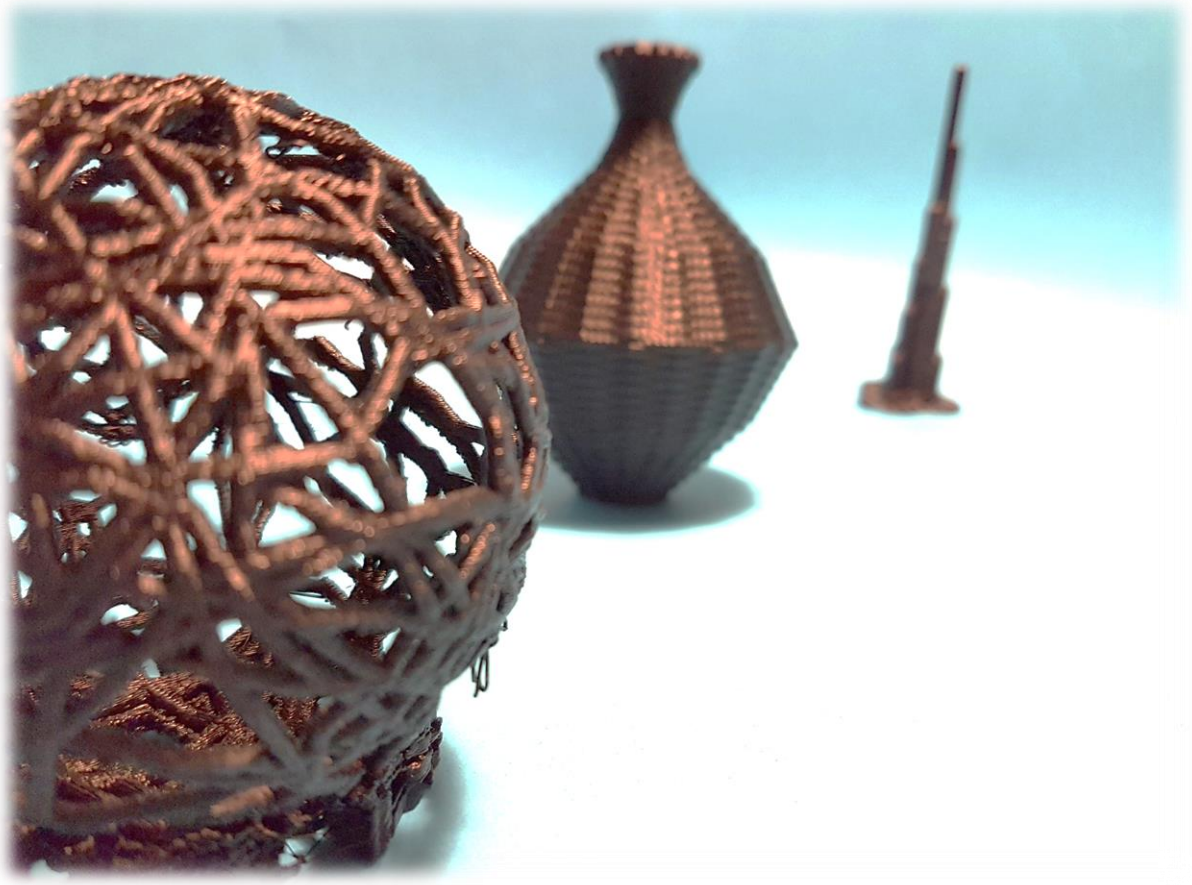


Zulassungsarbeit

3D-Druck in der Schule



eingereicht von

Manuel Riel

betreut von

Prof. Dr. Ralf Romeike

29. Oktober 2017

Inhaltsverzeichnis

Vorwort	3
1. Kurzübersicht rund um den 3D-Druck.....	5
1.1 Begriffsklärung und Grundlagen nicht-industrieller Nutzung	5
1.2 Lebensweltliche Ein- und Ausblicke	6
1.3 Startschwierigkeiten von 3D-Druck im nicht-industriellem Kontext.....	6
1.4 Fazit zum 3D-Druck aus sozio-technischer Sicht	7
2. „Elementarbausteine“ in Beetle Blocks.....	8
2.1 Regelmäßige n-Ecke als Grundflächen	9
2.1.1 Konstruktion über Außenwinkel.....	9
2.1.2 Kellerspeicher mit push & pop	10
2.1.3 Mathematische Funktionen	11
Exkurs: Problematische Mathematik im 3D-Druck	11
2.2 Wege in die dritte Dimension.....	12
2.2.1 Gestapelte Grundflächen	12
2.2.2 Rotation um x- oder y-Achse	15
Exkurs: Im Unterricht zu berücksichtigende Auswirkungen zweier Koordinatensysteme.....	17
2.3 Rotationssymmetrien	17
2.4 Fazit zu den ermittelten Konzepten	18
3. Einführung in die Algorithmik mit 3D-Druck	19
3.1 Rahmenbedingungen des Unterrichtsvorhabens	19
3.2 Weitere informatische Konzepte in Beetle Blocks	20
3.2.1 Bedingte Anweisungen in Beetle Blocks	20
3.2.2 change scale by als Alternative zu Variablen?!	23
3.3 Gestaltung der Unterrichtseinheit	24
Exkurs: Defizite und Bugs von Beetle Blocks im praktischen Unterrichtseinsatz.....	27
3.4 Fazit zur Unterrichtsplanung aus Lehrersicht.....	28
4. Evaluation der Unterrichtsreihe	29
4.1 Vorgehensweise bei der Datenerhebung.....	29
4.2 Auswertungsergebnisse	30
4.3 Exemplarische Vorstellung eines Schülerprojektes.....	35
Schlussbemerkungen.....	37
Abbildungsverzeichnis.....	38
Literaturverzeichnis.....	39
Anhang	40

Vorwort

„Dann können Sie sich ja selbst ihre eigenen Objekte ausdrucken!“ lautete die Reaktion einer Bibliothekarin der Universitätsbibliothek, als ich Bücher zum Thema „3D-Druck“ im Zuge der vorliegenden Arbeit auslieh. Ihr Ausspruch kann anekdotisch für das breite Interesse an der vielfach als „Zukunftstechnologie“ bezeichneten Thematik rund um den 3D-Druck stehen: Berichten schon längst Fernsehsender, wie der WDR (2017) in populärwissenschaftlichen Reportagen über 3D-Druck in Zusammenhang mit – meist pragmatisch als „3D-Scanner“ bezeichneten – bildgebenden Verfahren, durch die sich kleine Kunstrepliken menschlicher Probanden erstellen lassen. Weiterhin wird über mögliche medizinische Anwendungen, wie orthopädische Prothesen oder gar bionische Organe, spekuliert oder im Falle von ProSieben (2017) der kulinarische Mehrwert des medienwirksam inszenierten 3D-Druck-Restaurants *Food Ink* erörtert.

In Anbetracht der medialen Mannigfaltigkeit, in der über lebensweltliche Verwendungsmöglichkeiten von 3D-Druck berichtet wird, erstaunt es nicht, dass bereits vereinzelte Meldungen über den Einsatz von 3D-Druckern im Bildungssektor existieren. Bereits im Jahre 2013 schrieb Greb:

„Die Bandbreite der Nutzung [von 3D-Druck zu Bildungszwecken] reicht dabei von ersten Gehversuchen im Technik- oder Kunstunterricht, über kreative Schularbeitsgemeinschaften und Projektwochen bis hin zur professionellen Nutzung [...]. Noch handelt es sich um vergleichsweise wenige Leuchtturmprojekte, diese sind aber schon jetzt vielversprechend“.

Vier Jahre nach den von Greb erwähnten aussichtsreichen Ansätzen fällt das Fazit zum aktuellen Stand didaktischer Bemühungen rund um den 3D-Druck für Krisch, Kastl und Romeike (2017) in ihrem Unterrichtskonzept für 3D-Druck an Realschulen jedoch kritisch aus:

„An verschiedenen Schulen werden [...] 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.“

Begünstigt wird diese fragwürdige Entwicklung durch die zunehmende Anzahl von freiverfügbaren 3D-Objekten auf Tauschwebseiten wie *Thingiverse*, die von dekorativen Osterhasen bis hin zu maßgeschneiderten Handyhüllen fast jeden Geschmack bedienen können. Dabei verspricht gerade im Unterrichtskontext das eigenständige Kreieren und Gestalten von eigenen 3D-Objekten sowie das anschließende „Ausdrucken“ dieser besonders motivierend zu sein.

Aus informatikdidaktischer Sicht ergibt sich ergo ein erheblicher Forschungsbedarf, wenn 3D-Druck als allgemein antizipierte Technologie für die Zukunft auch im Informatikunterricht inhaltlich gehaltvoll und zugleich motivierend eingesetzt werden soll. Die hier vorliegende Arbeit schließt daher an den Beitrag von Krisch et al. (2017) an und stellt ein mehrfach erprobtes Unterrichtskonzept im dritten Kapitel vor, welches den gewinnbringenden Einsatz von 3D-Druck im Anfangsunterricht zur Algorithmik im Fach Natur und Technik in der 7. Jahrgangsstufe Bayerischer Gymnasien ermöglicht und vorangegangenen Aspekt der selbständigen Gestaltung und Programmierung von 3D-Modellen – im Kontrast zum bloßen „Konsum“ vorgefertigter Modelle – im schülerzentrierten Unterricht berücksichtigt. Zum Einsatz kommt die auf *Snap!* basierende, sich (laut offizieller Webseite) gegenwärtig noch im Alphastadium befindliche, visuelle Programmierumgebung *Beetle Blocks*, an deren Entwicklung bekannte Namen, wie Eric Rosenbaum, Bernat Romagosa und Jens Mönig, beteiligt sind.

Weiterhin ist das Thema „3D-Druck“ nicht nur aus Lehrer-, sondern insbesondere aus Schülersicht – welche teils durchaus durch die eingangs erwähnten populärwissenschaftlichen Beiträge geprägt ist –

für didaktische Forschungen zu beleuchten, weshalb sowohl zum konkreten Unterricht als auch zu 3D-Druck im Allgemeinen eine Schüler-Evaluation stattfand im vierten Kapitel dargelegt wird. Insbesondere erfasst wurden Selbsteinschätzungen von Merkmalsausprägungen, die Kreativität und Motivation indizieren.

Bevor dieses Unterrichtskonzept jedoch überhaupt entwickelt werden konnte, war eine intensive Beschäftigung mit basalen konzeptionellen Punkten im Bereich mathematisch- und informatikdidaktischer Voraus- und Zielsetzungen, die sich aus der Programmierplattform Beetle Blocks ergeben, notwendig: Zur Beschreibung von 3D-Objekten bedarf es gemessen an der Altersstufe relativ weitreichender Kenntnisse im Bereich der Mathematik – geometrischer und analytischer Natur. Strukturierende Überlegungen zu diesem nicht zu unterschätzendem Spannungsfeld zwischen mathematischen und informatischen Konzepten, die auch zur Planung ähnlich gearteten Unterrichtsvorhaben herangezogen werden sollten, finden sich im zweiten Kapitel.

Nur kurz umrissen werden allgemeine, historische und technische Aspekte des 3D-Drucks, da es bereits eine Vielzahl an praktischen Ratgebern für den Einstieg in die Welt des 3D-Drucks gibt. Das erste Kapitel bietet hierzu einen Ausgangspunkt für erste Beschäftigung mit dem Thema aus didaktischer Sicht und verweist auf geeignete Literatur für eine weiterführende Auseinandersetzung.

Danksagung

Zu besonderem Dank bin ich dem Institut der Informatikdidaktik in Erlangen, das mir beim Erstellen dieser Arbeit zur Seite stand, verpflichtet. Namentlich möchte ich hier Petra Kastl erwähnen, die mit mir bereits im Vorfeld an diese Arbeit, das *push-and-pop*-Verfahren aus Kapitel 2.1.2 ausführlich diskutierte. Für die praktische Umsetzung der Unterrichtseinheit wurde der Physiklehrer Rudolf Pausenberger nicht müde, mich mit pädagogischem, erfahrenem Sachverstand bei einem meiner ersten Unterrichtsversuche zu unterstützen; seine Kooperationsbereitschaft und Aufgeschlossenheit für neue Konzepte ermöglichten die erprobte Unterrichtsreihe überhaupt erst.

1. Kurzübersicht rund um den 3D-Druck

Praktische und technische Grundlagen dürfen beim Durchführen einer Unterrichtseinheit mit 3D-Druckern keineswegs vernachlässigt werden und beanspruchen in der Vorbereitung einen erheblichen Mehraufwand von Seiten der Lehrkraft. Wegen der im Vorwort erwähnten, vielen eröffneten lebensweltlichen Perspektiven und der kontinuierlichen Weiterentwicklung von 3D-Druckern ist es jedoch nicht möglich einen stets aktuellen, umfassenden oder gar vollständigen Überblick über die für Bildungseinrichtungen geeigneten Geräte zu geben oder sämtliche didaktische Anknüpfungspunkte zu nennen. An Büchern zum Themeneinstieg für Anfänger, die den Prozess beginnend beim Druckerkauf bis hin zum ersten Drucken (meist vorgefertigter Modelle) begleiten, gibt es zwar (noch) keine große Auswahl im üblichen Sinne, aber eine Handvoll fundierter, praktischer Ratgeber – beispielsweise der Autoren Nitz (2015), Rattat (2016) sowie Sommer, Schlenker und Lange-Schönbeck (2016). Auf sie sei in Hinblick auf tiefergehende Fragen an dieser Stelle verwiesen. Das ebenfalls erhältliche, aus dem amerikanischen übersetzte Buch „3D-Druck für Dummies“ von Hausman, Horne und Muhr (2014) verwirrt meines Erachtens nach mit einer Vielzahl (im deutschsprachigen Raum) wenig gebräuchlichen Fachbegriffen und unterscheidet sich in dieser Hinsicht von seinen Mitbewerbern.

1.1 Begriffsklärung und Grundlagen nicht-industrieller Nutzung

Bei „3D-Druck“ handelt es sich um einen Sammelbegriff, unter dem verschiedene additive Fertigungsverfahren – anschaulich benannt nach dem Prinzip des sukzessiven Hinzufügens, des „Addierens“ von weiterem Material kontrastierend zu dem Entfernen von Werkstoff beim Einsatz Fräsmaschinen oder der Erstellung von Skulpturen – zusammengefasst werden. Additive oder generative Fertigung bedeutet insbesondere auch, dass die entstehenden Werkstücke aus einem Guss sind (vgl. Rattat (2016), S.9). Die verschiedenen Vorgehensweisen sind jedoch keineswegs neu, wenn dies auch durch den medialen Hype suggeriert werden könnte, sondern kommen bereits seit in etwa den 1960ern in industrieller Fertigung zum Einsatz. Warum sich erst seit 2012 erste 3D-Drucker für Privathaushalte, Bildungseinrichtungen oder klein- und mittelständische Unternehmen auf dem Markt befinden, erklärt sich durch den Ablauf von Patenten (vgl. Nitz (2015), S. 16f.).

Von den vielen verschiedenen Prozessverfahren ist bislang nur das Schmelzschichtverfahren (engl. *Fused Deposition Modeling*, kurz FDM) im Schulkontext effektiv von Interesse: Ähnlich einer Heißklebepistole werden Schichten aus verflüssigtem Kunststoff übereinander aufgetragen und bilden nach und nach das fertige Objekt aus. Je nach Qualitätsansprüchen beginnen die Preise für einen 3D-Drucker für sog. „Open Source“-Bausatzdrucker bereits für unter 100 € und reichen bis zum inoffiziellen Goldstandard im FDM-Bereich zum Ultimaker 3 für über 3000 €. Das Filament – das Druckmaterial in Form von dünnem, aufgewickeltem Kunststoff – gibt es ebenso abhängig von Farb-, Struktur- und Qualitätsanforderung schon für weniger als 15 € pro Kilogramm. Unabhängig von der Qualitätsvariable lässt sich jedenfalls feststellen: Schmelzschicht-Drucker sind in Erwerbs- und Folgekosten verblüffend preiswert und daher auch für schulische Einsatzgebiete geeignet.

An dieser Stelle wird meinerseits jedoch keinerlei Kaufberatung für einen bestimmten Drucker erfolgen, sondern ich möchte den geneigten Leser, die geneigte Leserin anregen, sich selbst Gedanken über die eigenen Anforderungen an einen möglichen Drucker zu machen – zumal hierzu genügend Literatur bereitsteht. Besonderes Augenmerk auf verschiedenste Druckverfahren, die Auswahl eines Druckers sowie eine zusätzliche, ausführliche Auflistung verfügbarer Drucker samt jeweils tabellarischer Spezifikationen bietet hierbei in breitem Umfang Sommer et al. (2016) auf den Seiten 63 bis 104.

