

## 3D-Druck als Motivation für die Einführung in die Programmierung in der Realschule

Oliver Krisch<sup>1</sup>, Petra Kastl<sup>2</sup> und Ralf Romeike<sup>3</sup>

**Abstract:** 3D-Drucker gehören derzeit zu den Informatiksystemen, die für die Schüler noch neu sind, Neugier hervorrufen und sich somit gut einsetzen lassen, um sie zu motivieren. Erschöpft sich die Nutzung eines 3D-Druckers allerdings allein im Herunterladen und Ausdrucken vorgefertigter Modelle, haben die Schüler aber wenig über Informatik gelernt. In diesem Beitrag wird eine Unterrichtssequenz von 7 Doppelstunden zum Erlernen grundlegender Programmierkenntnisse vorgestellt, die die Programmierung dreidimensionaler Objekte mithilfe von Turtle-Grafiken in Beetle Blocks zum Ziel hat. Zudem sollte noch eine fächerübergreifende Verbindung zu den im Mathematikunterricht kennengelernten Funktionen geknüpft werden. Als Motivation diene das Erstellen eines 3D-Körpers, der am Ende der Sequenz mit Hilfe eines 3D-Druckers ausgedruckt werden sollte. Die dargestellten Erfahrungen zeigen, dass sich die Schüler von der Idee begeistern lassen, die Verknüpfung mit mathematischen Grundlagen allerdings eine Herausforderung darstellt.

**Keywords:** 3D-Druck, Realschule, Programmierung, Beetle Blocks

### 1 Einleitung

Nachdem lange Zeit der wahrnehmbare technische Fortschritt vor allem durch bessere Grafikleistungen bestimmt war, die in der Schule im Bereich der Computergrafik Schülerinnen und Schüler motivieren konnten, sind 3D-Drucker eine jüngere technische Entwicklung, die i.d.R. (noch) nicht im typischen Schülerhaushalt zu finden ist, aber durch erstaunliche Ergebnisse Schüler, Kollegen und Eltern zu faszinieren weiß. An verschiedenen Schulen werden deshalb 3D-Drucker angeschafft, um Schülern die beeindruckenden Möglichkeiten des 3D-Drucks zu eröffnen. In der Praxis beschränkt sich die Verwendung von 3D-Druckern allerdings häufig auf das Konfigurieren des Druckers und das Ausdrucken von aus dem Internet heruntergeladenen vorgefertigten Modellen. Für die Nutzung des fachübergreifenden Potenzials im Informatikunterricht wurden bisher wenig überzeugende Beispiele publiziert. Im Folgenden stellen wir ein Beispiel vor, in dem von Schülern algorithmisch erzeugte 3D-Modelle und deren 3D-Ausdruck als Gegenstand und Motivierung der Einführung in die Programmierung herangezogen werden. Dabei experimentie-

---

<sup>1</sup> Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, oliver.krisch@fau.de

<sup>2</sup> Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, petra.kastl@fau.de

<sup>3</sup> Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, ralf.romeike@fau.de

ren die Schüler mit ihnen aus dem Mathematikunterricht bekannten Konzepten, deren Verwendung in Verbindung mit einfachen algorithmischen Grundstrukturen erst die beeindruckenden Figuren ermöglicht und damit den Anwendungskontext außerhalb der Mathematik eröffnet.

## 2 Geometrische Formen und algorithmische 3D-Modelle

### 2.1 Turtlegrafik als Basis für algorithmische 3D-Modelle

Seymour Papert stellte in den 1980er Jahren mit Logo eine Programmierumgebung vor, mit der Kinder geometrische Formen erkunden können, indem sie diese mit einfachen Programmieranweisungen am Bildschirm selbst zeichnen. Papert [Pa82] unterstreicht mit dem von ihm gleichzeitig als Konstruktivismus vorgestellten Ansatz die Rolle des Schaffens konkreter Artefakte, die persönlich bedeutungsvoll sind und anderen gezeigt, erprobt und auch bewundert werden können. Der als Turtle-Grafik bekannte Ansatz liegt auch den folgenden Beispielen zugrunde, allerdings erweitert auf den dreidimensionalen Raum: Die Ausgabe des 3D-Druckers bzw. deren virtuelle Darstellung entspricht dem nachgezeichneten Weg der Schildkröte in Logo oder dem Stift eines Objekts in Scratch.

