Lehrerfortbildung

Der micro:bit als Informatiksystem

Verbindliche Ziele und Inhalte in Klasse 8 Sensorgesteuerte Anwendungen entwickeln (ca. 10 Unterrichtsstunden)

Eine Vielzahl alltäglicher Informatiksysteme wertet Daten von Sensoren aus. Anhand des Arbeitens mit Sensoren erkennen die Schülerinnen und Schüler die Analog-Digital-Wandlung als Grundlage der Digitalisierung und festigen ihre Fähigkeiten im Algorithmieren mit einer blockbasierten Programmiersprache.

Verbindliche Ziele und Inhalte	**Hinweise und Anregungen**
Software für ein Informatiksystem entwickeln • eine Spezifikation prüfen und erweitern	Eine Spezifikation ist die exakte Beschreibung der gewünschten Funktionalität eines Informatiksystems.
• eine Anwendung realisieren und dokumentieren	Die Dokumentation kann als Kommentarfunktion für die Implementation, als Prozessdokumentation oder als Bedienungsanleitung realisiert werden.
• eine Anwendung systematisch testen	Anhand der Testergebnisse sind Schlussfolgerungen für die Entwicklung zu ziehen.
Daten verarbeiten • Sensorwerte abfragen und verarbeiten	Die Schülerinnen und Schüler erkunden die Wirkungsweise von Sensoren eines Informatiksystems und ermitteln den Wertebereich der Sensoren.
• das Prinzip der Analog-Digital-Wandlung erläutern	

Verbindliche Ziele und Inhalte in Klasse 9 Sensorwerte erfassen und auswerten (ca. 10 Unterrichtsstunden)

Die blockbasierte Programmierung eines Mikrocontrollers oder eines anderen autonomen Informatiksystems für die Erfassung und Auswertung einer Messwertreihe bildet den Kontext für die Verwendung des Datentyps Liste. Mit dem EVAS-Prinzip lernen die Schülerinnen und Schüler ein einfaches, verallgemeinertes Modell eines Informatiksystems kennen.

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Daten verarbeiten	
• Listen zur systematischen Verarbeitung von Sensorwerten verwenden	Operationen zum Hinzufügen und Abfragen von Werten sind zu thematisieren.
Algorithmen strukturieren	
• Funktionen definieren und verwenden	nur Gy
Sensoren verwenden	
• Sensorwerte analysieren und bewerten	Die Analyse kann z. B. im Programm oder nachträglich in einer Tabellenkalkulation durchgeführt werden.
• Zuverlässigkeit von Sensorwerten beurteilen	Die Schülerinnen und Schüler leiten Schlussfolgerungen zu Fehleranfälligkeit, Sicherheit und Konstruktionsprinzipien von Informatiksystemen ab.
• Zusammenhang zwischen gemessener physikalischer Größe und Sensorwert erläutern	nur Gy
das EVAS-Prinzip erläutern	Die Schülerinnen und Schüler erkennen die Allgemeingültigkeit des EVAS-Prinzips für Informatiksysteme. Anhand der verwendeten Hardware identifizieren sie Sensoren als Eingabegeräte, den Prozessor als Verarbeitungseinheit, Aktoren als Ausgabegeräte sowie den Speicher.

Didaktische Informatiksysteme

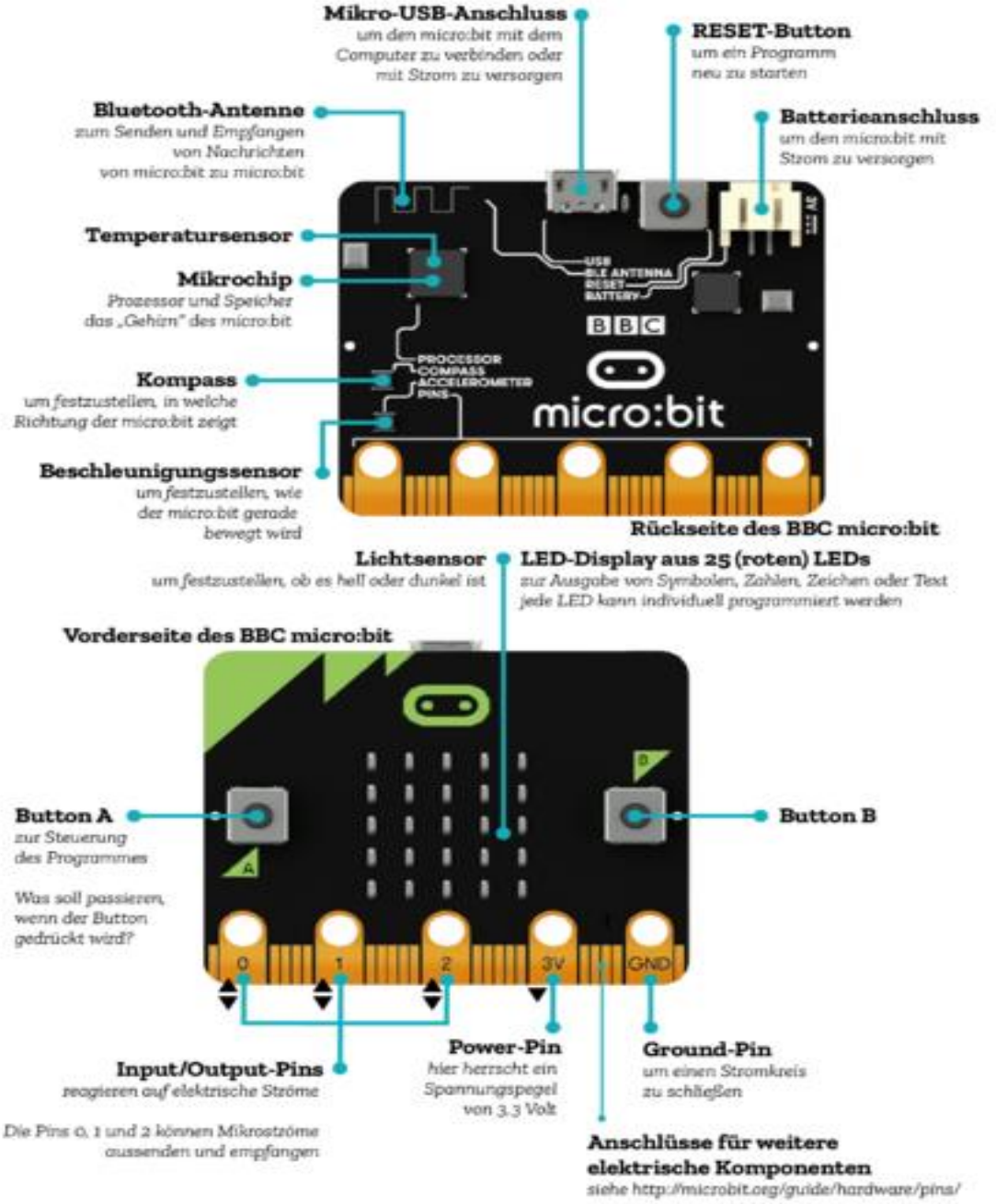
1. Microcontroller „Calliope mini“ + Scratch-ähnliche Entwicklungsumgebung
   - MakeCode-Editor (https://makecode.calliope.cc/)
   - Open Roberta Lab (https://lab.open-roberta.org)
2. Microcontroller „micro:bit“ + Scratch-ähnliche Entwicklungsumgebung
   - MakeCode-Editor (https://makecode.microbit.org/#editor)

