

VPLS - Vacation Plant Life Saver

<Autor> Andreas Meier

<Autorin> Sonja van der Byl



<Info>

<Schlagwörter> automatische Pflanzenbewässerung, Verwendung eines Mikrocontrollers zum Ansteuern einer Wasserpumpe

<Unterrichtsfächer> Informatik, Naturwissenschaften, Technik

<Altersgruppe> 10–14 Jahre

<Hardware> Computer (möglichst einer pro Schülerin bzw. Schüler), Calliope mini^[4] mit Feuchtigkeits- und Temperatursensor (einer pro Kleingruppe)

<Programmiersprache> Scratch^[2] (online oder offline), Editor für Calliope mini^[3] (online)

<Programmierniveau> leicht

<Zusammenfassung>

Die Kübelpflanzen im Schulgebäude vertrocknen regelmäßig in den Ferien, weil niemand sie gießt – wir brauchen also einen Vacation Plant Life Saver! Im Rahmen dieser Reihe bauen wir den Lebensretter für unsere Schulpflanzen virtuell und real.

<Vorstellung des Konzepts>

Das Projekt ist für alle naturwissenschaftlichen Bereiche geeignet, da es sich an Programmieranfängerinnen und -anfänger richtet. Das regelmäßige Gießen einer Pflanze wird dabei im ersten Teil des Projekts virtuell umgesetzt und führt aus Sicht der Informatik in Objektorientierung, Kontrollstrukturen (Wiederholungen, Bedingungen) und das Nutzen von Variablen ein. Dabei lässt es viel Raum für eigene Programmiererfahrungen der Schülerinnen und Schüler. Als Vorbereitung arbeiten sie sich online in Scratch ein („Erste Schritte“)^[4]. (Zeitbedarf ca. 45 Minuten)

Bei der Objektorientierten Programmierung (OOP) wird alles als Objekt gesehen. Diese besitzen Eigenschaften (Attribute) und Fähigkeiten (Methoden). Hier sind das die Katze „Sprite“, die Gießkanne und die Bühne.

Jedes Objekt gehört zu einer Klasse. Alle Objekte einer Klasse haben die gleichen Eigenschaften und Fähigkeiten. Man kann eine Klasse als Blaupause (blueprint) und ein Objekt als Instanz (instance), also konkrete Realisierung, auffassen. In Scratch sind Figuren Instanzen der Klasse „Sprite“. Man spricht deshalb auch von Sprites. Die Katze mit dem Namen „Sprite“ ist ein Sprite, also eine Instanz der Klasse „Sprite“. Eine Eigenschaft von Sprites ist deren Kostüm, in diesem Fall z. B. das Bild einer Katze. Eine andere Instanz der Klasse „Sprite“ könnte als Kostüm das Bild eines Menschen haben und „Günther“ heißen. Die Katze „Sprite“ und der Mensch „Günther“ sind also

beide Objekte bzw. Instanzen der Klasse „Sprite“ oder kurz: beide sind Sprites. In Scratch gibt es nur zwei Klassen: die Bühne (stage) und Figuren (sprites). In diesem Projekt werden die zwei Figuren (sprites) Katze und Gießkanne sowie die Bühne benutzt. (© 1)



© 1: Der virtuelle Vacation Plant Life Saver

Der zweite Teil des Projekts kann auf dem ersten Teil aufbauen, ist aber auch unabhängig davon durchführbar, wenn die Schülerinnen und Schüler die oben genannten Kontrollstrukturen kennen und bereits erste Programmiererfahrungen mit dem Calliope mini^[4] gesammelt haben. Statt der Variablen kommen hier Sensoren eines Mikrocontrollers zum Einsatz, die das Ventil einer Wasserpumpe ansteuern.

<Praktische Umsetzung>

Alle benötigten Materialien wie Arbeitsblätter und (zerlegte) Programme stehen digital zur Verfügung.^[5]

<Teil 1: Virtuelles regelmäßiges Gießen einer Pflanze>

Arbeitsschritt 1: Erstellen eines Programms, das nur die Zeit als Variable berücksichtigt; Kennenlernen von einseitigen Bedingungen und Wiederholungen (Zeitbedarf: 180 Minuten)

Nach dem Einstieg in das Problem (Wie könnte ein virtueller Vacation Plant Life Saver funktionieren?) sollten sich die Schülerinnen und Schüler zunächst Gedanken über die Grundstruktur eines Gießprogramms machen, diese in Form eines „Kochrepts“ (Algorithmus) notieren und ihre Ideen gegenseitig testen. Dann erst kann eine gemeinsame Grundstruktur für ein Gießprogramm festgelegt werden (siehe Arbeitsblatt 1^[5]). Diese Phase dient dazu, sich mit der grundlegenden Struktur des Programms, das entstehen soll, auseinanderzusetzen. Die Begriffe „Anweisungsfolge“, „Wiederholung“ und „Bedingung“ ergeben sich dabei aus dem Sachzusammenhang.

