Zufallsexperimente mit Calliope mini simulieren und Reflexionen anregen am Beispiel des manipulierten Würfels

Frederik Dilling – Julian Plack

In diesem Beitrag geht es um das Programmieren des Mikrocontrollers Calliope mini im Kontext von Zufallszahlen. Hierzu wird zunächst eine kurze Einführung in das Themenfeld der Pseudozufallszahlen im Kontext von Algorithmen sowie ein Überblick über wichtige Funktionen der grafischen Programmierumgebung MakeCode gegeben. Außerdem wird eine kurze Einführungsaufgabe für Schüler/innen vorgestellt. Der Kern des Artikels liegt in der Präsentation einer Schüler/innen-Aktivität, in welcher ein manipulierter Spielwürfel mit dem Calliope mini simuliert werden soll. Die Ausführungen werden unterstützt durch Ausschnitte aus einem Gespräch mit einem Schüler der 9. Jahrgangsstufe.

1 *Zufallszahlen und -experimente in MakeCode für Calliope mini*

Unter dem Zufall wird im Allgemeinen etwas verstanden, was man nicht vorhersehen kann (siehe z.B. Hehl, 2021). Es handelt sich hierbei eigentlich nicht um einen mathematischen Begriff und der Zufall lässt sich auch nicht objektiv definieren. Dennoch tritt der Begriff zuweilen im Mathematikunterricht auf, insbesondere im Rahmen von sogenannten Zufallsexperimenten. Hierbei handelt es sich um Prozesse, die mehrere mögliche Ausgänge, sogenannte Ergebnisse, haben und bei denen nicht vorhergesehen werden kann, welches Ergebnis auftritt. Vereinfacht gesagt sind Zufallsexperimente solche Prozesse, welche wir mit den Mitteln der Wahrscheinlichkeitsrechnung beschreiben.

Unter dem Begriff der Zufallszahl wird eine Zahl verstanden, welche man als Ergebnis eines Zufallsexperiments erhält. Als Generatoren von echten Zufallszahlen lassen sich zum Beispiel Würfel oder Münzen verwenden. Da diese Verfahren aber meist recht aufwändig sind, werden für viele Anwendungen, insbesondere im Bereich der Informatik, sogenannte Pseudozufallszahlen genutzt. Hierbei handelt es sich um scheinbar zufällige Zahlen, welche aber eigentlich Ergebnisse von Algorithmen sind. Der Algorithmus liefert eine deterministische und jederzeit reproduzierbare Folge von Zahlen. Die meisten Rechner arbeiten dabei mit demselben Verfahren. Beginnend mit einer Zahl wird eine Folge von Pseudozufallszahlen induktiv durch eine Vorschrift definiert. Die dabei entstehenden Zahlen sind von einer Zufallsfolge nahezu nicht unterscheidbar (vgl. Behrends, 2013).

Durch Algorithmen erzeugte Pseudozufallszahlen spielen in der Informatik im Kontext des Programmierens eine bedeutende Rolle. Dies können auch bereits Schüler/innen beim Arbeiten in einfachen blockbasierten Programmierumgebungen feststellen. Wir wollen in diesem Artikel einige Beispiele zum Mikrocontroller Calliope mini und der Programmierumgebung MakeCode vorstellen, in denen der Umgang mit Algorithmen und Pseudozufallszahlen im Vordergrund steht und die im Mathematik- aber auch Informatikunterricht als Lern- und Reflexionsanlässe über den Wahrscheinlichkeitsbegriff genutzt werden können.

Der Calliope mini besteht aus einer Platine mit Mikroprozessor sowie weiteren elektrotechnischen Bauelementen, wie z.B. einer Fläche aus 5 mal 5 einfarbigen LEDs. Das Ziel des Calliope mini ist, bereits Kindern ab der dritten Klasse einen Zugang zum Programmieren zu ermöglichen und dabei auch technische Bauteile einzubeziehen. Mit dem Mikrocontroller können Schüler/innen arbeiten, ohne dass beispielsweise Kurzschlüsse entstehen oder andere Fehler auftreten. Außerdem haben Lehrkräfte die Möglichkeit, ohne größeren Vorbereitungsaufwand mit der vorhandenen, vergleichsweise umfangreichen Hardware zu arbeiten.