Videos zum micro:bit – entwickelt von der PH Steiermark und der TU Graz in Zusammenarbeit mit weiteren PH

(1) Ausrüstung

(2) Einführung zum micro:bit

Die Hardware des micro:bit



Quelle:https://microbit.eeducation.at/wiki/Arbeiten_mit_dem_BBC_micro:bit

EVA-Prinzip am micro:bit

Eingabe	→	Verarbeitung	→	Ausgabe
über		durch		über
Sensoren		Algorithmen (Programme)		Aktoren

Es gilt:

Sensoren erfassen Daten, diese werden digital durch Algorithmen verarbeitet, entsprechend der Verarbeitung geben Aktoren etwas aus. [3]

Sensoren des micro:bit

Unter Sensoren versteht man Bauteile, die zur Messung von Umgebungsbedingungen geeignet sind. [2]

- Lichtsensor
- Temperatursensor
- Kompass
- Beschleunigungs-/Lagesensor
- Taster

Aktoren des micro:bit

Unter Aktoren versteht man Bauteile, die zur Ausgabe geeignet sind.

- intern:
  - LED-Matrix für die Anzeige
- extern:
  - Motor
  - Lautsprecher
  - Leuchtdioden

Überblick

Sensor →	(Mess)fühler	erfasst physikalische Größen liefert Informationen über physikalische Größen
Aktor →	"Macher", Stellglied	verändert physikalische Größe

Wofür werden Sensoren benötigt?

- steuern
- überwachen
- dokumentieren

Sensoren bei Menschen und Tieren

5 Sinne des Menschen: Schnittstellen zur Umwelt

▸ Sehen	*Augen*	$2 \cdot 10^8$ Sensoren
▸ Hören	*Ohren*	$3 \cdot 10^4$ Sensoren
▸ Riechen	*Nase*	$1 \cdot 10^7$ Sensoren
▸ Schmecken	*Zunge*	$1 \cdot 10^7$ Sensoren
▸ Fühlen		
▪ thermisch	*Haut*	$2 \cdot 10^5$ Sensoren
▪ mechanisch	*Haut*	$5 \cdot 10^5$ Sensoren
▪ Schmerz	*Haut*	$3 \cdot 10^6$ Sensoren
▪ Gleichgewicht	*Innenohr*	

Tiere:	▸ Lichtpolarisation	*Insekten*
	▸ Magnetfeld	*Vögel, Bienen*

[4]

Programmieren

Video von der TU Graz zur Erläuterung der Begriffe Maschinensprache, Programmiersprache und Algorithmus:

Programmiersprachen - Was ist das?

Online-Schulbuch zum micro:bit:
https://microbit.eeducation.at
(auch als pdf-Download vorhanden)

https://makecode.microbit.org

Projekte

1. Schrittzähler

   Einführung von Listen
2. Würfeln
3. Fruit-Catcher-Spiel

4. „A oder B“ Spiel (Quelle: [3], S. 35 ff)

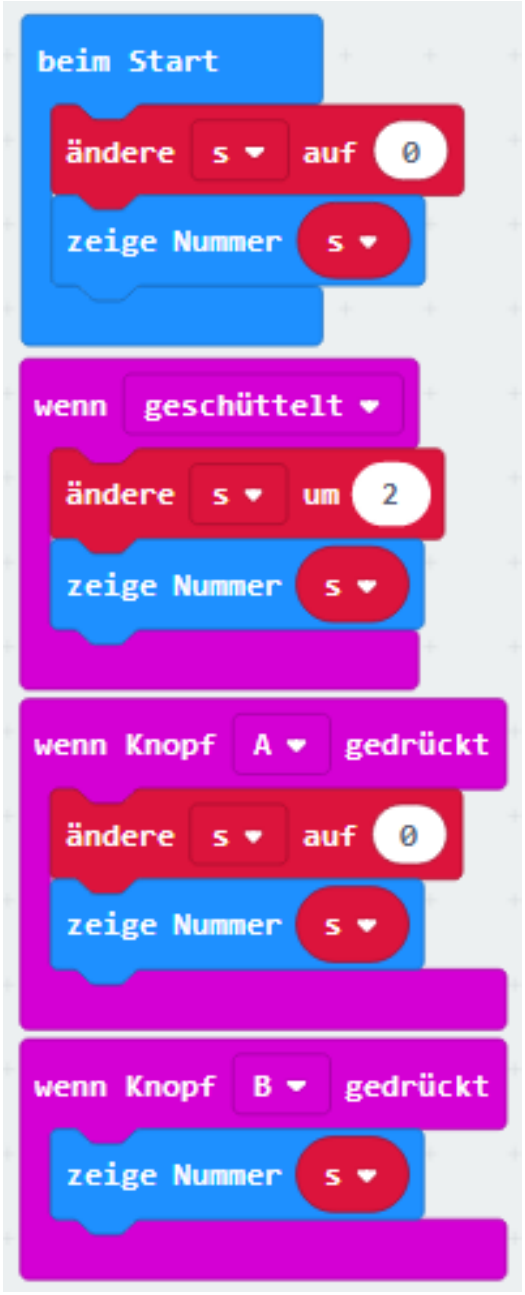
Analyse eines unbekanntes Programms

1. Schließe den micro:bit über ein Mini-USB-Kabel an deinen Rechner an. Sende das Programm *Was_ist_das.hex* an den Microbit (rechter Mausklick auf der Datei → Senden an MICROBIT).
2. Öffne das Programm auch im Microbit-Editor.
3. Was tut das Programm? Welche Sensoren und Aktoren werden in dem Programm verwendet?
4. Kannst du dir einen praktischen Nutzen für das Programm vorstellen?
5. Was muss an dem Programm geändert werden, damit es als Schrittzähler eingesetzt werden kann?
6. Teste den Schrittzähler bei dir. Welche Probleme tauchen auf?

