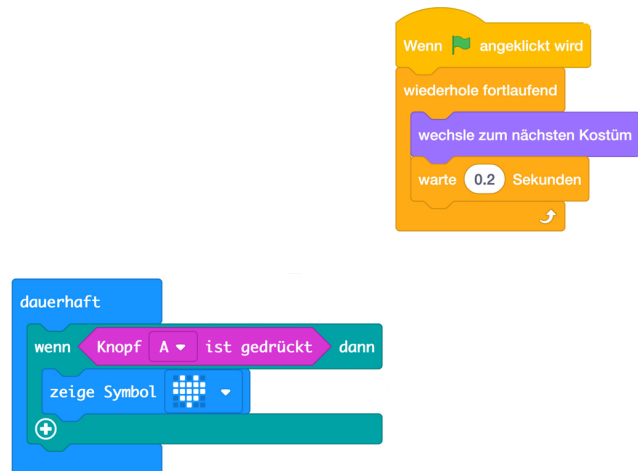
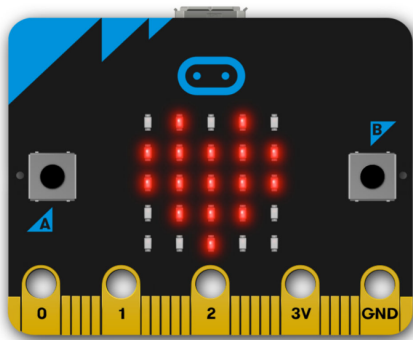




DigiMe Basis Workshop

Interreg



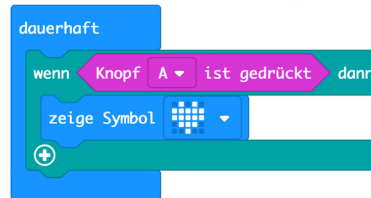
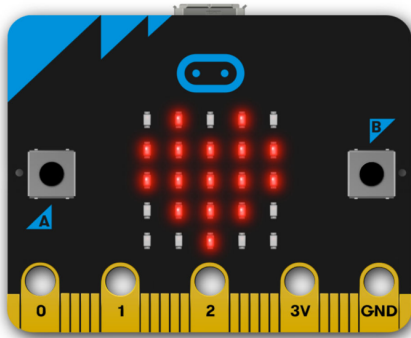
Österreich-Tschechische Republik

DigiMe

Europäischer Fonds für regionale Entwicklung

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

DigiMe Basis Workshop



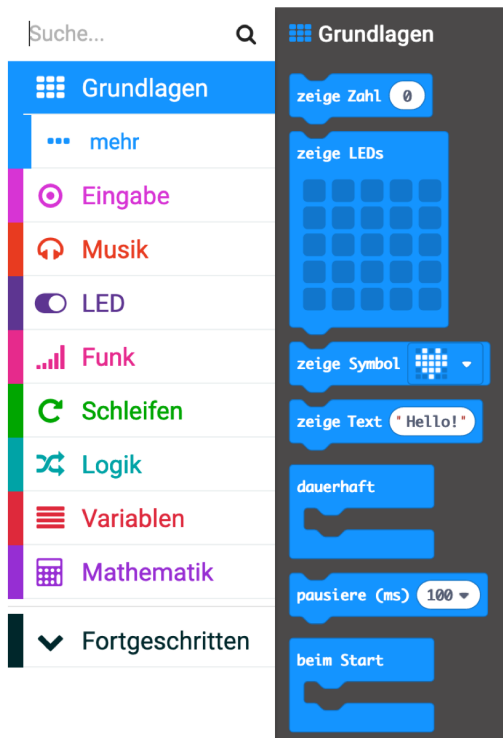
Workshopablauf

- ▶ Ziel + Benefit für den Unterricht
 - ◆ Erlernen der Basics einer Programmiersprache und des Mikrobits
 - ◆ Anwenden passender Beispiele für diverse Unterrichtsgegenstände
 - ◆ Motivation und Abwechslung im Unterricht
 - ◆ Hausübungen bzw. Flipped Classroom Szenarien durch Simulator
- ▶ Beispiele und Einsatzmöglichkeiten im Unterricht
- ▶ Aufbau Microbit (LEDs, Taster, Sensoren, ...)

Praxisteil

- ▶ Verwendung des Microbits in 3 Schritten:
 - ◆ Verbindung (Microbit – Device)
 - ◆ Programmierung
 - ◆ Übertragung des Programmes
- ▶ Teilen (Weitergabe) des Programmes
- ▶ Kennenlernen der Makecode Programmieroberfläche anhand von Beispielen
- ▶ Kennenlernen der Microbit-Scratch Kombination anhand der Programmierung von Flappy Bird

Kennenlernen der Makecode Oberfläche – 1



Zu einem Thema passende Codeblöcke sind in Kategorien zusammengefasst. Du kannst die Blöcke einfach per Drag&Drop auf die rechte Programmieroberfläche ziehen.

Einige Kategorien enthalten die Unterkategorie **... mehr**, die weitere Blöcke (welche nicht so oft benötigt werden) verbirgt.

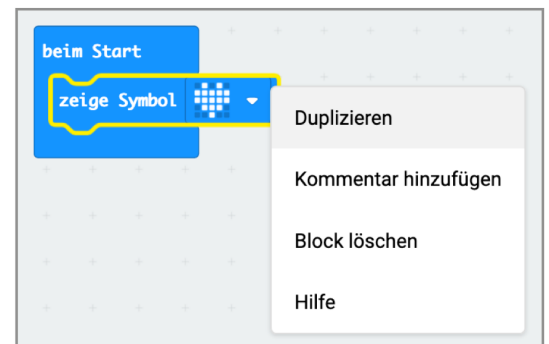
Mit einem Rechtsklick auf einen Block erhältst du folgende Auswahl:

[Duplizieren](#) ermöglicht dir einzelne, aber auch zusammenhängende Codeblöcke zu kopieren.

Bei komplizierten Programmen verhilft dir die Möglichkeit [Kommentar hinzufügen](#) deinen Code übersichtlich zu halten.

Solltest du einen [Block löschen](#) (oder auch eine Blockkombination) wollen, kannst du das hier machen oder deine Blöcke einfach in die Kategorieübersicht ziehen.

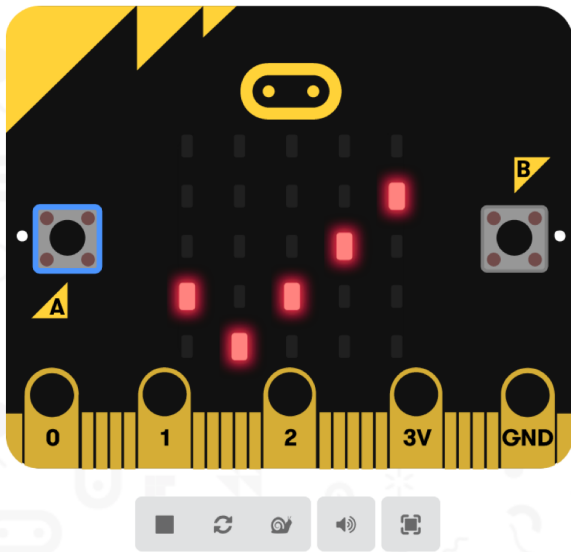
Solltest du [Hilfe](#) benötigen, verschafft dir ein Klick darauf die gewünschten Infos.