Mit Bezug auf die Software lässt sich sagen, dass der Calliope mini mit verschiedenen Programmierumgebungen bedient werden kann, welche jeweils unterschiedliche Zugänge ermöglichen. Es kann zwischen grafischen Programmieroberflächen und reinen Textprogrammierungen gewählt werden. Wir verwenden in diesem Beitrag die Programmierumgebung MakeCode (https://makecode.calliope.cc/). Hierbei handelt es sich um eine Webapplikation, welche sowohl block- als auch textbasierte Programmierung ermöglicht, wobei wir uns auf die Befehle der Blockprogrammierung beschränken. Es ist keine Installation oder die Erstellung eines Benutzerkontos notwendig. Mit MakeCode kann auch ohne Internetverbindung gearbeitet werden. Außerdem lassen sich die Algorithmen direkt im Programm an einem virtuellen Calliope mini testen.

Die in MakeCode zur Verfügung stehenden Codeblöcke sind nach verschiedenen Kategorien geordnet. Unter „Grundlagen“ werden zum Beispiel Blöcke gefasst, mit denen sich Texte, Zahlen oder Zeichen auf der LED-Fläche anzeigen lassen. Unter „Eingabe“ lassen sich die Blöcke finden, welche Vorgänge als Auslöser für Algorithmen darstellen. Hierzu zählen zum Beispiel das Drücken eines bestimmten Knopfes oder das Schütteln des

Calliope mini. Ebenfalls grundlegend für Programmierungen sind die Bereiche „Schleifen“ mit Wiederhole- oder Während-X-mache-Y-Blöcken sowie „Logik“ mit Wenn-ansonsten-, Größer/Kleiner/Gleich-, wahr/falsch- oder und/oder/nicht-Blöcken. Im Bereich „Variablen“ lassen sich Variablen definieren und ändern.

Für diesen Artikel steht der Bereich „Mathematik“ im Fokus. Hier lässt sich eine Vielzahl verschiedener Befehle finden wie Grundrechenarten, Minimal- und Maximalwerte, Wurzeln und trigonometrische Funktionen oder Runden. Hier lässt sich auch der Befehl zur Festlegung von Pseudozufallszahlen finden. Mit „wähle eine zufällige Zahl von X bis Y“ können (durch einen Algorithmus erstellte) gleichverteilte Zahlen zwischen zwei gewählten Grenzen (einsetzbar für X und Y) generiert werden. Im folgenden Artikel soll am Beispiel einer Schüleraktivität zum manipulierten Spielwürfel aufgezeigt werden, welches Potential die Programmierung von Calliope mini für den Mathematikunterricht im Kontext der Wahrscheinlichkeitsrechnung bietet.

2 Eine Aufgabe zum Einstieg

Zum Einstieg können die Schüler/innen einen einfachen Würfelwurf simulieren (siehe Code in Abb. 1). Dabei lernen sie die Grundlagen der Programmierung mit MakeCode für Calliope mini kennen. Zunächst benötig man einen Eingabe-Block – es bietet sich der Block „wenn geschüttelt“ an, da dies dem Werfen eines Würfels am nächsten kommt. Es können aber auch beliebige andere Eingabeparameter gewählt werden.

Auf die Eingabe folgt die Erstellung einer neuen Variable, welche wir in unserem Beispiel „Ergebnis“ nennen. Diese wird mit einem „setze *Variable* auf“-Befehl als Zufallsvariable zwischen 1 und 6 definiert. Anschließend wird mit einem „zeige Text“-Block im Algorithmus der Wert der Variable „Ergebnis“ – also eine Ziffer zwischen 1 und 6 – auf dem Display des Calliope mini angezeigt.