Erst jetzt wird der Algorithmus in die Programmiersprache Scratch^[2] übersetzt, indem die Schülerinnen und Schüler zu-

nächst ein in seine Einzelteile zerlegtes Programm^[5] zu einem funktionierenden Programm zusammensetzen.

So lernen sie die Funktionsweise von Scratch intensiver kennen:

- ↳ Objektorientierung (jede Figur hat ihr eigenes Skript, auch die Bühne)
- ↳ Strukturierung (Wie sieht die Struktur einer „Bedingung“ und einer „Wiederholung“ in Scratch aus?)
- ↳ Skript/Kostüm/Klänge können jeder einzelnen Figur zugewiesen werden

Zusätzlich wird die gemeinsam erarbeitete Grundstruktur des Gießprogramms vertieft, indem die Schülerinnen und Schüler darüber nachdenken, in welcher Reihenfolge sie die vorgegebenen Puzzleteile anordnen müssen, damit das Programm funktioniert. Insbesondere die Frage, welche Anweisungen sich innerhalb der Zählschleife (☉ 2 & 3) befinden müssen, kann durch Ausprobieren beantwortet werden.

☉ 2

☉ 3

Auf Grundlage des zusammengesetzten Programms dürfen die Schülerinnen und Schüler nun ein eigenes Programm erstellen, das in Abhängigkeit von der Zeit die Katze „Sprite“ losschickt, um die Pflanze auf der Bühne zu gießen. Dazu bekommen sie lediglich das Ausgangsbühnenbild als Datei zur Verfügung gestellt.^[5]

Dieser Schritt dient dazu, die Schülerinnen und Schüler zu ermutigen, die Programmiersprache eigenständig zu erforschen, Ideen auszuprobieren und kreativ zu arbeiten. Das ist wichtig, weil Anfängerinnen und Anfänger nur dann mit Spaß programmieren lernen, wenn sie das Gefühl haben, sich in der entsprechenden Programmiersprache (dem eigenen Wissensstand entsprechend) gut auszukennen.

Arbeitsschritt 2: Schrittweises Erstellen eines Programms, das die Variablen „Wasserstand“ und „Temperatur“ berücksichtigt
 [Zeitbedarf: 270 Minuten]

Als Einstieg in den zweiten Arbeitsschritt denken die Schülerinnen und Schüler über weitere Faktoren nach, die festlegen, wie oft eine Kübelpflanze gegossen werden muss. Die „Raumtemperatur“ und der „Wasserstand“ im Kübel spielen hier sicherlich eine Rolle als sich verändernde, also variable Größen. [Arbeitsblatt 2^[5]]

Die Schülerinnen und Schüler erhalten im Anschluss ein funktionierendes Programm, bei dem sich die Skripte für die Katze „Sprite“ und die Gießkanne kaum verändert haben.

Allerdings steuert das Skript für die Bühne jetzt die neu definierte Variable „Wasserstand“, die den Timer ersetzt. Auch gießt „Sprite“ die Pflanze nur einmal. Die Schülerinnen und Schüler sind also aufgefordert, sich einerseits Gedanken darüber zu machen, wie die Variable „Wasserstand“ erzeugt wurde und wie sie die Aktivität der Katze steuert. Andererseits sollen sie versuchen, das Problem zu lösen, dass nur einmal gegossen wird. Falls nötig steht ihnen hierfür eine Hilfsdatei zur Verfügung.^[5]

Durch die vorläufige Nutzung von nur einer Variablen bleibt das Programm übersichtlich. Die Organisation mehrerer Variablen würde sicher zu Anfang überfordern.

Die hier einzufügende Wiederholungsstruktur ist komplexer als zuvor, da mit ihr eine Bedingung verknüpft ist (☉ 4). Die Schülerinnen und Schüler müssen sich genau überlegen, was unter welcher Bedingung wie oft zu wiederholen ist.