Die Möglichkeit Blöcke mit dem + bzw. – zu vergrößern bzw. zu verkleinern ist besonders für den Unterricht mit Beamereinsatz sehr hilfreich.

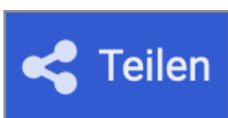
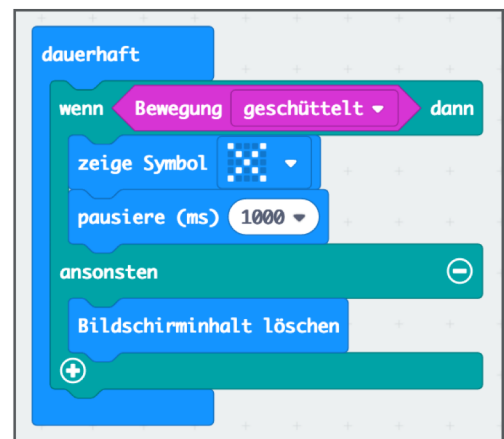
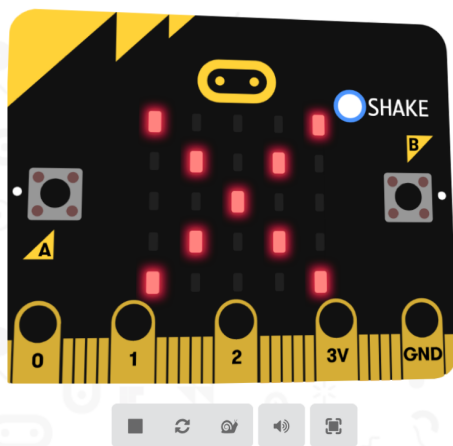
Hab keine Angst, mit dem Zurück-Pfeil kannst du Schritte auch wieder rückgängig machen.

Kennenlernen der Makecode Oberfläche – 2



Der Simulator ermöglicht dir Programme zu testen, auch wenn du keinen Microbit zur Verfügung hast. Neben Taster und LED Matrix (Display) stehen dir auch Sensoren für Licht, Temperatur und Neigung, sowie das Funkmodul zur Verfügung.

Diese werden eingeblendet, sobald du die entsprechenden Programmblöcke verwendest.



Der Teilen-Button ermöglicht dir deinen Programmcode zu archivieren bzw. mit deinen Freunden zu teilen.

Verschicke den Link und zeig her, was du Tolles geschaffen hast.

Unterrichtstipp

- Verwende das Teilen auch, um als LehrerIn Online Aufgaben zu erstellen. Beispielsweise können in diesen Aufgaben kleine Fehler eingebaut sein, die von den SchülerInnen gefunden werden müssen.
- Oder es sollen Teilprogramme ergänzt bzw. weiterentwickelt werden.

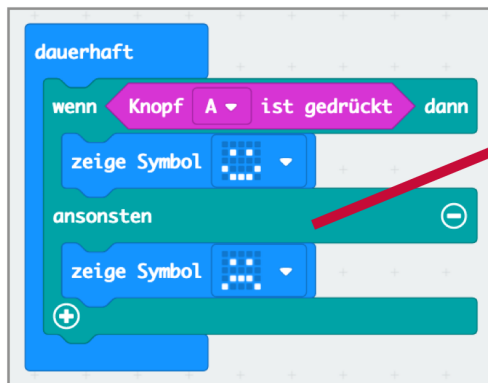
Auf diesem Weg können Hausübungen auch online "abgegeben" werden.

Tasterabfrage

Aufgabe:

Das Display des Microbits zeigt einen traurigen Smiley, wenn der Taster A nicht gedrückt ist.

Wird der Taster A gedrückt, lacht der Smiley, solange der Taster gedrückt bleibt.



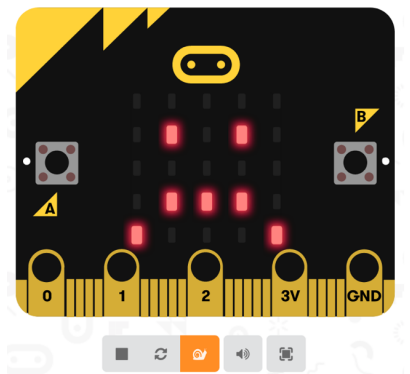
Ohne "ansonsten"-Teil würde der Smiley nach dem erstmaligen Tastendruck lachen und so verbleiben, auch nachdem der Taster losgelassen wird.

https://makecode.microbit.org/_cLTEcUJm1dyH

Hinweis

Da die Darstellung am Display (im "ansonsten" Teil) fast eine Sekunde dauert, muss der Taster etwas länger gedrückt werden, damit der Tastendruck erkannt wird.

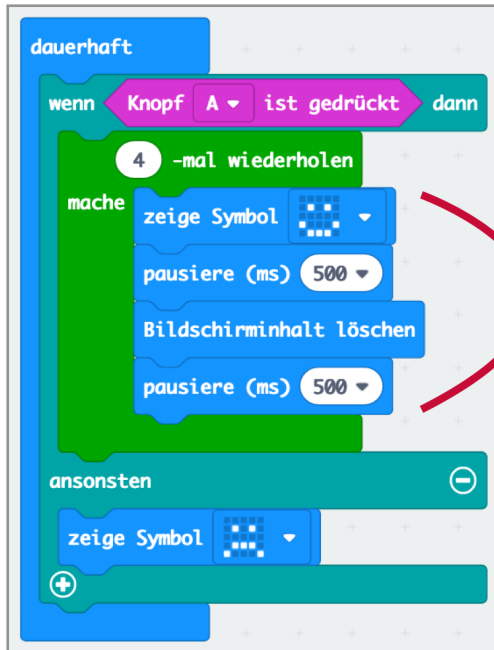
Mit dem "Schneckenbutton" im Simulator kannst du die Geschwindigkeit des Microbits verlangsamen, um diesen Effekt zu beobachten.



Schleifen

Aufgabe:

Beim Tastendruck blinkt der Smiley 4x (sonst gleich wie 1a).



Ohne Schleife müsste man die 4 Zeilen viermal verwenden => 16 Zeilen.

In diesem Fall sorgen Schleifen für einen übersichtlicheren und weniger fehleranfälligeren Code.

Bei einer Änderung der Schleifenwiederholungen muss man dies nur einmal im Schleifenkopf vornehmen.