Viele Gegenstände des alltäglichen Lebens lassen sich aus mathematischen Körpern und Formen aufbauen, welche sich sehr gut als Grundlage für Turtle-Grafiken eignen. Ausgangspunkt in unserem Beispiel war ein Salzstreuer, dessen Form einem Hyperboloid mit der Formel  $\frac{x^2}{5,5^2} - \frac{y^2}{2,5^2} = 1$  ähnelt (vgl. Abb. 1) und die zugrundeliegende Idee, Körper aus mathematischen Funktionen zu erstellen und mit einem 3D-Drucker auszudrucken. Zur Umsetzung wurde die blockorientierte Programmiersprache Beetle Blocks [KR12] verwendet.

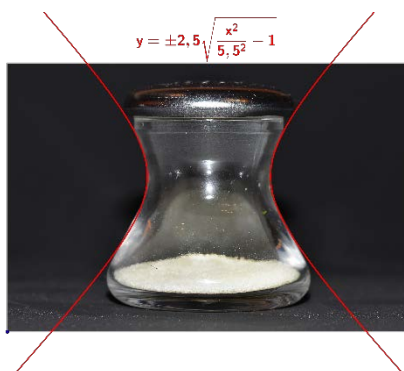


Abb. 1: Salzstreuer – Hyperboloid als Vorlage für 3D-Objekte.

Für den vorliegenden Salzstreuer diente der Kreis als Ausgangsfigur. Die Schülerinnen und Schüler erzeugten die Kreislinie mit Hilfe der Formel  $r^2 = x^2 + y^2$ . Diese stellten sie zuerst um zu  $y = \pm\sqrt{r^2 - x^2}$  und teilten das Zeichnen des Kreises auf in das Zeichnen zweier Halbkreise. Damit konnten die Schülerinnen und Schüler nun einen Zylinder zeichnen, indem sie die Höhe  $z$  langsam erhöhten. In einem zweiten Schritt lernten sie dann den Radius der einzelnen Kreise in Abhängigkeit von der Höhe zu verändern. Dies geschah exemplarisch erst durch einfache mathematische Operationen, die dann im Verlauf der Stunde durch komplexere ersetzt wurden. Der Vorteil, den Körper aus einzelnen Schichten aufzubauen, liegt darin, dass der 3D-Drucker seine Objekte auch durch Hinzufügen einzelner Schichten aufbaut. Die Schülerinnen und Schüler konnten so im Vorfeld sehen, wie aus einzelnen Schichten der Gesamtkörper entsteht.

## 2.2 Visuelle Programmierung mit Beetle Blocks

Mit der Einführung von Scratch [MRR10] wurde die Einstiegshürde für Programmieranfänger deutlich gesenkt, was insbesondere den Schülerinnen und Schülern an der Realschule sehr entgegen kommt. Durch die visuelle Darstellung der programmierten Objekte auf der Bühne wird die Programmierung im Sinne Paperts' Konstruktivismus unmittelbar erfahrbar, die multimediale Bandbreite ermöglicht den Schülerinnen und Schülern persönlich bedeutsame Ideen umzusetzen und die Folgen ihrer Programmierentscheidungen unmittelbar nachzuvollziehen. Durch die Repräsentation der Anweisungen als Blöcke werden zudem Syntaxfehler vermieden und die zur Verfügung stehenden Programmierkonstrukte sind unmittelbar zur Auswahl präsent. So wie Strecker [St15] auf der INFOS 2015 über den Einsatz grafischer Programmiersprachen im Abitur berichtete, überzeugt der visuelle Zugang aufgrund seiner Intuitivität und leichten Bedienung auch in der Realschule. Koschitz und Rosenbaum [KR12] übertragen nun die Konzepte und Bedienung von Scratch und die Idee der Turtle-Grafik unter Verwendung von Snap [HM10] in den dreidimensionalen Raum. Mit Beetle Blocks<sup>4</sup> können mit Hilfe der algorithmischen Grundstrukturen dreidimensionale Objekte programmiert werden, die anschließend als 3D-Modell exportierbar sind und mittels eines 3D-Druckers ausgedruckt werden können. Algorithmische, mathematische und künstlerische Objekte können somit unmittelbar „erfassbar“ werden. Im Unterricht haben wir gute Erfahrungen gemacht mit Schmuck, Alltagsgegenständen wie Vasen und Schalen und künstlerischen Figuren (vgl. Abb. 7). Das Potenzial eines an solchen Zielen orientierten Unterrichts liegt darin, dass auch vordergründig nicht an der Programmierung interessierte Schülerinnen und Schüler durch die kreative Erstellung von Produkten, die so erstmal nicht im Informatikunterricht erwartet wird, motiviert werden können.