Zu 3.)
Der Wert einer Variablen s wird beim Programmstart auf Null gesetzt. Beim Ereignis „Schütteln“ wird der Wert von s um Eins erhöht. Bei Drücken der Taste A wird s auf Null gesetzt. Jede Wertänderung von s wird auf dem micro:bit angezeigt.

Sensoren: Beschleunigungssensor, Knopf A

Aktoren: LED Matrix



Zu 4.)

z.B. als Schrittzähler, wenn man den micro:bit mit einem Batteriefach verbindet und beides an einem Bein in Fußnähe anbringt

Zu 5.)

Es wird nur jeder zweite Schritt gezählt, Variable s muss jeweils um 2 erhöht werden.

Knopf B wird gebraucht, um die aktuelle Schrittzahl anzeigen zu lassen, wenn der Wert 10 überschritten ist, wird nichts angezeigt (Wert ist einmal durchgelaufen)

Möglichkeiten zur Verbesserung des Schrittzählers

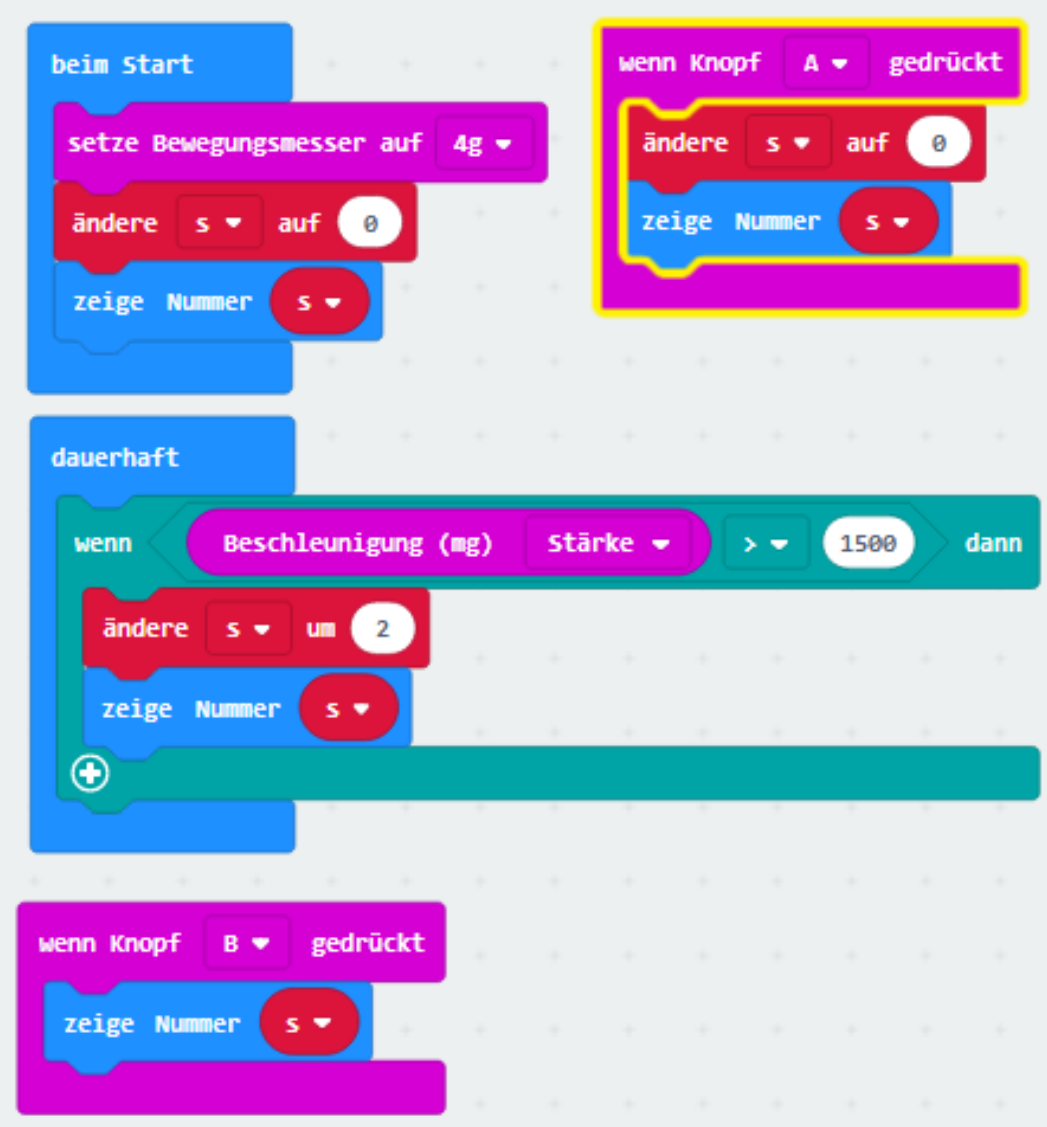
7. Wie könnte man den Schrittzähler verbessern?

A: Sensorwerte wechseln:
Lagesensor auf dem micro:bit, erkennt die x-, y- oder z-Richtung

8. Programmiere den Schrittzähler unter Einsatz des Lagesensors.

9. Was können professionelle Schrittzähler?

zu 8.)