☉ 4

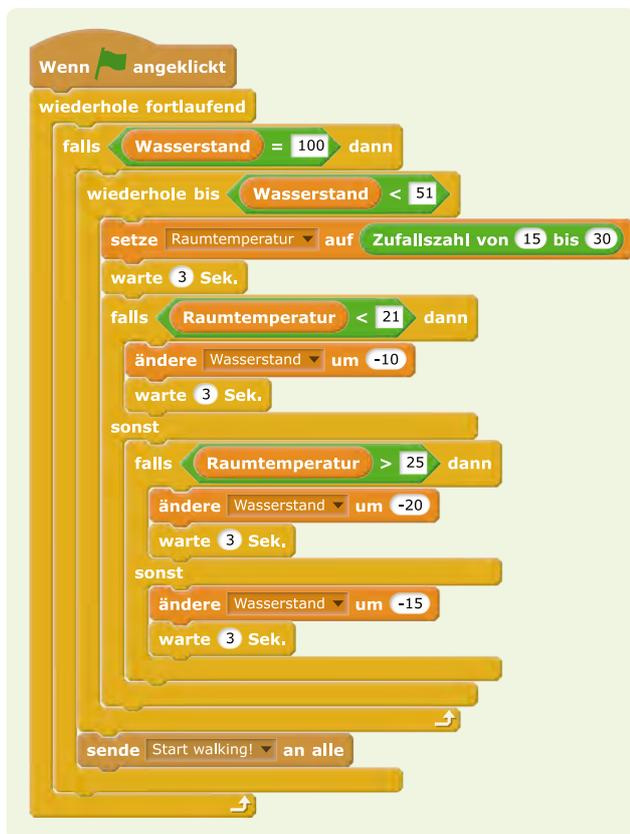
Die Fähigkeit zu strukturieren ist beim Erlernen jeder Programmiersprache von grundlegender Wichtigkeit. Hier wird sie auf

eine sehr spielerische Art und Weise vermittelt, denn die Schülerinnen und Schüler können einfach ausprobieren, was passiert, wenn sie eine falsche Wiederholungsstruktur nutzen bzw. nicht an die notwendige Bedingung zum Abbruch der Wiederholung denken.

Für Schülerinnen und Schüler mit rascher Auffassungsgabe besteht am Ende dieser Arbeitsphase die Möglichkeit, das Programm abzuändern und eigene Ideen auszuprobieren.

Erst in diesem sich anschließenden Schritt spielt auch die Variable „Temperatur“ im Programm eine Rolle. Ihr Wert wird per Zufallszahlengenerator zwischen 15 °C und 30 °C festgelegt. Der Wasserstand im Kübel ändert sich dann in Abhängigkeit von diesem Wert. (Arbeitsblatt 3^[5])

Da die Programmstruktur des „neuen“ Gießprogramms somit sehr komplex ist, soll zunächst wieder ein zerlegtes Programm zu einem funktionierenden Programm zusammengesetzt werden. Dabei werden der Zufallszahlengenerator und die zweiseitige Bedingung eingeführt, ebenso wie das Arbeiten mit Variablen und das Abfragen von Bedingungen wiederholt und vertieft. Erneut müssen sich die Schülerinnen und Schüler genau überlegen, was unter welcher Bedingung wie oft wiederholt werden sollte. (© 5)



© 5

Auch hier gibt es im Anschluss die Möglichkeit, das entstandene Programm zu individualisieren, wobei die Schülerinnen und Schüler entsprechend ihrer Ideen und erworbenen Programmierfähigkeiten arbeiten können.

Unbedingt sollten sie am Ende dieses ersten Projektteils, bei dem sie den Vacation Plant Life Saver virtuell umgesetzt haben, möglichst viele ihrer Programmierergebnisse präsentieren. Damit werden ihre Ideen gewürdigt und ihre Leistung anerkannt.

<Teil 2: Regelmäßiges Gießen einer Pflanze gesteuert durch einen Mikrocontoller>

Im Laufe des Projekts wurde schnell deutlich, dass sich einige Schülerinnen und Schüler nicht mit der virtuellen Lösung des Gießproblems zufriedengeben. Sie fragen nach Möglichkeiten, reale Pflanzen mithilfe ihres Programms wirklich zu gießen. Mit einem Mikrocontroller lässt sich das virtuelle Gießprogramm relativ leicht zur Steuerung eines realen VPLS verwenden, insbesondere da viele dieser Minicomputer ebenfalls mit Scratch oder einer vergleichbaren App programmierbar sind.