---

<sup>4</sup> <http://beetleblocks.com/>

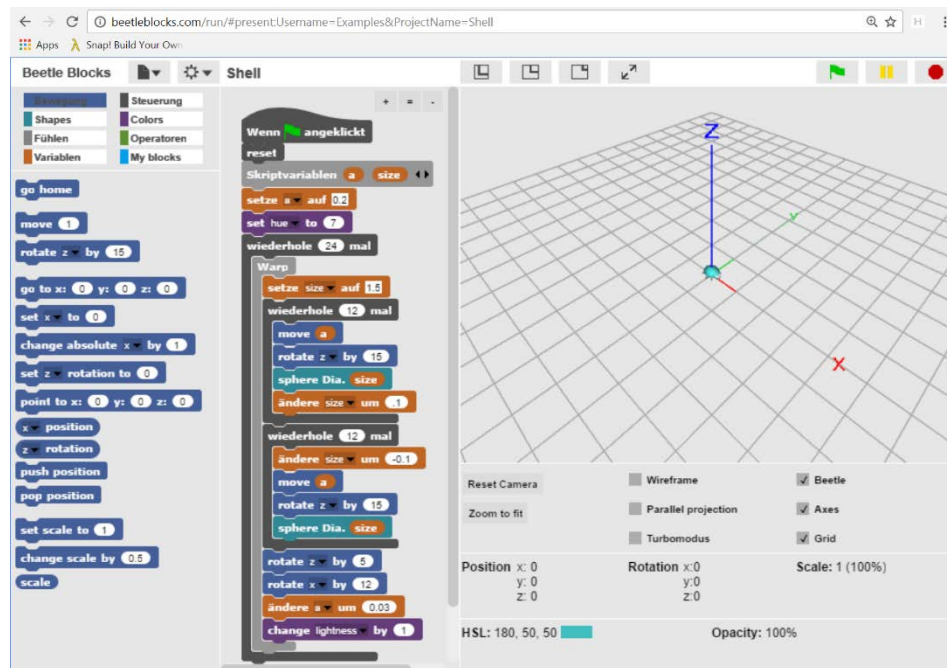


Abb. 2: Beetle Blocks.

### 3 Umsetzung

#### 3.1 Kontext und Einordnung

Spätestens in der zehnten Klasse an bayerischen Realschulen lernen die Schüler im Modul „Modellierung und Codierung von Algorithmen“ des Lehrplans für das Fach Informationstechnologie die Grundstrukturen Sequenz, Auswahl und Wiederholung kennen und anzuwenden. Ein weiterer Punkt ist die Implementierung von Algorithmen mit einem geeigneten Programmierwerkzeug. Die Motivierung der Schüler ist hierbei ein wesentliches Anliegen des Unterrichts. Entsprechend haben sich die zum Erlernen verwendeten Programmiersprachen im Laufe der Zeit gewandelt. So wurde an unserer Schule zuerst die Programmiersprache Pascal verwendet. Um einen größeren Anreiz zu schaffen, sind wir dann auf Delphi umgestiegen. Mit Delphi haben die Schüler am Ende der Unterrichtssequenz einen Taschenrechner oder Getränkeautomaten programmiert und dabei die neu erworbenen Fähigkeiten in einem komplexeren Kontext angewandt. So ein selbst programmierter Taschenrechner motivierte schon etwas, weckte aber in der Schülergruppe auch keine Begeisterungstürme. Grund dafür war zum einen die doch sehr statische Interaktion mit selbigem und zum anderen die hohe Fehlerquote wegen syntaktischer Fehler. Geändert

