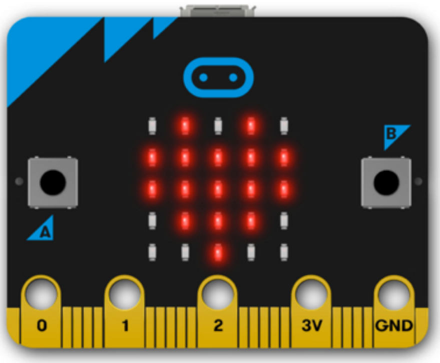
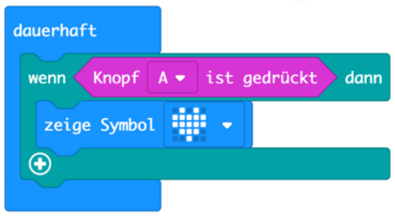


Wenn angeklickt wird
wiederhole fortlaufend
wechsle zum nächsten Kostüm
warte 0.2 Sekunden



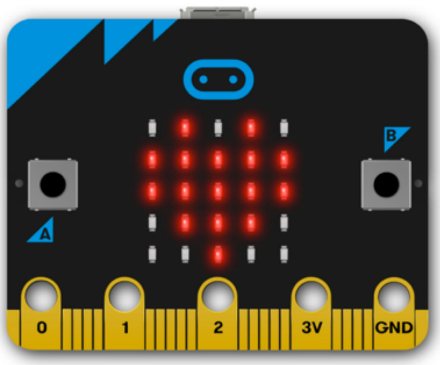
dauerhaft
wenn Knopf A ist gedrückt dann
zeige Symbol

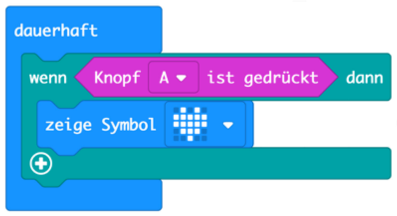
DigiMe Basic Workshop

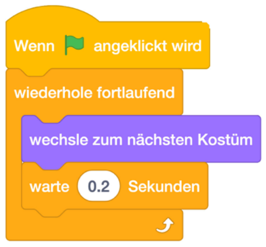
Interreg
EUROPÄISCHE UNION
Österreich-Tschechische Republik
DigiMe
Europäischer Fonds für regionale Entwicklung

Project co-funded by the European Regional Development Fund (ERDF)

DigiMe Basic Workshop







Workshop course

▶ Goal + Benefit for teaching

- Learning of the basics of a programming language and the micro:bit
- Use of suitable examples for various teaching subjects
- Motivation and variety in teaching
- Homework and flipped-classroom scenarios in the simulator

▶ Examples and opportunities for use in teaching

▶ Set-up of the Micro Bit (LEDs, buttons, sensors, ...)

Practical part

▶ Use of the Micro Bit in 3 steps:

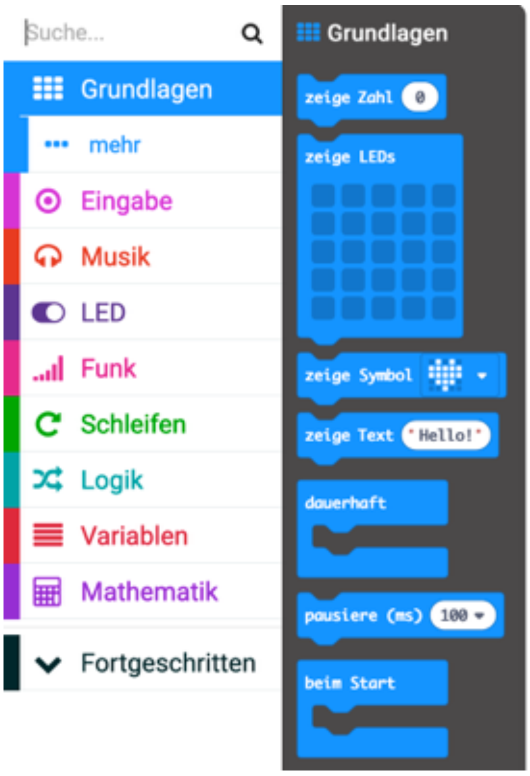
- Connection (Micro Bit - Device)
- Programming
- Transfer of the program

▶ Sharing (passing on) of the program

▶ Getting to know the Makecode programming interface by examples

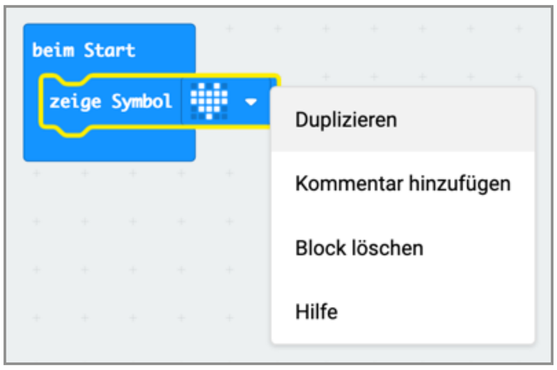
▶ Getting to know the Micro Bit Scratch combination by programming of Flappy Bird

Getting to know the Makecode interface – 1

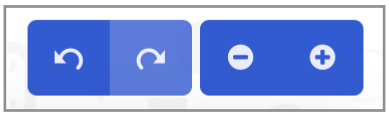


Code blocks are assigned to categories according to different topics. By drag & drop, the blocks can be easily moved to the programming interface on the right side.
Some categories have a sub-category **... more** with additional blocks (not used as often).

Right click on a block to use the following functions:

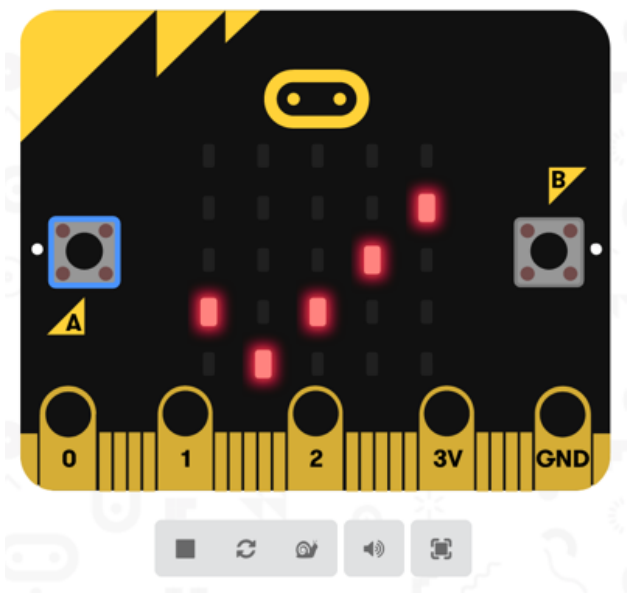


Duplicate enables copying of individual as well as connected code blocks.
For complicated programs, the Add comment function can be used to provide information and keep the code in order.
Blocks (or block combinations) can be deleted with Delete block or simply drawn into the category overview.
If you need any Help, this function will provide you with much needed information.



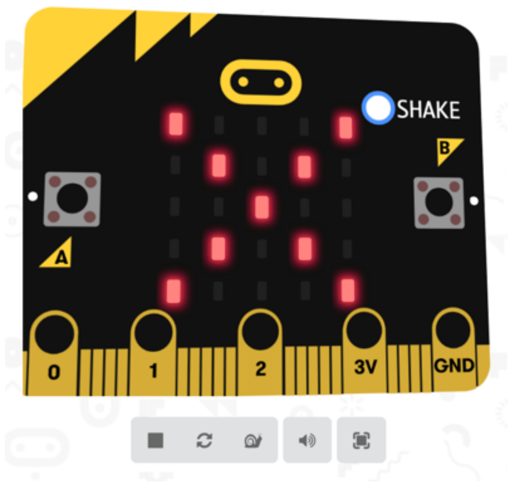
The function to increase and decrease the size of blocks with + and – is particularly useful for lessons with projectors.
Don't worry, steps can be easily undone with the back arrow.