Stellung bezogen werden soll zum von Nitz (2015) auf Seite 26f. diskutierten Vorschlag, externe Druckdienstleister zurate zu ziehen, statt selbst einen Drucker zu erwerben: Dies mag zwar eine in der Unterrichtsvorbereitung (und Druckerwartung) zeitsparende und für nur einzelne Druckexemplare benötigende Privathaushalte eine finanziell sinnvolle Lösung sein, raubt den Schülerinnen und Schülern niedriger Jahrgangsstufen jedoch die Möglichkeit sich den Fertigungsprozess konkret anzusehen. Letzterer trug in der durchgeführten Unterrichtsreihe in den ersten Stunden augenscheinlich zur beachtlichen Motivation der Schülerinnen und Schüler bei.

1.2 Lebensweltliche Ein- und Ausblicke

Die 3D-Druck-Technologie wird von verschiedensten Fachdisziplinen nahezu euphorisch erwartet, wie das Zitat des Chirurgen Kamolz (2016) in einer Fachzeitschrift verdeutlicht:

„Ich bin mir sicher, dass gerade in der Plastischen Chirurgen [sic!] 3D Technologien in naher Zukunft unverzichtbar sein werden[.]“

Nicht weniger spannend wird der 3D-Druck von Wirtschaftswissenschaftlern empfunden, die mit großer Erwartungshaltung den Begriff *rapid prototyping*, die Möglichkeit schnell kleine Vorserienprodukte in geringer Stückzahl ohne die Notwendigkeit des Konfigurierens oder gar Anfertigen spezieller Fertigungsmaschinen zu produzieren, prägen. 3D-Druck im Allgemeinen sowie das „Kopieren“ von Realweltobjekten mittels so genannter „3D-Scanner“ wurde unterdessen bereits aus urheberrechtlich-juristischer Fachsicht ausführlich von Moser (2015) beleuchtet.

Aus informatikdidaktischer Sicht sei daher angemerkt, dass sich eine Vielzahl an fächerübergreifenden Anknüpfungspunkten ergeben und zukünftig ergeben werden. Schienen mir z.B. urheberrechtliche Fragestellungen zunächst weit von der durchgeführten Unterrichtsreihe mit *Beetle Blocks* hergeholt, so programmierte eine Schülerin ein geschütztes Markenzeichen als Schlüsselanhänger nach - was ein-drucksvoll die lebensweltliche Einbettung der Gesamthematik unterstreicht.

1.3 Startschwierigkeiten von 3D-Druck im nicht-industriellem Kontext

Ähnlich euphorisch wie Kamolz' Zitat lesen sich die Einführungskapitel – wenn nicht sogar der gesamte Textkorpus – der Ratgeber für 3D-Druck im privaten Einsatz von Nitz (2015), Rattat (2016) und Sommer et al. (2016). Jedoch kann auch einige Jahre nach den ersten erhältlichen nicht-industriellen 3D-Druckern zum Verfassungszeitpunkt dieser Arbeit nicht konstatiert werden, dass 3D-Drucker inzwischen zum jetzigen *state of the art* im privaten, schulischen oder mittelständischen Umfeld gehörten. Als Gründe hierfür wird von Gerber (2017) der unattraktive Preis für den Privatgebrauch – der jedoch für Unternehmen und auch Schulen nur bedingt eine Rolle spielen kann – angeführt. Mindestens im gleichen Maße dürften „Kinderkrankheiten“ von 3D-Druckern, insbesondere FDM-Modellen, viele Interessenten abschrecken. Aus der gesammelten Erfahrung verschiedenster Beteiligter in Universität und FabLab mit günstigeren und hochwertigeren Druckern lässt sich subsumieren, dass kleinere, teils auch größere Unzulänglichkeiten ein gewisses Maß an Tüftelei von Seiten der Benutzer erfordern – ein Umstand der in den sonst so detaillierten Ratgebern in der überwiegenden Mehrheit relativ euphemistisch dargestellt wird. Rattat (2016) hingegen widmet dem ungeliebten *Troubleshooting* ein eigenes Unterkapitel mit folgenden, einleitenden Worten:

„Den niedrigen Preis der FDM-3D-Drucker [Anm.: im Vergleich zu anderen Drucktechnologien] bezahlt man mit Problemen und Nachteilen, die man mit anderen 3D-Druckern in dieser Form nicht hat. Diese Probleme in den Griff zu bekommen ist nicht trivial. Es erfordert ein gutes Verständnis der Druckmaterialien [...] sowie der Funktionen des 3D-Druckers.“ (S. 216)

Rattats fundamentale Kritik am FDM-Verfahren beachtend ist der Umstand, dass ausgerechnet seinem Buch ein Vorwort der im FDM-Druck als Vorreiter geltenden Firma Ultimaker vorangeht, überaus bemerkenswert.

1.4 Fazit zum 3D-Druck aus sozio-technischer Sicht

An das Thema „3D-Druck“ werden von vielen Seiten höchste Anforderungen gestellt. Ihnen allen ist eine äußerst hohe Erwartungshaltung gemein – die jedoch in den letzten Jahren nicht erfüllt wurde: Zu kompliziert und zu teuer seien die bisherigen Geräte. Ein Blick in die FabLabs jedoch zeigt, dass bereits jetzt 3D-Druck günstig und effektiv verwendet werden kann, sofern ein hohes Maß an Frustrationstoleranz von Seiten der Benutzer vorhanden ist. Für Lehrkräfte, die den Einsatz im Unterricht planen, bietet es sich ganz im Sinne des Modelllernens *learning by doing* an, praktische Tipps in FabLabs oder bei bereits in die Problematik des FDM-Drucks involvierten Kollegen und Kolleginnen einzuholen.

Zwar werde ich mich nicht an den vielgestaltigen Zukunftsvisionen und -spekulationen über die Auswirkungen von 3D-Druck auf unser alltägliches Leben beteiligen – auch weil diese meines Erachtens nach inzwischen mehr als überstrapaziert sind – jedoch möchte ich hervorheben, dass der weiterhin eher geringen Verbreitung des 3D-Drucks im privaten Umfeld zum Trotz keinerlei Aussagen zu finden sind, die der Technologie und ihren Weiterentwicklungen das breite Spektrum an potentiellen Anwendungsmöglichkeiten absprechen würden. Unabhängig von weiteren, antizipierten Optionen zeigt sich auch schon zum gegenwärtigen Zeitpunkt, dass 3D-Druck im Unterricht als Kontext viele fächerübergreifende Aspekte bietet und, wie im Laufe dieser Arbeit gezeigt wird, schon heute gewinnbringend im Informatikunterricht verwendet werden kann.

2. „Elementarbausteine“ in *Beetle Blocks*

Die didaktisch-konzeptionellen Vorteile visueller Programmiersprachen sind hinlänglich erforscht und im Falle von *Snap!* und *Scratch* weitbekannt, wenn diese auch leider noch immer nicht von sämtlichen Lehrkräften im Unterricht verwendet werden. Die vorteiligen Aspekte werden wegen der guten Forschungslage hier nicht weiter aufgegriffen, sondern vielmehr werden im Laufe dieser Arbeit die Überschneidungen, Chancen, aber vor allem auch die Fallstricke ausführlich erläutert, die sich in dem *Snap!*-Ableger *Beetle Blocks* aufgrund der Verschmelzung von erprobter visueller Programmierung und mathematischen Inhalten bzw. Anwendungen ergeben.

In diesem Kapitel sollen allgemeine Ideen und Konzepte, die als Art „Grundbausteine“ für Projekte auf *beetleblocks.com* dienen – stets in Hinblick auf die angestrebte Algorithmik-Einführung in der 7. Jahrgangsstufe – herausgestellt und auf ihren mathematischen und informatischen Anspruch hin analysiert werden. Zielsetzung des Unterrichtsvorhabens war es von Beginn an, den Schülerinnen und Schülern die Möglichkeit zu geben, selbstständig eigene Projekte zu entwerfen, was die Frage nach den „Grundbausteinen“ unvermeidlich macht. In Abgrenzung dazu stehen konkretere Planungen für die Unterrichtseinheit, die im Kapitel 3 ausgeführt werden.

Insbesondere, aber nicht ausschließlich, sind die auf *beetleblocks.com* unter den Kategorien *Examples* und *Beetle Blocks Team Favorites* aufgeführten Projekte in die folgenden Betrachtungen eingeflossen; die ermittelten „Elementarbausteine“ sind aufgrund der Beschaffenheit von 3D-Modellen geometrischer Natur und in drei Hauptkategorien zu fassen, die folgend vorgestellt werden.



Abbildung 1: „Elementarbausteine“ von *Beetle Blocks* schematisch visualisiert

In obiger schematischer Darstellung angedeutet ist bereits das Ineinandergreifen der einzelnen Konzepte: Existieren sie schließlich in der gleichen Rahmenumgebung, *Beetle Blocks*, und basieren auf gemeinsamen mathe-informatischen Designentscheidungen, wie der Verwendung zweier räumlicher Bezugssysteme, wie in diesem Kapitel folgend ausgeführt wird.

Warum aber überhaupt *Beetle Blocks* und keine alternative Programmierumgebung?

Pointiert ausgedrückt hebt sich *Beetle Blocks* durch den namensgebenden Käfer, der von den Schülern und Schülerinnen gesteuert werden kann, von visuellen Alternativen, wie *BlocksCAD* – das einen blockbasierten Einstieg in das von vielen FabLabs favorisierte *OpenSCAD* ermöglicht – ab. Das Extrudieren

von Formen entlang des zurückgelegten Weges unterstützt die geometrische Vorstellung der Lernenden mehr als die sonst übliche abstraktere (De-)Komposition von Objekten – ganz speziell unter Berücksichtigung der erst ca. 12-jährigen Schülerinnen und Schüler in der 7. Jahrgangsstufe an Bayerischen Gymnasien. Weiterhin stellen Romagosa, Rosenbaum und Koschitz (2016) ihre Software *Beetle Blocks* in die Tradition von Seymour Paperts *Logo* und zugleich in die direkte Vererbungslinie mit den de-facto-Standards visueller Programmierung, *Scratch* und *Snap!*.

2.1 Regelmäßige n-Ecke als Grundflächen

Bevor „wirklich“ dreidimensionale Gegenstände programmiert werden können, sollte sich zunächst die Frage nach der Grundfläche dieser gestellt werden. Neben ästhetischen Wünschen bieten sich regelmäßige n-Ecke auch aus Gründen der Parametrisierbarkeit als Grundfläche geradezu an. Im Lehrplan Mathematik der 7. Jahrgangsstufe¹ „erschließt sich ihnen [gemeint sind die Schülerinnen und Schüler] *Symmetrie als ein Ordnungsprinzip*“, was die regelmäßigen (und somit insbesondere symmetrischen) Vielecke in ihrer strukturierenden Bedeutung anerkennt. In *Beetle Blocks* kristallisieren sich im Wesentlichen drei Möglichkeiten zur Umsetzung dieser heraus.²

2.1.1 Konstruktion über Außenwinkel

Die intuitivste und algorithmisch einfachste Lösung stellt die Wiederholung des Legens einer Seite und des anschließenden Drehens um den Außenwinkel des n-Eckes dar. Ausgangspunkt für die Bewegung des Käfers ist hierbei eine Ecke des zu legenden Vielecks.

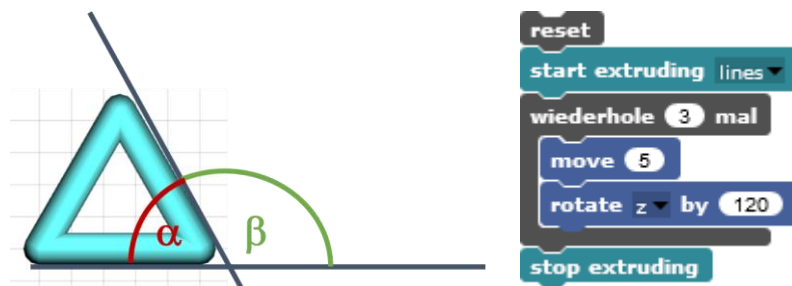


Abbildung 2: Erzeugen eines Dreiecks durch Rotation des Beetle um den Außenwinkel

Vorteilhaft erscheinen auf den ersten Blick die einfachen mathematischen Mittel, die zum Verständnis benötigt werden: Es handelt es sich im Wesentlichen um das Konzept des Innen- und Nebenwinkels. In der 7. Jahrgangsstufe Mathematik werden Innenwinkel von Drei- und Vierecken behandelt; die unterrichteten Winkelbetrachtungen umfassen Neben-, Stufen- und Wechselwinkel. An Dreiecken werden außerdem Schwerpunkt sowie Um- und Inkreis mit Zirkel und Lineal konstruiert. Den Schülerinnen und Schülern sollte des Weiteren bekannt sein, dass im gleichseitigen n-Eck Umkreis-, Inkreismittelpunkt sowie der Schwerpunkt des Dreiecks zusammenfallen.

¹ Zwar wird in dieser Arbeit explizit auf den gegenwärtigen Lehrplan Mathematik Bezug genommen, die Inhalte sind gegenüber dem „Lehrplan Plus“ der adressierten Jahrgangsstufe aber vollkommen unverändert. Aussagen zum neuen G9-Lehrplan können indes nicht getroffen werden, da dieser noch nicht vorliegt.

² Die Anweisungen unter der Kategorie *Shapes* bieten vorgefertigte Objekte, wie Quader, Kreise, usw., an – und seien an dieser Stelle außenvorgelassen. Meines Ermessens nach widersprechen diese dem Grundgedanken von *Beetle Blocks*: Proklamieren Romagosa, Rosenbaum und Koschitz (2016) doch den steuerbaren Käfer als entscheidenden Vorteil ihres Ansatzes anstelle der Komposition vorgegebener geometrischer Objekte. Für fortgeschrittene Anwender können diese vereinfachenden, vorgegebenen Bausteine durchaus sinnvoll sein, im Anfangsunterricht hingegen sorgen sie für ein weiteres Konzept (neben dem extrudierendem Käfer), das für Verwirrung sorgen kann.

Für allgemeine n-Ecke muss in obigen Code lediglich die Anzahl der Wiederholungen auf die Eckenanzahl gesetzt und der Rotationswinkel (in Abbildung 2 bezeichnet mit β) mittels des simplen Terms

$$\text{Rotationswinkel} = \frac{360^\circ}{\text{Anzahl der Ecken}}$$

berechnet werden. Problematisch wird diese Vorgehensweise dagegen, sobald den Objekten eine Höhe verliehen werden soll, da der – meist angestrebte – Umkreismittelpunkt nicht für alle regelmäßigen n-Ecke mit elementargeometrischen Verfahren zu finden ist; ausgeführt wird dies im Abschnitt 2.2.

2.1.2 Kellerspeicher mit *push & pop*

Beetle Blocks bietet mit den unter der Kategorie *Bewegung* aufgeführten Anweisungen *push position* und *pop position* einen versteckten, leider (noch) nicht visualisierten Stapelspeicher für die Position³ des *Beetle* an. Ausgangspunkt für ein alternatives Verfahren ist nicht mehr wie zuvor eine Ecke des zu implementierenden Vielecks, sondern der (Umkreis-)Mittelpunkt. Von diesem aus bewegt sich der *Beetle* zu einer Ecke, legt diese Position auf den Stapelspeicher, bewegt sich zurück zum Umkreismittelpunkt, dreht sich einmal um den Innenwinkel (einer Ecke) des Vielecks und fährt so für die übrigen Ecken fort. Die zuerst gespeicherte Eckenposition muss zuletzt ein weiteres Mal auf den Stapel gelegt werden. Nun kann sich der Käfer mittels *pop position* an den Ort einer Ecke „beamen“ und folgend mittels Extrusion und n-maligem Aufruf von *pop position* die Ecken verbinden. Abschließend muss der Beetle das Extrudieren beenden und zum Umkreismittelpunkt zurückkehren. Letzteres kann elementar geschehen, da er sich auf der Umkreislinie befindet und daher nur „Rückwärtsgehen“ muss.