Mit diesem einfachen Algorithmus lässt sich bereits der Würfelwurf simulieren. Schüttelt man den Calliope mini, erscheint auf dem Display eine Zufallszahl zwischen 1 und 6. Durch das Programmieren können die Schüler/innen bereits wesentliche Elemente eines Algorithmus kennenlernen: Auf eine Eingabe (das Schütteln des Calliope mini) folgt eine feste Reihenfolge von Schritten (Zufallsvariable bestimmen, Text anzeigen), die schließlich zu einer Ausgabe führt (Text auf dem Display).

Abb. 1. Algorithmus für einen einfachen Würfelwurf in MakeCode

Der einfache Würfelwurf-Simulator kann unter dem folgenden Link getestet und verändert werden: https://makecode.calliope.cc/_6kdHUbgyme91

Der einfache Würfel kann von den Schüler/inne/n auf verschiedene Weise verändert werden. So lässt sich beispielsweise anstelle des üblichen 6-seitigen Würfels eine andere Anzahl an gleichverteilten Ergebnissen wählen. Als haptische Würfel in Gesellschaftsspielen sind entsprechend den platonischen Körpern auch 4-, 8-, 12- und 20-seitige in Gebrauch. Der Calliope-Würfel ermöglicht diese, aber auch jede andere Gleichverteilung – z.B. einen „Würfel“ mit 7 Seiten. Dies kann für die Schüler/innen einen ersten Schritt hin zu einem formaleren Verständnis des Konzepts der Laplace-Wahrscheinlichkeit darstellen.

Eine weitere Möglichkeit der kreativen Veränderung des Würfel-Algorithmus kann die Veränderung der Ausgabe auf dem Display sein. So lässt sich anstelle der Anzeige der Ziffern auch eine Darstellung als Punktmuster wählen (siehe Abb. 2).

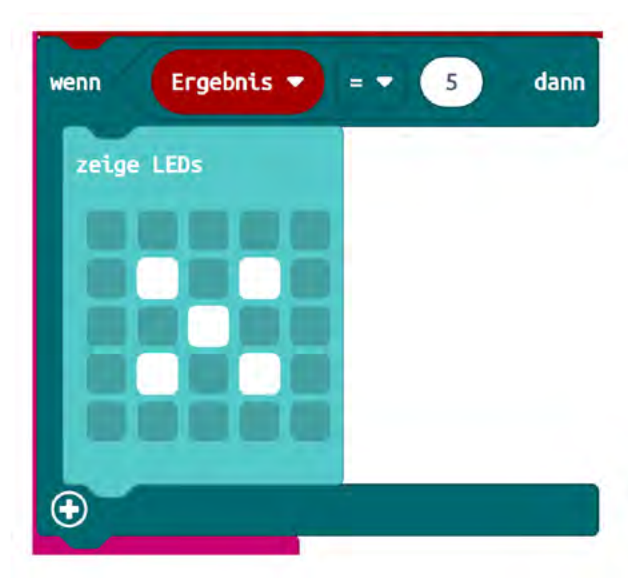

Abb. 2. Würfel-Punktmuster im Algorithmus für den Calliope mini

3 Programmieren eines manipulierten „Würfels“

Ein spannendes Thema im Mathematikunterricht ist die Betrachtung von manipulierten Zufallsgeräten. Bekannt sind zum Beispiel „gezinkte“ Würfel, welche aussehen wie normale Würfel mit einer annähernden Gleichverteilung der Ergebnisse, tatsächlich aber deutlich häufiger bestimmte Zahlen zeigen. Dies wird bei haptischen Würfeln meist durch ein Gewicht („gezinkt“) auf der entsprechenden Seite des Würfels erzielt.