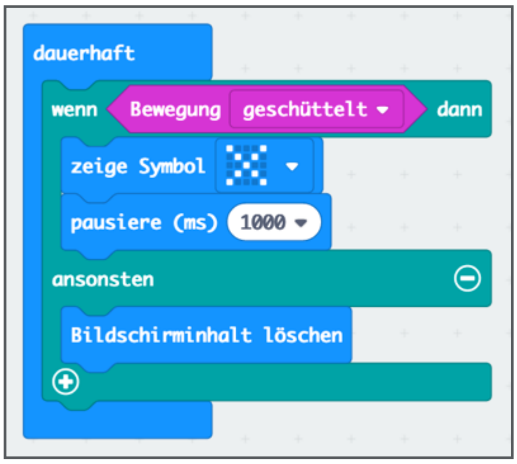
Getting to know the Makecode interface – 2



The simulator enables testing of programs even without a Micro Bit. Besides buttons and the LED matrix (LED), sensors for light, temperature and inclination as well as the radio communication module are available.

These functions are displayed as soon as the respective program blocks are used.





```
dauerhaft
    wenn  Bewegung geschüttelt ▾  dann
        zeige Symbol  ▾
        pausiere (ms) 1000 ▾
    ansonsten
        Bildschirminhalt löschen
```

The share button enables archiving of the program code and sharing it with friends.

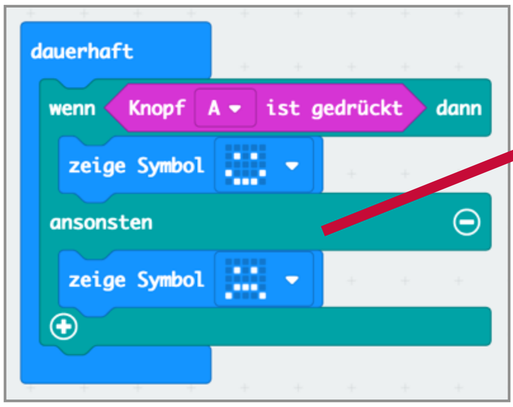
Simply send the link to show others your great work.

Tips for teaching

- The share function can also be used by teachers to assign tasks online. For example, these tasks may include minor errors that need to be found by the students. Or parts of programs may need to be completed or developed.
- This way, homework can also be "handed in" online.

Button function

Assignment:
At the Micro Bit, a sad smiley is displayed if button A is not pressed. If button A is pressed, the smiley smiles for as long as the button is pressed and held.



```
dauerhaft
  wenn Knopf A ist gedrückt dann
    zeige Symbol
  ansonsten
    zeige Symbol
```

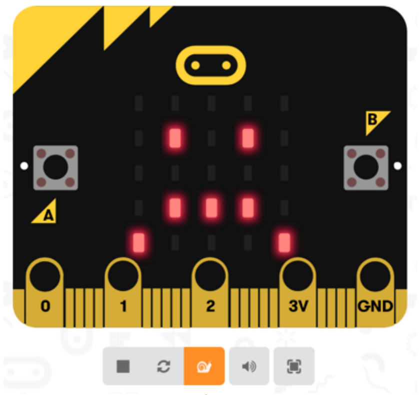
Without the "otherwise" part, the smiley would start and remain smiling after the button is pressed and even after it is released.

https://makecode.microbit.org/_cLTEcUJm1dyH

Please note

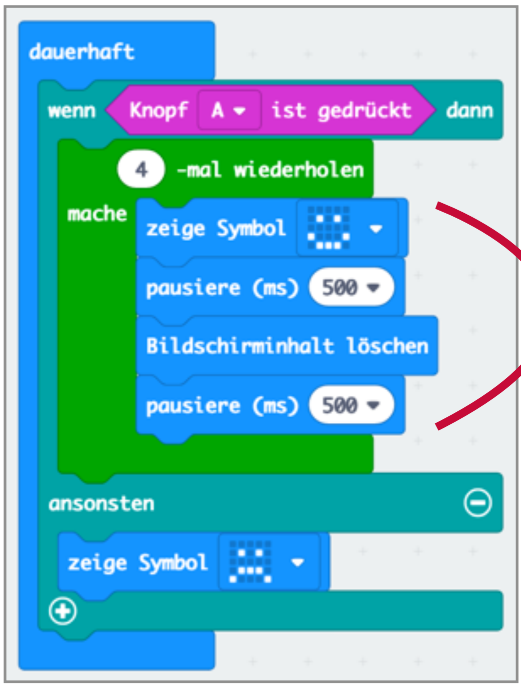
As it takes almost one second for display (in the "otherwise" part), the button must be pressed a bit longer to be recognised.

With the "snail" button in the simulator, the speed of the Micro Bit can be reduced to observe this effect.



Loops

Assignment:
If the button is pressed, the smiley flashes 4x (otherwise identical to



Without a loop, the 4 lines would have to be used four times => 16 lines.

In this case, loops make sure that the code is clearer and less prone to errors.

If the loop repetition is changed, this change only needs to be done once in the head of the loop.

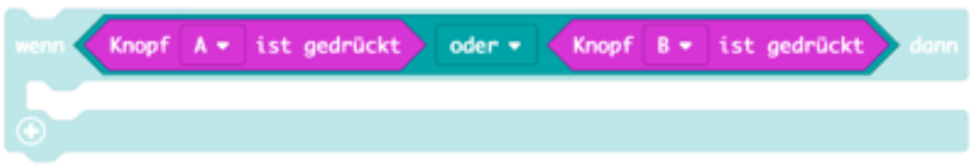
https://makecode.microbit.org/_3id2wociRUbD

Please note

This is the easiest and most common use of loops.
Loops are also used in other assignments described in a different worksheet.

Additional information

For the Micro Bit to react if any of the two buttons is pressed, the "or" block is required.

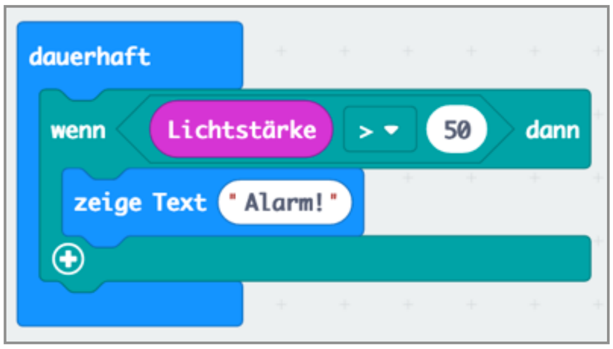


If the Micro Bit is only to react if both buttons are pressed simultaneously, an "and" link is required.

Alarm system (light sensor)

Assignment:
Program an alarm system using the light sensor.
The Micro Bit is inside a dark drawer. If the drawer is opened, light shines on the sensor and the alarm system is triggered => "Alarm" is displayed.

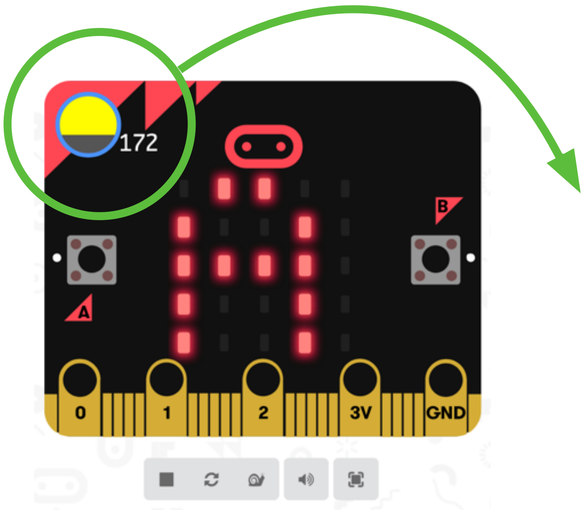
Optional: Instead of the displayed alarm, a sound is played.


```
dauerhaft
    wenn  Lichtstärke  >  50  dann
        zeige Text "Alarm!"
```

The "Light sensor" block can be found in the Eingabe category.