hat sich das Ganze mit der Einführung der visuellen Programmiersprache Scratch. Mit dieser Programmiersprache konnten die Schüler sehr schnell und ohne syntaktische Fehler zu einem ansprechenden Ergebnis kommen. Die ersten kleinen Spiele waren sehr schnell programmiert und konnten mit nach Hause genommen werden. Mit Beetle Blocks verhielt es sich ebenso wie mit Scratch. Mit ein paar Befehlen konnten die Schülerinnen und Schüler sehr schnell den Käfer über den Bildschirm bewegen. Dies sorgte für Erheiterung in der Gruppe.

### 3.2 Unterrichtsziele und -gliederung

Ziel des Unterrichts war es, aus einfachen mathematischen Grundformen, wie z. B. Kreis, Quadrat oder gleichseitigem Dreieck, dreidimensionale Objekte zu entwickeln und diese zu programmieren. Als Vorlage konnten die Schüler sich an bekannten Gegenständen des täglichen Lebens orientieren, gleichfalls durften sie ihre 3D-Körper frei wählen, da sie unterschiedliche Vorstellungen hatten, welchen Gegenstand sie gern kreieren wollten. Ein Ziel hierbei war auch, dass den Schülerinnen und Schülern die Möglichkeit gegeben werden sollte, selbständig ihre eigene Kreativität zu nutzen und eigene Vorstellungen umzusetzen. Hierzu mussten die Schüler ihre Kenntnisse aus der Geometrie heranziehen. Für die Grundformen kamen neben dem allseits bekannten Satz des Pythagoras auch weitere grundlegende Kenntnisse, wie z. B. das Teilungsverhältnis der Höhen in einem gleichseitigen Dreieck, zum Einsatz. Mit diesem Wissen waren dann die Grundformen sehr schnell erstellt.

Für die dritte Dimension kamen mathematische Funktionen zum Einsatz: Mit Hilfe von Sinus, Kosinus und anderen Funktionen wurde die Größe der Grundformen in Abhängigkeit von der Höhe verändert, um den Objekten eine schöne geschwungene Form zu geben. In Abb. 3 sieht man eine eingefärbte Schnecke, die aus gleichseitigen Dreiecken aufgebaut ist. Bei der Schnecke wurde die Seitenlänge sukzessive konstant verkleinert.

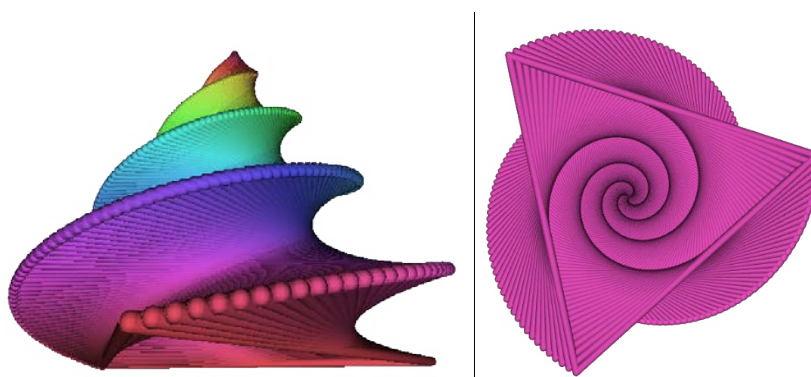


Abb. 3: Schnecke aus gleichseitigen Dreiecken.