In MakeCode können ausgehend von gleichverteilten Zufallsvariablen Algorithmen erstellt werden, die nicht gleich-

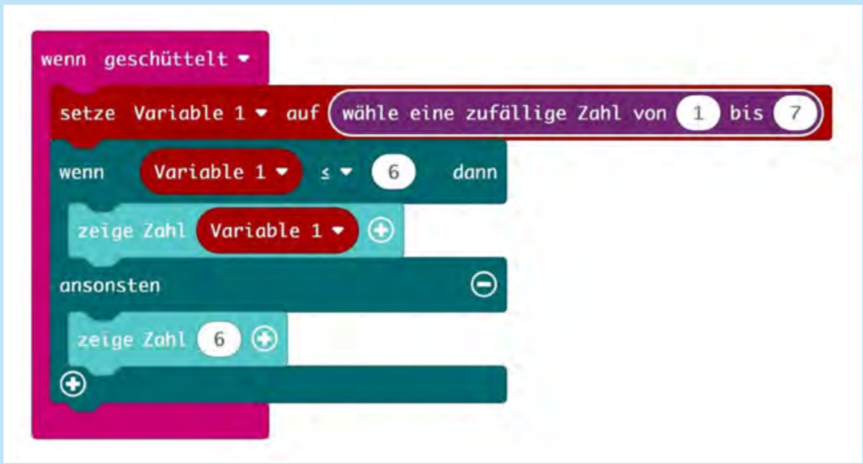

Abb. 3. Algorithmus für einen manipulierten Würfelwurf in MakeCode

verteilte Variablen ausgeben. Ein Beispiel hierfür ist in Abb. 3 zu sehen (Download: https://makecode.calliope.cc/_fDA0P-P5aXK3f).

Eine Variable wird als Zufallszahl zwischen 1 und 7 definiert. Hieran knüpft ein „wenn-ansonsten"-Block an (auch bekannt als if-else-Befehl). Wenn die zuvor zufällig bestimmte Variable einen Wert zwischen 1 und 6 hat, wird sie als Zahl auf dem Display angezeigt. Wenn dies nicht der Fall ist, also die Zahl größer als 6 ist (in unserem Beispiel nur die Zahl 7), wird laut dem Algorithmus immer die Zahl 6 angezeigt. Auf diese Weise erhält man einen Würfel, bei dem die Zahl 6 doppelt so wahrscheinlich auf dem Display auftritt wie die übrigen Zahlen. Erhöht man das Intervall der Zufallszahlen im Algorithmus, so wird die Häufigkeit der Zahl 6 auf dem Display im Vergleich zu den anderen Zahlen noch größer.

Das Besondere an der Calliope-Simulation des manipulierten Würfels ist, dass man es dem Mikrocontroller äußerlich nicht ansieht, ob der Würfel manipuliert ist oder nicht. Hierfür ist ein frequentistischer Zugang notwendig, bei dem die Schüler/innen die Simulation sehr häufig wiederholen und die einzelnen aufgetretenen Ergebnisse notieren. Eine andere Möglichkeit liegt in der Analyse des zugrundeliegenden Algorithmus. Neben dem oben beschriebenen Algorithmus sind viele weitere Algorithmen denkbar, welche die gleichverteilte(n) Zufallsvariable(n) so verändern, dass schlussendlich nicht gleichverteilte Ergebnisse entstehen (hier einige Beispiele:
https://makecode.calliope.cc/_cpiXM59XLbeL;
https://makecode.calliope.cc/_HqxhWHKmfhyA;
https://makecode.calliope.cc/_aPKHD40iR9eE).
Eine Aufgabe für die Schüler/innen könnte es sein, die Manipulation in ihrem Algorithmus möglichst gut vor den anderen Schüler/inne/n zu verstecken. Dies ist eine offene Aufgabenstellung, welche Problemlösestrategien und Wissen aus dem Bereich der Kombinatorik fordern und fördern soll. Die verschiedenen Lösungen können dann zu spannenden Bedeutungsaushandlungen zwischen den Schüler/inne/n über den Wahrscheinlichkeitsbegriff führen (vgl. Pielsticker, 2019).
Wir wollen an dieser Stelle eine kurze Beispielsequenz zeigen, in welcher der Schüler Marco einer 9. Klasse den in Abb. 3 dargestellten Algorithmus analysiert, um die Manipulation aufzudecken bzw. zu erklären.