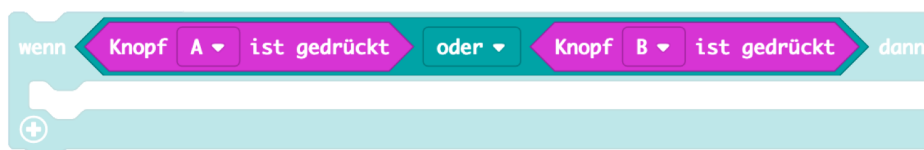
https://makecode.microbit.org/_3id2wociRUbD

Hinweis

Dies ist die einfachste und häufigste Verwendung von Schleifen. Schleifen werden ebenfalls für andere Aufgaben verwendet, die in einem weiteren Arbeitsblatt vorgestellt werden.

Zusatzinfo

Damit der Microbit reagiert, egal welcher der beiden Taster gedrückt wird, benötigst du den Block "oder".



Soll der Microbit nur reagieren, sofern beide Taster gleichzeitig gedrückt werden, benötigst du eine "und" Verknüpfung.

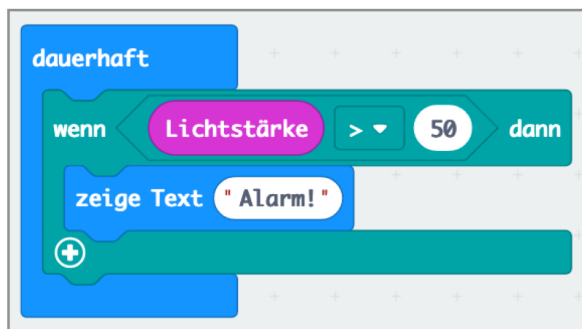
Alarmanlage (Lichtsensord)

Aufgabe:

Programmiere eine Alarmanlage mit Hilfe des Lichtsensors.

Der Microbit liegt in einer Schublade, in welcher es dunkel ist. Sobald die Schublade geöffnet wird, fällt Licht auf den Sensor und die Alarmanlage löst aus => es erscheint "Alarm!" am Display.

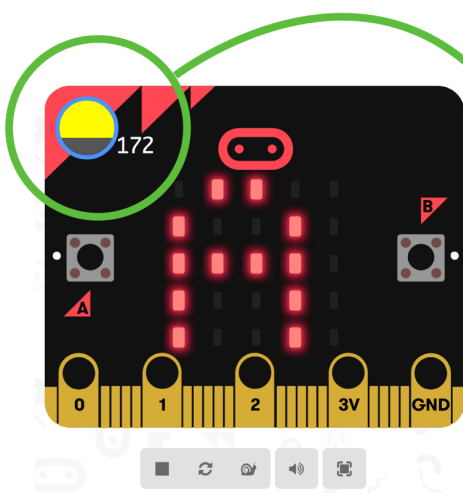
Optional: Statt der Anzeige des Alarms wird eine Melodie abgespielt.



Den Block "Lichtsensord" findest du in der Kategorie **Eingabe**.

Verwende den Vergleichsoperator  aus der Kategorie **Logik**, um den Lichtsensor auf einen bestimmten Wert hin abzufragen.

https://makecode.microbit.org/_DmTLTs100V00



Hinweis

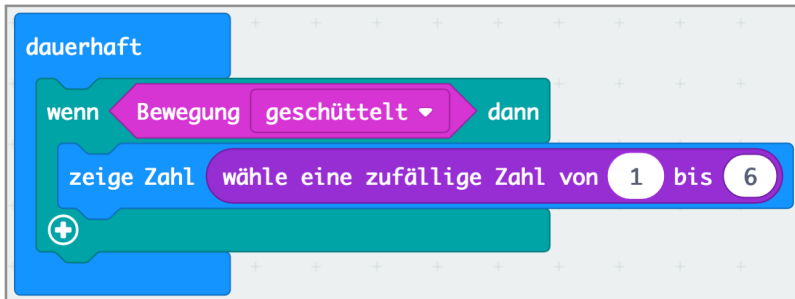
Verwendest du keinen "echten" Microbit, kannst du den Helligkeitswert im Simulator mit der Maus verändern. Sobald der Wert 50 überschritten wird, erscheint "Alarm!" am Display.

Optional: In der Kategorie **Musik** findest du die Blöcke für das Ausgeben einer Melodie (im Simulator abspielbar). Die Ausgabe über einen Lautsprecher direkt am Microbit wird in einem separaten Arbeitsblatt behandelt.

Würfel (Zufallszahl)

Aufgabe:

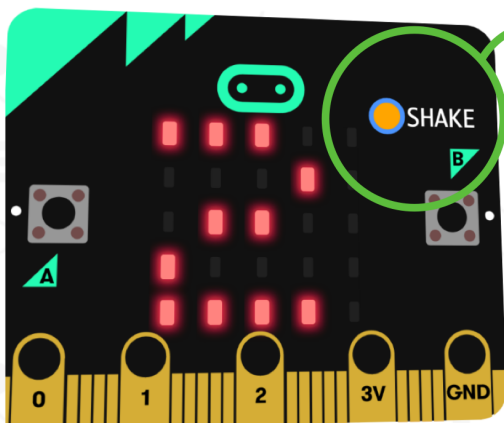
Beim Schütteln des Microbits erscheint eine Zufallszahl zwischen 1 und 6.



In der Kategorie **Eingabe** findest du neben anderen Sensoren auch den Block für den Neigungssensor **Bewegung geschüttelt**.

https://makecode.microbit.org/_HkFbbRJ48Ywa

Hinweis



Solltest du keinen "echten" Microbit verwenden, wird dir die "Schüttelfunktion" im Simulator angezeigt, sobald du den entsprechenden Block in deinem Programm verwendest.