Use the relational operator 0 > 0 from the Logik category to query the light sensor for a specific value.

https://makecode.microbit.org/_DmTLTs100V00



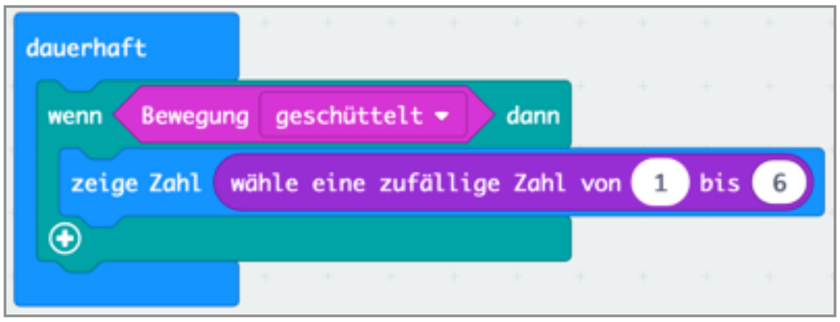
Please note

If the used Micro Bit is not a "real" Micro Bit, the brightness can be changed with the mouse in the simulator. As soon as the value 50 is exceeded, "Alarm!" is displayed.

Optional: In the Musik category, the blocks for playing sounds (can be played in the simulator).
Playing sounds on a speaker directly at the Micro Bit is described in a separate worksheet.

Die (random numbers)

Assignment:
When shaking the Micro Bit, a random number between 1 and 6 is displayed.

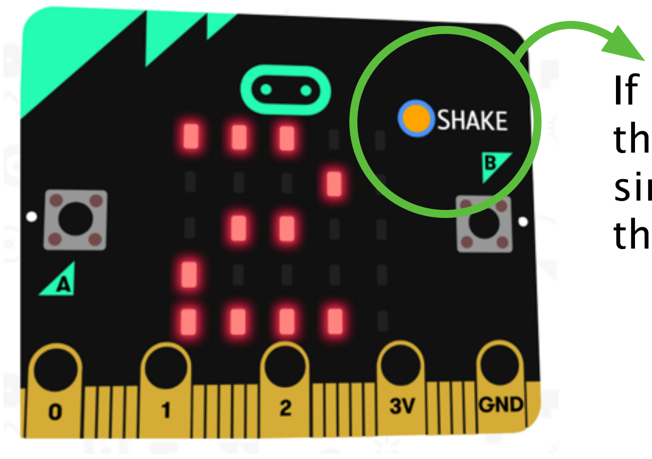


```
dauerhaft
  wenn Bewegung geschüttelt ▾ dann
    zeige Zahl wähle eine zufällige Zahl von 1 bis 6
```

Besides other sensors, the block for the Bewegung geschüttelt ▾ can also be found in the ⊙ Eingabe category.

https://makecode.microbit.org/_HkFbbRJ48Ywa

Please note

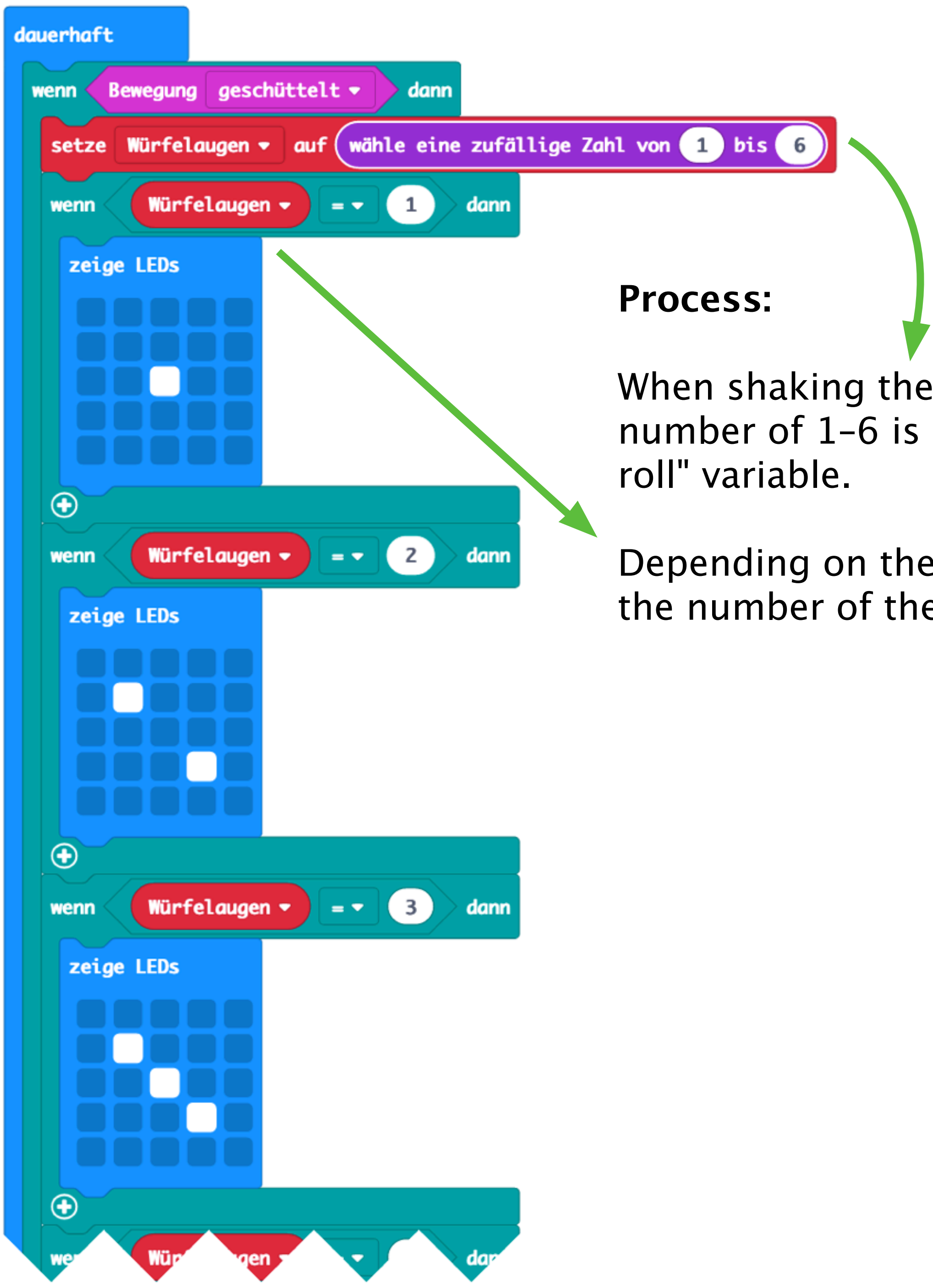


If the used Micro Bit is not a "real" one, the "shaker function" is displayed in the simulator if the respective block is used in the program.

Die (variable)

Assignment:
When shaking the Micro Bit, a random number is displayed.