> Marco: Wenn geschüttelt, wird eine Zahl von 1 bis 7 gemacht. Warum da jetzt ne 7 ist, weiß ich noch nicht ganz. Ok. Und wenn die Variable 1, ist ja diese Zahl von 1 bis 7, kleiner oder gleich die 6 ist, dann wird Variable 1 gezeigt. Das heißt, also alle Zahlen 1 bis 6 werden angezeigt, ok genau. Und dann wird nochmal 1 bis 7/ Ansonsten zeige Zahl 6. Das müsste manipuliert/ Nee, es gibt ja gar kein ansonsten, weil es kann ja nur 1 bis 6 sein. Also kleiner als gleich heißt ja 6 und alles was da drunter ist. Das heißt 1, 2, 3, 4, 5 und auch die 6. Deshalb glaub ich, das ist kein manipulierter Würfel.

Zunächst beschreibt Marco den Algorithmus. Dabei fällt ihm auf, dass anders als beim „normalen" Würfel (siehe Abb. 1) die gewählte Variable einen Wert zwischen 1 und 7 annehmen kann. Weiterhin erkennt er, dass die Zahlen 1 bis 6 „angezeigt werden". Dabei bezieht er sich vermutlich auf den Teil „wenn Variable 1 ≤6 dann zeige Zahl Variable 1". Der Teil „ansonsten zeige Zahl 6" ist für den Schüler nicht relevant, da seiner Meinung nach eine Zahl größer als 7 nicht vorkommen darf. Bei seiner Überlegung verwechselt Marco vermutlich die Eingabewerte des Algorithmus, welche zufällig zwischen 1 und 7 gewählt werden, und die am Ende ausgegebenen Werte, welche zwischen 1 und 6 liegen.

Nachdem Marco darauf aufmerksam gemacht wird, dass Variable 1 auch die Zahl 7 annehmen kann, ändert er seine Antwort:

> Marco: Achsooo. Ja dann ist er doch manipuliert, weil wenn dann da nämlich die 7 kommt, dann wird's ja ne 6. Dann hat die 6 ja zwei Wahrscheinlichkeiten. Die 6 zu ziehen sind dann 2/7. Oder?
> Lehrer: Aha. Und die anderen?
> Marco: Die sind jeweils 1/7, weil 7 kann man ja dann nicht ziehen, weil die automatisch zur 6 wird.

Nach dem kurzen Impuls ist Marco schnell klar, dass die Zahl 6 häufiger erscheint als die anderen Zahlen. Dies begründet er damit, dass „die 6 ja zwei Wahrscheinlichkeiten hat". Damit meint er vermutlich, dass zwei der möglichen Einträge von Variable 1, nämlich die 6 und die 7, zur Ausgabe der Zahl 6 führen. Er schlussfolgert, dass die Wahrscheinlichkeit für die 6 2/7 beträgt und für die anderen Zahlen jeweils 1/7. Die 7 könne man nicht „ziehen". Marco zeichnet hierzu die folgende Grafik, die seine Berechnung der Wahrscheinlichkeiten verdeutlichen soll.
Die Zahlen 1 bis 6 werden direkt auf die Zahlen 1 bis 6 aufgeteilt. Dies versucht Marco durch die Schweifklammer zu visualisieren. Auf der rechten Seite berechnet er entsprechend, dass mit einer Wahrscheinlichkeit von 6/7 die Zahlen 1 bis 6 vorkommen und jeweils zu 1/6 die einzelnen Zahlen erscheinen. Als