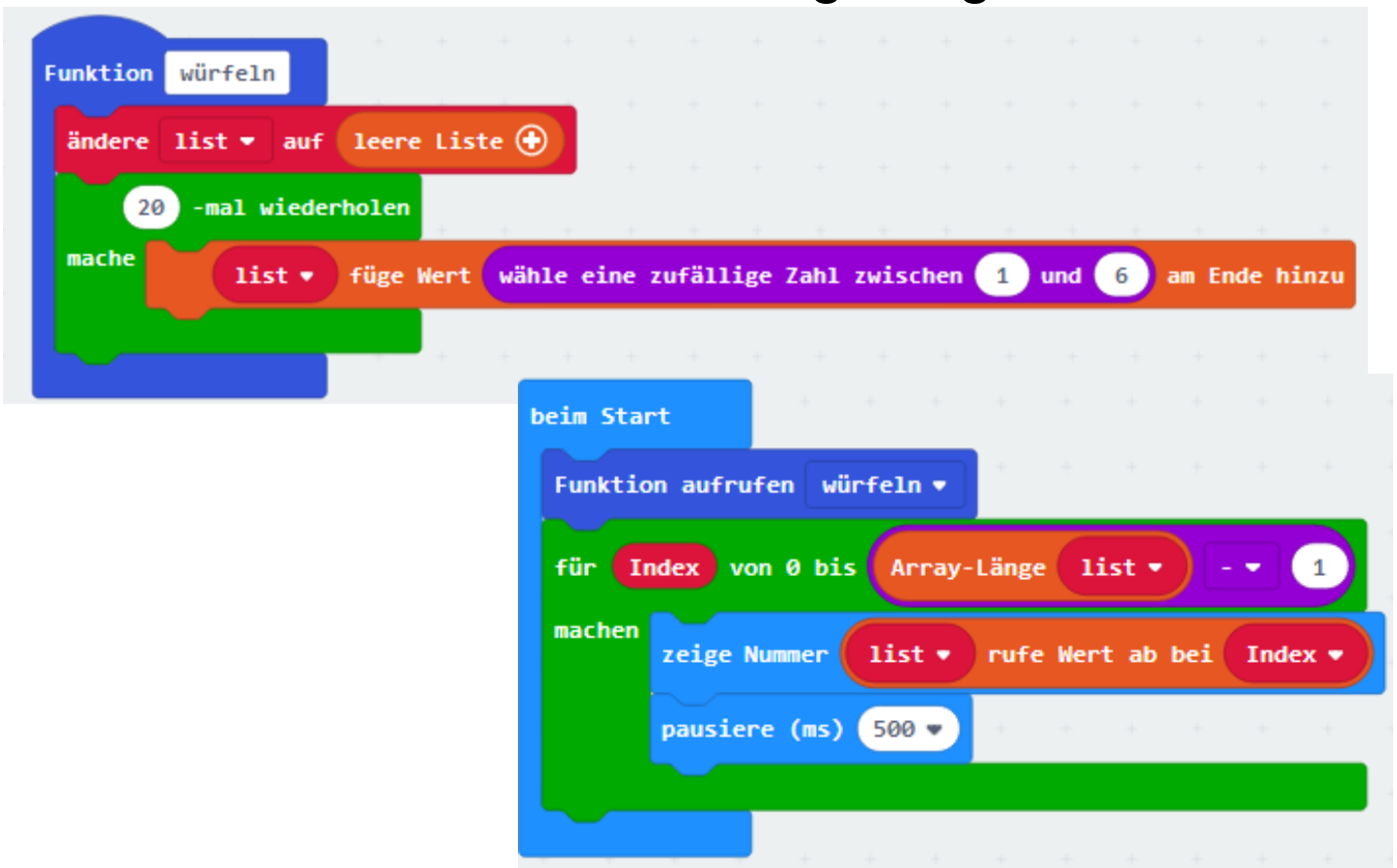

beim Start
setze Bewegungsmesser auf 4g
ändere s auf 0
zeige Nummer s
wenn Knopf A gedrückt
ändere s auf 0
zeige Nummer s
dauerhaft
wenn Beschleunigung (mg) Stärke > 1500 dann
ändere s um 2
zeige Nummer s
wenn Knopf B gedrückt
zeige Nummer s

Einführung von Listen

So könnte man das nächste mal die Klassensprecherwahl durchführen. Jeder Schüler darf einmal schütteln. ☺

