

CoALA - Code a Little Animal

<Autor> Mirek Hančl

<Autorin> Julia Winckler



<Info>

<Schlagwörter> Simulation, IPO model (input-processing-output – Eingabe, Verarbeitung, Ausgabe), Messungen, Computational Thinking, Making

<Unterrichtsfächer> Naturwissenschaften, Biologie, Informatik

<Altersgruppe> 9–13 Jahre

<Hardware> Calliope mini^[1] oder BBC micro:bit^[2]

<Werkstatt A> Krokodilklemmen, roter Bastelkunststoff, USB-Kabel und Batterie für den Calliope mini, selbstklebendes Kupferband (5 mm), Pappe, Klebstoff, Schere, kleines Wasserglas, Poster mit Tierbildern

<Werkstatt B> Krokodilklemmen, USB-Kabel und Batterie für Calliope mini, Feuchtigkeitssensor (Grove Moisture Sensor), Touchsensor mit vier Fühlern (Grove I2C Touch Sensor), Grove NFC, Grove I2C hub^[3], Pappe, roter Bastelkunststoff, kleines Wasserglas, Poster mit Tierbildern

<Programmiersprache> MakeCode^[4]

<Programmierniveau> leicht

<Zusammenfassung>

Welches Kind wünscht sich kein Haustier? Um herauszufinden, welches das richtige ist, bauen Schülerinnen und Schüler einen Simulator, der mithilfe eines Einplatinencomputers und externer Sensoren die Bedürfnisse eines Haustiers nachahmt.

<Vorstellung des Konzepts>

Das Thema „Haustiere“ steht nicht nur in der Grundschule auf dem Lehrplan – auch an weiterführenden Schulen wird im Biologieunterricht besprochen, wie der Wolf zum Hund wurde, welche Grundbedürfnisse ein Haustier hat und welche Anforderungen damit an die Besitzerin oder den Besitzer gestellt werden. Oft werden dafür Texte im Schulbuch oder Lehrfilme rezipiert, denn ein echtes Tier kann für den Unterricht nicht extra angeschafft werden. Daher ist ein elektronischer Simulator für die Grundbedürfnisse eines Haustiers – Essen, Trinken, Bewegung, Streicheleinheiten und richtige Körpertemperatur – anschaulich und lehrreich zugleich.

Für die Simulation wird im Projekt weder ein fertiges Gerät eines kommerziellen Lehrmittelproduzenten, das nur vorgegebene Programme erlaubt, verwendet, noch ein simples Spielzeug eingesetzt, wie das in den 90ern weltweit erfolgreiche Tamagotchi. Stattdessen konstruieren, bauen und programmieren die Schülerinnen und Schüler mithilfe eines Einplatinencomputers, hier Calliope mini^[1] oder BBC micro:bit^[2], und Bastelmaterial wie Pappe, Kupferklebeband sowie externen Sensoren ihren eigenen Simulator in Gestalt ihres Lieblingstiers, inklusi-

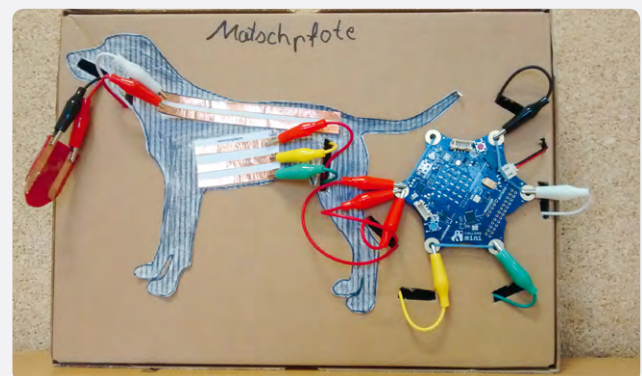
ve Bild! Dessen Grundbedürfnisse werden mit den Sensoren erfasst und durch einen selbstprogrammierten Algorithmus ausgewertet. Je nach Ablauf des Algorithmus zeigt der Tier-Simulator mit entsprechenden Smileys an, wie er sich momentan fühlt, oder spielt selbstkomponierte Melodien ab.