This assignment can be solved by using an auxiliary variable.
In the Mathematik category, a block for "selection of a random number" with possible specification of the limit of the value range can be found.



```
dauerhaft
  wenn Bewegung geschüttelt dann
    setze Würfelaugen auf wähle eine zufällige Zahl von 1 bis 6
    wenn Würfelaugen = 1 dann
      zeige LEDs
        . . . . .
        . . . . .
        . . # . .
        . . . . .
        . . . . .
    ⊕
    wenn Würfelaugen = 2 dann
      zeige LEDs
        . . . . .
        . # . . .
        . . . . .
        . . . # .
        . . . . .
    ⊕
    wenn Würfelaugen = 3 dann
      zeige LEDs
        . . . . .
        . # . . .
        . . # . .
        . . . # .
        . . . . .
    ⊕
```

Process:

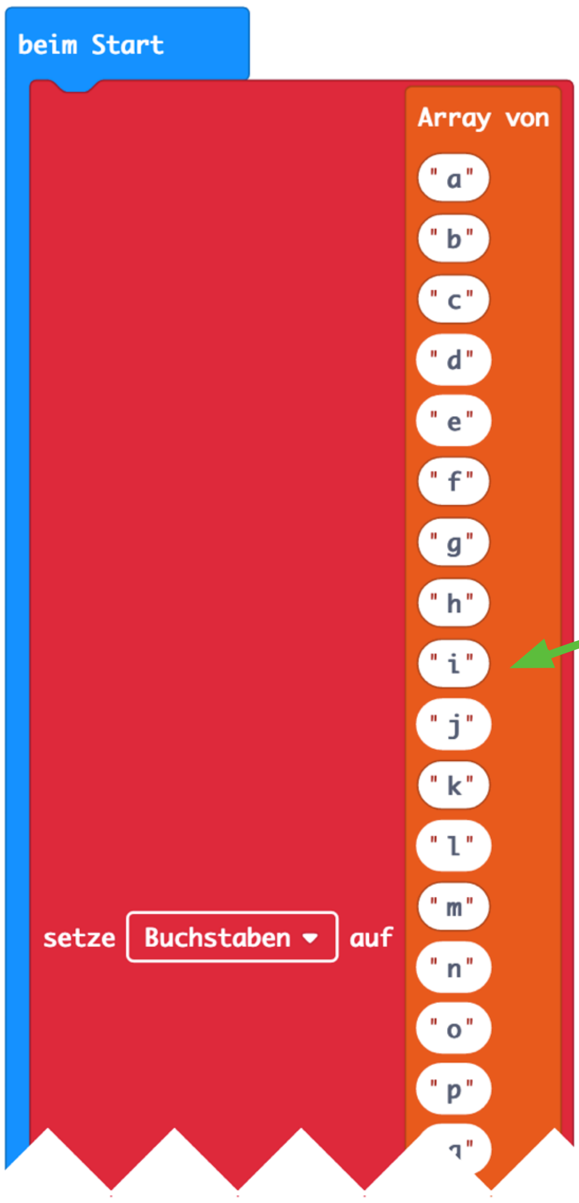
When shaking the Micro Bit, a random number of 1-6 is assigned by the "die roll" variable.

Depending on the random number, the number of the roll is displayed.

https://makecode.microbit.org/_PE9H8vFKV3eH

City – Country – River (array)

Assignment:
If a button is pressed, a random letter is displayed.



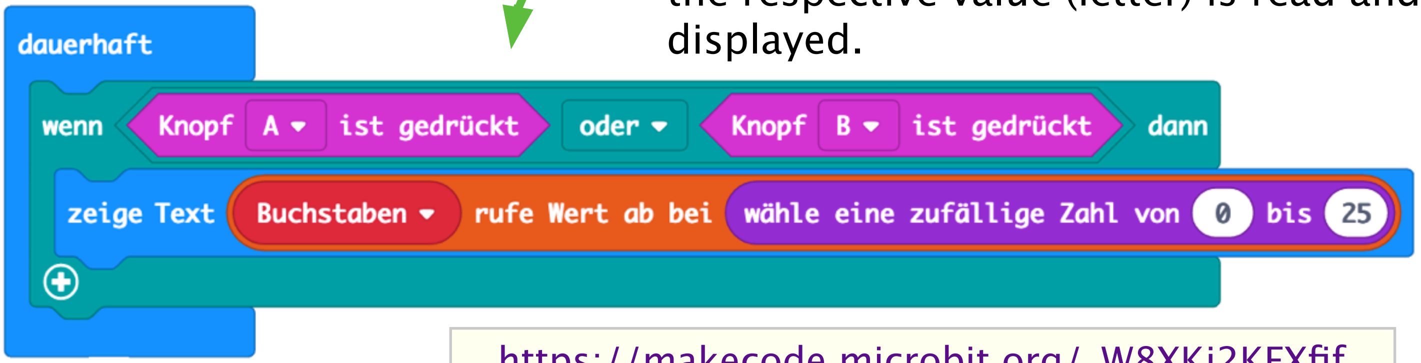
Variable vs Array

An array is a group of multiple variables.

In this case, the array "Letters" with the 26 letters of the alphabet is displayed on start.

Please note
At an array, the numbering of the elements starts at 0 instead of 1.

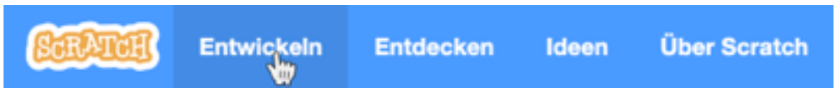
If a button is pressed, a random number between 0 and 25 is generated.
In the "Letters" array, the position of the random number is accessed and the respective value (letter) is read and displayed.

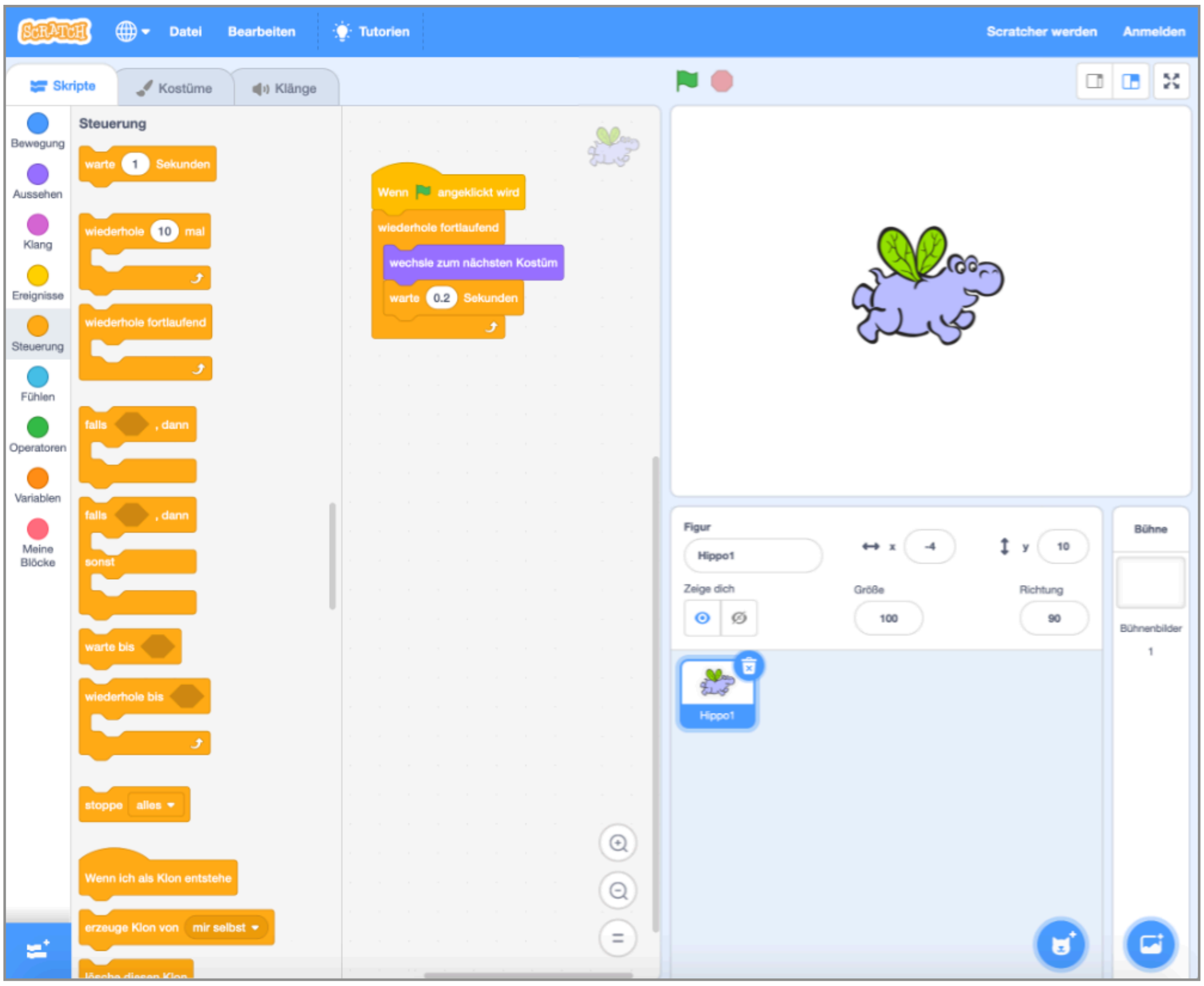

dauerhaft
wenn Knopf A ist gedrückt oder Knopf B ist gedrückt dann
zeige Text Buchstaben rufe Wert ab bei wähle eine zufällige Zahl von 0 bis 25

https://makecode.microbit.org/_W8XKj2KFXfif

Scratch interface

Scratch can be started by clicking on "Create" on the Scratch website https://scratch.mit.edu .