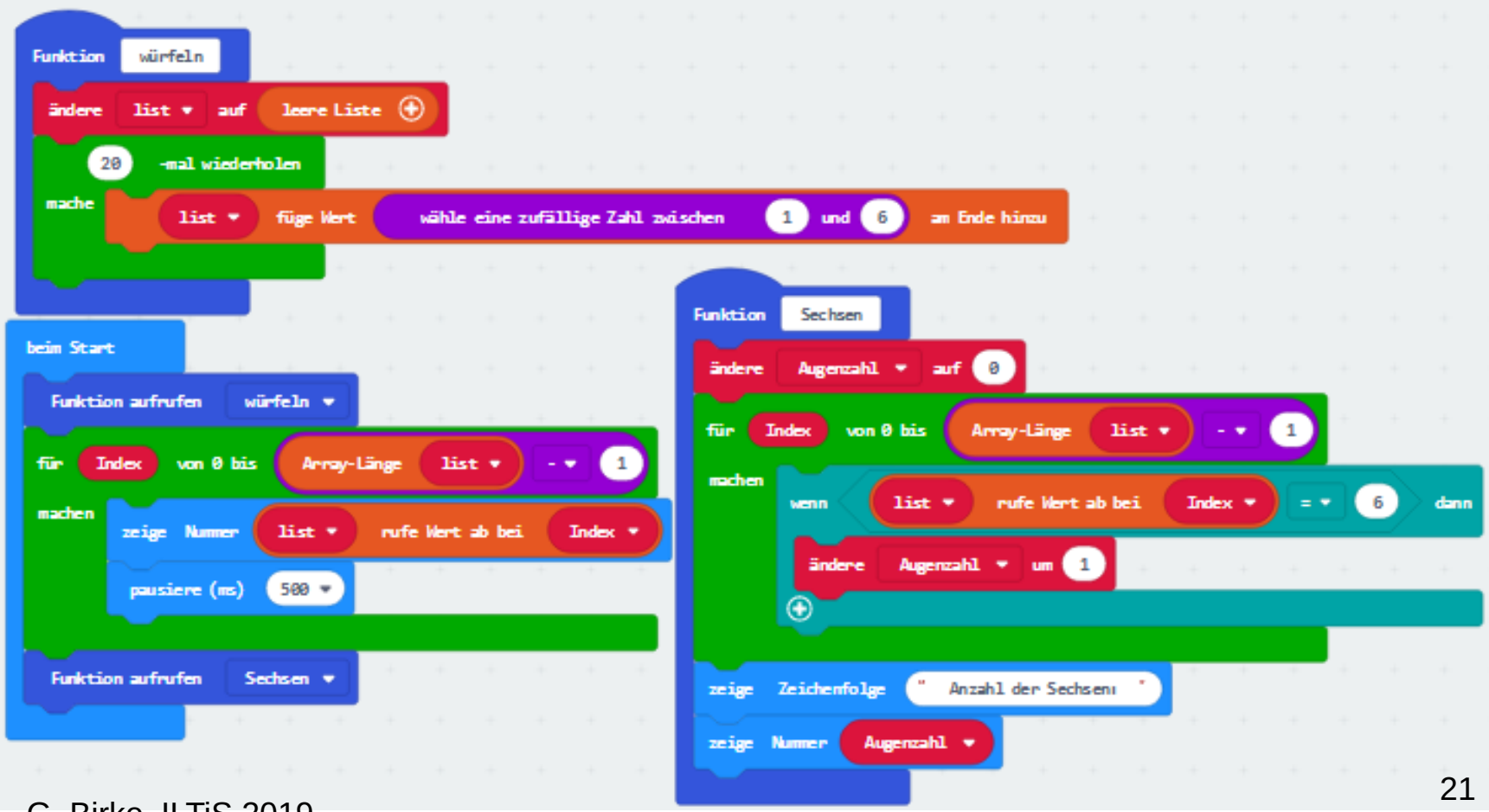
Projekt Würfeln

Ein Würfel wird 20-mal geworfen. Die Ergebnisse werden in eine Liste geschrieben. Anschließend werden die Listenelemente angezeigt.



Projekt Würfeln

In dem Projekt ist eine Funktion Sechsen zu ergänzen, die die Anzahl der in der Liste vorkommenden Sechsen ermittelt.



Das Fruit-Catcher-Spiel

Idee:

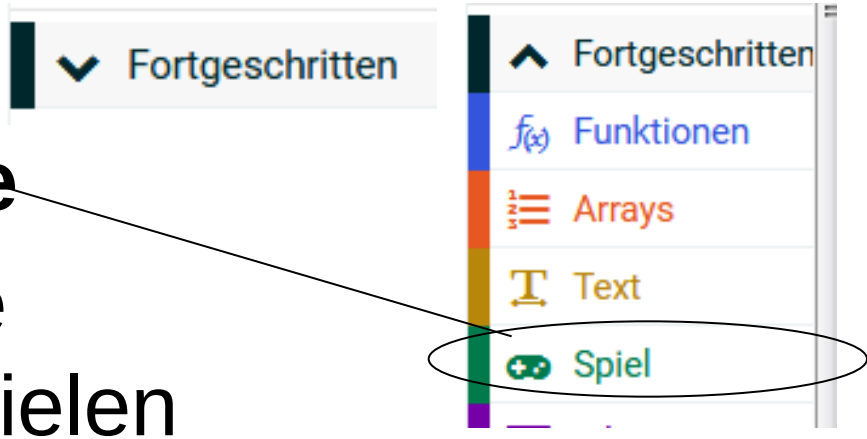
Der Spieler, dargestellt als einfaches leuchtendes Pixel am Boden des Displays, versucht immer schneller fallende Früchte – andere Pixel, die an der Spitze des Displays starten und abwärts fallen – zu fangen, bevor sie den Boden des Displays treffen. (Quelle: [6])

Das Fruit-Catcher-Spiel

Nutzung einer Kategorie aus der Erweiterungstoolbox,

die **Spiel-Kategorie**

→ spezifisch für die Entwicklung von Spielen



Das Fruit-Catcher-Spiel → **die Spiel-Kategorie**

enthält eingebaute Werkzeuge

- zum Nachverfolgen des Spielstandes
- zur Darstellung einer „Game over“-Anzeige
- zur Erstellung einer Ein-Pixel-Figur (sprite) → Objekt, das auf dem Display bewegt werden kann und
- zur Erkennung, ob eine Figur mit einer anderen Figur oder dem Displayrand kollidierte

Das Fruit-Catcher-Spiel

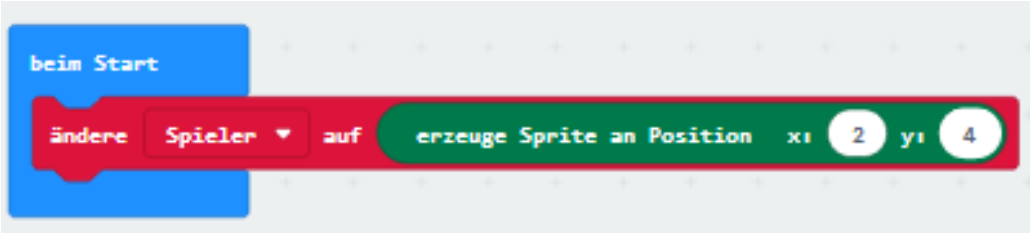
- Startseite →

- Projektnamen festlegen

- [beim Start]-Block und [dauerhaft]-Block werden gebraucht

Das Fruit-Catcher-Spiel → die Einrichtung (Setup)

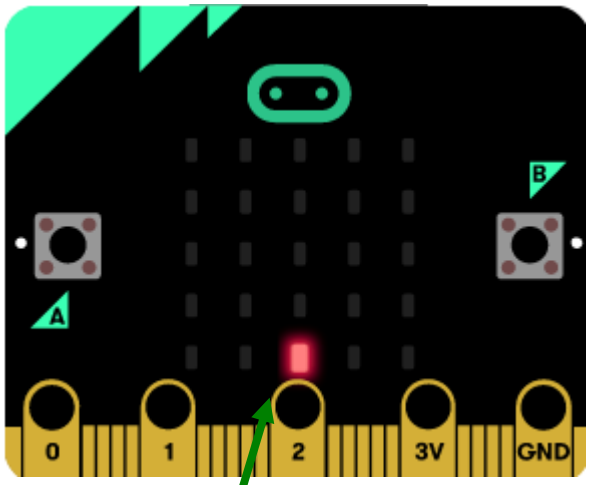
- Start des Spiels mit Einstellen der Spieler-Figur und Initialisierung des Spielstands
- Variable `Spieler` erzeugen (Variablen-Kategorie)
  - → [ändere Spieler auf 0]-Block in den [beim Start]-Block ziehen
  - → [erzeuge Sprite an Position x:2 y:2]-Block (Spiel-Kategorie) auf die 0 ziehen Position [2, 4] setzen