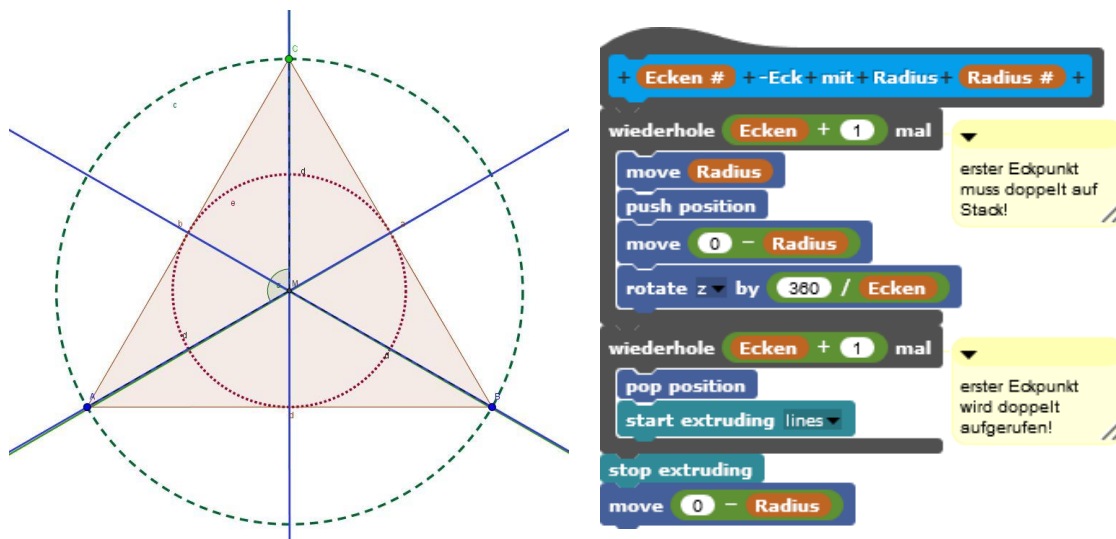


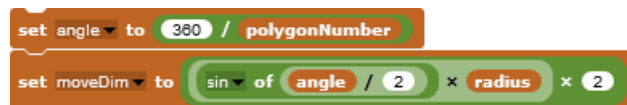
Abbildung 3: Parametrisiertes Legen eines Vielecks mittels Stackspeicher

³ Der etwas unklare Begriff „Position“ bezieht sich in diesem Zusammenhang nicht nur auf die x-, y-, z-Koordinaten des Käfers, sondern ebenso auf seine Rotation bezüglich des (ortsfestens) Koordinatensystems. Der Umstand, dass diese gespeicherte Rotation beim Aufruf von *pop position* wiederhergestellt wird, findet in den Implementierungen der Projekte überwiegend keine sinntragende Verwendung.

Der aufmerksame Leser, die aufmerksame Leserin dürfte an dieser Stelle zurecht die Berechtigung dieses Verfahrens hinterfragen: Werden zwar in der 7. Jahrgangsstufe gelehrt, jedoch eindeutig schwierigere mathematische Kenntnisse benötigt; darüber hinaus bedarf es eines Stapelspeichers, einer in der Informatik zwar fundamentalen Datenstruktur, die im Anfangsunterricht zur Algorithmik allerdings etwas deplatziert erscheint. Wie bereits erwähnt sei erneut auf Abschnitt 2.2 verwiesen; es wird gezeigt, dass diese komplizierte Vorgehensweise eindeutige Vorteile beim „Weg in die Höhe“ bietet und daher je nach Rahmenbedingungen auch im Einführungsunterricht Verwendung finden kann.

2.1.3 Mathematische Funktionen

In einer nicht geringen Anzahl an *Beetle*-Projekten finden sich zum Anlegen von Grundflächen recht schwierig anmutende Berechnungen unter Einsatz polynomialer Funktionen von Grad ≥ 2 oder trigonometrischer Zusammenhänge. Wie bereits angedeutet hängt dies mit dem Wunsch, den Umkreismittelpunkt der zumeist regelmäßig n-eckigen Grundfläche zu erhalten, zusammen (vgl. Abschnitt 2.2). So ist auch unter den offiziellen *Examples* das Projekt „Cup“, welches auf den Sinus zur Berechnung der Grundfläche zurückgreift:



```
set angle to 360 / polygonNumber
set moveDim to sin of angle / 2 x radius x 2
```

Abbildung 4: Codeauszug aus dem Example-Projekt Cup

Aus didaktischer Sicht ist dies sehr kritisch zu betrachten, da somit bereits für Grundflächen auf Mathematik der Sekundarstufe II nicht verzichtet werden kann; für die 7. Jahrgangsstufe unterdessen verbietet sich dieser wenig anschauliche Zugang sogar, da nur lineare Gleichungen bzw. Funktionen aus dem Mathematikunterricht bekannt sind.

Exkurs: Problematische Mathematik im 3D-Druck

Schon der bisherige Teils dieses Kapitels, insbesondere Abschnitt 2.1.3, zeigt auf, dass unter der bunten Oberfläche der meisten *Beetle Blocks*-Projekten nicht triviale Mathematik anzutreffen ist – ein gravierender Gegensatz zu *Scratch* und *Snap!*. Gerade dieser Umstand macht 3D-Druck im Unterricht – noch mehr als die erwähnten technischen Unwägbarkeiten – zu einem sorgfältig zu planenden Unterfangen unabhängig der zu adressierenden Jahrgangsstufe oder Schulart. Aus ihren Erfahrungen mit der (mathematisch) funktionalen Beschreibung von Objekten in der Abschlussklasse einer Realschule folgern Krisch et al. (2017):

„Für die Wiederholung der Unterrichtssequenz empfiehlt es sich, die Erstellung der Grundfiguren zunächst mit einfacheren mathematischen Mitteln auszuprobieren.“

Eine Erkenntnis, die umso mehr in jüngeren Jahrgangsstufen, sei es auch am Gymnasium, berücksichtigt werden muss. Der Wunsch nach einfacherer Mathematik wird im Grunde genommen durch den elementar-geometrisch steuerbaren Käfer unterstützt – jedoch durch die mathematisch-funktionalen Berechnungen konterkariert: Ein kleiner Blick in die etablierten Grundvorstellungen von Funktionen der Mathematikdidaktik nach Rudolf von Hofe zeigt, dass für die Vorstellung von – womöglich noch zu transformierenden – Graphen eine ausgeprägte Objektvorstellung von Funktionen benötigt wird; diese baut ihrerseits auf den anderen beiden Grundvorstellungen, der Zuordnungs- und Kovariationsvorstellung, auf (vgl. Schroeders (2017), S. 148ff.). Knapper formuliert: Ein Informatikunterricht, der solch hohe mathematische Einblicke voraussetzt, kann nur für eine sehr eingeschränkte Zielgruppe attraktiv sein.

Anstatt obiger, funktionaler Zusammenhänge könnten an dieser Stelle Gleichungen, wie der Satz des Pythagoras, in ihrer Eignung für *Beetle Blocks* diskutiert werden – ein Ansatz, der in eine falsche, überwunden geglaubte Richtung zielen würde:

„Der Umstand, dass Computer (zu deutsch: Rechner) ursprünglich vor allem für mathematische Berechnungen verwendet wurden, scheint bis heute einen erstaunlichen Einfluss auf die informativische Bildung auszuüben: Viele Probleme entstammen der Mathematik. Fraglich ist, ob Schüler im Informatikunterricht nicht besser lernen sollten, Probleme der Informatik mit Methoden der Informatik zu lösen, da sich das Fach sonst seiner Legitimation beraubt.“ (Romeike (2008), S. 55)

Fragestellungen ganz ohne mathematische Bezugspunkte lassen sich aufgrund der räumlichen Dimension als konstitutives Merkmal von 3D-Objekten zwar kaum aufwerfen, jedoch kann sich Informatikunterricht – auch mit 3D-Druck – auf absolutes Grundlagenwissen⁴ im Bereich der elementaren Geometrie stützen und diesen mathematischen Basiskenntnissen neue, sinnhafte Anwendungskontexte darbieten. Diese Maxime versucht die entwickelte Unterrichtseinheit, vorgestellt in Kapitel 3, zu verwirklichen.

2.2 Wege in die dritte Dimension

Mit etwas kreativem Einsatz lassen sich allein durch die bereits vorgestellten Grundflächen ästhetische 3D-Drucke herstellen; das eigentlich Faszinierende am 3D-Druck stellt aber weniger das Drucken ebener Strukturen als das Erzeugen räumlicher Objekte dar. Insofern sollte dieser Aspekt im Unterricht genügend Raum erhalten – ein spannendes Vorhaben, bedenkt man, dass die Schülerinnen und Schüler erst in der 11. Jahrgangsstufe mit dem dreidimensionalen kartesischen Koordinatensystem in Berührung kommen. Um den in 2.1 vorgestellten Grundflächen Tiefe zu geben, bilden sich zwei Optionen heraus.

2.2.1 Gestapelte Grundflächen

Die naheliegendste Idee zum Erstellen von „echt“ dreidimensionalen Körpern ist sicher das Übereinanderlegen von Grundflächen.⁵ In Zusammenhang mit 3D-Druckern ergibt sich der weitere Vorteil, „*dass der 3D-Drucker seine Objekte auch durch Hinzufügen einzelner Schichten aufbaut*“ (Krisch et al. (2017)). Somit vollzieht der programmierte Käfer ähnliche Bewegungen wie der Druckkopf später, was – gerade wenn, wie im Regelfall, das *Slicing*-Verfahren zum Erzeugen von *G-Code* nicht im Unterricht thematisiert wird – das Verständnis von der Funktionsweise eines (Schmelzschicht-)Druckers fördert.

⁴ Keineswegs gleichzusetzen sind die hier erwähnten Grundkenntnisse mit den an vielen Schulen eingeführten sog. „Grundwissenskatalogen“ des Faches Mathematik; letztere sind viel ausführlicher und stellen im Mathematikunterricht eher ein wünschenswertes Soll- als ein tatsächlich vorhandenes Ist-Wissen dar.

⁵ Ganz präzise ausgedrückt besitzen natürlich auch die vom *Beetle* extrudierten Flächen eine Höhe und stellen damit keine Flächen im mathematischen Sinne dar; aus letzteren könnte durch Aufsichten schließlich auch kein Körper entstehen. Zugunsten eines verständlicheren Sprachgebrauchs wird in diesem Dokument auf eine penible Differenzierung im mathematischen Sinne verzichtet.

Im Falle einer über Außenwinkel konstruierten Grundfläche aus Abschnitt 2.1.1 zeigt sich nun, dass die mathematischen Schwierigkeiten nicht umgangen, sondern nur ins Dreidimensionale verlegt wurden.

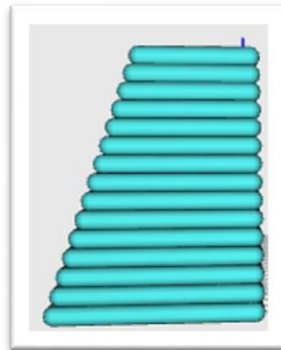


Abbildung 5: Turm gestapelter Vierecke ohne Achse durch Umkreismittelpunkt

Ein einfaches Übereinanderlegen (und Verkleinern) von z. B. Vierecken führt zu dem schiefen Turm aus Abbildung 5, da die Höhenachse durch den Ausgangspunkt des *Beetle* beim Erstellen der Grundfläche, also durch eine Ecke, läuft. Um wieder ein symmetrisches Gesamtbild zu erzeugen, könnte entweder der Turm als Ganzes (hier: viermal um 90°) rotiert werden, so dass ein größerer Turm mit inneren Querstreben entsteht. Naheliegender kann bei Vier- und Sechsecken der Umkreismittelpunkt elementargeometrisch – über die (halbe) Seitenlänge und den bekannten respektive über einfache Winkelbetrachtungen herleitbaren Innenwinkel von 90° bzw. 60° – ermittelt werden. Der entstehende Turm entspricht den Erwartungen:

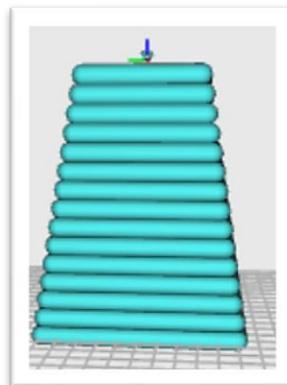


Abbildung 6: Turm mit Lotachse durch Umkreismittelpunkt der quadratischen Grundfläche

Fortsetzung siehe nächste Seite.

Vergleicht man den Quellcode der beiden Türme aus Abbildung 5 und Abbildung 6, ergibt sich, dass das an sich einfache Übereinanderlegen, welches nur durch den Befehl *change absolute z by* realisiert wird, für ansprechende Ergebnisse mit weiteren Anweisungen angereichert werden muss:



Abbildung 7: Quellcode des schiefen Turmes (links) sowie seines symmetrischen Pendants (rechts)

Dies hebt die Vorteile der n-Ecke aus 2.1.2 hervor: Muss bei diesem Ansatz zwar zunächst vieles an Überlegungen erfolgen und ein klares algorithmisches Vorgehen unter Einsatz eines Kellerspeichers entwickelt werden, gestaltet sich der Weg ins Dreidimensionale um ein Vielfaches einfacher. Anschließend an den Exkurs auf Seite 11 kann dem *push-and-pop*-Ablauf im Dreidimensionalen eine eher informatisch, den n-Ecken aus 2.1.1 eine eher mathematisch motivierte Hintergrundproblematik attestiert werden. Der Schwierigkeitsgrad ist bei den über Außenwinkel konstruierten Vielecken zu Beginn sehr einstiegshilfreich und beim Verleihen von räumlicher Tiefe erhöht, während sich dies bei den mittels eines Stapelspeichers implementierten n-Ecken genau invers dazu verhält.

2.2.2 Rotation um x- oder y-Achse

Romagosa (2016) stellt in seinen Vortragsfolien für katalanische Workshops folgendes Schneckengehäuse vor, das mit erstaunlich wenig Programmbausteinen auskommt:

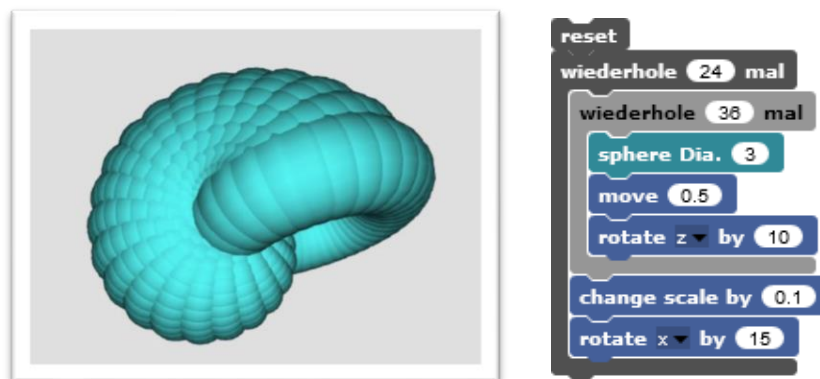


Abbildung 8: Schneckengehäuse nach Romagosa (2016) mit übersetzten Blöcken

Auf die Anweisung *change scale by* wird später noch näher eingegangen; gegenwärtig soll das Räumlichkeit verleihende *rotate x by* eingehend betrachtet werden, da es durch das Rotieren des Käfers um eine weitere Achse eine Möglichkeit zum Generieren dreidimensionaler Objekte aus Grundflächen im Unterricht darzustellen scheint. Hierzu ist es nötig, die Funktionsweise der Anweisung in Ermangelung eines Handbuches⁶ für *Beetle Blocks* selbst näher in Augenschein zu nehmen.

Strukturelle Basis für Rotationen sind die zwei Koordinatensysteme von *Beetle Blocks*: Ein „klassisches“, ortsfestes Koordinatensystem sowie ein bewegliches Koordinatensystem, das sich mit dem *Beetle* mitbewegt.