The Scratch interface has 3 columns:

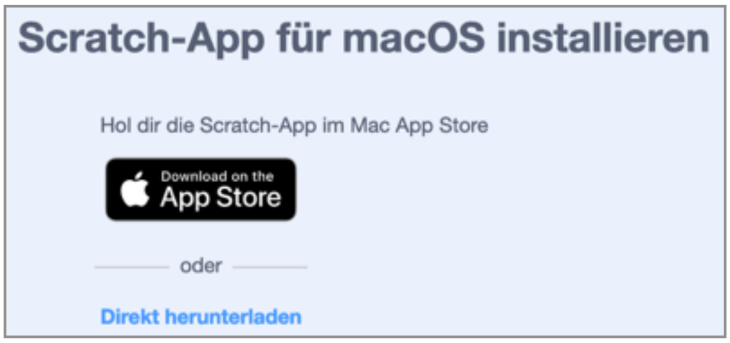
- The left column contains all coding blocks grouped in categories and sorted by colour.
- The column in the middle is the programming environment where the program blocks can be moved by drag & drop and connected similar to the pieces of a jigsaw puzzle.
- The right colour is the stage where the program is visualised together with the figures used in the project and the background of the stage.

Scratch + Micro Bit

In only 2 steps, the Micro Bit can be embedded in the Scratch program to bring games interactively to live by controlling the figures by buttons or the inclination sensor, output of the score at the display and many in many other ways.

1. Download and installation of the Scratch-App
The current Scratch app is available for download from https://scratch.mit.edu/download .

Available for:
- Windows
- macOS
- ChromeOS
- Android

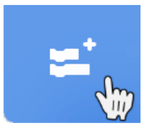

Scratch-App für macOS installieren

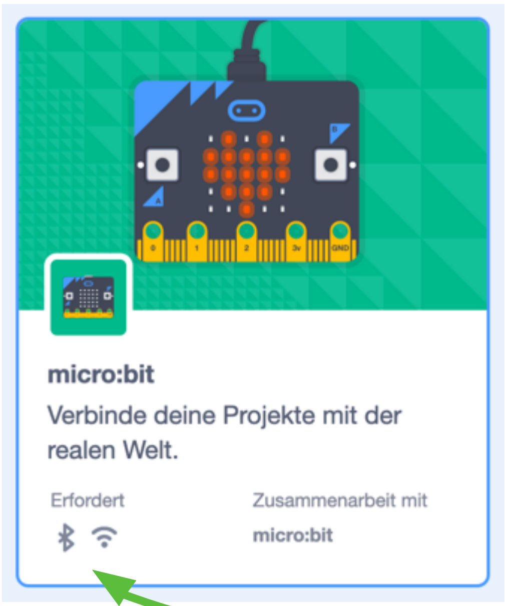
Hol dir die Scratch-App im Mac App Store

oder

Direkt herunterladen

2. After clicking on "Add extension" (bottom left), select the micro:bit extension.




micro:bit

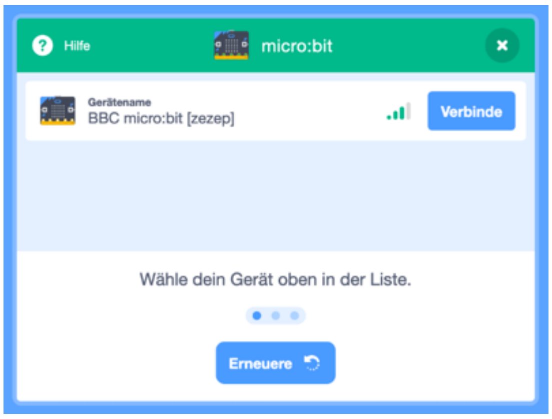
Verbinde deine Projekte mit der realen Welt.

Erfordert

Zusammenarbeit mit micro:bit

Please note
To use the Scratch app, Bluetooth and internet connections are required.

The Micro Bit connected to the PC or tablet is recognised and made available by clicking on "Connect" on the Scratch interface.



Gerätename
BBC micro:bit [zezep]

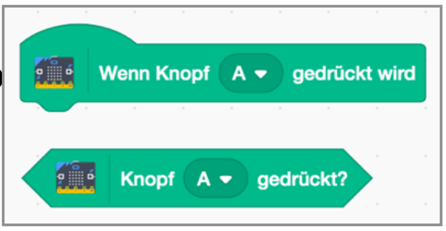
Wähle dein Gerät oben in der Liste.

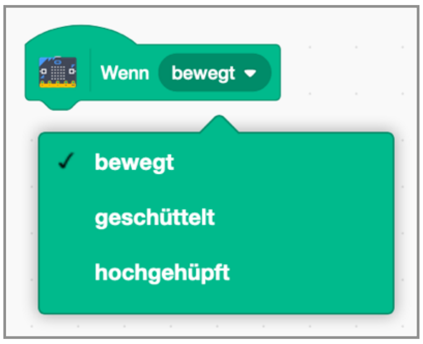
Micro Bit functions in Scratch

Unfortunately, not all functions of the Micro Bit are also available in Scratch.

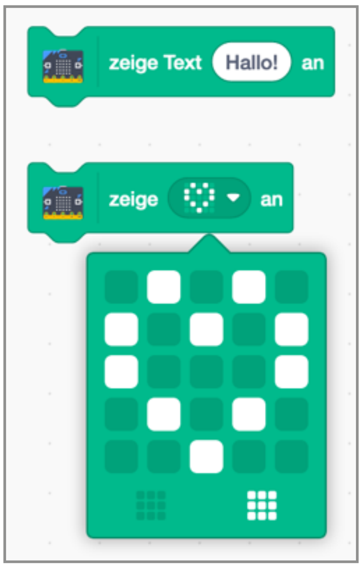
<u>The following functions are available:</u>

Buttons A and B can be queried in two ways.

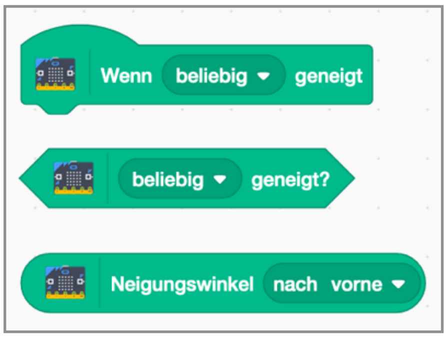




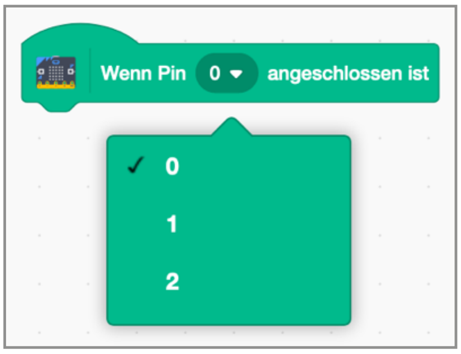
Vibrations can be recognised by the Micro Bit.



Vibrations can be recognised by the Micro Bit.