Der Unterricht gliedert sich in 3 Phasen:

1. Kennenlernen der algorithmischen Grundstrukturen Sequenz und Wiederholung und Anwendung/Übung mit der Programmiersprache Beetle Blocks. Hierbei erstellen die Schülerinnen und Schüler einfache geometrische Figuren, wie z. B. Rechtecke oder Quadrate (3 Doppelstunden)
2. Erweiterung in die dritte Dimension, indem die Schüler die einfachen geometrischen Figuren als Turm aufbauen (1 Doppelstunde)
3. Mathematische Veränderung des Turms (3 Doppelstunden)

### **3.3 Unterrichtsverlauf und Beobachtungen**

Bei der 10. Klasse handelte es sich um Schülerinnen und Schüler aus dem wirtschaftlichen Realschulzweig. Daher war die anfängliche Begeisterung der Schülerinnen und Schüler Programmieren zu lernen nicht gerade sehr groß. Dass das Ganze dann auch noch mit dem allseits geliebten Fach Mathematik verknüpft werden sollte, trug auch erstmal nicht gerade zur Motivationssteigerung bei. Dies zeigte sich z. B. in der Aussage der Schülerin C.: „Ich kann eh kein Mathe und jetzt muss ich auch noch damit Programmieren lernen.“ Entsprechend gestalteten sich die Anfangsstunden etwas zäh. Die Schülerinnen und Schüler sollten zuerst einmal den Begriff „Sequenz“ kennen und anwenden lernen. Dazu programmierten wir einfache geometrische Figuren, wie z. B. ein Rechteck. Etwas aufwändiger war die Programmierung eines gleichseitigen Dreiecks, da hier mit Hilfe des Satzes von Pythagoras die Koordinaten des dritten Eckpunkts erst einmal berechnet werden mussten.

Die vierte Doppelstunde war für die Schülerinnen und Schüler etwas leichter, da hier nur die bis dahin nicht verwendete Variable  $z$  neu hinzukam. Bis zu diesem Zeitpunkt sind sie noch ohne die Wiederholung ausgekommen. Bei der Erstellung des Turms haben die Schülerinnen und Schüler die Höhe ihres Turms durch wiederholte Erhöhung der Variable  $z$  mit einer Zählerschleife realisiert.

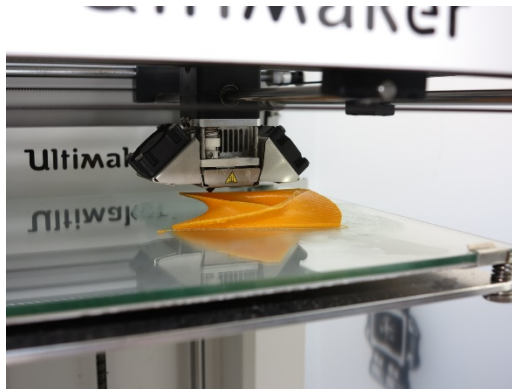


Abb. 4: Ausdruck eines Beetle Block-3D-Modells mit einem Ultimaker 2.

Ab der fünften Doppelstunde stieg die Motivation der Schülerinnen und Schüler deutlich an. Ich habe einen von mir mit dem 3D-Drucker ausgedruckten Körper mitgebracht und den Schülerinnen und Schülern versprochen, ihre Objekte auch mit dem 3D-Drucker auszudrucken. Nach dem konstruierenden Einstieg im Sinne des bottom-up verfolgte der weitere Unterricht nun einen experimentell-dekonstruierenden Ansatz. Hierzu wurden den Schülerinnen und Schülern verschiedene Methoden in Form von Beetle Blocks-Blöcken zum Experimentieren zur Verfügung gestellt, die zusätzlich zu den von den Schülerinnen und Schülern entwickelten Blöcken die Möglichkeit bereitstellten, dass die geometrischen Figuren noch um einen Winkel  $\alpha$  gedreht werden konnten (vgl. Abb. 5).