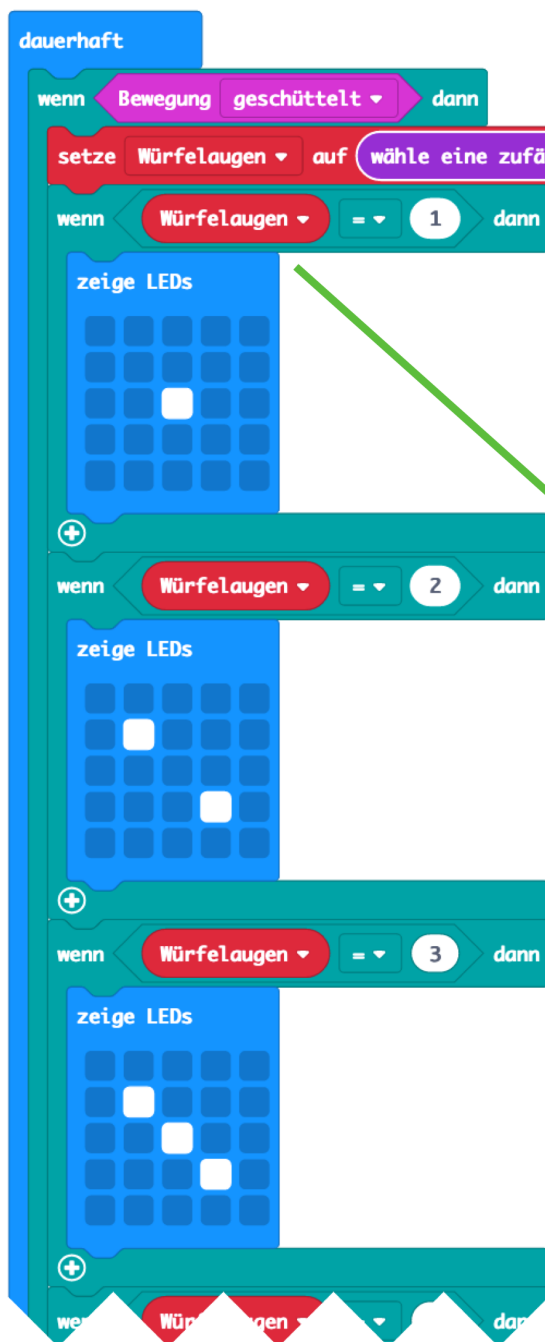
Würfel (Variable)

Aufgabe:

Beim Schütteln des Microbits erscheint zufällig eine Augenzahl.

Eine Möglichkeit diese Aufgabe zu realisieren ist das Verwenden einer Hilfsvariablen.

In der Kategorie **Mathematik** findest du den Block "wähle eine zufällige Zahl", bei dem du die Grenzen des Wertebereiches angeben kannst.



Ablauf:

Beim Schütteln des Microbits wird der Variablen "Würfelaugen" eine Zufallszahl von 1-6 zugewiesen.

Abhängig von der Zufallszahl wird die Augenzahl am Display ausgegeben.

https://makecode.microbit.org/_PE9H8vFKV3eH

Stadt – Land – Fluss (Array)

Aufgabe:

Bei Tastendruck erscheint am Display ein Buchstabe per Zufall.



Variable vs Array

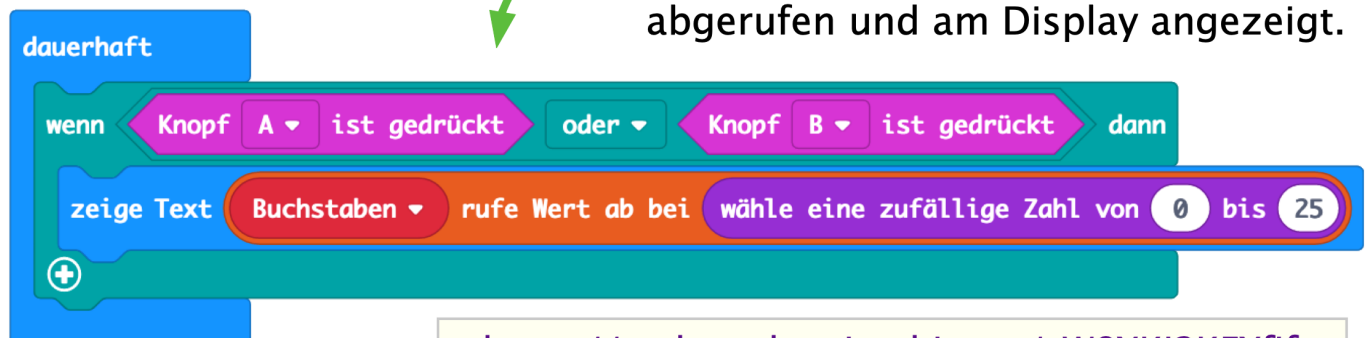
Wenn du mehrere Variablen zu einer Gruppe zusammenfasst, erhältst du ein Array.

In unserem Fall erstellen wir beim Start das Array "Buchstaben" mit den 26 Buchstaben des Alphabets.

Hinweis

Bei einem Array beginnt die Nummerierung der Elemente bei 0 und nicht bei 1.

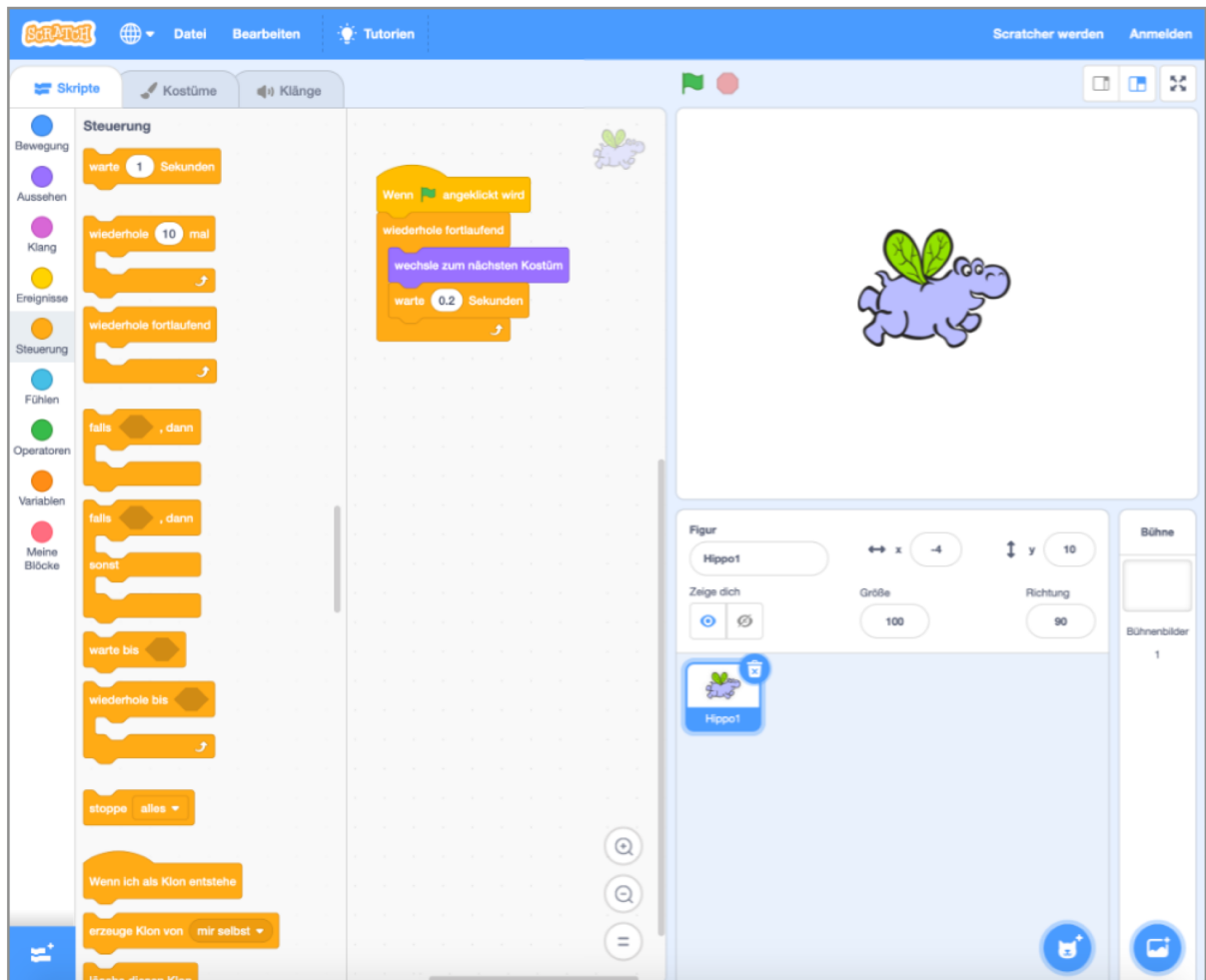
Bei Tastendruck wird eine Zufallszahl zwischen 0 und 25 erzeugt. Im Array "Buchstaben" wird auf die Stelle der Zufallszahl zugegriffen, der entsprechende Wert (Buchstabe) abgerufen und am Display angezeigt.