In unserem Projekt kommt der Calliope mini^[4] zum Einsatz, eine Weiterentwicklung des BBC micro:bit^[6], der zusätzlich über leicht nutzbare „Plug-and-play“-Funktionen wie Berührungssensoren und Motorenanschlüsse verfügt. Aber auch mit allen anderen in der Schule üblicherweise verwendeten Mikrocontrollern wie LEGO EV3, LEGO NXT, Arduino, Raspberry Pi, Teensy, ... kann die Steuerung des VPLS erfolgen^[5].

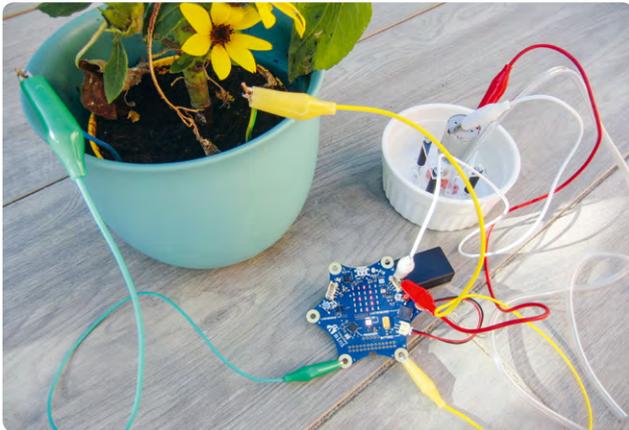
Um den Calliope mini zu programmieren, wird er einfach per USB-Kabel mit dem Computer verbunden. Als Programmieroberfläche eignet sich Open Roberta Lab^[7], welche verschiedene Mikrocontroller unterstützt (alternative Editoren sind im Zusatzmaterial aufgeführt^[5]). Die Programmieroberfläche ist in verschiedenen Sprachen verfügbar. Diese lassen sich schnell mit einem Klick auf das Erdkugel-Symbol ändern, nachdem man den Mikrocontroller ausgewählt hat, in unserem Fall den Calliope mini.

Eine einfache Version des VPLS könnte folgendermaßen funktionieren:

1. Die Feuchtigkeit der Erde wird ständig gemessen.
2. Ist die Erde zu trocken, wird eine bestimmte Menge Wasser über eine Pumpe zugeführt, bis der Boden feucht genug ist.

Der Mikrocontroller muss also die Feuchtigkeit des Bodens messen und den Motor einer Wasserpumpe betreiben können.

Der Calliope mini verfügt über vier Berührungssensoren, die aus physikalischer Sicht lediglich die elektrische Leitfähigkeit zwischen den Anschlusspunkten messen. Da Wasser elektrischen Strom leitet, besitzt feuchter Boden eine höhere Leitfähigkeit als trockener Boden. Als Sensoren verwendet man zwei Kupferdrähte, die man mit etwas Abstand in den Blumentopf steckt. Diese werden durch Krokodil-Kabel mit dem Calliope mini an den Eckkontakten [-, P1] verbunden. Der Sensor „analoger Pin“ P1 liefert nun einen Wert zwischen 0 und 1023. Sinkt die Leitfähigkeit unter einen bestimmten Wert, dann wird die Pumpe aktiviert, die die Pflanze mit Wasser versorgt. [📷 6]



📷 6: Calliope mini steuert den VPLS

Wesentlicher Bestandteil der Pumpe ist ein kleiner Elektromotor, der je nach benötigter Leistung direkt oder mithilfe eines zusätzlichen Motortreibers an den Calliope mini angeschlossen wird. Zwei Motorausgänge [A, B] stehen unter dem Menü Aktion/Bewegung zur Verfügung. Mit dem Wert „Tempo %“ lässt sich die Wassermenge der Pumpe einstellen.

Sowohl beim Bau der Pumpe als auch beim Aufbau des Motortreibers ist etwas handwerkliches Geschick gefragt, im Online-Material [5] stellen wir entsprechende Bauanleitungen zur Verfügung. Die Kosten für die Pumpe belaufen sich auf wenige Euro.

Die Arbeit mit dem realen VPLS bietet den Schülerinnen und Schülern zahlreiche Fragestellungen zur Optimierung und Vertiefung, beispielsweise:

- ↳ Benötigen unsere Pflanzen viel oder wenig Wasser?
Wie groß muss der gesamte Wasservorrat sein?
- ↳ Welcher ist der beste Gießzeitpunkt? Ist es besser nur einmal am Tag viel zu gießen oder mehrmals wenig?
- ↳ In welcher Bodentiefe sollten die Feuchtigkeitssensoren am besten messen? Welches ist der beste Abstand für die Sensoren?