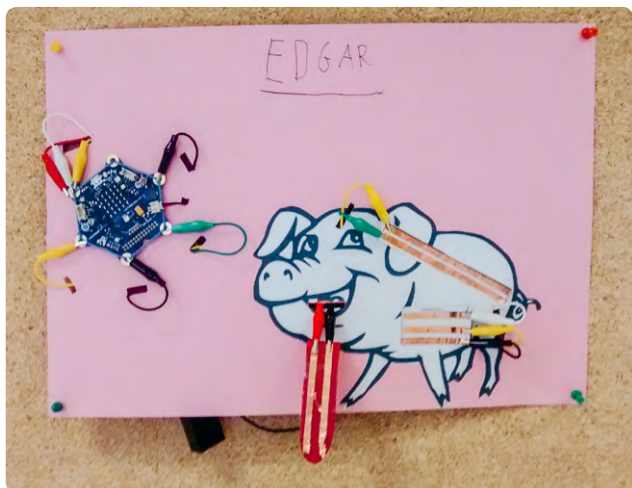
Das Projekt ist als Werkstatt konzipiert. Die praxiserprobten OER-Unterrichtsmaterialien (Open Educational Resources) bestehen aus drei Teilen. Zuerst werden die Schülerinnen und Schüler in die Grundlagen der Algorithmik und die Handhabung des Calliope mini^[1] eingeführt. Im zweiten Teil lernen sie explorativ die Grundbedürfnisse eines Haustiers kennen und einschätzen. Im dritten Teil bauen sie aus Pappe ihr Lieblingstier, stattdessen es mit dem Einplatinencomputer und passenden Sensoren aus und erstellen in einer grafischen Programmiersprache geeignete Algorithmen.

Um den unterschiedlichen Anforderungen in Primar- und Sekundarstufe gerecht zu werden, gibt es zwei Versionen: Für die Grundschule (Werkstatt A) werden die Messwertaufnahmen für Essen, Trinken und Streicheln mit leitfähigem Kupferklebeband realisiert, für die weiterführenden Schulen (Werkstatt B) werden externe Sensoren zur Feuchtigkeitsmessung (Trinken), für Multitouch (Streicheln) und zum drahtlosen Auslesen von NFC-Chips (Essen) verwendet. NFC steht für Near Field Communication. Die Messungen zur Bewegung und zur Temperatur erfolgen in beiden Versionen mit im Einplatinencomputer vorhandenen Sensoren. Sämtliche Werkstattmaterialien und Beispiele für die Programmierumgebung MakeCode^[4] sind online frei verfügbar.^[5]

<Praktische Umsetzung>

Für den Haustiersimulator suchen sich die Schülerinnen und Schüler ein Bild ihres Lieblingstiers aus oder fotografieren es selbst. Das Bild wird auf Pappe geklebt und mit Kupferklebeband (Werkstatt A) oder externen Sensoren (Werkstatt B) versehen. Das Kupferklebeband bzw. die Sensoren werden mit Drähten an die Anschlüsse des Einplatinencomputers verkabelt und ein Programm auf den Computer geschrieben, um ihn „intelligent“ zu machen. Im Folgenden wird am Beispiel „Essen“





```

wenn Knopf B gedrückt
wenn Pin P0 ist gedrückt und nicht Pin P1 ist gedrückt
dann zeige Zeichenfolge "Maus"
wenn nicht Pin P0 ist gedrückt und Pin P1 ist gedrückt
dann zeige Zeichenfolge "Wurst"
wenn Pin P0 ist gedrückt und Pin P1 ist gedrückt
dann zeige Zeichenfolge "Vogel"
wenn Pin P3 ist gedrückt
dann zeige Zeichenfolge "Wasser!"
    zeige Symbol [Icon]
    pausiere (ms) 2000
    Bildschirminhalt löschen
  
```

gezeigt, wie sich die beiden Versionen voneinander unterscheiden und wie die Programmierumgebung verwendet wird.

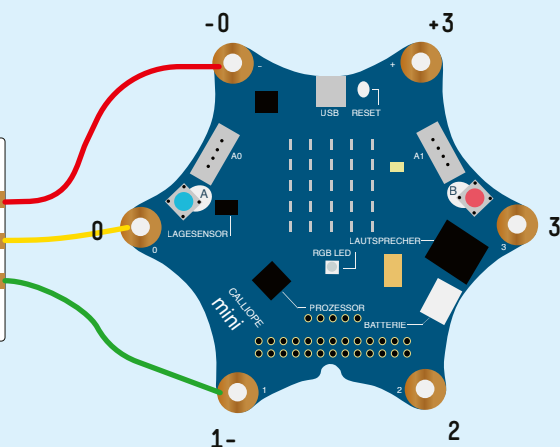
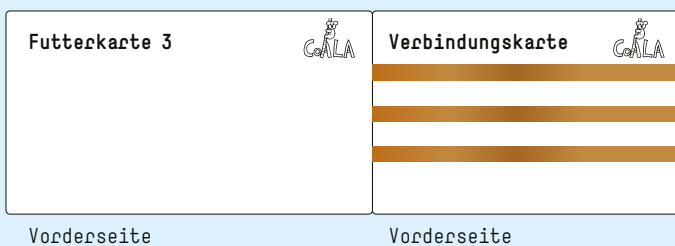
Wird ein Haustiersimulator in Mundnähe gefüttert, nutzt dieser keine Geschmackssensoren. Stattdessen „liest“ ein passender Sensor die vorgehaltene Nahrung und der Algorithmus wird über bedingte Verzweigungen so gesteuert, dass die Ausgabe dem zu erwartenden Verhalten des Haustieres entspricht. So zeigt ein Katzensimulator ein lachendes Gesicht, wenn eine Maus gefüttert wird, und ein trauriges für einen Knochen. Diese Verzweigungen sind bei beiden Versionen gleich.