https://makecode.microbit.org/_W8XKj2KFXfif

Scratch Oberfläche

Auf der Scratch Webseite <https://scratch.mit.edu> und einem Klick auf "Entwickeln" startest du Scratch.



Die Scratch Oberfläche besteht aus 3 Spalten:

- Die linke Spalte beinhaltet alle Codingblöcke, die in Kategorien zusammengefasst und farblich sortiert sind.
- Die mittlere Spalte ist die Programmierumgebung, in die die Programmblöcke per Drag&Drop hineingezogen und ähnlich einem Puzzle ineinander verschachtelt werden können.
- Die rechte Spalte zeigt einerseits die sogenannte Bühne, die dein Programm visualisiert und andererseits darunter deine im Projekt verwendeten Figuren und den Hintergrund der Bühne.

Scratch + Microbit

In nur 2 Schritten kannst du deinen Microbit in Scratch Programme einbinden und damit deine Spiele interaktiv beleben, z.B. Steuerung der Figuren mit Taster und Neigungssensor, Ausgabe des Punktestands am Display und vieles mehr.

1. Download und Installieren der Scratch-App

Auf <https://scratch.mit.edu/download> findest du die aktuelle Scratch-App zum Download.

Verfügbar für:

- Windows
- macOS
- ChromeOS
- Android

Scratch-App für macOS installieren

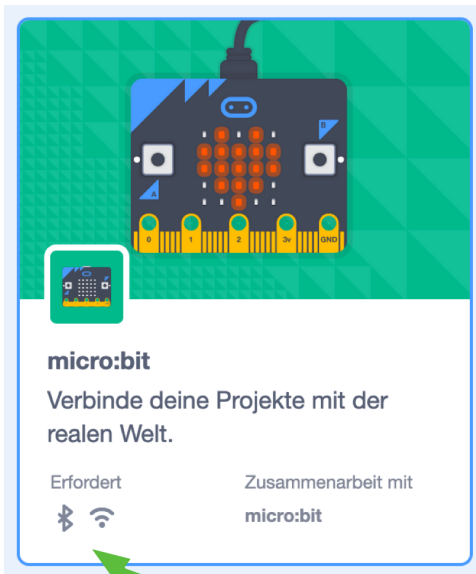
Hol dir die Scratch-App im Mac App Store



oder

[Direkt herunterladen](#)

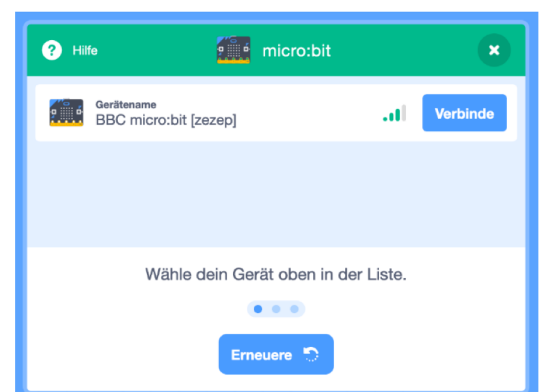
2. Nach einem Klick auf den Button "Erweiterung hinzufügen" (links unten) wählst du die Erweiterung **micro:bit**.



Hinweis

Für die Verwendung der Scratch-App benötigst du eine Bluetooth- und Internetverbindung.

Der mit dem Computer oder Tablet verbundene Microbit wird erkannt und steht mit einem Klick auf "Verbinde" der Scratch Oberfläche zur Verfügung.

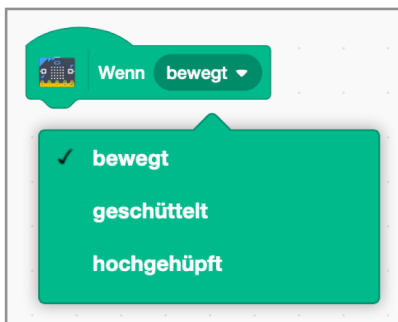
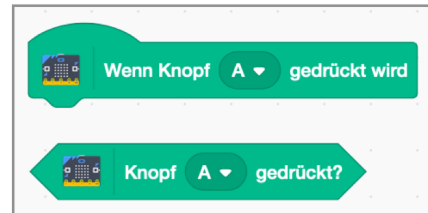


Funktionen des Microbits in Scratch

Leider stehen nicht alle Möglichkeiten des Microbits auch in Scratch zur Verfügung.

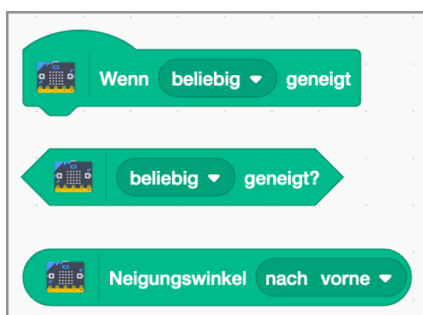
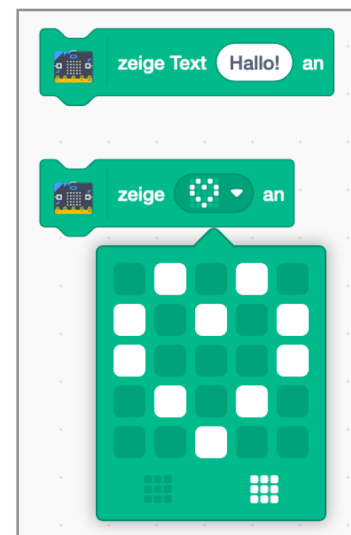
Folgende Funktionen können verwendet werden:

Die Taster A und B können auf zwei Varianten abgefragt werden.