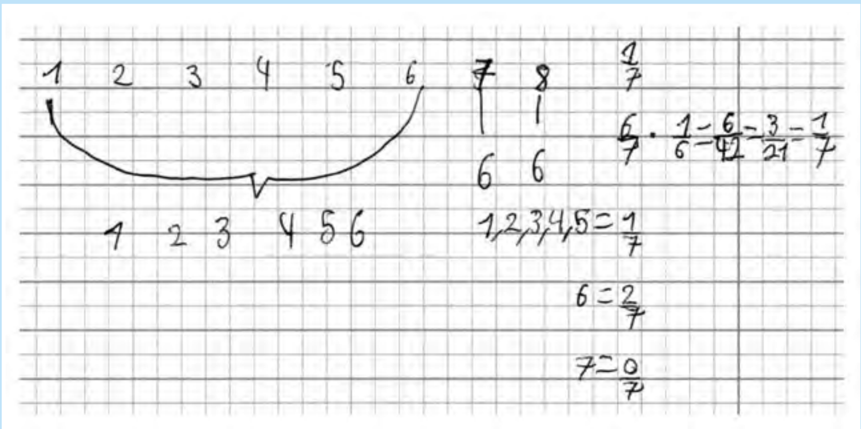

Abb. 4. Skizze des Schülers Marco

Produkt erhält man eine Wahrscheinlichkeit von 1/7 für die Zahlen 1 bis 6. Die 7 wird direkt auf die 6 abgebildet, weshalb er für die 6 als Gesamtwahrscheinlichkeit 2/7 angibt. Für die 7 notiert er eine Wahrscheinlichkeit von 0/7. Es scheint dem Schüler wichtig zu sein, für jeden Eingabewert (Zufallszahlen von 1 bis 7) auch die passende Wahrscheinlichkeit als Ausgabewert zu notieren, obwohl die 7 eigentlich kein mögliches Ergebnis des Zufallsexperiments darstellt. Die Nähe der Zeichnung des Schülers zum Algorithmus des manipulierten Würfels ist klar zu erkennen.

Marco erklärt zudem, wie man durch Änderungen am Algorithmus die Wahrscheinlichkeit für die 6 erhöhen könnte:

> Lehrer: Wie müsste man den jetzt verändern, damit die Wahrscheinlichkeit für die 6 noch größer wird?
> Marco: Das ist relativ einfach. [...] Man kann hier dann damit sie sehr wahrscheinlich ist eine höhere Zahl einfach eingeben. Zum Beispiel, wenn man sie nur etwas wahrscheinlicher haben will, kann man so ne 8.
> Lehrer: Ok. Und warum ist die dann wahrscheinlicher?
> Marco: Weil es dann nämlich wieder, wenn wir hier zum Baumdiagramm kommen. Dann kommt hier nämlich einfach was dazu. Die 8 und die wird dann nämlich auch automatisch zur 6. Und dann hat man hier nämlich 2/8 plus diese 1/6 ähm 1/8. Joa.
> Lehrer: Ok, und 3/8 ist größer als 2/7.
> Marco: Ja. Und man kann zum Beispiel auch mal 100. Und dann wird jetzt wahrscheinlich fast nur die 6 kommen. (probiert einige Male die Simulation aus). Und irgendwann wir auch mal ne andere Zahl, aber sie ist halt sehr, sehr unwahrscheinlich.

Der Schüler macht deutlich, dass er erkennt, dass sich die Wahrscheinlichkeit für die 6 im Vergleich zu den Zahlen 1 bis 5 erhöhen lässt, indem man das für die Variable 1 gewählte Zufallszahlen-Intervall vergrößert. Er kann am Beispiel der Zahl 8 auch die Wahrscheinlichkeit nennen („2/8 plus diese 1/6 ähm 1/8") und verdeutlicht die neuen Wahrscheinlichkeiten an seiner Zeichnung (siehe Zahl 8 in Abb. 1). Außerdem erklärt er, dass bei einer sehr großen Zahl wie beispielsweise der 100 „fast nur die 6 kommen" würde.