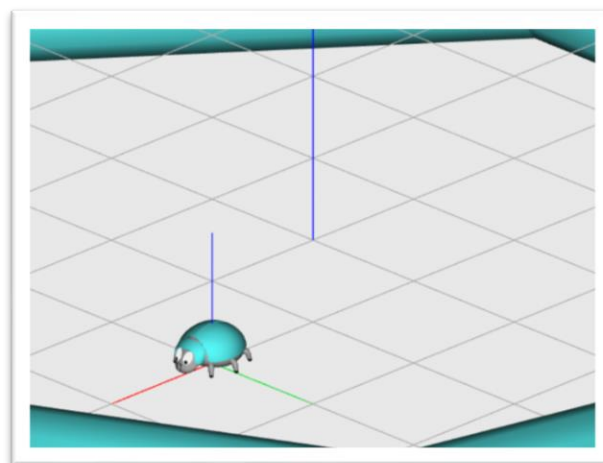


Abbildung 9: Die zwei Koordinatensysteme in Beetle Blocks, gut erkennbar anhand der jeweiligen blauen z-Achse

Der Bedarf an zwei verschiedenen Bezugssystemen ergibt sich aus dem Wunsch, die Position des Käfers auf „traditionelle“ Weise angeben zu können und doch gleichzeitig aus der Sicht des *Beetle* programmieren zu können – letzteres als bedeutsamer Unterschied zu Konkurrenzsoftware. Durch die Verwendung dieser beiden Koordinatensysteme, die bei erster Betrachtung auch sehr intuitiv scheint,

⁶ Rosenbaum und Koschitz arbeiten derzeit an *The Beetle Blocks Primer*, einer Art Einstiegshilfe für *Beetle Blocks*, in der auch eine Systematisierung von Projektbausteinen unternommen wird. Interessenten sei die Lektüre zwar empfohlen, bisweilen ist die Zielgruppe des Dokuments jedoch unklar.

ergeben sich bei fortgeschrittener Verwendung durch die fehlende explizite Trennung der beiden Koordinatensysteme Inkonsistenzen, die es im Unterricht zu vermeiden gilt.⁷ Exemplarisch soll eine Wendel programmiert werden, die aus „verschobenen“ Quadraten besteht; für diese genügt es nämlich nicht den Käfer „nach hinten zu lehnen“ (*rotate y by 10*), wie folgendes Schema illustriert:

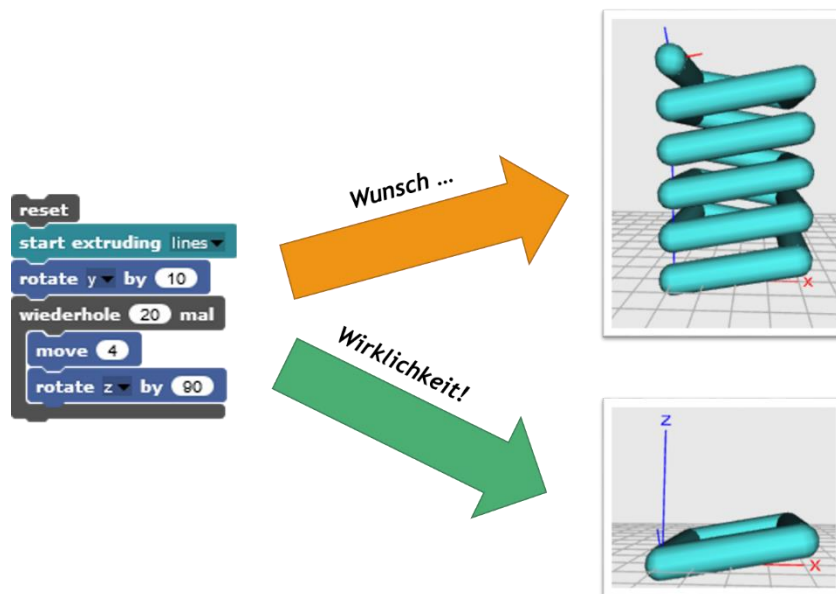


Abbildung 10: Fehlvorstellung resultierend aus der Existenz zweier Bezugssysteme

Die obige Fehlvorstellung, der gerade Programmieranfänger schnell erlegen können, besteht darin, dass die Rotation des Beetle nicht nur durch *rotate y by 10*, sondern auch den für das Quadrat nötige *rotate z by 90* verändert wird – und zwar in Bezug auf sein bewegliches Koordinatensystem. Die Problematik lässt sich im Unterricht (und nicht nur dort) gut mit einem mit Achsen versehenen Stofftier motivieren – sofern sich die Lehrkraft überhaupt für das Thematisieren der Achsenrotation als zweiten Weg ins Dreidimensionale entscheidet.

Zwar funktioniert in Abbildung 8 der verfolgte, rotierende Ansatz gut, jedoch ist er nach obigen Beispiel kaum auf anders geartete Exempla übertrag- und wenig für den (Anfangs-)Unterricht systematisierbar, da einerseits beim Rotieren um mehr als eine Achse zwangsläufig von dem gewohnten (ortsfesten) Koordinatensystem stets zum zweiten, beweglichen Koordinatensystem gedanklich gewechselt werden muss; andererseits sind die Rotationsanweisungen (um verschiedene Achsen) in ihrer Reihenfolge nicht kommutativ – mathematisch dem Umstand der im Allgemeinen nicht-kommutativen Multiplikation von Transformationsmatrizen geschuldet. Dies kann mit den Schülerinnen und Schülern aber kaum auf diesem Niveau kommuniziert werden und stiftet (trotz veranschaulichendem Stofftier) schnell Verwirrung.

Um auf die Wendel aus Abbildung 10 zurückzukommen: Bei der (korrekten) Implementierung dieser ist ein wiederholtes „Nachrotieren“ um die y-Achse (aus *Beetle*-Perspektive) nötig, um das gewünschte Resultat zu erzielen. Im Kontrast zu dem eingangs vorgestellten Schneckengehäuse wird die Umsetzung der Wendel durch die zwei Koordinatensysteme sogar erschwert.

⁷ Mit dem Phänomen zweier Bezugssysteme kommen Schülerinnen und Schüler in der Schule übrigens kaum in Berührung; am naturwissenschaftlichen Zweig wird dies in Zusammenhang mit der Zentrifugalkraft im Fach Physik in der 10. Jahrgangsstufe erwähnt.

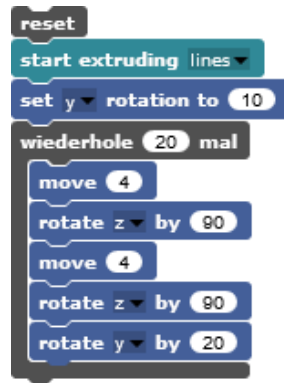
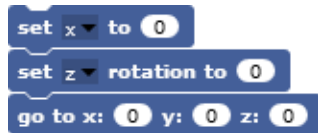


Abbildung 11: Korrekte Implementierung der Wendel mit größerem Codeumfang

Exkurs: Im Unterricht zu berücksichtigende Auswirkungen zweier Koordinatensysteme
 Wie in 2.3 noch erläutert wird, bringt der Einsatz zweier Koordinatensysteme in *Beetle Blocks* – neben dem nicht zu verachtenden Alleinstellungsmerkmal, der Programmierung aus Käfersicht – noch weitere positive Aspekte mit sich; ungeachtet dessen bleiben die in 2.2.2 ausgeführten Rotationen um eine Achse weiterhin im Unterricht als kritisch zu betrachten. Von diesen Rotationen abgesehen, könnten jedoch viele problematische Punkte der zwei Bezugssysteme verringert, wenn nicht sogar vermieden werden, würden die vorhandenen Anweisungen unter der Kategorie *Bewegung* nochmals unterteilt werden: Mit Bezugnahme auf das ortsfeste bzw. bewegliche Koordinatensystem.

Beispiel für Befehle aus ortsfixierter Sicht



Entsprechungen für Befehle aus Beetle-Sicht

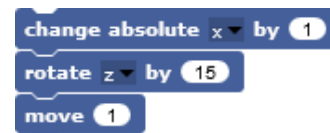


Abbildung 12: Exemplarischer Vergleich von redundanten Befehlen

Die bisher vorherrschende Vermischung dieser beiden Gruppen an Befehle erschwert das eigenständige, explorative Lernen erheblich und macht eine Art Vorauswahl von Anweisungen für den Unterricht beinahe unabdingbar.

2.3 Rotationssymmetrien

Während in 2.2.2 am Beispiel der Wendel dargelegt wird, dass durch den Einsatz zweier Koordinatensysteme manche Objekte in ihrer Implementierung sogar erschwert werden, existieren diametral dazu ebenso viele Muster, die durch das zusätzliche Käfer-Koordinatensystem überhaupt erst richtig ermöglicht werden. Schon beim Durchstöbern der Projekte auf *beetleblocks.com* fällt auf, wie „schön“ – fachlicher ausgedrückt „wie (rotations-)symmetrisch“ – viele dieser sind.

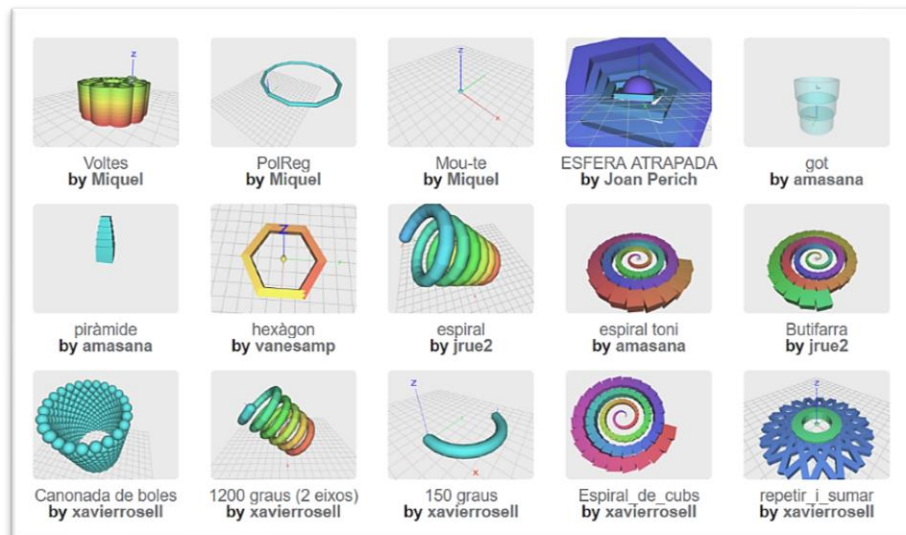


Abbildung 13: Screenshot von beetleblocks.com im Juli 2017

Die häufige Verwendung von Rotationssymmetrie, wie sie in komplexerer Form auch beim Schneckengehäuse aus Abbildung 8 vorliegt, erklärt sich dadurch, dass es durch das *Beetle*-Koordinatensystem genügt, ein (vom Ursprung des ortsfixierten Bezugssystems aus gedachtes) Muster zu legen, den Käfer zum Ursprung zurückkehren und sich rotieren zu lassen, um das Muster zu wiederholen. Gerade in den ersten Unterrichtsstunden kann dies zur Motivation durch das einfache Erstellen symmetrischer Ergebnisse beitragen.

2.4 Fazit zu den ermittelten Konzepten

So vielfältig die verschiedenen Projekte auf beetleblocks.com auch sein mögen, ist es doch gelungen, eine praktikable Einteilung in „Elementarbausteine“ zu finden, die sich überraschend stabil über etliche Projekte hinweg herauskristallisierte. Das auf Seite 8 erwähnte Ineinandergreifen der „Bausteine“ bedingt allerdings auch eine gewisse Trennschärfe, ähnlich wie das Kategorisieren von Ideen auch fast immer eine schwankende Granularität der genannten Prinzipien verursacht. In diesem Sinne sei auf das zu Kapitelbeginn vorgestellte Forschungsanliegen, das Finden einer praktischen Einteilung in Hinblick auf den algorithmischen Anfangsunterricht in der 7. Jahrgangsstufe, verwiesen. Für weiterführenden Unterricht können sicher noch mehrere solcher „Bausteine“ identifiziert werden: Beispielsweise könnte die Vielzahl an Projekten, die Rekursion und Fraktale verknüpfen, ein interessanter Anknüpfungspunkt für weitere Forschung sein.

3. Einführung in die Algorithmik mit 3D-Druck

Nach den in Kapitel 2 geklärten Voraussetzungen von *Beetle Blocks* kann die Planung der eigentlichen Unterrichtseinheit beginnen, deren Prozess in diesem Kapitel dargelegt wird. Als wichtiger Punkt werden Hintergründe des entwickelten und erprobten Unterrichtsmaterials vorgestellt sowie die Erfahrungen aus Lehrerperspektive vorgestellt. Eine tiefergehende Evaluation schließt sich im vierten Kapitel an.

3.1 Rahmenbedingungen des Unterrichtsvorhabens

Zielgruppe der Unterrichtsreihe sind 7. Klassen mit jeweils in etwa 28 Schülerinnen und Schülern eines Bayerischen Gymnasiums, für die der Lehrplan im Fach *Natur und Technik* unter dem Punkt „NT 7.2.3 Beschreibung von Abläufen durch Algorithmen“ (ISB (2004)) unter anderem folgendes für einen circa 13-stündigen Unterricht vorschlägt:

- Grundprinzipien der automatischen Informationsverarbeitung
- altersgemäße Visualisierung der Einzelschritte
- Bausteine von Algorithmen: Anweisung, Sequenz, bedingte Anweisung, Wiederholung
- Programmieren eines einfachen Informatiksystems unter Verwendung dieser Bausteine

Bereits zum Zeitpunkt der Unterrichtsplanung war absehbar, dass der nach den Pfingstferien im Juni stattfindenden Lerneinheit keine 13 Unterrichtsstunden in den beiden Erprobungsklassen zur Verfügung gestellt werden können, sondern nur sieben Unterrichtsstunden, die meist in Form von Doppelstunden eingeräumt werden können: Antizipierter Unterrichtsausfall wegen „Hitzefrei“ oder anderer Aktivitäten – vorhergesehener oder unvorhergesehener Natur – wie Sportfesten oder dem „Abischerz“, stellen zu Schuljahresende, an dem die Computerräume in mehrwöchigem Vorfeld zu reservieren sind, stets ein gewisses Hindernis für produktiven Unterricht dar. Dennoch bietet es sich wegen der vielfältigen Bezugnahme auf im Mathematikunterricht der 7. Jahrgangsstufe zu erwerbende Basiskompetenzen nicht an, die Unterrichtseinheit (zu) früh im Verlauf des Schuljahrs durchzuführen.

Die beiden 7. Klassen unterschieden sich hinsichtlich ihres Sozialverhaltens und Leistungsfähigkeit so sehr – die eine beinahe mustergültig und überaus leistungsstark, die andere zwar freundlich, jedoch deutlich pubertär und mit unterdurchschnittlichen Leistungen – dass der ursprünglich angedachte Ansatz, eine Klasse „konventionell“ – an dieser Schule leider noch üblicherweise mit *Robot Karol* – und die andere mit dem neuen Konzept zum 3D-Druck zu unterrichten, verworfen werden musste und beide Klassen am 3D-Druck teilhaben durften. Im Vorfeld des 3D-Druck-Unterrichts hospitierte ich im Unterricht zur Physik – Informatik war schon länger nicht mehr unterrichtet worden – und sprach daraufhin die Lehrkraft auf den klar lehrerzentrierten Unterricht an, welcher plausibel aus Zeitmangel begründet wurde. Ungeachtet dessen, ist die geplante und durchgeführte Unterrichtseinheit eindeutig für schülerorientierten Unterricht zugeschnitten kumulierend in einem kleinen, kreativen „Mini-Projekt“, in dem die Schülerinnen und Schüler ein eigenes Objekt kreieren konnten.

Als technische Ausstattung im Computerraum fand sich nur pro Schülerpaar ein Computer, was die Jugendlichen zwar gewohnt sind und dennoch didaktisch die Möglichkeiten verschiedener Sozialformen in „erzwungener Partnerarbeit“ deutlich einschränkt; aus diesem Grund wird auch folgend auf diese kaum Bezug genommen. Das nahe gelegene FabLab NüLand e. V. stellte freundlicherweise einen kleinen, günstigen und insbesondere portablen 3D-Drucker, den *Up mini* von PP3DP, dem trotz des Aktionspreises von knapp 170 € bei einem bekannten Kaffeeröster ein flottes Arbeitstempo und „akzeptable Ergebnisse“ (König (2015)) bescheinigt werden, zur Verfügung. Er kann für den Unterricht

aufgrund der kompakten Abmessungen direkt in den Computerraum gebracht werden, wovon gerne Gebrauch gemacht wurde.

3.2 Weitere informatische Konzepte in *Beetle Blocks*

Wurden im vorausgegangenen Kapitel bereits „Grundbausteine“ von Projekten in *Beetle Blocks* untersucht, gilt es noch Voraussetzungen für weitere informatische Inhalte in der Programmierumgebung zu untersuchen: Bemerkenswerterweise besitzt eine in anderen Programmiersprachen elementare Kontrollstruktur eine deutlich weniger bedeutsame Rolle in *Beetle Blocks*, für ein anderes informatisches Konzept existiert eine mathematisch motivierte (Schein-)Alternative.

3.2.1 Bedingte Anweisungen in *Beetle Blocks*

if-Anweisungen stellen nicht nur eine fundamentale Idee der Informatik dar, sondern werden auch explizit im Lehrplan (s. o.) – ganz im Gegensatz zu der hier vorliegenden Arbeit – erwähnt. Dies hängt damit zusammen, dass tatsächlich nur in etwa einem Drittel der über zwanzig *Beetle Blocks Team Favorites* auf beetleblocks.com überhaupt (!) eine bedingte Anweisung vorkommt. Ein ähnliches Bild liefern die *Examples*.

Daher ist es Aufgabe, die Kontexte, in denen ein *if* sinnvoll verwendet werden kann, zu finden und auf ihre didaktische Eignung zu überprüfen. Eine sicherlich sinnhafte, jedoch für die adressierte Jahrgangsstufe völlig unpassende Anwendung bieten Abbruchbedingungen von Rekursionen bei Fraktalen, wie sie in einer großen Anzahl von Projekten vorkommen. Folgend werden drei Projekte, die in ihrer Verwendung der bedingten Anweisung exemplarisch für eine Gruppe von Projekten stehen können, vorgestellt:

1. Räumliche Begrenzung eines Zufallsmusters

2d random walk within boundary by Examples



Abbildung 14: Screenshot und Codeauszug des Projekts *2d random walk within boundary by Examples*

Die bedingte Anweisung wird in diesem sogar unter den *Examples* aufgeführten Projekt als räumliche Begrenzung eines *Random Walk* unter Verwendung der Kreisgleichung eingesetzt. Zwar steht die Kreisgleichung als solche in der 7. Jahrgangsstufe noch nicht zur Verfügung, aber lineare Ungleichungen zur

Beschreibung quadratischer oder rechteckiger Flächen wären bereits in dieser Jahrgangsstufe⁸ vorstellbar. Darüber hinaus ist keine spezielle Einführung von Variablen notwendig. Problematisch ist hingegen trotz der sinnvollen Motivierung der Begrenzung für sämtliche Zufallsbeispiele der fehlende persönliche Bezug zum Resultat. Diese sind in der Regel auch wenig ästhetisch, hier vorgestelltes Beispiel ist wegen seiner filigranen – das heißt schlecht bis gar nicht druckbaren – Strukturen ein Sonderfall.