Grundlegend verschieden sind dagegen die verwendeten Essen-Sensoren: In Werkstatt A wird Kupferklebeband so auf mit passenden Bildern versehenen Futterkarten geklebt, dass ein an der Zunge angebrachtes Kupferklebeband eine binärcodierte Zahl liest, wenn die Futterkarte draufgehalten wird. Da die Anschlüsse des „Lesers“ an einzelne Pins des Einplatinencomputers angeschlossen sind, kann im Algorithmus direkt abgefragt werden, ob die Pins jeweils kurzgeschlossen sind oder nicht: Die Futterkarten schließen also in unterschiedlichen Kombinationen an den Pins Stromkreise kurz.

In Werkstatt B liest ein externer Sensor mit NFC-Chip und Funkantenne drahtlos Zeichenketten aus einem NFC-Tag aus. Dieser Tag kann in einem Klebeetikett oder in einer Chipkarte untergebracht sein. Im Unterschied zur Werkstatt A wird nun nicht eine (binärcodierte) Zahl ausgelesen, sondern der Name der Nahrung, z. B. „Fisch“ oder „Knochen“. Dadurch steigen die Umsetzungsmöglichkeiten, aber auch die Komplexität deutlich an. Im Algorithmus erfolgt nun die bedingte Verzweigung durch direkten Vergleich des ausgelesenen Werts mit einer vorgegebenen Zeichenkette. Das Beschreiben des NFC-Tags erfolgt mit einer App, das Auslesen wird didaktisch reduziert mit einem einzigen Block in MakeCode^[4] realisiert, der als Erweiterung in der Programmierumgebung nachgeladen wird.

```

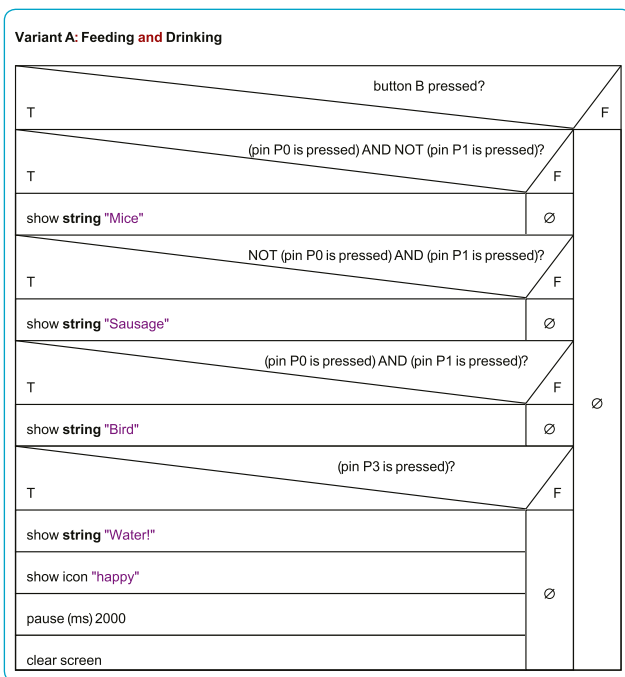
dauerhaft
wenn vergleiche lies Textnachricht aus NFC-Tag mit "Maus"
dann zeige Symbol [Icon]
wenn vergleiche lies Textnachricht aus NFC-Tag mit "Knochen"
dann zeige Symbol [Icon]
  
```



<Algorithmen in anderen Programmiersprachen>

Die Beispiele^[5] können im MakeCode Programmiereditor^[4] hochgeladen und direkt verwendet werden. Durch Umschalten der Ansicht von block- auf textbasiert wird der Quelltext nach JavaScript konvertiert und kann so bequem in anderen Programmiersprachen für Calliope mini ^[1] oder BBC micro:bit^[2] übernommen werden. Die Zusatzblöcke für MakeCode, um den Multitouch- und NFC-Sensor zu verwenden, funktionieren auch für MakeCode für den BBC micro:bit.

Schließlich werden die Beispielprogramme auch als Struktogramme auf der Webseite angeboten, sodass die Algorithmen leicht nachvollzogen und auf anderen Plattformen und Programmierumgebungen, z. B. Arduino, portiert werden können.