Das Fruit-Catcher-Spiel

x-, y-Koordinaten für die LEDs im 5x5-Raster:

```
0, 0  1, 0  2, 0  3, 0  4, 0
0, 1  1, 1  2, 1  3, 1  4, 1
0, 2  1, 2  2, 2  3, 2  4, 2
0, 3  1, 3  2, 3  3, 3  4, 3
0, 4  1, 4  2, 4  3, 4  4, 4
```



Startposition des Spielers

Das Fruit-Catcher-Spiel - Setup

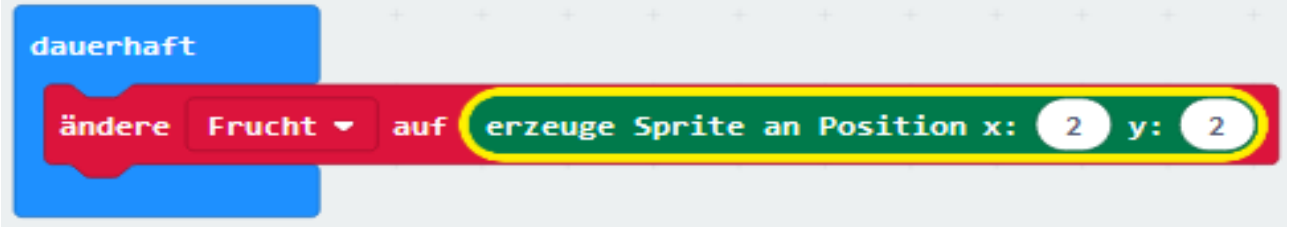
- Initialisierung des Spielstands bei 0
  - Spiel-Kategorie: [setze Punktestand auf 0]-Block am Ende im [beim Start]-Block ergänzen
- Kontrolle der Spielgeschwindigkeit mit Hilfe einer Variablen Verzögerung
  - Variablen-Kategorie: Variable Verzögerung erzeugen, dann den [Ändere Verzögerung auf 0]-Block am Ende im [beim Start]-Block ergänzen und den Wert auf 1000 ändern

Das Fruit-Catcher-Spiel – die Hauptprogramm-Schleife

- nächste Phase: Spiel selbst wird entwickelt
- analog zur Figur Spieler wird eine Figur Frucht erzeugt
- Variable `Frucht` erzeugen (Variablen-Kategorie)

  → [ändere Frucht auf 0]-Block in den [dauerhaft]-Block ziehen

  → [erzeuge Sprite an Position x:2 y:2]-Block (Spiel-Kategorie) auf die 0 ziehen

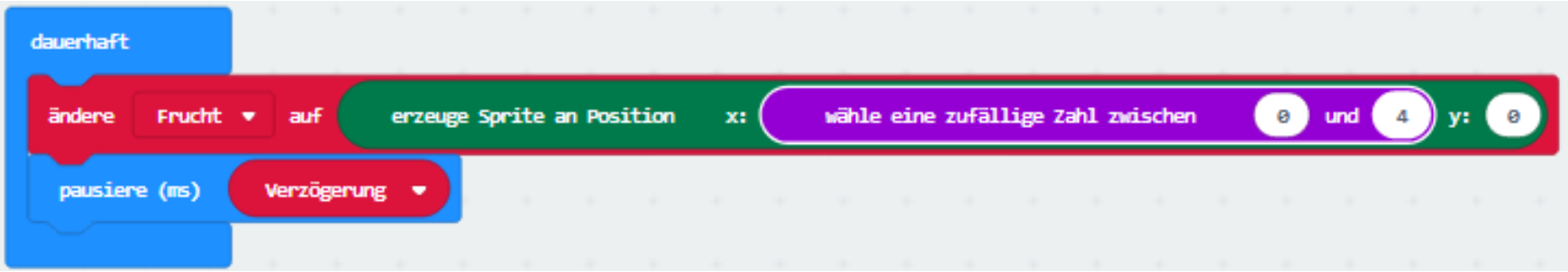


Das Fruit-Catcher-Spiel

- Position der Frucht: zufälliger Wert in der ersten Zeile
- 1. Zeile: y=0
- x-Koordinate: Zufallszahl von 0 bis 4 (Mathematik-Kategorie)

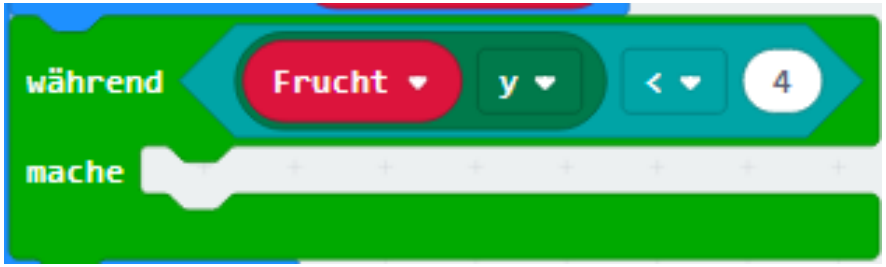
Das Fruit-Catcher-Spiel

- ein Spiel, bei dem die Früchte so schnell fallen wie möglich, wäre Millionen mal schneller als der Mensch spielen kann → eine Verzögerung muss eingebaut werden
- Grundlagen-Kategorie