2. Strukturierung von Code mit Variablen

Anna Hillier 83 Coin by anna



Abbildung 15: Screenshot und Codeauszug des Projekts Anna Hillier 83 Coin by anna

Schon der kleine Auszug der Programmblöcke dieses sehr aufwändigen, beeindruckenden Projektes wirkt relativ kompliziert. Offenbar wurde die Münze per Hand geplant und entsprechend danach implementiert. Um den Code etwas übersichtlicher zu gestalten, zu strukturieren, wird auf Variablen, die in der Kombination mit *if*-Anweisungen den Kontrollfluss steuern, zurückgegriffen. Für den an fundamentalen Ideen orientierten Informatikunterricht ist diese wenig anschauliche, schlecht zu motivierende Vorgehensweise zur Algorithmikeinführung jedoch nicht geeignet.

3. Variable zur Mustererzeugung

2d staggered grid by Examples



Abbildung 16: Screenshot und Codeauszug des Projekts 2d staggered grid by Examples

⁸ Lineare Ungleichungen werden im Mathematikunterricht erst in der 8. Jahrgangsstufe explizit behandelt, da die Unterrichtseinheit aber gegen Schuljahresende stattfindet, stellt die Kurzeinführung dieser keinen Hinderungsgrund in der 7. Jahrgangsstufe dar.

Ein sinnvoll zu motivierendes Beispiel ist sicherlich das in vielen Kontexten einsetzbare Erzeugen eines Musters mithilfe einer Variable, *if*-Anweisung und häufig einer Bedingung, die den Modulo-Operator verwendet. Allerdings ist auch hier die didaktische Eignung für einen Anfangsunterricht nur bedingt gegeben, müssen doch Variablen in ihren Veränderlichenaspekt und Bedingungen nicht nur bekannt, sondern auch gut verstanden sein. Per se ist diese Anwendung außerdem wenig anschaulich und muss entsprechend zugänglich gemacht werden.

Notwendigkeit von weiteren Beispielen

Keine der drei vorgefunden Möglichkeiten, bedingte Anweisungen in *Beetle Blocks* zu verwenden, scheinen sich wirklich gut für den Anfangsunterricht zu eignen, weshalb von Lehrerseite eine Anwendungsmöglichkeit „konstruiert“ werden muss. Dies stellt ein Manko dar, für das höchstens die in *Beetle Blocks* umso anschaulicher zu implementierenden Wiederholungen entschädigen können. Nachfolgend werden daher zwei von mir erdachte Ansätze gezeigt, die im Unterricht als Beispiel für *if*-Anweisungen herangezogen werden können.

Das Bedürfnis nach einer bedingten Anweisung sollte sich organisch aus dem schulischen Beispielen heraus ergeben, weshalb bedingte Anweisungen im Zusammenhang mit Fehlerbehandlung bzw. robuster Programmierung als Ansatzpunkt dienen können. Der *Swirl* aus der von *Logo* inspirierten *Turtle Art Gallery*⁹ lässt sich auch ansprechend dreidimensional umsetzen.

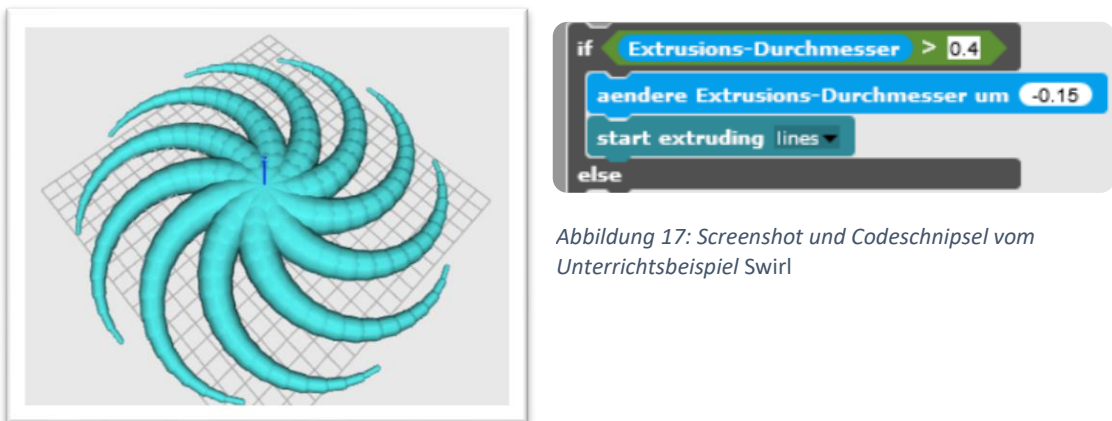


Abbildung 17: Screenshot und Codeschnipsel vom Unterrichtsbeispiel Swirl

Idee ist, den Extrusionsdurchmesser, der als selbst erstellter Block verfügbar gemacht und in einem ebenfalls selbst erstellten Block geändert werden muss,¹⁰ für die einzelnen Tentakelarme immer weiter zu verringern ohne, dass dieser negativ wird. Wird der Durchmesser dennoch negativ, so zeigt Beetle Blocks in der gegenwärtigen Version undefiniertes Verhalten.¹¹

⁹ Verfügbar unter <https://turtleart.org/gallery/index.html>, Stand: 21. Sep. 2017.

¹⁰ Ärgerlich ist nicht nur, dass der Extrusionsdurchmesser nicht als *Reporter* vorhanden ist, sondern ebenso, dass er nur dann geändert werden kann, wenn der Beetle nicht extrudiert. Dies bedeutet, dass zwischen *start extruding* und *stop extruding* keine Änderungen an diesem übernommen werden. Dem Team von *Beetle Blocks* wurde dies zusammen mit anderen Fehlern bereits mitgeteilt.

¹¹ Abbrüche des Programmes mit Fehlermeldung, blinkendes Vorschaufenster, seltsam extrudierte Formen, etc.

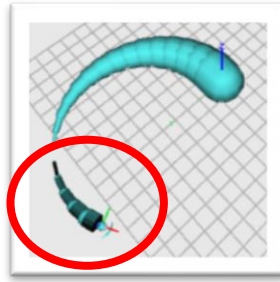


Abbildung 18: undefinierbare extrudierte Formen (links unten) bei negativem Durchmesser

Umsetzung im Unterricht und Zwischenfazit

Zwar bietet der *Swirl* ein ansprechendes Objekt zum Ausdrucken und verdeutlicht die Sinnhaftigkeit bedingter Anweisungen gut – jedoch ist die Anwendung im Kontext dieser *if*-Anfrage auf andere Projekte nur sehr schwer übertragbar und unterstützt damit das Endziel des Unterrichts, dem am Ende der Unterrichtseinheit geplanten Raum für kreative Mini-Projekte der Schülerinnen und Schüler, nicht. Deshalb wurde entschieden, den *Swirl* „nur“ als Projektanregung zum Schluss zu verwenden und stattdessen im regulären Unterricht die Bodenfläche von n-eckigen Grundflächen¹² auszulegen (vgl. Unterrichtsmaterial im Anhang).

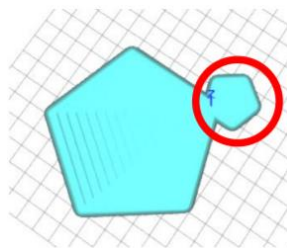


Abbildung 19: Motivation einer bedingten Anweisung zum Ausfüllen einer fünf-eckigen Grundfläche

Ein herausragend leistungsstarker Schüler griff statt der *if*-Anweisung gerne auch auf mathematische Beziehungen bei der Angabe der Schleifendurchläufe unter Verwendung von Operatoren zurück, während aber das Gros der Klasse weiterhin auf die im Unterricht erarbeitete Lösung zurückgriff. Als wirklich „ideal“ kann die vorgestellte Lösung dennoch nicht bezeichnet werden. Ob sich eine solche finden lässt, ist indes ungewiss, da die vorausgegangenen Beobachtungen zeigen, dass *if*-Anweisungen in *Beetle Blocks* bei weitem nicht so genuin sind wie Schleifen.

3.2.2 *change scale by* als Alternative zu Variablen?!

Da Variablen als solche im Kontrast zu bedingten Anweisungen in der 7. Jahrgangsstufe nicht Teil des Lehrplanes sind, ist fraglich, ob diese – besonders angesichts des geringen Studienpensums – überhaupt in einer Unterrichtsreihe mit 3D-Druck unterrichtet werden sollten. Für das im Arbeitsmaterial und unter 3.2.1 kurz vorgestellte Beispiel, dem Auslegen von Grundflächen, sind diese jedoch – wie auch für das Erzeugen sich in die Höhe verjüngender Türme – zwingend notwendig.¹³ Seite 15 aber zeigte als Beispiel für mehrachsige Rotationen das von Bernat Romagosa entworfene Schneckengehäuse, welches trotz sich in ihrer Größe verändernder Kreise ganz ohne Variablen auskommt. Ein weiteres Beispiel für diese verführerisch kurzen Programme bietet das sich unter den *Examples* von *beetleblocks.com* befindliche Projekt *Simple Cup*.

¹² Die gleichseitigen n-Ecke sind zentrales Element der Unterrichtseinheit, weshalb sich das Einführen der *if*-Anweisung anhand dieser mehr anbietet als der *Swirl*.

¹³ Für den *Swirl* könnte man diese mit den vorgefertigten Blöcken „umschiffen“, weshalb er als primäres Beispiel für *if*-Anweisungen von mir lange Zeit nicht verworfen wurde.

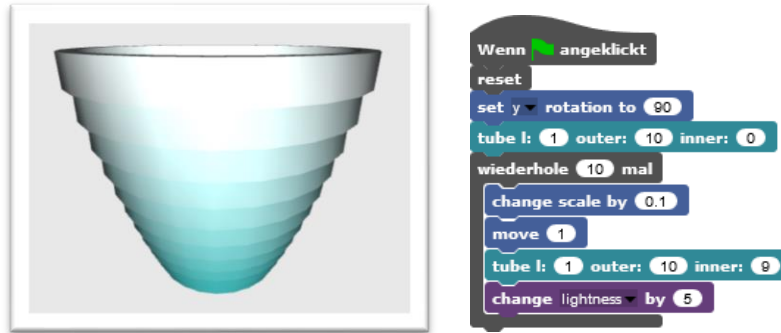


Abbildung 20: Screenshot des Simple Cup's mit Quellcode

Die „Magie“ des kurzen Programmes findet dabei in der Anweisung *change scale by* statt, welche nach ihrem Aufruf (für den Nutzer aber ohne Extrusion und *Beetle*-Bewegung unsichtbar) das interne *Beetle*-Koordinatensystem in seiner Skalierung ändert.¹⁴ Damit einhergehend verändern sich anhand vom Beispiel *Simple Cup* sowohl der Innendurchmesser der *tubes* als auch deren Dicke. Dies stellt bereits einen gewichtigen Kontrapunkt dieses Befehls dar, werden durch ihn die an sich grundverschiedenen Konzepte von Durchmesser bzw. Dicke von (durch den Käfer extrudierten) Formen und Abstände im Koordinatensystem aneinandergelüpft – was außerdem auch noch die Wiederverwendbarkeit der Anweisung einschränkt: Mitunter kommt es bei falscher Anwendung zu „schwebenden“ Schichten ohne Verbindungsstück. Weiterhin handelt es sich um ein nicht triviales mathematisches Konzept, nämlich einen zeitlich während des Programmablaufs nicht-konstanten, sich verzerrenden Raum, weshalb den informatischen Variablen mit darüber hinaus allgemeinbildendem Anspruch der Vorzug gegenüber dem äußerst ambivalenten *change scale by* gegeben werden sollte. Ergo ist der *Simple Cup* nur dem oberflächlichen Anschein nach tatsächlich simpel.

3.3 Gestaltung der Unterrichtseinheit

Der Verlauf der Unterrichtsreihe folgt im Groben der untenstehenden Abbildung, in der zu den einzelnen (Doppel-)Unterrichtsstunden die jeweiligen Seiten der Arbeitsblätter angegeben sind. Das Arbeitsmaterial¹⁵ ist recht ausführlich gestaltet, damit die Schülerinnen und Schüler im Wesentlichen selbstständig vorgehen und andere Lehrkräfte den Stundenverlauf bei Wiederholung der Unterrichtseinheit besser einsehen können. Die einzelnen Stunden werden deshalb hier nicht im Detail beschrieben, sondern nur auf Besonderheiten aufmerksam gemacht und erste Erfahrungen aus Lehrerperspektive mitgeteilt. Für eine abschließende Bewertung ist zusätzlich die Evaluation in Kapitel 4 (mit Befragung der Jugendlichen) heranzuziehen.

¹⁴ Für meinen Geschmack ist es außerdem in Hinblick auf weitere Einsatzmöglichkeiten wenig intuitiv, dass eine Skalierung, ein „Zoom“ additiv angesprochen wird statt multiplikativ, d. h. ein zweimaliger Aufruf von *change scale by 10* führt beispielsweise zu einem *scale* von 120 und nicht etwa 121.

¹⁵ Die Nummerierung der Arbeitsblätter im Anhang folgt ihrer eigenen Reihenfolge; sie sind samt dieser Seitenzahlen so im Unterricht eingesetzt worden.

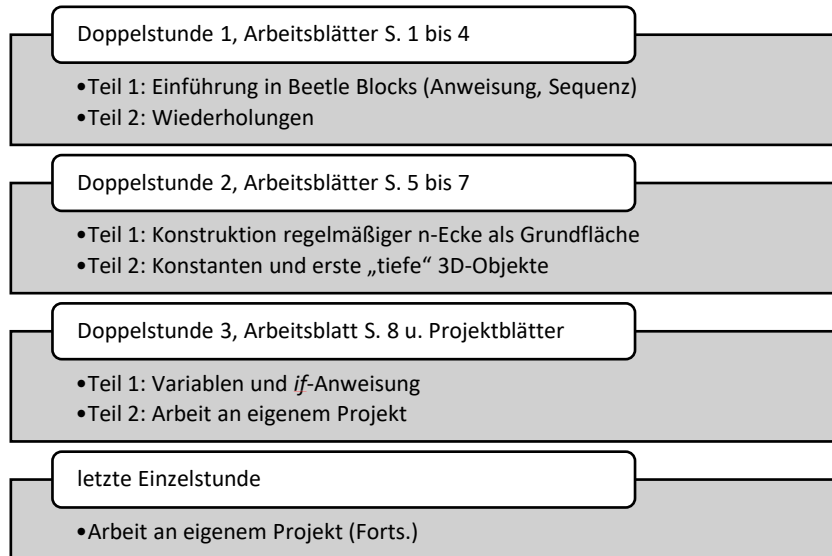


Abbildung 21: Kurzgefasste Stundenübersicht

Erste Doppelstunde – Motivation als tragendes Element von Beginn an

In Anbetracht der geringen Stundenanzahl und der hohen Stoffdichte war es wichtig, die Schülerinnen und Schüler von Anfang an für das Thema zu begeistern. Wir zeigten dazu zu Beginn der ersten Stunde ausgedruckte 3D-Objekte, wie sie die Jugendlichen zum Ende der Unterrichtsreihe auch selbst kreieren können. Wichtiges Element ist außerdem der 3D-Drucker und das Versprechen, die schönsten Initialen¹⁶ (siehe Arbeitsblatt S. 2) nach 45 Minuten noch in der Doppelstunde „live“ auszudrucken. Diese Aufgabe mit *Challenge*-Charakter wurde mit Begeisterung umgesetzt. In der ca. 5-minütigen Zäsur zwischen erster und zweiter Unterrichtsstunde hatten interessierte Schülerinnen und Schüler die Option, den 3D-Drucker in seiner Arbeitsweise beim Ausdrucken einer weiteren Initialie zu beobachten statt eine Pause einzulegen – eine Möglichkeit, von der ein Großteil der Klasse wahrgenommen wurde.