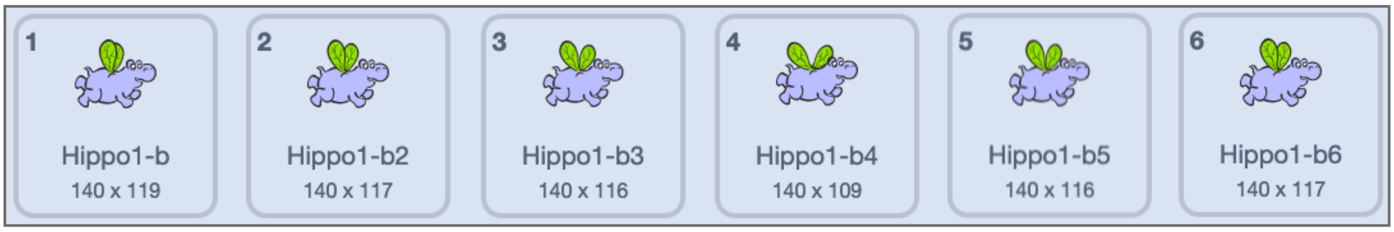


The inclination to the front, back, left and right can be queried as well as specific inclination angles.

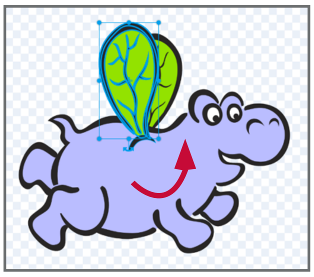


Any connections between Pin0, Pin1 or Pin2 with GND are also recognised by the Micro Bit.

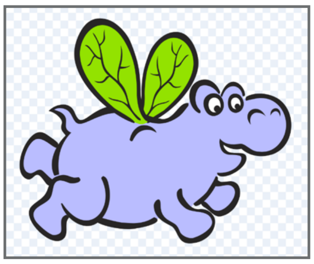
Flappy Bird (Scratch) – Flying motion



To make sure that the flying motion of the Flappy Bird look real, the different wing positions are run through quickly one after the other (just like in a flip book).
[The hippo is an original figure of Scratch]

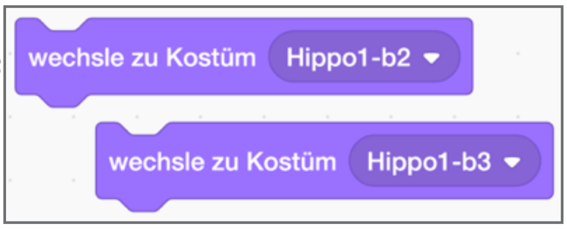


First, copy the first costume in Scratch. Afterwards, select the wing to be turned with the mouse. The wing simply needs to be turned and moved to the desired position.

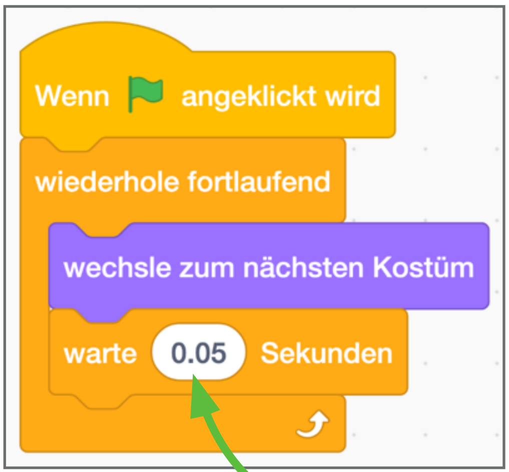


Repeat this for the other wing and the second costume is ready.
The flying motion of the Flappy Bird will become even more realistic if three or four costumes with different wing positions are created.

One way of doing this would be to call the desired costumes one after the other.



```
wechsle zu Kostüm Hippo1-b2
wechsle zu Kostüm Hippo1-b3
```



```
Wenn [Flagge] angeklickt wird
wiederhole fortlaufend
    wechsle zum nächsten Kostüm
    warte 0.05 Sekunden
```

There is an even easier way of doing this:

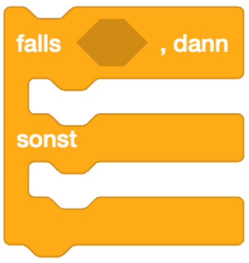
The block `wechsle zum nächsten Kostüm` enables automatic switching to the next costume in an infinite loop. The following break defines the speed for flapping the wing. The longer the break the slower the flying motion of the Flappy Bird.

Please note
Do not forget about the decimal point!

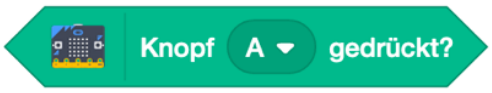
https://scratch.mit.edu/projects/421455884

Flappy Bird (Scratch) – Control

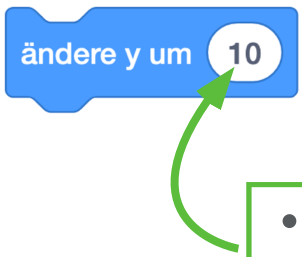
Conditions: Without any interaction, the Flappy Bird is slowly sinking to the ground.
Every time, button A of the Micro Bit is pressed, the "bird" should move up a little bit.



For this, the "if – otherwise" query is needed.

Additionally, this block 

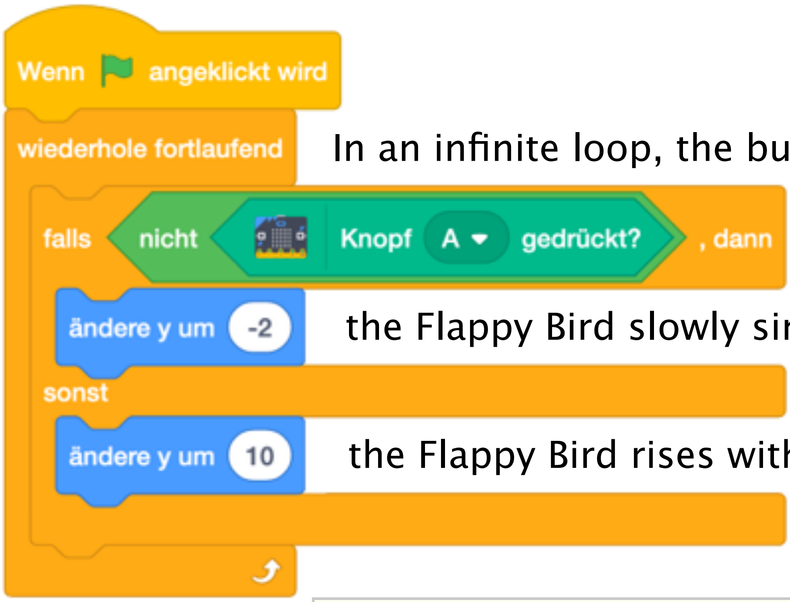
from the micro:bit category is required.



Sinking and rising of the "bird" is achieved by increasing or decreasing the value of the Y coordinate. This concept is known from mathematics in school.

- The value defines the speed of sinking/rising
- The sign defines whether the Flappy bird is rising or

The control block looks as follows:



In an infinite loop, the button is queried.