4 Fazit und Ausblick

Die Aufgaben und der Fall des Schülers Marco zeigen, dass man mit MakeCode auf verschiedene Weise mit Pseudozufallszahlen arbeiten kann. Die Grundlage bildet jeweils der „wähle eine zufällige Zahl von X bis Y"-Block, welcher gleichverteilte Pseudozufallszahlen zwischen den angegebenen Grenzen generiert. Diese Gleichverteilung kann dann mit einem Algorithmus verändert werden, z.B. um einen manipulierten Spielwürfel zu simulieren. Auf diese Weise sammeln die Schüler/innen aktiv Erfahrungen mit Wahrscheinlichkeitsverteilungen und dem Begriff der Wahrscheinlichkeit. Es lassen sich auch explizit Wahrscheinlichkeiten für die einzelnen Zahlen berechnen, wie es auch Marco in dem kurzen Interviewausschnitt getan hat. Dabei arbeitet Marco eng am Algorithmus und einer vergleichsweise formalen Auffassung der Laplace-Wahrscheinlichkeit, welche auch im Programm so angelegt ist.

Nach unserer Einschätzung lohnt es sich in Bezug auf Pseudozufallszahlen im Anschluss an die obigen Aktivitäten, auch den Begriff des Zufalls im Unterricht explizit mit den Schüler/inne/n zu diskutieren. Hierbei entwickeln die Schüler/innen zum Teil spannende Ansätze. Beispielsweise erklärt Marco:

> Marco: Ich hätte eigentlich einfach gedacht, dass die Wahrscheinlichkeit gleich groß ist zwischen 1 und 6. Das ist ja so. Und dass der Computer das halt denkt. [...] Er hat keinen Algorithmus, weil das wär ja dann glaube ich manipuliert. [...] Weil, wenn es ein Algorithmus ist, dann kommt zum Beispiel 1, 6, 3, 4, 5, 2 und dann kommt wieder 1, 6, 3, 4, 5, 2 und das geht dann immer so weiter. Das wär ja vorhersehbar und dann könnte man zum Beispiel drauf wetten. Und das wär ja nicht so gut.

Für die Diskussion im Unterricht hilft es, wenn die Schüler/innen gezielte Reflexionsfragen an die Hand bekommen und hierzu recherchieren können:

- Überlege: Wie könnte der Block „wähle eine zufällige Zahl von X bis Y" funktionieren?
- Kannst du einen Algorithmus erstellen, der (ohne den Wahrscheinlichkeitsblock) Zufallszahlen generiert?
- Du konntest im Programm die Wahrscheinlichkeiten manipulieren. Entstehen dann überhaupt noch zufällige Ergebnisse?
- Welche Grenzen haben die Manipulationen der Wahrscheinlichkeiten im Programm?

Hierbei handelt es sich um vielseitige Fragen, welche eine tiefe Auseinandersetzung mit dem Wahrscheinlichkeitsbegriff anregen können. Sie beziehen sich unmittelbar auf die Erfahrungen der Schüler/innen in MakeCode, so dass vielfältige Antworten zu erwarten sind. Die Aufgabe der Lehrkraft besteht dann darin, diese in Richtung des zu entwickelnden Wahrscheinlichkeitsbegriffs weiterzuführen.

Literatur

BEHRENDS, E. (2013). *Elementare Stochastik. Ein Lernbuch von Studierenden mitentwickelt*. Wiesbaden: Springer Spektrum.

HEHL, W. (2021). *Der Zufall in Physik, Informatik und Philosophie. Zufall als Fundament der Welt*. Wiesbaden, Heidelberg: Springer Vieweg.

PIELSTICKER, F. (2019). MatheWelt. Spiel mit selbstgedruckten Würfeln. In I. WITZKE, & J. HEITZER (Hg.), *3D-Druck. mathematik lehren, 217*. Seelze: Friedrich Verlag.

Dr. FREDERIK DILLING, dilling@mathematik.uni-siegen.de, ist wissenschaftlicher Mitarbeiter an der Universität Siegen und Leiter des Forschungs- und Entwicklungsprojekts DigiMath4Edu.

JULIAN PLACK, plack@mathematik.uni-siegen.de, ist wissenschaftlicher Mitarbeiter und Doktorand an der Universität Siegen.