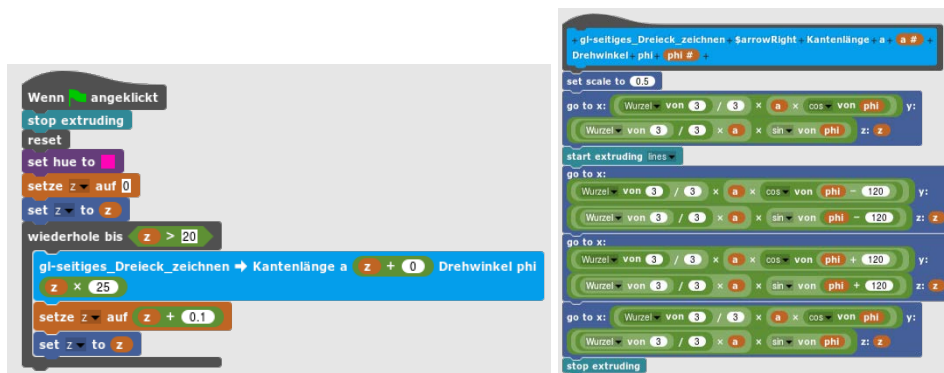


Abb. 5: Programmierung des Turms unter Verwendung eines vorgegeben Blocks (rechts).

Der Grund für die Bereitstellung war, dass die Schülerinnen und Schüler die zugrunde liegende mathematische Modellierung noch nicht im Mathematikunterricht behandelt hatten. Außerdem ist die Zeit mit sieben Doppelstunden knapp bemessen und es sollte den Schülerinnen und Schülern die Möglichkeit gegeben werden, mehr Zeit auf die Entwicklung ihrer eigenen 3D-Objekte zu verwenden. Dabei vertieften sie die bis dahin erlernte Vorgehensweise über den Aufbau der einzelnen Körper aus Schichten. Im Laufe der Zeit

änderte sich auch die Einstellung der Schülerin C. Mit Aussagen wie „Welche mathematische Funktion gibt es noch, die ich ausprobieren kann?“ wollte C. immer weitere Variationen ihres Turms ausprobieren. So kamen nicht nur einfach gedrehte Türme heraus, sondern auch z. B. kleine Schalen. Andere Schülerinnen und Schüler bauten ineinander geschachtelte Körper.

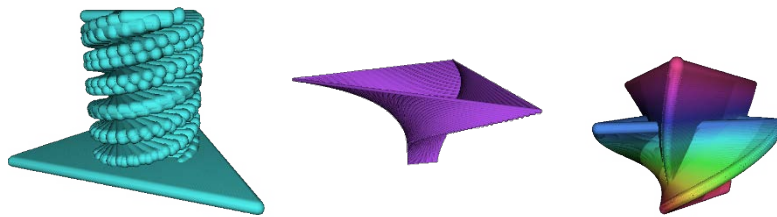


Abb. 6: Schülerergebnisse: Variationen der Türme.

Das in meinen Augen schönste Modell kreierte Schülerin C. Ein Schneckenhaus, welches aus gedrehten, gleichseitigen Dreiecken besteht, die nach oben immer eine kleinere Seitenlänge besitzen (vgl. Abb. 3). Alle Ergebnisse wurden dann mit einem 3D-Drucker ausgedruckt und konnten von den Schülerinnen und Schülern mit nach Hause genommen werden.

#### 4 Fazit und Ausblick

Als Resümee lässt sich feststellen, dass die Motivation der Schülerinnen und Schüler stieg, als klar wurde, dass die 3D-Objekte, die sie programmierten, dreidimensional ausgedruckt werden sollten. Ihr Engagement, sich mit dem Erlernen von Programmierstrukturen zu beschäftigen, war vor diesem Zeitpunkt vergleichbar mit dem einer Gruppe, die mit Scratch das Programmieren erlernt. Auch in der Geschwindigkeit des Erlernens konnte kein Unterschied zu Scratch festgestellt werden.