If button A is not pressed –

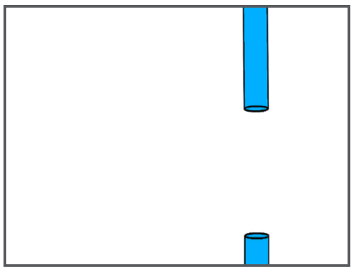
the Flappy Bird slowly sinks to the ground (change y

If the button is pressed (otherwise) –

the Flappy Bird rises with a jump (change y by 10).

https://scratch.mit.edu/projects/421459885

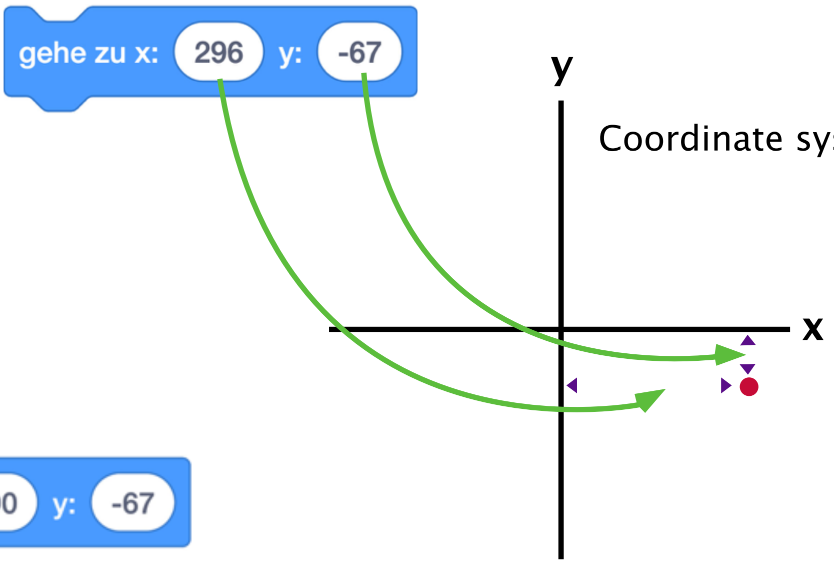
Flappy Bird (Scratch) – Obstacle



Conditions:

The tubes move from the right to the left edge of the screen.
The gap between the top and bottom ends of the tubes must be clearly larger than the Flappy Bird so that it can fly through.

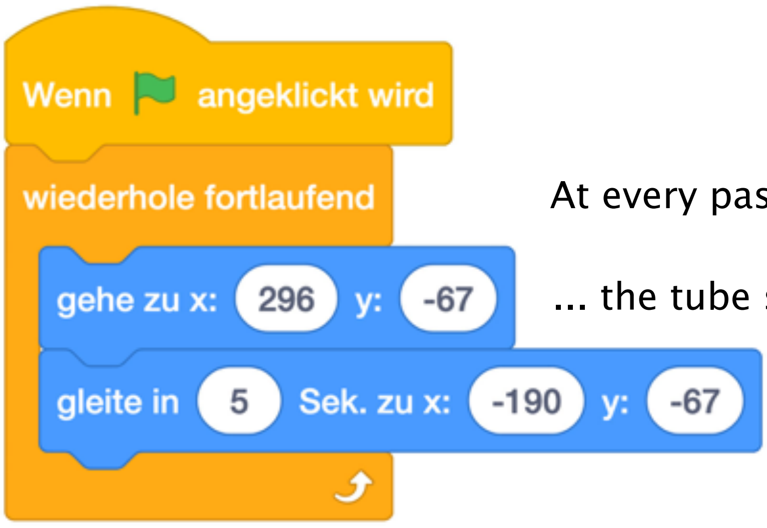
For the tube to appear at the right edge of the screen at every pass, the respective coordinates must be entered in the "go to" block.



Enter the end coordinates of the tube (left edge of the screen) and the time of the motion in the "float" block.

<u>Hint</u>: The shorter the time the quicker the motion of the tube

The control block looks as follows:



```
Wenn [flag] angeklickt wird
wiederhole fortlaufend
    gehe zu x: 296 y: -67
    gleite in 5 Sek. zu x: -190 y: -67
```

At every pass...

... the tube starts at the right edge of the screen.

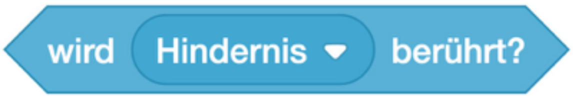
and floats in 5 seconds to the left edge.

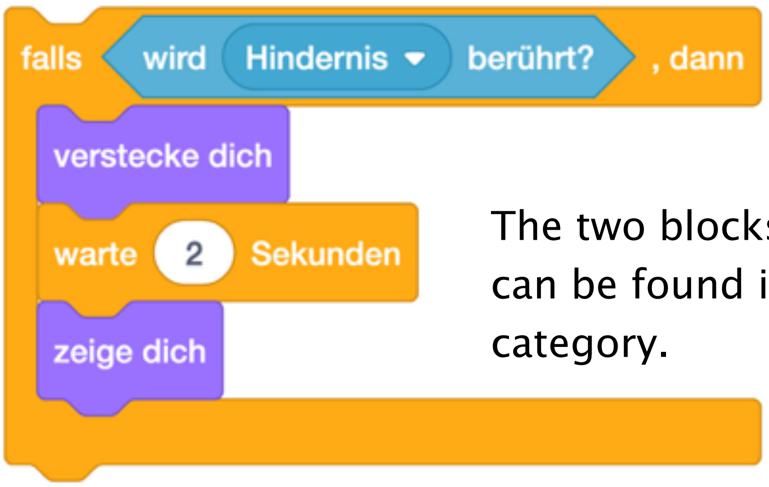
https://scratch.mit.edu/projects/421461139

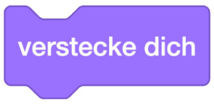
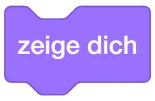
Flappy Bird (Scratch) – Mistake query / 1

Conditions:

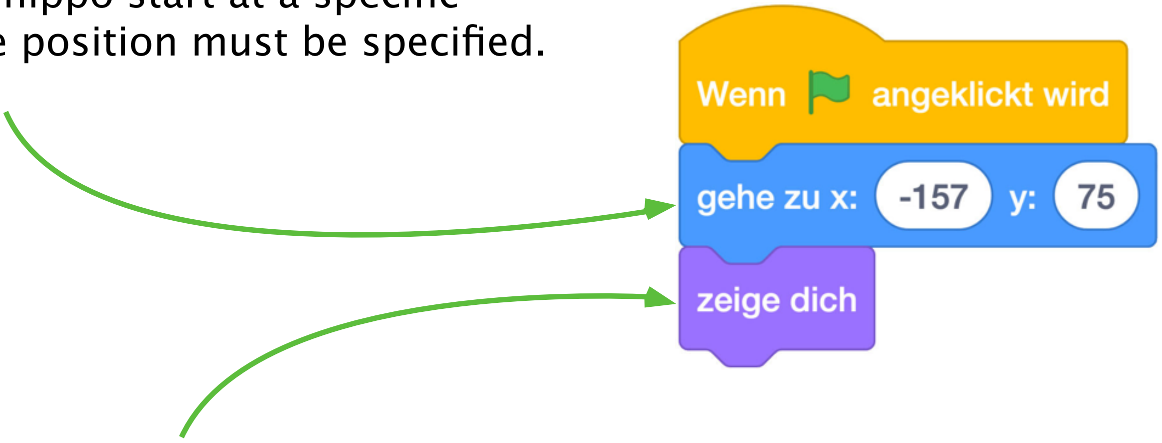
If the hippo touches the tube, it disappears and then reappears after 2 seconds. This is realised with an "if" query for touching in the tube (obstacle).





The two blocks  and  can be found in the "Appearance" category.