↳ Wie lange reicht die Energieversorgung des VPLS? Lässt sich die Energieeffizienz der Pumpe steigern, sodass der VPLS während der gesamten Ferien gießt?

Den Schülerinnen und Schülern bieten sich verschiedenste Ansätze aus den Bereichen Biologie und Physik zur Weiterarbeit am VPLS-Projekt an. Interessant wäre sicher auch eine Anbindung und Überwachung über das Internet. Dies ginge weit über die von uns mit dem VPLS angestrebte Einführung ins Programmieren hinaus, zeigt aber die Möglichkeiten, die sich aus einer einfachen Fragestellung ergeben können.

<Übertragbarkeit in andere Programmiersprachen>

Sehr leicht lässt sich das Projekt in Snap! [8], eine Weiterentwicklung von Scratch [2], übertragen. Programmbeispiele sind online erhältlich [5]. Diese beiden Programmiersprachen eignen sich besonders gut für ein Projekt, das sich an Programmieranfängerinnen und -anfänger richtet, weil sie anschaulich sind und dazu ermutigen, Ideen per „drag and drop“ einfach auszuprobieren.

<Fazit>

Programmieranfängerinnen und -anfänger machen im Laufe dieses alltagsnahen Projekts ihre ersten Programmiererfahrungen und lernen wichtige Grundlagen einer Programmiersprache kennen. Dabei liegt der Fokus nicht darauf, deren Syntax und das Vokabular zu lernen, sondern die Wirkung bestimmter Strukturen auszuprobieren: „Was funktioniert wie und warum?“. Fehler sind hier durchaus erwünscht, denn sie lassen sich in der Regel leicht finden und erklären und tragen somit dazu bei, zu verstehen, wie eine Programmiersprache funktioniert.

Zudem gibt der vorgegebene Rahmen den Schülerinnen und Schülern einerseits Sicherheit (wer unsicher ist, löst eben nur das Puzzle), andererseits gibt es für diejenigen, die die „Pflichtaufgaben“ schnell erledigen viel Raum für Kreativität.

Aus dem Projekt heraus sind einige Ideen entstanden, z. B. einen „echten“ Gießroboter zu bauen oder ein Gießspiel zu entwickeln. In jedem Fall haben die Schülerinnen und Schüler positive erste Programmiererfahrungen gemacht, die hoffentlich lange nachwirken.

Schwierig war mitunter der an unserer Schule vorgegebene Zeitrahmen von 45 Minuten. Manchmal hat allein der rein organisatorische Teil (Anmelden am Rechner, Dateien öffnen, Dateien speichern, Abmelden vom Rechner) 15 Minuten gekostet, sodass die echte Programmier- und Arbeitszeit zu kurz kam. Man kann als Lehrkraft für ein Projekt einen Lehrerzugang bei Scratch [2] anfordern, unter dem man eine Klasse einrichten und Materialien hinterlegen kann, was sehr hilfreich ist.

<Kooperationsmöglichkeiten>

Da es sich beim VPLS um eine Einführung ins Programmieren handelt, bieten sich zunächst kaum Möglichkeiten einer Kooperation an. Sobald das Projekt auf die reale Steuerung eines Mikrocontrollers übertragen wird, sind Kooperationen unter Schülerinnen und Schülern hilfreich, da sich der Schwierigkeitsgrad des Problems durch den Einsatz von zusätzlichen Komponenten (Sensoren, Pumpe) erhöht. Möglich wäre etwa, dass ältere Schülerinnen und Schüler beim Bau der Pumpe helfen. Auch eine internationale Kooperation zwischen Schulen, die am VPLS-Projekt arbeiten, ist denkbar.

Es wäre möglich, gemeinsam eine Datenbank für unterschiedliche Pflanzen zu erstellen, sodass sich der VPLS leicht auf die individuellen Eigenschaften verschiedener Pflanzenarten anpassen lässt. Gelingt die Anbindung des VPLS über das Internet könnten sogar internationale Gieß-Patenschaften entstehen!

<Quellen und Hinweise>

- [1] www.calliope.cc
- [2] www.scratch.mit.edu
- [3] www.calliope.cc/los-geht-s/editor
- [4] https://scratch.mit.edu/projects/editor/?tip_bar=getStarted [29.11.2018]
- [5] Sämtliches Zusatzmaterial ist erhältlich auf www.science-on-stage.de/coding-materialien.
- [6] www.microbit.co.uk/home
- [7] <https://lab.open-roberta.org>
- [8] <https://snap.berkeley.edu>

<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