Das Fruit-Catcher-Spiel - Bedingungsschleife

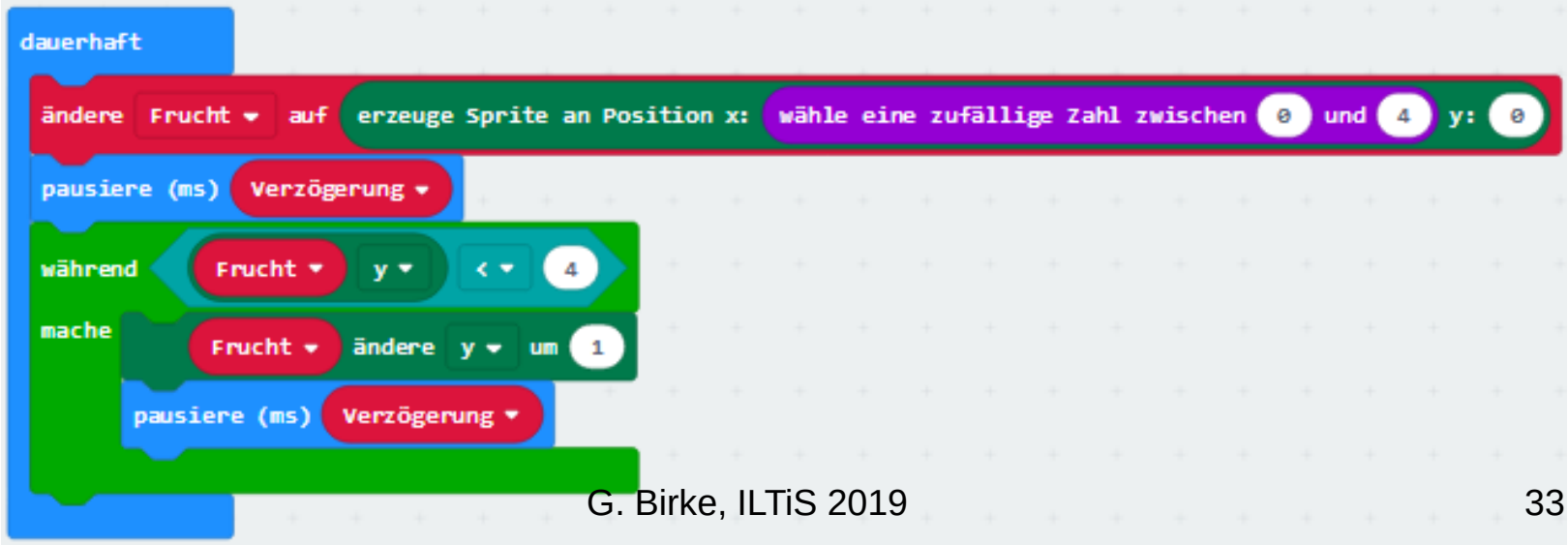
- Realisierung des Fallprozesses der Frucht: y-Koordinate muss schrittweise um 1 wachsen
- Fallprozess ist beendet, wenn y=4
- Schleifen-, Logik- und Spiel-Kategorie werden gebraucht:



Das Fruit-Catcher-Spiel - Fallprozess

- aus der Spiel-Kategorie [sprite ändere x um 1]-Block verwenden

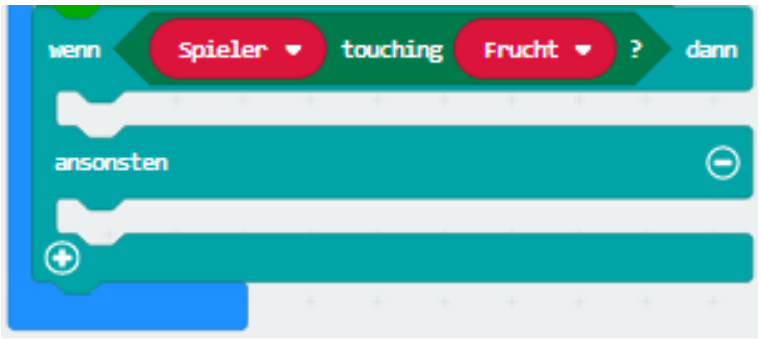
- Verzögerung einbauen, damit das Fallen sichtbar wird



Das Fruit-Catcher-Spiel – bedingte Anweisungen

- Prüfung, ob der Spieler die Frucht „berührt“ (Spieler und Frucht haben dieselbe Position)
- Logik-Kategorie →
  [wenn wahr dann ansonsten]-Block am Ende im [dauerhaft]-Block einfügen
- Spiel-Kategorie →
  die 

-Bedingung wird zwischen wenn und dann eingefügt, für sprite wird Spieler und vor ? Frucht eingebaut

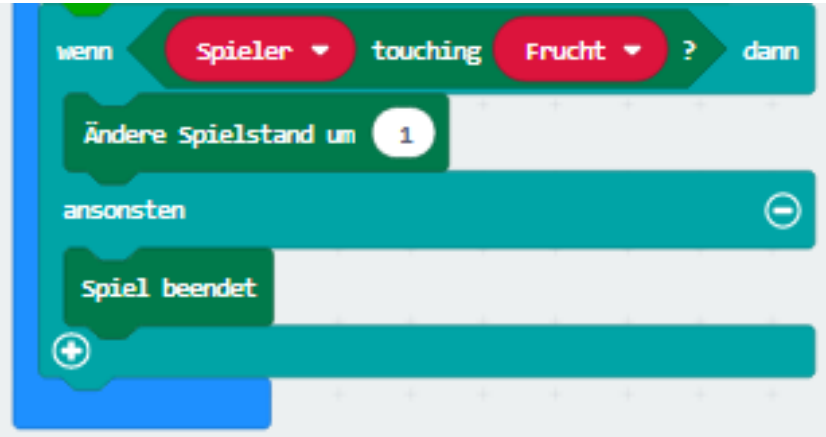


Das Fruit-Catcher-Spiel

- der 

-Block prüft, ob die Spieler-Figur (beim Start des Spiels erzeugt) die Frucht-Figur berührt hat, wenn die Frucht den Boden des Bildschirms erreicht hat
- hat der Spieler die Frucht berührt, so hat er sie „gefangen“ → der Punktestand des Spielers muss um 1 erhöht werden
- aus der Spiel-Kategorie
  - [Ändere Spielstand um 1]-Block im dann-Teil ergänzen
  - hat der Spieler die Frucht verpasst, ist das Spiel vorbei
    → [Spiel beendet]-Block im ansonsten-Teil ergänzen

Das Fruit-Catcher-Spiel

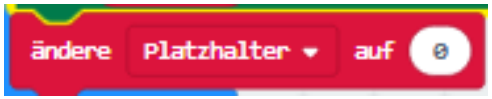


- wurde die Frucht „gefangen“, muss sie anschließend „ausgeblendet“ werden → dazu wird die Helligkeit der Figur auf 0 gesetzt
- Spiel-Kategorie: 