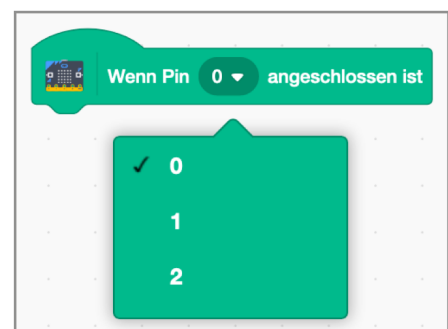
Erschütterungen können mit dem Microbit erkannt werden.

Am Display (LED Matrix) des Microbits kann einerseits Text ausgegeben werden – andererseits kannst du mit den 25 LEDs auch einfache Muster selbst gestalten.

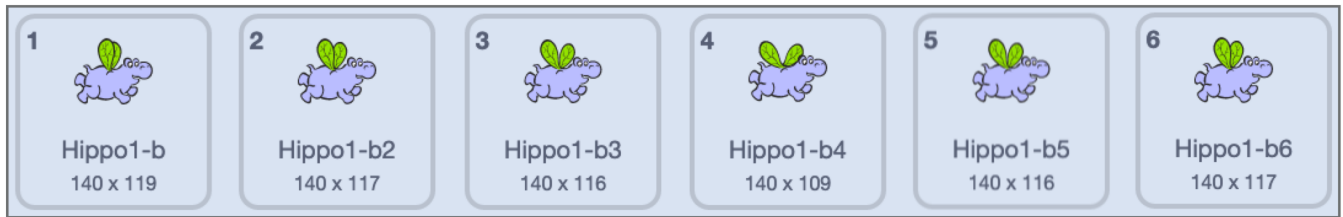


Es kann sowohl die Neigung nach vorne, hinten, links und rechts abgefragt werden als auch ein bestimmter Neigungswinkel.

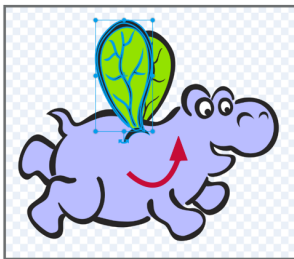
Der Microbit erkennt, wenn Pin0, Pin1 oder Pin2 mit GND verbunden wird.



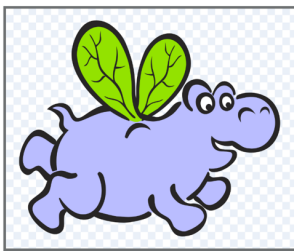
Flappy Bird (Scratch) – Flugbewegung



Damit die Flugbewegung des Flappy Birds echt aussieht, werden (wie bei einem Daumenkino) die unterschiedlichen Flügelpositionen schnell hintereinander abgespielt. *[Bei dem Nilpferd handelt es sich um eine Originalfigur von Scratch]*

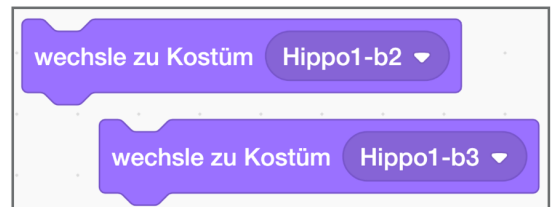


Zuerst kopierst du in Scratch das erste Kostüm. Anschließend wählst du mit der Maus den Flügel, den du drehen möchtest aus. Danach musst du den Flügel nur noch drehen und an die gewünschte Stelle schieben.



Dies machst du ebenfalls für den anderen Flügel und dein zweites Kostüm ist fertiggestellt. Noch realistischer sieht der Flug des Flappy Birds aus, wenn du drei oder vier Kostüme mit unterschiedlichen Flügelpositionen erstellst.

Eine Möglichkeit wäre es, die gewünschten Kostüme nacheinander aufzurufen.



Noch einfacher geht es auf diese Weise.

Mit dem Block **wechsle zum nächsten Kostüm** wird in einer Endlosschleife automatisch zum nächsten Kostüm gesprungen. Die anschließende Pause gibt das Tempo des Flügelschlages an. Je länger du die Pause wählst, desto langsamer bewegen sich die Flügel des Flappy Birds.

Hinweis

Achte auf den Dezimalpunkt bei Kommazahlen!

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

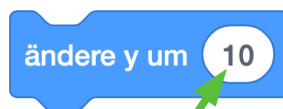
Flappy Bird (Scratch) – Steuerung

Anforderung: Der Flappy Bird schwebt bei keiner Interaktion langsam zu Boden. Jedesmal, wenn man den Button A des Microbits drückt, soll der "Vogel" ein kleines Stück aufsteigen.



Zu diesem Zweck benötigst du die Abfrage "falls-sonst".

Ausserdem benötigst du diesen Block aus der Kategorie **micro:bit**.



Das Absenken bzw. Ansteigen des "Vogels" erreichst du, indem du den Wert der y-Koordinate erhöhst bzw. verringerst. Du kennst dies sicher noch vom Koordinatensystem aus dem Mathematikunterricht.

- der Wert bestimmt das Tempo des Sinkens/Steigens
- das Vorzeichen, ob der Flappy Bird steigt oder sinkt

So sieht der Steuerungsblock aus:



In einer Endlosschleife wird der Taster abgefragt.

Wird der Taster A nicht gedrückt –

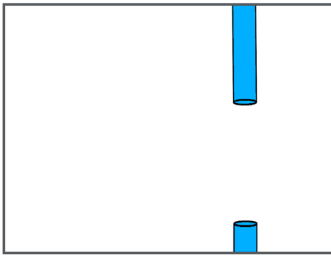
sinkt der Flappy Bird langsam (ändere y um -2).

Wird der Taster gedrückt (sonst) –

steigt der Flappy Bird sprungartig (ändere y um 10).

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

Flappy Bird (Scratch) – Hindernis

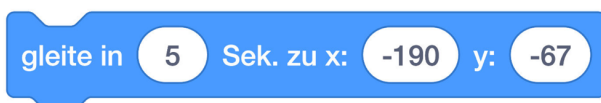
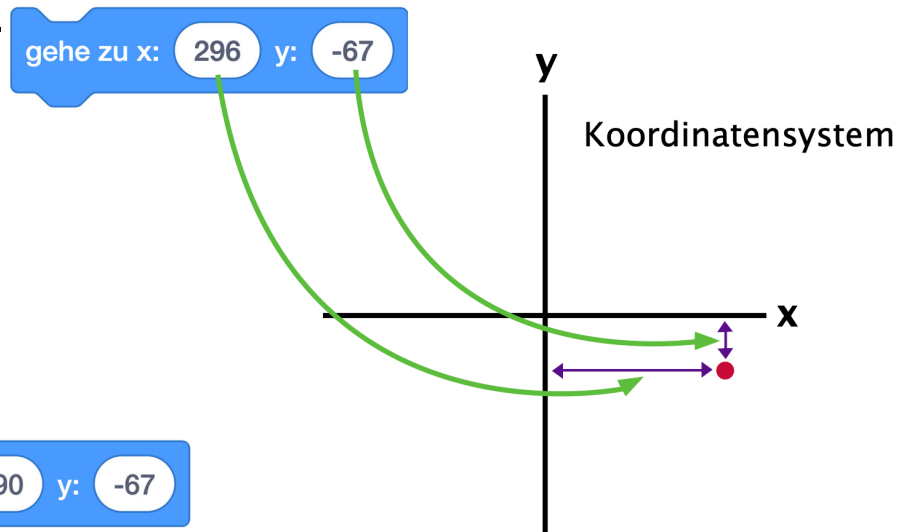


Anforderung:

Die Röhren gleiten vom rechten zum linken Bildschirmrand.

