
3 # # # #
4 # # # #
wait ms 500
else
show image 0 1 2 3 4
0 # # # #
1 # # # #
2 # # # #
3 # # # #
4 # # # #
wait ms 500
else if button B pressed?
    
```

```

else if button B pressed?
do
+ if Random = 0
do
show image 0 1 2 3 4
0 1 2 3 4
1 # # # #
2 # # # #
3 # # # #
4 # # # #
wait ms 500
else
show image 0 1 2 3 4
0 # # # #
1 # # # #
2 # # # #
3 # # # #
4 # # # #
wait ms 500
    
```

Info

Variables are great.

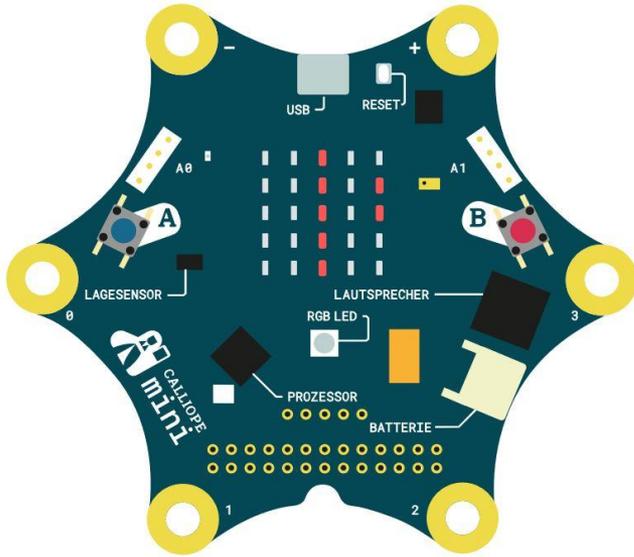
Let's **save** a value so that we can read it again and again later.

Try it out!

Great. Now you can play for real!

Let's expand the game and count the captured moles.

You'll learn how to do so on the next flashcard.



Info

With **functions** you can design your program clearer.

Each programmed function is available as its own program text. You can access functions from the main program by **calling** the functions, i.e. you mention the name of the function in the main program.

With functions you can divide the main program into logical "single programs".

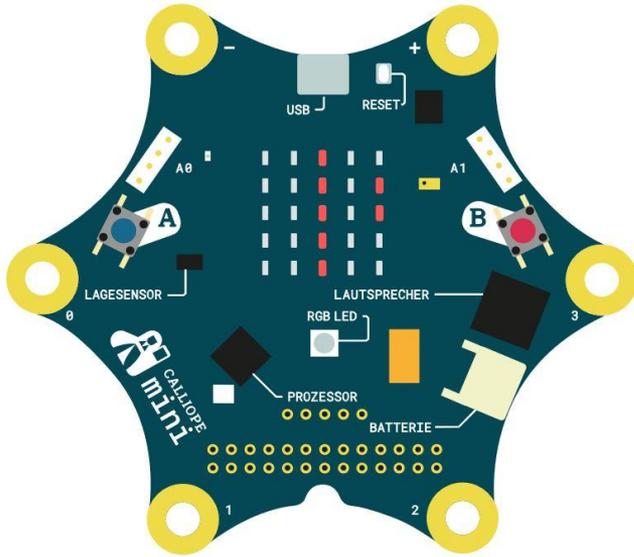
Task

We want to count how many times we caught the mole. We can do so with our own variable.

1. Add another variable and call it *score*.
2. If the mole appears on the left and button A and "not"-button B are pressed, the score is to be increased by 1 and a sound is to be heard (Advanced Functions > Sound > Game Frequency; Frequency = 440). Otherwise the picture with the cross should be displayed. The same applies to button B if the mole appears on the right (button B pressed and "not"-button A pressed).
3. For this purpose we introduce two functions **checkLeft** and **checkRight**. We store all tasks to be performed when the mole appears on the left in the function **checkLeft**. We outsource all tasks to be performed when the mole appears right to the **checkRight** function.
4. Display the score for 500 ms when the Calliope is shaken.

You will find the solutions on the next page.

3.4b Mole game: score



Probier's aus!

Wow! You've now programmed a real game - even with a score.

To make the game even more complicated, here's an idea:

Expand the game so that two moles can appear at the same time, which you have to catch with A and B at the same time. Do you have any idea how to do that? You'll find a solution on the next flashcard.

You can find the solution here