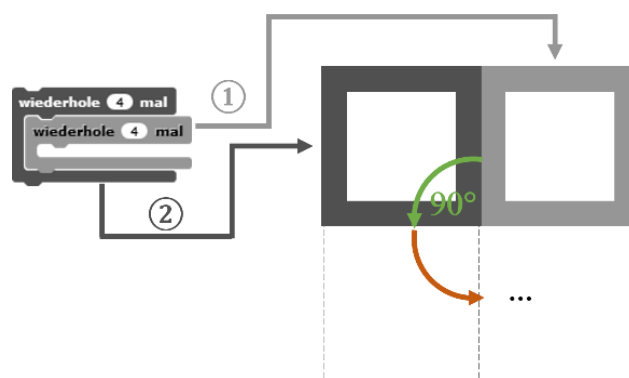


Abbildung 22: Zusammenhang geschachtelte Wiederholung und entstehende Figur (aus dem Unterrichtsmaterial S. 4)

¹⁶ Im Allgemeinen haben die so entstandenen Initialen keine Verbindung, würden also wenig attraktiv in kleine Einzelstücke zerfallen. Hier kann ein Nachteil des FDM-Drucks zum Vorteil reichen: Um den gefürchteten *Warping*-Effekt (ein Abheben unterer, abgekühlter Kunststoffschichten durch die thermische Kontraktion sich zusammenziehender, frisch gedruckter, oberer Filamentschichten) zu vermeiden, wird oft ein sogenanntes *Raft* als Unterlage gedruckt, welches in diesem Fall nicht unbedingt entfernt werden muss.

Im zweiten Doppelstundenteil wurden Wiederholungen und deren Schachtelung (!) eingeführt. Dieses sehr zügige Voranschreiten im geplanten Stoff ist sicherlich ambitioniert, doch sind für Programme in *Beetle Blocks*, wie bereits mehrfach erwähnt, absolut elementar: Ohne sie ist man in der Programmierung sehr eingeschränkt, kaum motivierende Beispiele ließen sich finden. Zwischen Kontrollfluss und entstehendem Muster besteht außerdem eine gewisse Synergie (siehe Abbildung 22). Ein außerdem wichtiger Hinweis zum Einstieg: *Beetle Blocks* besitzt eine Vielzahl, wie auf Seite 17 erläutert, teils redundanter Anweisungen. Es hat sicher daher bewährt, dass die Schülerinnen und Schüler in den Anfangsstunden eine Vorlage laden, welche die Anweisungen (in diesem Fall auf die auf der ersten Seite abgebildeten Befehle im Arbeitsmaterial) beschränkt, um Verwirrung zu vermeiden.

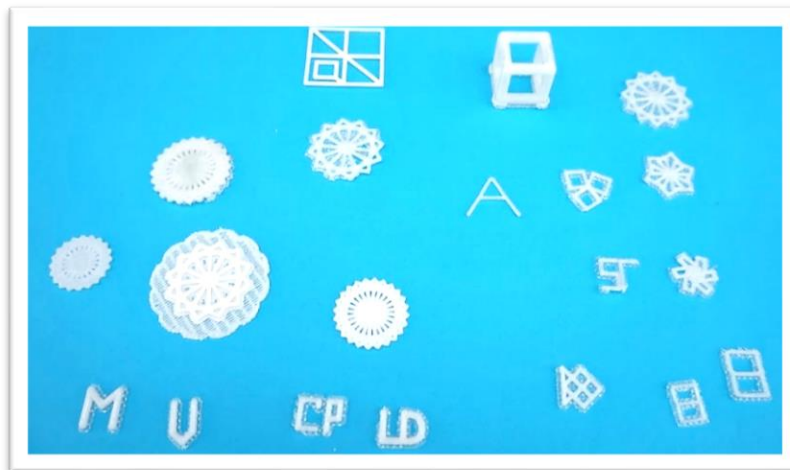


Abbildung 23: Beispiel für Schülerresultate nach der ersten (!) Doppelstunde

Zweite Doppelstunde – Mathematik intuitiv angewandt

Nach der obligatorischen Wiederholung der Schleifen und insbesondere der geschachtelten Wiederholungen zu Beginn der Stunde sollte es um die durch Außenwinkel zu konstruierenden n -Ecke gehen. Letztlich wurde sich gegen die auf Seite 10 vorgestellten *push-pop*-Grundflächen entschieden, da der Rahmen von lediglich sieben Unterrichtsstunden eine fachgerechte Einführung dieser nicht zuließ. Die mit den n -Ecken einhergehende Mathematik, insbesondere die Formel für den Rotationswinkel des Käfers, muss von allen Schülerinnen und Schülern umgesetzt werden können, weshalb auf eine elementar-induktives Schließen dieser gesetzt wurde: Aus konkreten Vielecken und wird die Beziehung von Eckenanzahl und Rotationswinkel hergeleitet (vgl. S. 5, 3 c) im Unterrichtsmaterial). Die vorangehenden Einstiegsaufgaben zu dem gleichseitigen Dreieck und dem Nebenwinkel (siehe S. 5 3 a) und b) im Unterrichtsmaterial) lässt schnell das absolute Basiswissen der 7. Jahrgangsstufe Mathematik wiederholen. Die Schülerinnen und Schüler lernen außerdem im Verlauf der Stunde verstehen, dass ein Kreis durch ein Vieleck mit zunehmender Eckenanzahl¹⁷ immer besser angenähert werden kann.

Variablen werden in dem zweiten Teil dieser Doppelstunde nur in ihrer Funktion als Konstanten vorgestellt, weil die Vielzahl an Konzepten nicht zu einer Überforderung der Lernenden führen soll. Wie im vorangehenden Unterkapitel 3.2 erläutert, lässt sich nur schwerlich auf diese im Rahmen eines allgemeinbildenden Informatikunterrichts verzichten.

¹⁷ Bezüglich der Eckenanzahl sollte auch beim Kreis nicht übertrieben werden, da sonst beim Übereinanderstapeln der n -Ecke schnell ein Performanceproblem verursacht wird.

Projektblätter nur als Anregung zum Endprojekt

Zum didaktischen Vorgehen bezüglich der *if*-Anweisungen wurde bereits auf Seite 20f. berichtet, weshalb an dieser Stelle keine weiteren Erläuterungen dazu erfolgen. Die drei Projektblätter ohne Nummerierung wurden den Schülerinnen und Schülern zur Auswahl und Anregung zur Verfügung gestellt, damit sie sehen konnten, was möglich ist. Es stand ihnen jedoch frei, die Beispiele zu modifizieren, oder etwas ganz anderes zu programmieren – was gerne angenommen wurde! Die endgültigen Projekte werden im folgenden Kapitel ausgeführt.

Kritisch muss die Projektidee „Vasen, Türme und Becher“ mit einfachem Schwierigkeitsgrad betrachtet werden, da sie von den Schülerinnen und Schülern als wenig attraktiv empfunden wurde. Für leistungsschwächere Schülerinnen und Schüler sollte daher eher auf zweidimensionale Objekte zurückgegriffen werden, da sich hier trotz geringerer Schwierigkeit schnell ansprechende Ergebnisse erzielen lassen.

Exkurs: Defizite und Bugs von *Beetle Blocks* im praktischen Unterrichtseinsatz

Wie auf Seite 8 ausgeführt, stellt *Beetle Blocks* gegenwärtig zweifelsfrei das Tool der Wahl für den Unterrichtseinsatz von 3D-Druck eingebettet in die Einführung in die Algorithmik dar – und dennoch gibt es einige Kritikpunkte, die zumindest zur Diskussion gestellt werden müssen: Die bereits erwähnten beiden Koordinatensysteme stiften schnell Verwirrung, da nicht offensichtlich ist, welcher Befehl, auf welches Koordinatensystem bezugnimmt (vgl. S. 17f). Wünschenswert wäre sicherlich auch die Möglichkeit, den *Beetle* als Art „Radierer“, d. h. als Löscher bestehender Strukturen, einzusetzen; im durchgeführten Unterricht offenbarten sich allerdings schnell „Kleinigkeiten“, die den Ablauf der Unterrichtsstunde erheblich verzögern können: Das lokale Speichern der Projekte am Rechner muss über einen Klick auf das Dateisymbol gefolgt von einem weiteren Klick auf „Download project as...“ sowie der obligatorischen Bestätigung des Browser-Download-Dialogs erfolgen – und stellt mitunter eine Hürde für die junge Zielgruppe dar. Im Unterricht war es nötig, den elementaren Speichervorgang mehrfach im Plenum vorzuführen. Verstärkt wird diese Problematik durch die Notwendigkeit auch bereits kleinere Projekte mit „tiefen“ Objekten, also schon ab der zweiten Doppelstunde, häufiger zu sichern, da hier *Beetle Blocks* auf den Schulrechnern zum Absturz neigt. Weiterhin sind die aus didaktischer Sicht ohnehin problematischen Rotationswinkel (vgl. S. 15) nicht wohldefiniert implementiert:



Abbildung 24: Reporter liefert falsches Ergebnis.

Die Abfrage in obiger Abbildung liefert ein „wahr“, wenn als Vergleichswinkel minus 180° angegeben ist – geometrisch betrachtet ist dies aber völlig äquivalent.

Obwohl für das Team von *Beetle Blocks* durchaus noch Handlungsbedarf besteht, so ist die Vielzahl der angemerkten Defizite jedoch behebbar, d. h. oftmals nicht konzeptioneller Natur. Aus diesem Grund stehen Sven Jatzlau und ich zum jetzigen Zeitpunkt in Dialog mit dem Entwicklungsteam, allen voran Bernat Romagosa und Jens Mönig, um die Software noch weiter zu verbessern. Ein Zwischenergebnis der Beobachtungen findet sich im Anhang, Teil C.

3.4 Fazit zur Unterrichtsplanung aus Lehrersicht

Unter realen Unterrichtsbedingungen müssen immer Ziele abgewogen und gerade unter Zeitmangel Schwerpunkte gesetzt werden. Eine der hier eingegangenen, schwerwiegenden Konzessionen ist der nur verbal im Unterricht vermittelte Algorithmus-Begriff. In Bezug auf diesen sind auch die vorgestellten Beispiele und Projekte – gerade auch in Zusammenhang mit den selten verwendeten *if* – kritisch zu sehen: Sicher sind die resultierenden Programme endlich, eindeutig, ausführbar – jedoch weniger allgemein, d. h. sie lösen nicht zwangsläufig, wie dies aber für einen Algorithmus charakteristisch wäre, eine Klasse an Problemen. Ein Umstand, der im Anfangsunterricht nicht unbedingt ungewöhnlich ist, z. B. fehlt einem Animationsfilm in *Scratch* dieser Aspekt ebenso. Am ehesten wird ein Algorithmus (im Gegensatz zum bloßen „Design“) durch das parametrisierte *n*-Eck und das Ausfüllen dessen als Bodenfläche in der vorgestellten Unterrichtsreihe umgesetzt. Da aber *Beetle Blocks* als *Snap!*-Derivat auch Benutzereingaben ohne Weiteres ermöglicht, lässt sich der Algorithmusaspekt problemlos betonen.

Insgesamt – dies sei der Evaluation des kommenden Kapitels vorweggenommen – ist der Verlauf der Unterrichtsreihe äußerst zufriedenstellend: Die räumliche Orientierung im Koordinatensystem bereitete keinerlei Probleme, was einerseits der Beschäftigung mit Konstruktion und Symmetrien im zweidimensionalen Koordinatensystem in der 7. Jahrgangsstufe, andererseits dem Verzicht auf räumliche Rotationen als Weg ins Dreidimensionale zu verdanken ist. Der Schnelleinstieg mit *Beetle Blocks* gelang problemlos – auch für 3D-Modelle spielt die visuelle Programmierung ihre Vorteile aus. Einzig auf den *reset* Befehl am Programmstart musste mehrfach hingewiesen werden. Etwas umständlich ist der Download und Import eigener Projekte, wenn diese lokal gespeichert werden – ein aber letztlich zu verschmerzender Umstand. Bemerkenswert war die enorme Motivation der Schülerinnen und Schüler unterstützt durch den sofortige Resultate liefernden 3D-Drucker im Computerraum selbst, sich zeigend im großen Interesse an der neuen 3D-Druck-Technik, auch im lebensweltlichen Kontext; nach dem Nahrungsmitteldrucker oder orthopädischen Prothesen wurde mehrmals gefragt. Gar nicht genug hervorgehoben werden kann der gestalterisch-kreative Aspekt des 3D-Drucks: Schnell sprachen die Schülerinnen und Schüler von „meinem Modell“ und tauschten in (Zwischen-)Pausen selbst gefundene Lösungen für besonders gelungene Strukturen aus. Dass kein einziger Schüler, keine einzige Schülerin als Endprojekt auf eine bloße vorgegebene Projektidee zurückgriff, sondern mindestens eigene Modifikationen einbrachte, bestätigt dies.

4. Evaluation der Unterrichtsreihe

Neben den bereits erwähnten aus Lehrersicht gewonnen Erkenntnissen wurden die eigentlichen Adressaten des Unterrichts, die Schülerinnen und Schüler, über den Unterricht und über das Thema „3D-Druck“ an sich befragt. Schwerpunkte der Evaluation liegen auf der Selbsteinschätzung der Lernenden von Motivation, Kreativität und Verständnis. Insbesondere wurde für die durchgeführte Unterrichtseinheit untersucht, ob die aus Kapitel 2 bekannten, nicht trivialen mathematischen Basiskonzepte als ausreichend eingestuft werden.

4.1 Vorgehensweise bei der Datenerhebung

Aus den auf Seite 19 ausgeführten Gründen waren die Voraussetzungen für einen Vorher-/Nachher- oder einen Klassenvergleich leider nicht gegeben, weshalb im Anschluss an die Unterrichtseinheit mittels eines Fragebogens, einer Analyse der Endprojekte sowie Interviews ausgewählter Schülerinnen und Schüler verschiedener Leistungsniveaus evaluiert wurde.

Der eingesetzte Fragebogen (vgl. Anhang, Teil B) orientiert sich in Teilen stark an den von Romeike (2008) verwendeten, seinerseits auf Fragebögen der Informatikdidaktik der Universitäten Siegen und Paderborn basierenden Bogen. Der von mir eingesetzte Fragebogen ist speziell an die jüngere Zielgruppe – 7. statt 11. Jahrgangsstufe – mittels einer bunten Farbkodierung und Emoticons angepasst worden, da bereits im Unterricht offensichtlich wurde, dass die umfangreichen Arbeitsblätter – die absichtlich eher den Charakter eines Kapitels eines Lehrbuches besitzen – teils nicht genau gelesen wurden. Einhergehend mit dieser Gestaltung könnte allerdings auch eine nicht intendierte Beeinflussung der Probanden erfolgt sein – konkrete oder auch allgemeine Hinweise darauf ließen sich aber nicht finden, sei jedoch zur Reflektionszwecken erwähnt. Der Fragebogen gliedert sich in die in folgender Abbildung dargestellten Teilbereiche.

Fragebereich Erfassungsart	Seite im Bogen
eigenen Person Geschlecht, Alter, Vorleistungen, Unterrichtsversäumnisse	1
Motivation und Kreativität grob skalierte Single-Choice-Fragen nach Spaß oder Mitarbeit im Unterricht anschließende Pro-und-Contra-Liste mit freier Antwortmöglichkeit	1
Verständnis und Schwierigkeitsgrad (Selbsteinschätzung) grob skalierte Single-Choice-Fragen gefolgt von einer Tabelle schwerer und leichter Unterrichtsinhalte	2
Unterrichtsmaterial und 3D-Druck grob skalierte Single-Choice-Fragen	2
Unterrichtsmethodik feiner skalierte Single-Choice-Fragen zum Schwierigkeitsgrad mathematischer und informatischer Inhalte sowie zur Selbsttätigkeit	3
Möglichkeit freier Beurteilung Option einer frei formulierten Antwort anhand gegebener Leitfragen	4

Abbildung 25: Übersicht über Struktur des Fragebogens

Insgesamt wurden 50 Fragebögen¹⁸ ausgegeben, von denen nur zwei – beide in der als „pubertär“ beschriebenen Klasse – nicht sachgemäß ausgefüllt wurden. Davon abgesehen war die Beteiligung der Schülerinnen und Schüler qualitativ vorbildlich, d. h. unter anderem die Felder zur Äußerung ihrer freien Meinung wurden auf fast allen ausgewerteten Fragebogen angenommen.

Die getätigten Interviews mit insgesamt zwölf ausgewählten Schülerinnen und Schülern jeder Leistungsfähigkeit umfassten neben dem Fragenkatalog des Bogens außerdem ein gemeinsames Durchgehen des programmierten Endprojekts – meist gelang es, dies (nach Notenschluss) nicht wie eine Prüfungssituation wirken zu lassen. Durch die Interviews fiel die Interpretation der Fragebögen leichter und half (anhand der Schülererklärung des Endprojekts) bei einer relativ genauen Einschätzung der tatsächlich erworbenen Kompetenzen.