Der Abstand zwischen dem oberen und unteren Röhrenende muss deutlich größer als der Flappy Bird sein, damit dieser hindurchfliegen kann.

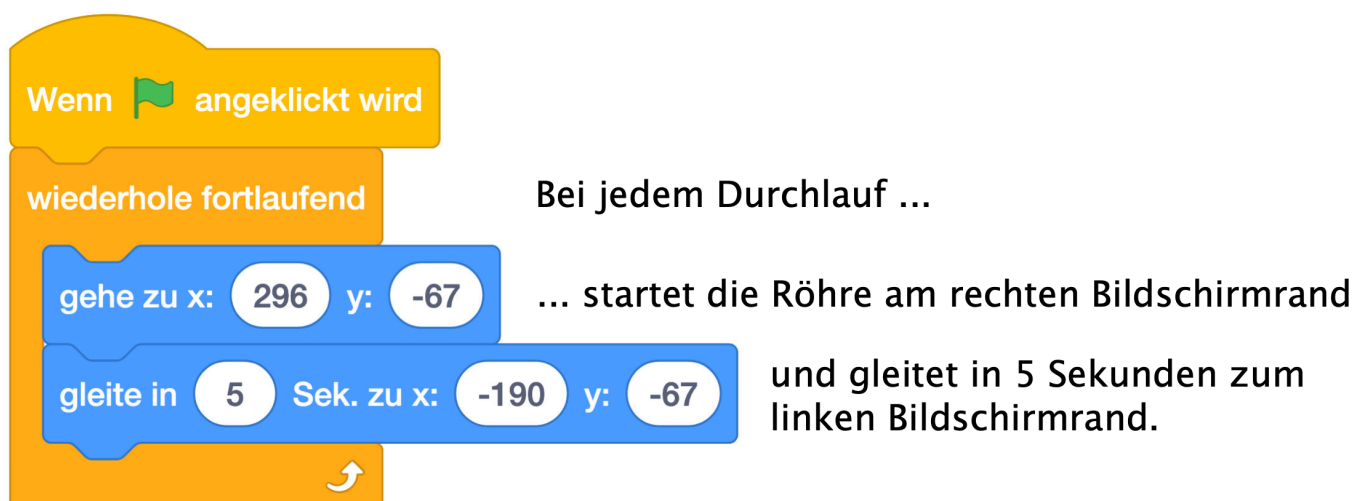
Damit die Röhre bei jedem Durchlauf rechts am Bildschirmrand erscheint, musst du in dem "gehe zu"-Block die entsprechenden Koordinaten angeben.



In dem "gleite"-Block gibst du die Endkoordinaten der Röhre (linker Bildschirmrand) an und die Zeitdauer der Bewegung.

Tipp: Je kürzer die Zeitdauer - desto schneller bewegt sich die Röhre

So sieht der Steuerungsblock aus:

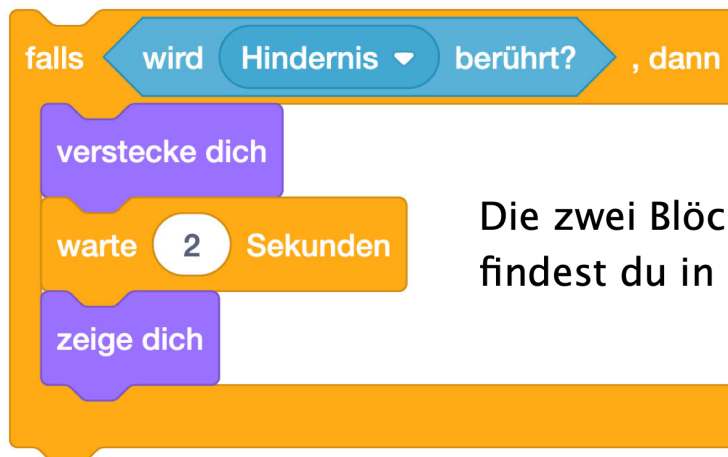


<https://scratch.mit.edu/projects/421461139>

Flappy Bird (Scratch) – Fehlerabfrage / 1

Anforderung:

Wenn das Nilpferd die Röhre berührt, verschwindet es und erscheint nach 2 Sekunden wieder. Dies wird mit einer "falls"-Abfrage umgesetzt, in der die Röhre (Hindernis) auf Berührung abgefragt wird.



Die zwei Blöcke **verstecke dich** und **zeige dich** findest du in der Kategorie "Aussehen".

Damit zu Beginn das Nilpferd an einer bestimmten Position startet, muss dessen Position angegeben werden.