```
+ start
- variable Random : Number ← 0
- variable Gamescore : Number ← 0

repeat indefinitely
do
+ if get shaking gesture
do
show text Gamescore

show image


|   |  |  |   |  |
|---|--|--|---|--|
| 0 |  |  | # |  |
| 1 |  |  |   |  |
| 2 |  |  | # |  |
| 3 |  |  | # |  |
| 4 |  |  | # |  |



wait ms 500
set Random to random integer from 0 to 1

+ if Random = 0
do
show image


|   |   |  |   |  |
|---|---|--|---|--|
| 0 |   |  | # |  |
| 1 | # |  |   |  |
| 2 | # |  |   |  |
| 3 |   |  | # |  |
| 4 |   |  | # |  |



wait ms 500
else if Random = 1
do
show image


|   |  |  |   |   |
|---|--|--|---|---|
| 0 |  |  |   |   |
| 1 |  |  | # | # |
| 2 |  |  | # | # |
| 3 |  |  | # |   |
| 4 |  |  | # |   |



wait ms 500

+ checkLeft
+ if button A pressed?
do
+ if Random = 0
do
set Random to Random + 1
play frequency Hz 440
duration ms 500
else
show image


|   |   |   |   |   |
|---|---|---|---|---|
| 0 | # |   |   | # |
| 1 |   | # |   |   |
| 2 |   |   | # | # |
| 3 |   | # |   |   |
| 4 | # |   |   | # |



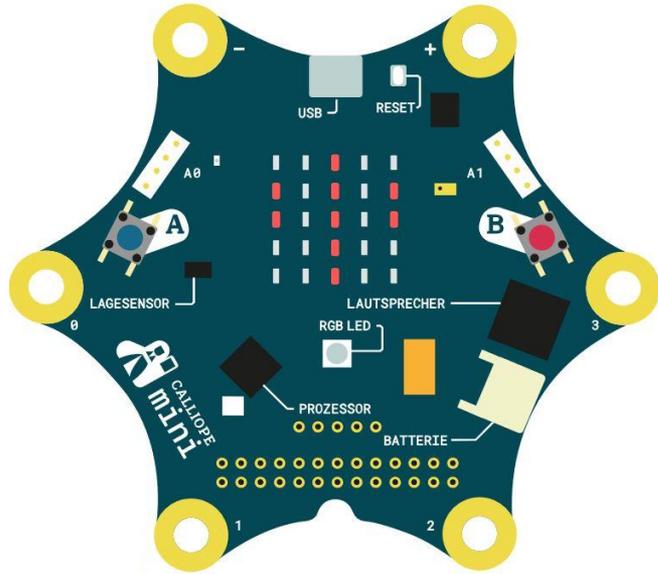
wait ms 500

+ checkRight
+ if button B pressed?
do
+ if Random = 1
do
set Random to Random + 1
play frequency Hz 440
duration ms 500
else
show image


|   |   |   |   |   |
|---|---|---|---|---|
| 0 | # |   |   | # |
| 1 |   | # |   |   |
| 2 |   |   | # | # |
| 3 |   | # |   |   |
| 4 | # |   |   | # |



wait ms 500
```



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.

Info

Now there are three possibilities:

1. Mole on the **left** (random = 0)
2. Mole on the **right** (random = 1)
3. There are two moles (random = 2)

Therefore you have to choose a random number between 0 and 2.

```

+ start
- variable Random : Number -> 0
- variable Gamescore : Number -> 0

repeat indefinitely
do
+ if get shaking gesture
do show text Gamescore

show image


|   |  |   |  |  |
|---|--|---|--|--|
| 0 |  | # |  |  |
| 1 |  | # |  |  |
| 2 |  | # |  |  |
| 3 |  | # |  |  |
| 4 |  | # |  |  |



wait ms 500
set Random to random integer from 0 to 2

+ if Random = 0
do show image


|   |   |   |  |  |
|---|---|---|--|--|
| 0 |   | # |  |  |
| 1 | # |   |  |  |
| 2 | # |   |  |  |
| 3 | # |   |  |  |
| 4 | # |   |  |  |



wait ms 500

else if Random = 1
do show image


|   |   |   |  |  |
|---|---|---|--|--|
| 0 |   | # |  |  |
| 1 | # |   |  |  |
| 2 | # |   |  |  |
| 3 | # |   |  |  |
| 4 | # |   |  |  |



wait ms 500

else if Random = 2
do show image


|   |   |   |  |  |
|---|---|---|--|--|
| 0 |   | # |  |  |
| 1 | # |   |  |  |
| 2 | # |   |  |  |
| 3 | # |   |  |  |
| 4 | # |   |  |  |



wait ms 500

checkLeft
checkRight
checkBoth
    
```

```

+ checkBoth
+ if button A pressed? and button B pressed?
do
+ if Random = 2
do
set Random to Random + 1
play frequency Hz 440
duration ms 500

else show image


|   |   |  |  |  |
|---|---|--|--|--|
| 0 | # |  |  |  |
| 1 | # |  |  |  |
| 2 | # |  |  |  |
| 3 | # |  |  |  |
| 4 | # |  |  |  |



wait ms 500
    
```

Congrats

Your game is ready! Great. Stop the time and see how often you can catch the mole in a minute.

Do you have any ideas on how to expand the game? Have fun with it!

Everything done yet?

Today you developed your first real game step by step.

Now try out for yourself what is possible. At the end of the lesson you can show the others in your class what you have done.

You could add more rules, for example:

- The game ends after a certain time.
- The game ends after ten points.
- The game ends after 10 moles have appeared.
- ...