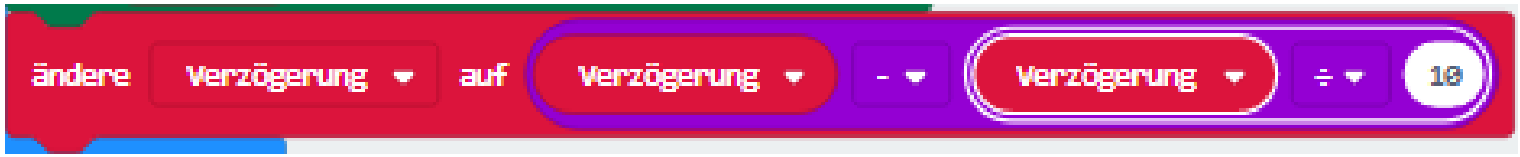
-Block am Ende des [dauerhaft]-Block einfügen und für x die Eigenschaft Helligkeit einstellen

Das Fruit-Catcher-Spiel

- um das Spiel interessanter zu machen, soll nach jedem erfolgreichen „Fang“ einer Frucht die Spielgeschwindigkeit erhöht werden
  → Variablen-Kategorie:
  → Mathematik-Kategorie: [0 - 0]- und [0 : 0]-Block



- Gesamt-Block am Ende des [dauerhaft]-Block einfügen



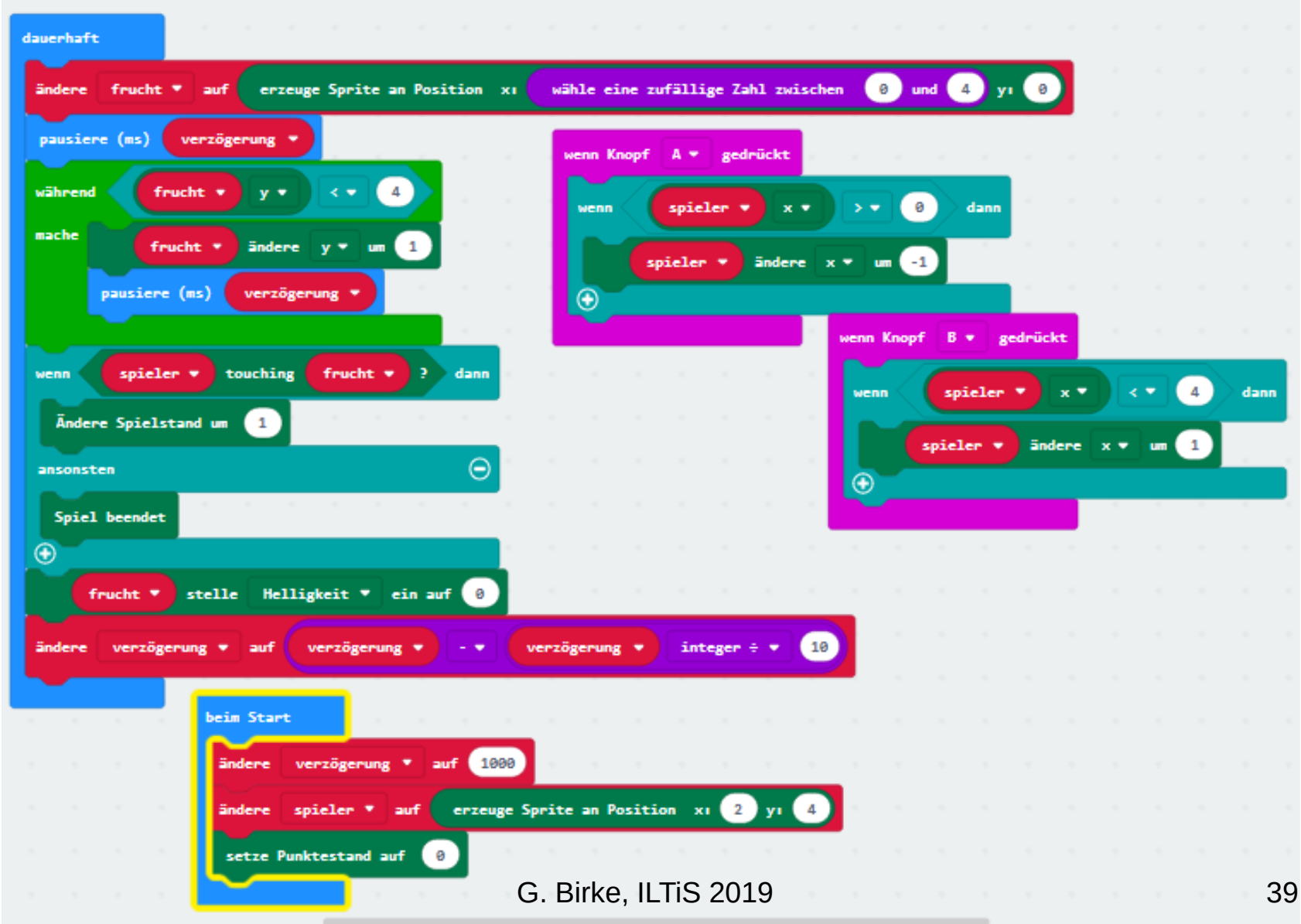
Das Spiel wird in jeder Runde um 10 % schneller.

Das Fruit-Catcher-Spiel

- Die Hauptprogramm-Schleife ist fertig.
- Es gibt noch ein Problem: Man kann die Spieler-Figur noch nicht kontrollieren.

  Aufgabe:
  Ergänze einen [wenn Knopf A gedrückt]- und einen [wenn Knopf B gedrückt]-Block (Eingabe-Kategorie), um mit Hilfe dieser Knöpfe den Spieler nach links bzw. rechts zu bewegen.

Das Fruit-Catcher-Spiel -Endlösung



Weitere Ideen

- Das Online-Schulbuch zum micro:bit: https://microbit.eeducation.at

- Für Fortgeschrittene:

  https://www.robotfreak.de/blog/der-bbc-microbit-teil-1-erste-eindruecke/

Quellen

(1) https://microbit.eeducation.at/
(2) https://www.inf-schule.de/vernetzung/calliope
(3) M. Hancl (2018):Informatische Bildung Klasse 7/8. Coding, Making und vernetzte Welten. Auflage 1. Berlin: Cornelsen
(4) https://www.fh-dortmund.de/de/fb/3/personen/lehr/hahn/medien_pw/Sensorik_Aktorik_Einfuehrung.pdf
(5) Gareth Halfacree: The Official BBC micro:bit User Guide (Englisch). Indianapolis: Wiley