4.2 Auswertungsergebnisse

Motivation und Kreativität

Meines Erachtens nach stellt die der Unterrichtsansatz mit *Beetle Block* einen überaus motivierenden Zugang zum 3D-Druck dar – eine Einschätzung, die sich absolut mit der der Schülerinnen und Schüler deckt, wie nachfolgende Abbildung¹⁹ verdeutlicht:

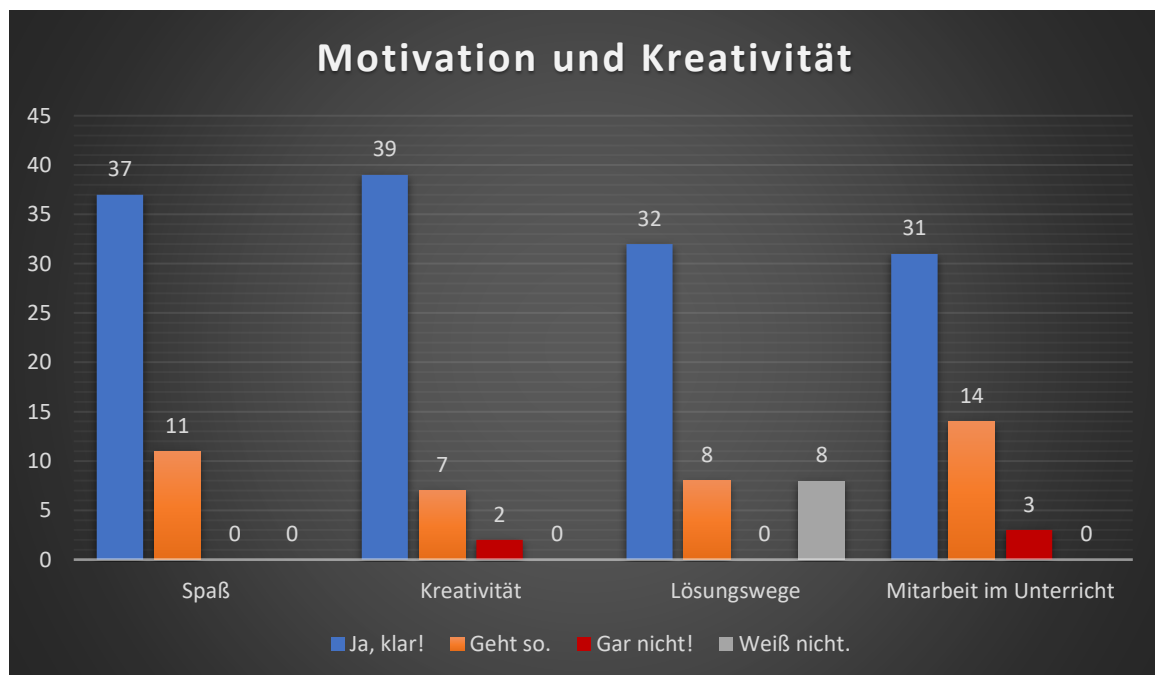


Abbildung 26: Multiple-Choice-Ergebnisse zur Motivation und Kreativität

¹⁸ Die Gesamtheit der 56 Schülerinnen und Schüler konnte zu Schuljahresende wegen erläutelter Umstände (AK-Treffen, usw.) auch im Regelunterricht leider nicht erreicht werden.

¹⁹ In Abbildung 26 fehlen absichtlich die Ergebnisse der Frage nach dem Beeindrucken des sozialen Umfelds durch 3D-Objekte, da sich in den Interviews zeigte, dass die Schülerinnen und Schüler dieses Alters die Frage nicht sinnvoll einordnen konnten; trotz der fragwürdigen Aussagekraft hier der Vollständigkeit halber die Resultate: 20 Stimmen „Ja, klar!“, 7 „Geht so.“, 2 „Gar nicht“ – sowie 17 der Antworten in die Kategorie „Weiß nicht“, die durch das Missverständnis erklärt werden können.

Neben den hohen Zustimmungsraten ist außerdem zu bemerken, dass in der leistungsschwächeren, „schwierigeren“ Klasse keine signifikante Abweichung zu der leistungsstärkeren in obigen Punkten gab.²⁰ Untermauert wird die indizierte hohe Motivation im Unterricht durch die stichpunktartigen Kurzwörter in der Tabellenspalte „gut gefallen“ – knapp 65% der gegebenen Antworten indizieren eine Selbsteinschätzung von Kreativität (vgl. folgende Abb.).

Beispielphrase der Schüler/innen	Anzahl (nicht disjunkt)
„Ich kann ganz viel selber machen! “	11
„... dass wir viel ausprobieren durften.“	11
„Man konnte seiner Kreativität freien Lauf lassen“	11
„ Projekte , die man selber machen konnte.“	8

Insgesamt **31 der 48** Fragebögen (disjunkt betrachtet) enthalten mind. ein Merkmal dieser Phrasenkategorien!

Abbildung 27: Auswertung der Freitextantworten hinsichtlich Kreativität

In diesem Zusammenhang wenig verwunderlich ist, dass mehr als 77% der Jugendlichen ein selbständiges Arbeiten an einem eigenen Projekt, d. h. ohne Arbeitsauftrag, bevorzugen. Verschiedene Lösungswege in *Beetle Blocks* werden weiterhin erkannt, obwohl diese nicht im Unterricht thematisiert wurden.

Als weitere motivierende Aspekte wird häufiger die Abwechslung zum üblichen Unterricht und die guten Erklärungen mit je sechs Stimmen sowie die ausgedruckten Modelle mit acht Stimmen genannt. Kritisch angemerkt wird außerdem von fünf bzw. vier Probanden, dass 3D-Druck kompliziert sei oder *Beetle Blocks* häufig zum Abstürzen neige; letztere Einschätzung wird von lehrender Seite uneingeschränkt geteilt.

Verständnis und Schwierigkeitsgrad

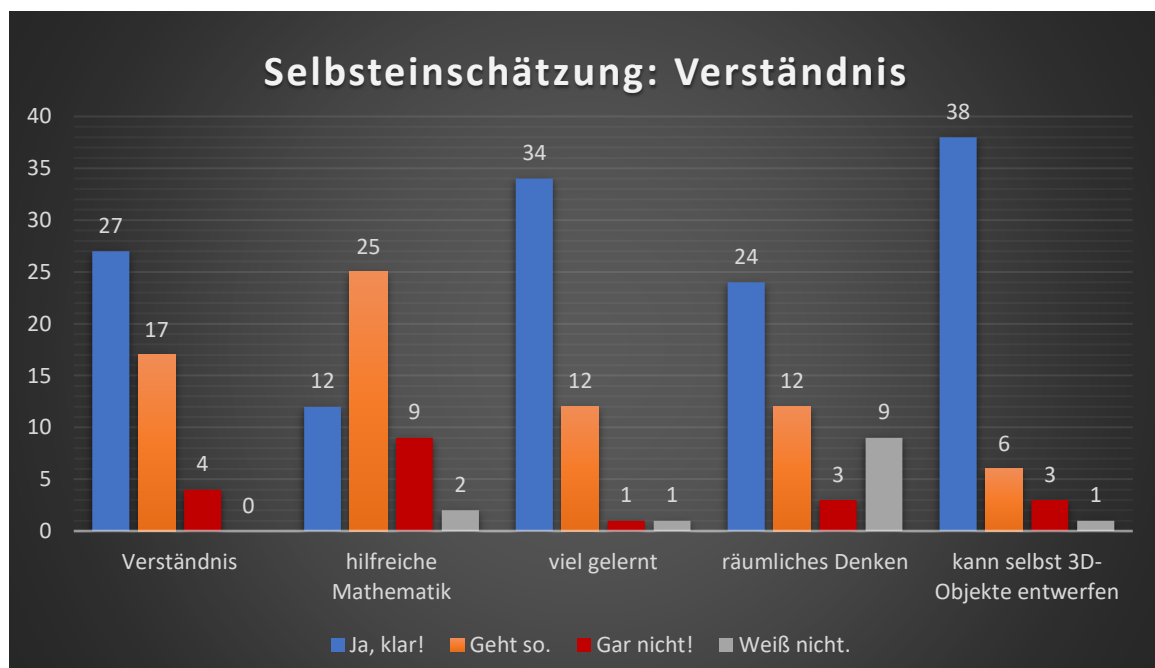


Abbildung 28: Multiple-Choice-Ergebnisse zur Einschätzung des Verständnisses

²⁰ Ein fernerliegender Unterschied ist, dass die Jugendlichen der „schwierigen“ Klasse das häufige Sperren des Bildschirms durch die Lehrpersonen mit einer Anzahl von fünf Stimmen kritisierten; dies ist jedoch der in dieser Klasse vorhandenen Notwendigkeit des konsequenteren „sich-Gehör-Verschaffens“ geschuldet.

Eine weniger eindeutige Datenlage liegt bezüglich der Selbsteinschätzung des Verständnisses vor, auch wenn die Zustimmung zu „Ich kann jetzt selbständig 3D-Objekte entwerfen!“ erfreulicherweise sehr hoch ist. Die Einschätzung zum Verständnis ist am ehesten auf die als schwer empfundenen bedingten Anweisungen mit sieben Stimmen im Freitext sowie besonders auf den auch als Lehrkraft spürbaren Zeitdruck in der Unterrichtseinheit zurückzuführen: Mehr als ein Drittel (!) der Schülerinnen und Schüler, 19 an der Zahl, vermerkten (meist auf der letzten Seite des Fragebogens), dass die Zeit im Unterricht bzw. zum vertiefenden Üben zu knapp ist. Für eine Wiederholung der Unterrichtsreihe sollte dementsprechend dringend auf einen umfangreicheren zeitlichen Rahmen geachtet werden, da konstatiert werden kann, dass sieben Unterrichtsstunden für das Stoffpensum nicht ausreichend sind. Wie viel Zeit aber tatsächlich zum effektiven Festigen des Erlernten bzw. zum spielerischen Lernen benötigt wird, sollte Gegenstand weiterer Untersuchungen sein: Geben doch mehr als 71% der Schülerinnen und Schüler an, sie hätten (trotz des straffen Zeitplans) viel gelernt (siehe vorangehende Abbildung).

Im Schwierigkeitsgrad als unproblematisch wahrgenommen werden hingegen Winkel, Wiederholungen (je fünf Stimmen) sowie zweidimensionale Muster (mit 11 Stimmen).

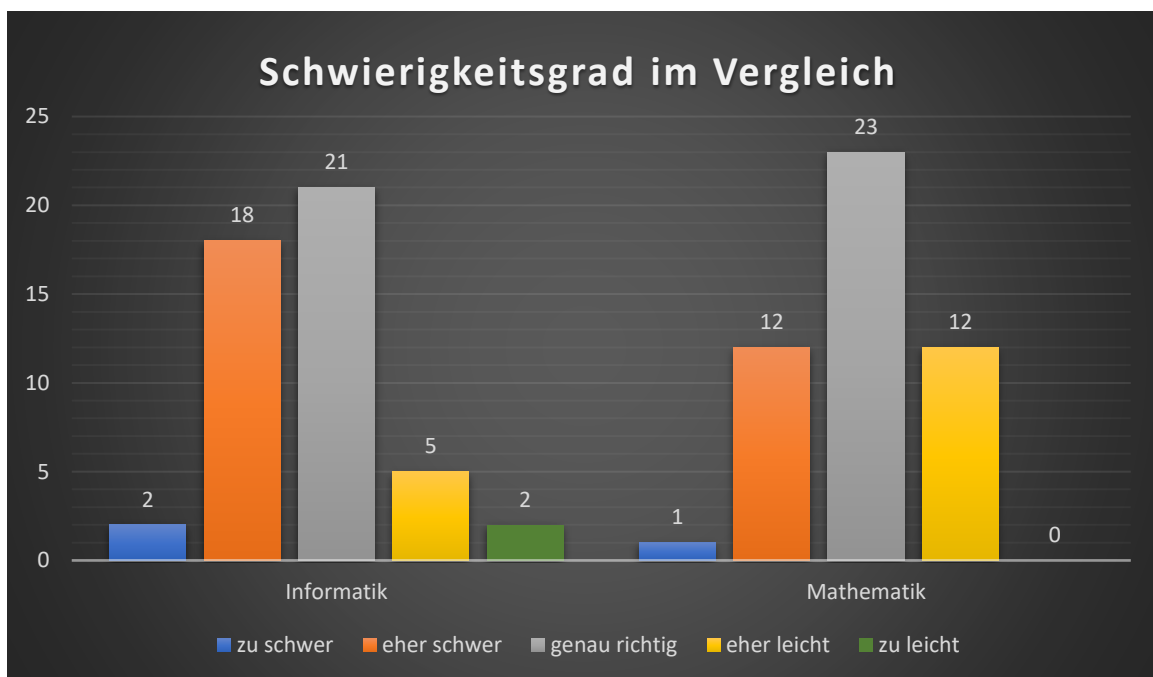


Abbildung 29: Multiple-Choice-Ergebnisse zur Einschätzung des Schwierigkeitsgrades informatischer vs. mathematischer Kompetenzen in der Unterrichtseinheit

Werden mathematische Kenntnisse in Abbildung 28 eher als mittelmäßig hilfreich eingestuft, so zeigt sich dessen ungeachtet in obigem Diagramm, dass die mathematischen Basiskenntnisse auch aus Schülersicht für den durchgeführten Unterricht (in beinahe diskret normalverteilter Weise) genügen. Der eigentliche „Knackpunkt“ wird auf informatischer Seite wahrgenommen – so wie dies im Informatikunterricht sein sollte. Insofern kann das Integrieren mathematischer Grundlagen in den 3D-Druck-Unterricht als gelungen bezeichnet werden.

3D-Druck als Unterrichtsthema

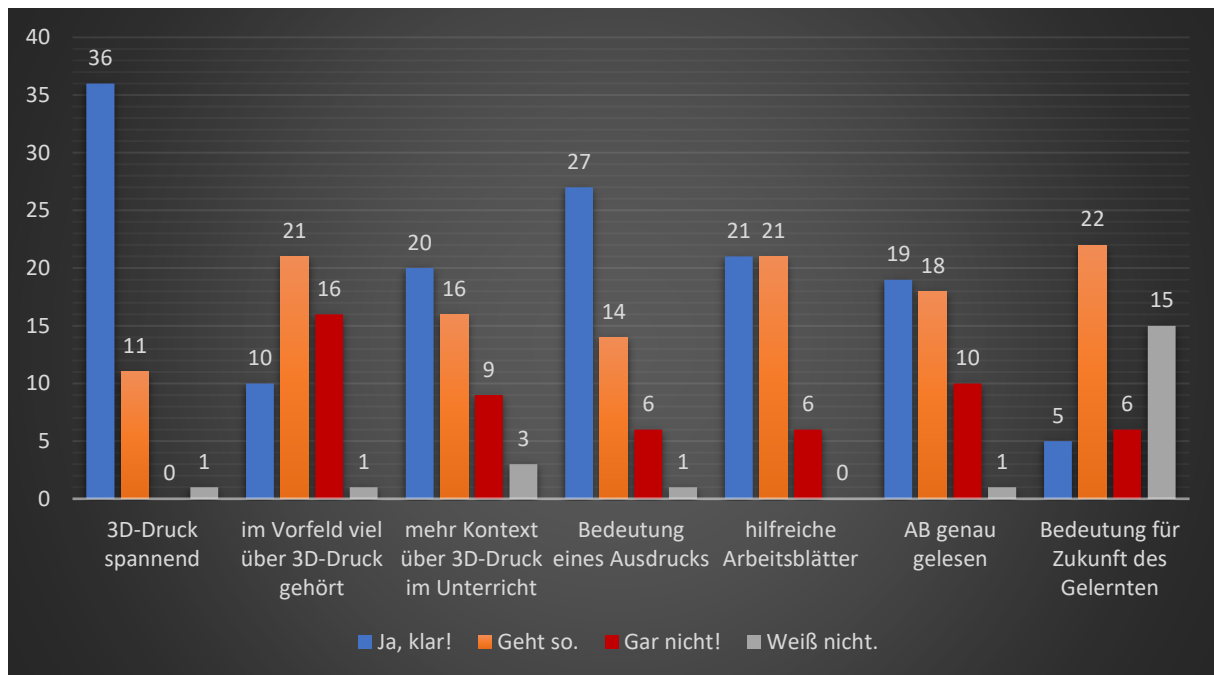


Abbildung 30: Multiple-Choice-Ergebnisse zum 3D-Druck als Unterrichtsthema und zum Unterrichtsmaterial

Die (implizite) Hypothese aller Unterrichtsansätze, 3D-Druck sei ein für Jugendliche spannendes, motivierendes Unterrichtsthema, kann mit 75%-iger Zustimmung (zumindest in der hier vorliegenden Altersstufe) als wahr angenommen werden. Auf lebensweltliche Kontexte wurde im Unterricht nicht eingegangen. Auch scheinen diese, obwohl sie medial in verschiedenen Reportagen inszeniert werden, für einen ansprechenden Unterricht nicht notwendig zu sein: Nur etwas mehr als 40% der Jugendlichen hätten sie im Unterricht gerne behandelt. Insgesamt „waren die Aufgaben und das kreative Tun offenbar motivierend genug, dass die Schüler lernten und dabei sogar Spaß hatten“, schreibt Romeike (2008) (S. 93) über seine durchgeführte, kreative Unterrichtseinheit zur Einführung in die Algorithmik in einer 11. Jahrgangsstufe – ein Ergebnis das für den 3D-Druck in der 7. Jahrgangsstufe ebenso vollumfänglich gilt.