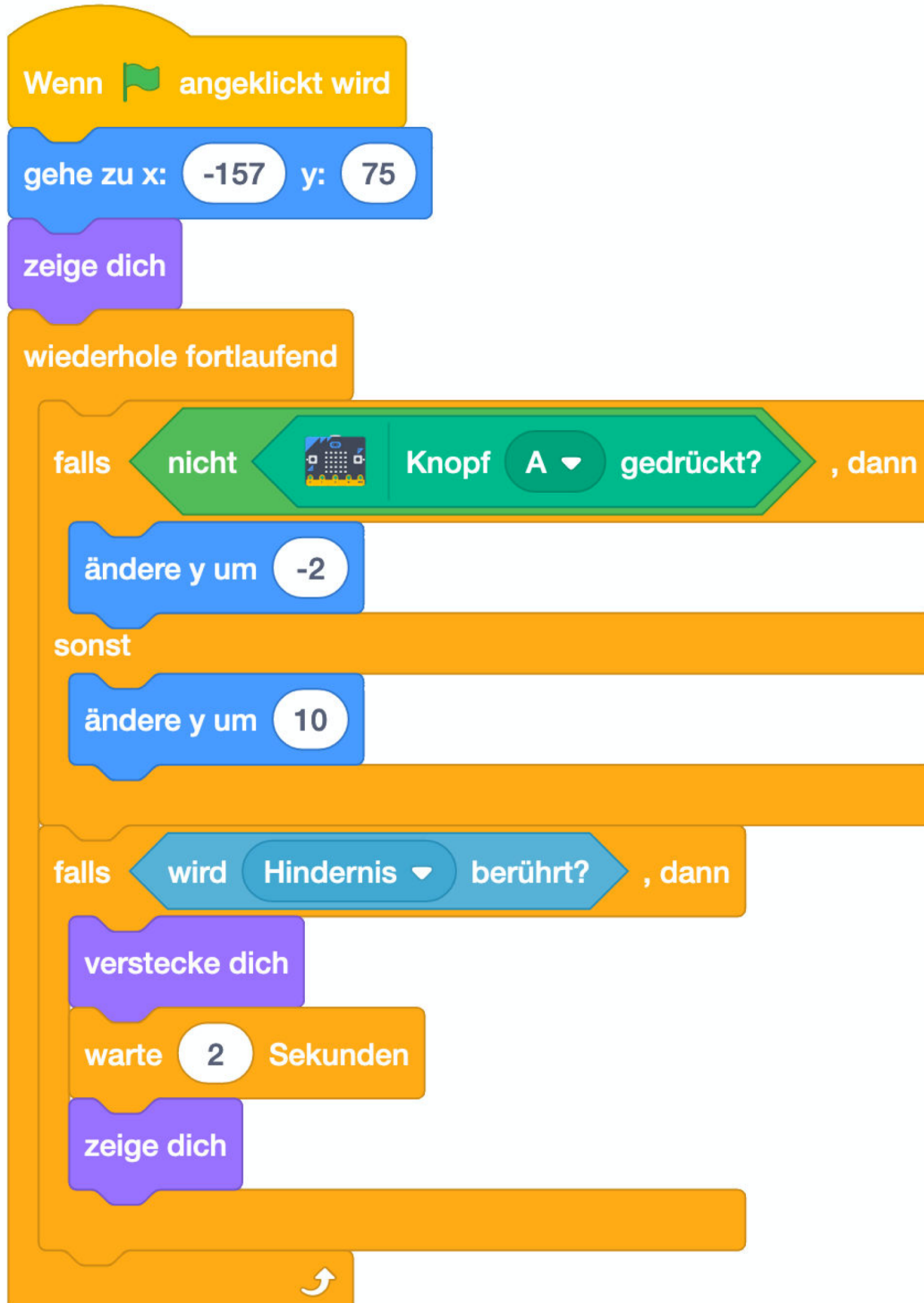


Hinweis:

Sollte das Spiel beendet werden, während das Nilpferd nicht sichtbar ist, würde es bei einem Neustart des Spiels (grüne Flagge) auch nicht sichtbar sein. Dieser Ausnahmestand muss zu Beginn mit "zeige dich" ausgeschlossen werden.

Flappy Bird (Scratch) – Fehlerabfrage / 2

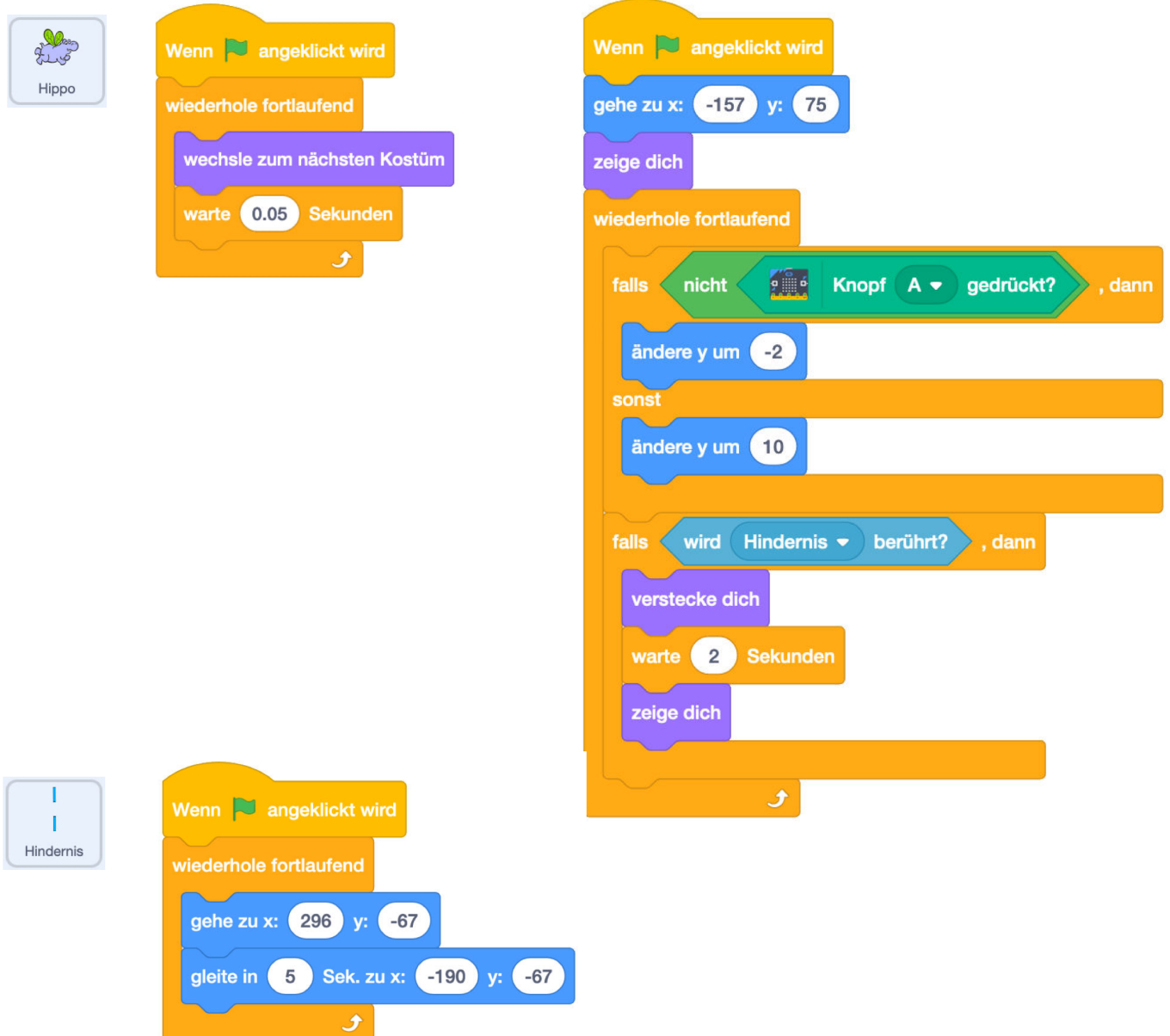
So sieht der Steuerungsblock der Röhre aus:



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

Flappy Bird (Scratch) – Das Spiel

Programmiercode und Link zum Spiel



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

mögliche Erweiterungen:

- Unterschiedliche Hintergründe
- automatisches Hochzählen von Fehlern
- Steigern des Tempos alle 10 Sekunden
- Verkleinern des Abstandes zwischen den Röhren alle 10 Sekunden
- und vieles mehr