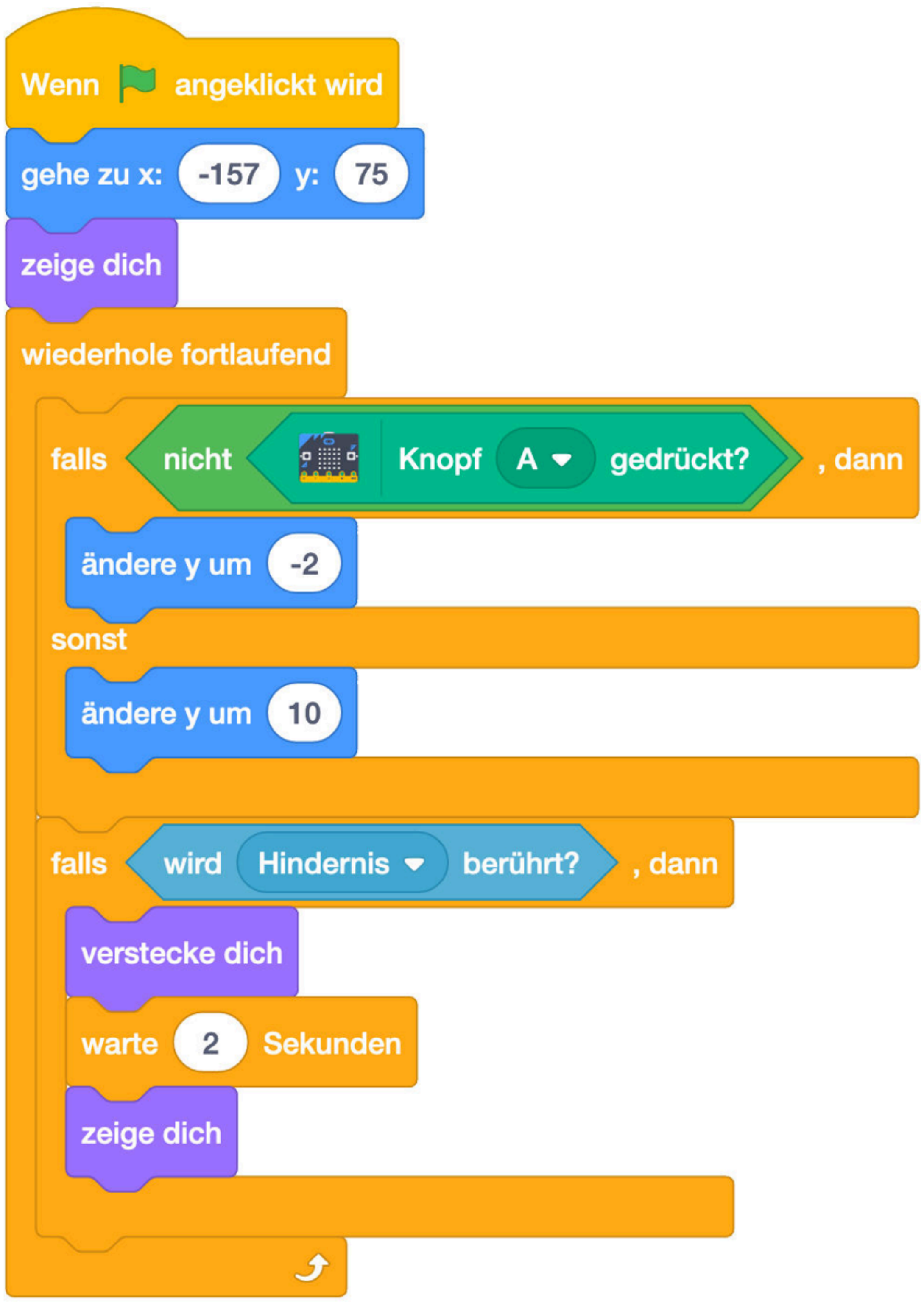
To have the hippo start at a specific position, the position must be specified.



Please note:
If the game is ended while the hippo is not visible, it would also not be visible if the game is restarted (green flag). This exception must be excluded in the beginning by "show".

Flappy Bird (Scratch) – Mistake query / 2

The control block of the tube looks as

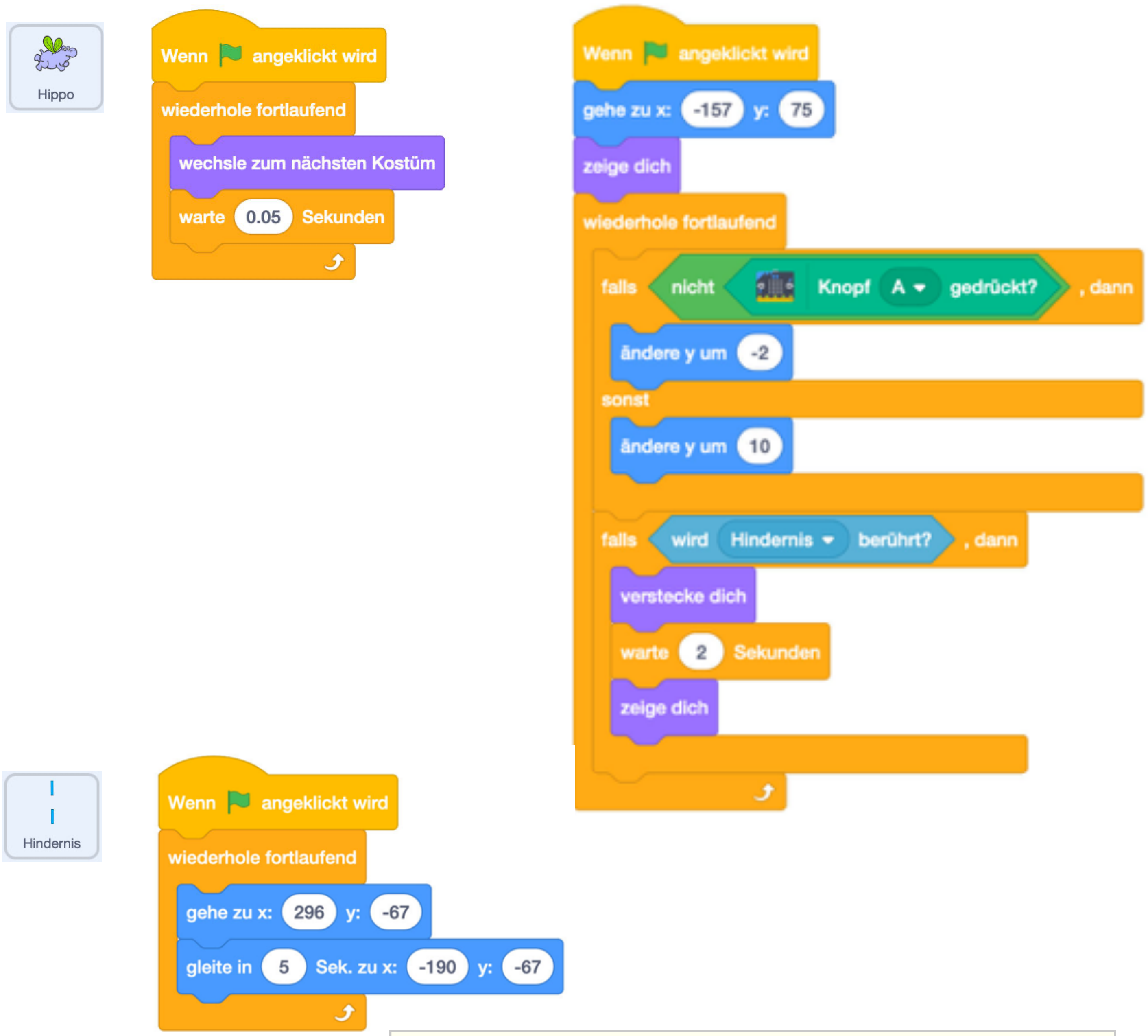


```
Wenn 🏳 angeklickt wird
gehe zu x: -157 y: 75
zeige dich
wiederhole fortlaufend
    falls nicht Knopf A gedrückt? , dann
        ändere y um -2
    sonst
        ändere y um 10
    falls wird Hindernis berührt? , dann
        verstecke dich
        warte 2 Sekunden
        zeige dich
```

https://scratch.mit.edu/projects/423073862

Flappy Bird (Scratch) – Das Spiel

Programming code and link to the game



```
Hippo

Wenn [Flagge] angeklickt wird
wiederhole fortlaufend
    wechsle zum nächsten Kostüm
    warte (0.05) Sekunden

Wenn [Flagge] angeklickt wird
gehe zu x: (-157) y: (75)
zeige dich
wiederhole fortlaufend
    falls <nicht <[micro:bit] Knopf (A) gedrückt?>>, dann
        ändere y um (-2)
    sonst
        ändere y um (10)
    falls <wird (Hindernis) berührt?>, dann
        verstecke dich
        warte (2) Sekunden
        zeige dich

Hindernis

Wenn [Flagge] angeklickt wird
wiederhole fortlaufend
    gehe zu x: (296) y: (-67)
    gleite in (5) Sek. zu x: (-190) y: (-67)
```

https://scratch.mit.edu/projects/423073862

Possible extensions:

- Different backgrounds
- Automatic counting of mistakes
- Increase in speed every 10 seconds
- Reduction of the gap between the tubes every 10 seconds
- And much more