Bei den ersten geometrischen Grundfiguren wurde es schwieriger. Die Schülerinnen und Schüler konnten zwar noch ohne Probleme die Koordinaten der Ecken eines Quadrates berechnen, aber bei einem gleichseitigen Dreieck ergaben sich gravierende Probleme. Hier muss für die Zukunft eine andere Vorgehensweise, mit einfacheren mathematischen Mitteln gefunden und erprobt werden. Die Anwendung der Mathematik wich hier doch von dem gewohnten Umgang im Mathematikunterricht ab. Vielleicht haben an dieser Stelle Schülerinnen und Schüler aus dem mathematischen Zweig oder am Gymnasium weniger Schwierigkeiten. Eine Gruppe aus zwei Schülern ist hier etwas anders vorgegangen. Anstatt die Koordinaten zu berechnen, haben sie ein gleichseitiges Dreieck auf einem Blatt Papier konstruiert, ungefähr in den Schwerpunkt des Dreiecks den Ursprung des Koordinatensystems gelegt und die Koordinaten des gleichseitigen Dreiecks abgelesen. In der



dritten Dimension konnten die beiden Schüler dann aber nicht mehr die Größe der Seite verändern.



Abb. 7: Vase, Kunstobjekt und Schnecke aus dem 3D-Drucker.

Der Weg in die dritte Dimension war hingegen wieder einfacher. Da die Grundfiguren so programmiert waren, dass ihre Größe von der Seitenlänge abhängt, konnte diese sehr leicht mit Hilfe von mathematischen Funktionen verändert werden. Die Schülerinnen und Schüler ließen hier ihrer Kreativität freien Lauf. Sie konnten dabei die Funktionen, die sie bis jetzt kennengelernt hatten, anwenden. In diesem Schritt variierten ein paar Schüler die Dicke der Linien, die vom Käfer gezeichnet wurden. Eine andere Schülergruppe veränderte mit der Höhe auch die Farbe der gezeichneten Linien. Zwei Schülerinnen haben ihre 3D-Körper mit einer Bodenplatte versehen, um eine Vase zu erhalten. Dieser Teil hat den Schülerinnen und Schülern besonderen Spaß gemacht. Eine Verknüpfung mit anderen Modulen aus dem Lehrplan, z. B. mit Arduinos (Modul: Mikrocontroller) als Steuerung für eine Leuchte (vgl. Abb. 7), wäre gut vorstellbar.

Für die Wiederholung der Unterrichtssequenz empfiehlt es sich, die Erstellung der Grundfiguren zunächst mit einfacheren mathematischen Mitteln auszuprobieren. Die Verknüpfung der Höhe mit mathematischen Funktionen hat sich bewährt. Hier sehen die Schülerinnen und Schüler, dass die Mathematik nicht nur trocken ist, sondern auch eine Anwendung hat, um der eigenen Kreativität Raum zu geben und neue Gegenstände zu entwickeln. Ein weiterer neuer Ansatzpunkt wäre, reale Gegenstände (vgl. Salzstreuer) in mathematische Grundkörper zu zerlegen und diese nachzuprogrammieren. In diesem Ansatz wäre wieder eine Verknüpfung zum Modul „Technisch Zeichnen“ gegeben.

## Literaturverzeichnis

- [HM19] Harvey, B. and Mönig, J.: Bringing “no ceiling” to scratch: Can one language serve kids and computer scientists. Proc. of Constructionism Paris, 2010.
- [KR12] Koschitz, D., and Rosenbaum, E.: Exploring algorithmic geometry with “beetle blocks:” a graphical programming language for generating 3d forms. 15th International Conference on Geometry and Graphics Proceedings, Montreal. 2012.
- [MRR10] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The scratch programming language and environment. ACM Transactions on Computing Education (TOCE), 10(4), 16, 2010.
- [Pa82] Papert, S.: Mindstorms: Kinder, Computer und Neues Lernen. Birkhäuser Verlag, Basel, 1982.
- [St15] Strecker, K.: Grafische Programmiersprachen im Abitur. In: Gallenbacher, J. (Hrsg.): INFOS 2015: Informatik allgemeinbildend begreifen (16. GI-Fachtagung Informatik und Schule, Darmstadt). Bonn: Köllen, 2015.