<Fazit>

Schülerinnen und Schüler machen sich im Projekt mit den fundamentalen Konzepten der Algorithmik bekannt: Anweisungen, Sequenzen, bedingte Verzweigungen, Schleifen, Variablen. Sie lernen diese nicht durch Auswendiglernen und Wiedergeben, sondern in einem spannenden Unterrichtsprojekt mit hohem Alltagsbezug. Dafür erstellen sie aus einfachen Materialien einen Tiersimulator, den sie nach eigenen Vorstellungen zum Leben erwecken. Die zur Verfügung gestellten Materialien vermitteln die informatischen Kompetenzen in didaktisch reduzierter Form und bieten zugleich unterschiedliche Anforderungsstufen für heterogene Lerngruppen oder höhere Jahrgänge. Beide Werkstattversionen können problemlos gemischt eingesetzt werden.

Das Material wurde neben dem Calliope mini^[1] auch mit dem BBC micro:bit^[2] erfolgreich getestet. Für die in Werkstatt B verwendeten externen Grove-Sensoren zur Messung von Feuch-



tigkeit, NFC oder Multitouch ist für den BBC micro:bit jedoch eine günstige Erweiterungslösung notwendig.^[3]

<Kooperationsmöglichkeiten>

Die CoALA-Werkstatt kann in verschiedenen Kooperationsformen eingesetzt werden. Da das Material für Grund- und Sekundarschulen vorliegt, kann hier ein Austausch stattfinden. Dieser ist nicht nur für die Schülerinnen und Schüler, sondern auch für die Lehrkräfte beider Schulformen bereichernd! Werkstatt A zielt auf einfache, logische Ja/Nein-Unterscheidungen ab, Werkstatt B auf kombinierte, komplexere Bedingungen, Variablen und Zeichenkettenoperationen.

Das Werkstattmaterial kann auch gemischt werden, um die Zusammenarbeit in einer heterogenen Lerngruppe zu fördern. Die leichter nachvollziehbaren Sensoren der Werkstatt A können dann in einer weniger leistungsstarken Lerngruppe eingesetzt werden, oder Leistungsstärkere erklären die Sensoren aus Werkstatt B (Kommunikationstraining).

Im CoALA-Pilotprojekt fand eine länderübergreifende Kooperation weiterführender Schulen statt, in der sich Lerngruppen aus Deutschland und Spanien in Videokonferenzen über ihre Haustiersimulatoren austauschten. Neben Tipps zur Problemlösung bei der Programmierung wurden vor allem die Namen und Bedürfnisse der Hausiere ausgetauscht, neben Englisch auch in der jeweiligen Landessprache. Coding in den Naturwissenschaften, Sprachkurs inklusive!

<Quellen und Hinweise>

- [1] <https://calliope.cc>
- [2] www.microbit.co.uk/home
- [3] Beim BBC micro:bit benötigt man zusätzlich das Grove Shield für den micro:bit.
- [4] <https://makecode.calliope.cc> oder <https://makecode.microbit.org>
- [5] Sämtliches Zusatzmaterial ist erhältlich auf www.science-on-stage.de/coding-materialien.

<Impressum>

<Entnommen aus>

Coding im MINT-Unterricht
www.science-on-stage.de/coding

<Herausgeber>

Science on Stage Deutschland e.V.
Am Borsigturm 15
13507 Berlin

<Revision und Übersetzung>

Translation-Probst AG

<Gestaltung>

WEBERSUPIRAN.berlin

<Illustration>

Rupert Tacke, Tricom Kommunikation und Verlag GmbH

<Text- und Bildnachweise>

Die Autorinnen und Autoren haben die Bildrechte für die Verwendung in dieser Publikation nach bestem Wissen geprüft und sind für den Inhalt ihrer Texte verantwortlich.

<Bestellungen>

www.science-on-stage.de
info@science-on-stage.de

<ISBN PDF-Fassung>

978-3-942524-60-5

Diese Publikation ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz:
<https://creativecommons.org/licenses/by-sa/4.0/>.



1. Auflage 2019

© Science on Stage Deutschland e.V.

Ein Projekt von



Hauptförderer von
Science on Stage Deutschland



Science on Stage Deutschland - The European Network for Science Teachers

... ist ein Netzwerk von Lehrkräften für Lehrkräfte aller Schularten, die Mathematik, Informatik, Naturwissenschaften und Technik (MINT) unterrichten.
... bietet eine Plattform für den europaweiten Austausch anregender Ideen und Konzepte für den Unterricht.
... sorgt dafür, dass MINT im schulischen und öffentlichen Rampenlicht steht.

Science on Stage Deutschland e.V. wird maßgeblich gefördert von think ING., der Initiative für den Ingenieurwachstum des Arbeitgeberverbandes GESAMTMETALL.

Machen Sie mit!

www.science-on-stage.de

www.facebook.com/scienceonstagedeutschland

www.twitter.com/SonS_D

Bleiben Sie informiert!

www.science-on-stage.de/newsletter

Mit freundlicher Unterstützung von