Wegen der vollständigen Einbettung der Algorithmik in den Kontext des 3D-Drucks – und der mit dem geringen Alter der Schülerinnen und Schüler einhergehenden, weniger ausgeprägten Reflektionsfähigkeit – befinden von ihnen nur 10% (!) das Gelernte als von Bedeutung für die Zukunft.²¹ Etwas überraschend ist möglicherweise, dass nur ca. 56% der Jugendlichen angeben, ein Ausdruck ihres 3D-Modells sei ihnen wirklich wichtig. Betrachtet man diese Aussage aber mit der expliziten Nennung von realen, „ausgedruckten“ 3D-Modellen als Motivationsfaktor (vgl. S. 31) und dem Hintergrund, dass in der durchgeführten Unterrichtsreihe mehrmals „gedruckt“ wurde (Initialen in der ersten Doppelstunde, Rotationsmuster in der zweiten Doppelstunde sowie später das Endprojekt), nivelliert sich dies bis zu einem gewissen Grad.²²

²¹ Ein herausragend leistungsstarker Schüler erkannte in einem Interview jedoch, 3D-Druck sei nur das „Drumherum“ für die „Programmierung“.

²² Vermutlich hätte ein einerseits höheres Stundenpensum für das Endprojekt (für einen engeren persönlichen Bezug dazu), andererseits eine explizite Thematisierung des Druckvorgangs an sich (Stützstrukturen, *raft*, uvm.) das Bedürfnis nach einem Ausdruck steigern lassen.

Evaluation des Unterrichtsmaterials und der -methodik

Abbildung 30 kann entnommen werden, dass die Arbeitsblätter als solche sehr kontrovers beurteilt werden. Diese Ambivalenz wird umso kritischer, beachtet man die 16 Fragebögen, in denen das Unterrichtsmaterial in handschriftlichen Kommentaren als besonders hilfreich gelobt (!) wird. Am ehesten erklärt werden kann dieses Auseinanderklaffen der Beurteilungen durch die angestrebte schülerzentrierte Unterrichtsmethodik, die ein eigenständiges Durchlesen der umfangreichen Arbeitsblätter erfordert – während im (leider noch immer) üblichen, rezeptiven Unterricht kein so umfangreiches Material zur selbstständigen Bearbeitung zum Einsatz kommt.²³ Ob die Arbeitsblätter überhaupt (in ausreichender Form) gelesen werden, stellte sich mir bereits im Unterricht aufgrund der Nachfragen offensichtlich auf den Arbeitsblättern vermerkter Information von zumeist weniger leistungsstarker Schülerinnen und Schülern. Von den 27 Jugendlichen, die das Material eher kritisch beurteilten, geben 10 an, das Material gar nicht genau gelesen zu haben – während dies von den Schülerinnen und Schülern, welche das Material für gut befinden, kein einziger vermerkte.



Abbildung 31: Präferenz eigenständiger Arbeit vs. Lehrererklärung

Ähnlich zeigt sich, dass von den Probanden, die den Stoff als gut verstanden befinden, kein einziger mehr Lehrerklärungen wünscht – während dies beinahe jeder zweite der restlichen Gruppe wünscht. Insgesamt kann das Verhältnis von eigener Arbeit zu Erklärungen durch die Lehrkraft als angemessen bezeichnet werden, dem ein heterogenes Bild an Schülerpräferenzen und -Lerntypen zugrunde liegt (vgl. obiges Kreisdiagramm). Der allgemeingültige Hinweis für Lehrkräfte, das Arbeitsmaterial im Anhang für eine eventuelle Wiederholung der Unterrichtsreihe an die Gegebenheiten der eigenen Klasse und des eigenen Unterrichtsstils anzupassen bzw. neu zu arrangieren, ist hier besonders zu berücksichtigen.

²³ In den Schülerinterviews wird diese Erklärung stützend primär der Umfang des Materials als Grund für die teils negativen Bewertungen dessen angegeben; jedoch sollte sich keine Illusion gemacht werden, dass die Jugendlichen mir persönlich gegenüber in diesem Punkt völlig unbeeinflusst antworten könnten.

4.3 Exemplarische Vorstellung eines Schülerprojektes

Abschließend wird hier ein Endprojekt des Schülers Adam²⁴ vorgestellt, welches allgemeiner für eine Gruppe von Projekten und in seiner Leistung für einen Großteil der Klasse stehen kann und den Erwartungshorizont insgesamt voll erfüllt. Das folgende Zitat (aus einer E-Mail) des „Natur und Technik“-Lehrers, der die Schülerinnen und Schüler ein ganzes Jahr begleitete, liefert einen Anhaltspunkt, wie sich die Aktivität der Jugendlichen bei der Programmierung mit *Beetle Blocks* veränderte – weiterhin sei auf die Kapitel 4.2 verwiesen, da sich Schüler- und Lehrersicht bezüglich der Motivation zu einem konsistenten Gesamtbild ergänzen:

„Adam (als Beispiel für weitere) war zu Beginn des Schuljahres mit Freude und kindlicher Neugier bei der Physik; leider hat er sich nach und nach zurückgezogen, je zusammenhängender der Stoff wurde. Deswegen bin ich jetzt wirklich froh, dass er bei Beetle Blocks wieder mit neuer, frischer Begeisterung losgelegt hat.“

Da konnte er sich nach einem einfachen Einstieg gerade die Strukturen herausuchen, die seinem Abstraktionsgrad entsprechen. Wie manch andere hat er – ohne Wiederholungsanweisung und mit hohem Aufwand – noch nach dem Gong und zu Hause seine Objekte weiterprogrammiert und uns mit großem Eifer die fertigen Dateien zum Ausdrucken geschickt. Mit dieser Unterrichtssequenz ist seine unbekümmerte Fröhlichkeit in unserem Fach wieder aufgeblüht.“

Über die gesamte Unterrichtsreihe hinweg entwickelte Adam (als treibende Kraft zusammen mit seinem Banknachbarn) ein Modell einer Shoppingmall, vom Schüler kurz „SAPPL-Haus“ genannt. Die anfangs nicht verwendeten Wiederholungen hat Adam im Laufe der Arbeit an seinem Projekt noch an vielen der geeigneten Stellen ergänzt.²⁵ Sein Projekt bot im Unterricht ein motivierendes Beispiel für ihn und seine Klassenkameraden/innen, Wiederholungen zu verwenden, weil der Code sonst schnell sehr umfangreich wird.



Abbildung 32: Screenshot von Adams Abschlussprojekt

²⁴ Der Name wurde aus Datenschutzgründen geändert und auch in nachfolgendem Zitat angepasst.

²⁵ Natürlich hätten sich aus informatischer Sicht mit vermehrtem Variableneinsatz noch mehr Codezeilen „einsparen“ lassen, doch dies kann wegen der nur sehr knappen Einführung dieser nicht spontan von Schülerseite aus erwartet werden.

Aus informatischer Sicht werden Wiederholungen, zumeist nicht geschachtelt, zum Erstellen der Wände des Hauses eingesetzt, die Standfläche und das Flachdach wurde mittels des *if*-Ansatzes zum Auslegen von n-Ecken mustergültig gelöst. Adam setzt außerdem viele nicht im Unterricht vorgestellte, selbst in seiner Freizeit ausprobierte Befehle ein, wie am auffälligsten der Einsatz der 2D-Schrift. Die im Lehrerzitat angesprochene „unbekümmerte Fröhlichkeit“ zeigt sich allein schon in der Namensgebung von Adams Projekt, in die für den Jugendlichen bedeutsame Umstände seiner Gedankenwelt einfließen: „SAPPL“ sei ein Kofferwort bestehend aus den Herstellernamen seines Smartphones und des seines Banknachbarn, Samsung und Apple, wie er mir in einem Interview erklärte.

Viele weitere kreative Ideen

Die Schülerinnen und Schüler gestalteten eine Vielzahl von verschiedenen Objekten, kleine architektonische Modelle standen dabei hoch im Kurs: Beispielsweise gestaltete eine Gruppe von drei Schülerinnen den Burj Khalifa nach, welcher im Hintergrund des Titelbildes dieser Arbeit zu sehen ist. Eine herausragende, aber sicher in keiner Weise erwartbare Leistung, war weiterhin die mittels *push* und *pop* konstruierte Vase aus sternförmigen Querschnittsflächen einer vermutlich hochbegabten Schülerin, die in der Mitte des Coverphotos abgebildet ist.



Abbildung 33: Das „Burj Khalifa“-Modell einer Schülerinnengruppe vor dem Foto seines Vorbilds

Schlussbemerkungen

Zusammenfassung

In der hier vorliegenden Arbeit wurden erste Grundlagen allgemeiner Natur zur Programmierung in *Beetle Blocks* (vor allem in Kapitel 2) identifiziert sowie die praktische Machbarkeit einer Unterrichtseinheit zur Algorithmeinführung insbesondere hinsichtlich schulmathematischen Vorwissens gezeigt. Mit dem Unterrichtsmaterial und den dargelegten Erfahrungen in der Evaluation ist ab sofort ein Anhaltspunkt für Wiederholungen dieser Unterrichtsreihe gegeben.

Insgesamt ist die Arbeit als erster Einstieg in Welt des 3D-Drucks im Informatikunterricht zu sehen, viele weitere, vielversprechende Anwendungsmöglichkeiten, wie Fraktale und Rekursion, sind noch zu erforschen.

Warum aber überhaupt 3D-Druck?

Auch wenn bereits gut erprobte, in mancherlei Hinsicht sogar besser funktionierende Unterrichtskonzepte, z. B. hinsichtlich bedingter Anweisungen, existieren – wie dies beim kreativen Einsatz von *Scratch* der Fall ist – so bietet 3D-Druck dennoch eine besondere Verknüpfung von zur Lebenswelt durch das Generieren von realen Objekten. Weniger sind damit Kontexte, wie Medizin oder Technik – die für die Jugendlichen nur bedingt interessant scheinen (vgl. Abbildung 30 auf S. 33) – gemeint, sondern mehr ist die der Realwelt entspringende Motivation gemeint. Wirklich verdeutlicht wurde mir dies erst durch folgende Anekdote: Der Schüler Nadir²⁶ schrieb mir kurz vor Ende der Sommerferien eine Mail, sein Airhockey-Puck sei kaputtgegangen – er habe einen neuen programmiert und hängte die XML-Datei an. Der „Natur und Technik“-Lehrer der Klasse reflektierte mir gegenüber in einer Mail:

„Mich hat in all den Jahren zwar noch kein Schüler der folgenden Jahrgangsstufen mehr nach Robot Karol gefragt, aber Nadir mit seinem Puck ist dagegen ein eindeutiges Beispiel: Er hat eine lebensweltliche Notwendigkeit identifiziert, seine erworbenen Fähigkeiten nicht nur angewendet, sondern dabei auch gleich wiederholt und gefestigt, und damit ein funktionales Ergebnis und einen passgenauen, persönlichen Erfolg erzielt; aus eigenem Impuls!“

Weitergedacht betrifft dies nicht „nur“ funktionale Gegenstände im engeren Sinne, sondern lässt sich auch problemlos auf selbst gestaltete Kunstobjekte erweitern: Wie wäre es im kommenden Jahr statt mit einem im Kunstunterricht in Kalligraphie *abgeschriebenen* Gedicht zum Muttertag mit einem im Informatikunterricht *selbst gestalteten* Herzdekorelement aus dem 3D-Drucker?

²⁶ Name geändert.

Abbildungsverzeichnis

Abbildung 1: „Elementarbausteine“ von Beetle Blocks schematisch visualisiert	8
Abbildung 2: Erzeugen eines Dreiecks durch Rotation des Beetle um den Außenwinkel	9
Abbildung 3: Parametrisiertes Legen eines Vielecks mittels Stackspeicher	10
Abbildung 4: Codeauszug aus dem Example-Projekt Cup	11
Abbildung 5: Turm gestapelter Vierecke ohne Achse durch Umkreismittelpunkt	13
Abbildung 6: Turm mit Lotachse durch Umkreismittelpunkt der quadratischen Grundfläche	13
Abbildung 7: Quellcode des schiefen Turmes (links) sowie seines symmetrischen Pendants (rechts)	14
Abbildung 8: Schneckengehäuse nach Romagosa (2016) mit übersetzten Blöcken.....	15
Abbildung 9: Die zwei Koordinatensysteme in Beetle Blocks, gut erkennbar anhand der jeweiligen blauen z-Achse	15
Abbildung 10: Fehlvorstellung resultierend aus der Existenz zweier Bezugssysteme	16
Abbildung 11: Korrekte Implementierung der Wendel mit größerem Codeumfang.....	17
Abbildung 12: Exemplarischer Vergleich von redundanten Befehlen	17
Abbildung 13: Screenshot von beetleblocks.com im Juli 2017	18
Abbildung 14: Screenshot und Codeauszugdes Projekts 2d random walk within boundary by Examples	20
Abbildung 15: Screenshot und Codeauszug des Projekts Anna Hillier 83 Coin by anna.....	21
Abbildung 16: Screenshot und Codeauszug des Projekts 2d staggered grid by Examples	21
Abbildung 17: Screenshot und Codeschnipsel vom Unterrichtsbeispiel Swirl.....	22
Abbildung 18: undefinierbare extrudierte Formen (links unten) bei negativen Durchmesser	23
Abbildung 19: Motivation einer bedingten Anweisung zum Ausfüllen einer fünfeckigen Grundfläche	23
Abbildung 20: Screenshot des Simple Cup's mit Quellcode.....	24
Abbildung 21: Kurzgefasste Stundenübersicht	25
Abbildung 22: Zusammenhang geschachtelte Wiederholung und entstehende Figur (aus dem Unterrichtsmaterial S. 4)	25
Abbildung 23: Beispiel für Schülerresultate nach der ersten (!) Doppelstunde	26
Abbildung 24: Reporter liefert falsches Ergebnis.....	27
Abbildung 25: Übersicht über Struktur des Fragebogens	29
Abbildung 26: Multiple-Choice-Ergebnisse zur Motivation und Kreativität	30
Abbildung 27: Auswertung der Freitextantworten hinsichtlich Kreativität	31
Abbildung 28: Multiple-Choice-Ergebnisse zur Einschätzung des Verständnisses	31
Abbildung 29: Multiple-Choice-Ergebnisse zur Einschätzung des Schwierigkeitsgrades informatischer vs. mathematischer Kompetenzen in der Unterrichtseinheit.....	32
Abbildung 30: Multiple-Choice-Ergebnisse zum 3D-Druck als Unterrichtsthema und zum Unterrichtsmaterial	33
Abbildung 31: Präferenz eigenständiger Arbeit vs. Lehrererklärung.....	34
Abbildung 32: Screenshot von Adams Abschlussprojekt	35
Abbildung 33: Das „Burj Khalifa“-Modell einer Schülerinnengruppe vor dem Foto seines Vorbilds....	36

Literaturverzeichnis

- Gerber, T. (2017, 1. März). *Umfrage: 3D-Drucker für Privathaushalte noch zu teuer*. Zugriff am 18.09.2017. Verfügbar unter <https://www.heise.de/newsticker/meldung/Umfrage-3D-Drucker-fuer-Privathaushalte-noch-zu-teuer-3639908.html>
- Greb, T. (2013, 22. Juli). *3D-Drucker in der Schule*. Zugriff am 18.09.2017. Verfügbar unter <http://www.digital-lernen.de/nachrichten/technik/einzelansicht/artikel/3d-drucker-in-der-schule.html>
- Hausman, K. K., Horne, R. & Muhr, J. (2014). *3D-Druck für Dummies. [drucken Sie sich die Welt, wie sie Ihnen gefällt ; auf einen Blick: die Typen von 3D-Druckern und ihre Anwendungen kennenlernen ; 3D-Modelle erstellen und drucken ; neue Geschäftsfelder entdecken ; einen sich selbst drucken- den 3D-Drucker bauen]* (... für Dummies, 1. Aufl.). Weinheim: Wiley-VCH.
- ISB. (2004). *Lehrplan für Natur und Technik Jahrgangsstufe 7*. Zugriff am 21.09.2017.
- Kamolz, L. (2016). Der Einsatz von 3D-Techniken in der Plastischen Chirurgie. *Handchirurgie, Mikrochirurgie, plastische Chirurgie : Organ der Deutschsprachigen Arbeitsgemeinschaft für Handchirurgie : Organ der Deutschsprachigen Arbeitsgemeinschaft für Mikrochirurgie der Peripheren Nerven und Gefäße : Organ der V...*, 48 (2), 85–86. <https://doi.org/10.1055/s-0042-104585>
- König, P. (2015, 17. Dezember). *FDM-3D-Drucker im Test: Up Mini*, Heise Medien. Zugriff am 21.09.2017. Verfügbar unter <https://www.heise.de/make/artikel/3D-Drucker-Test-Up-Mini-3046629.html>
- Krisch, O., Kastl, P. & Romeike, R. (2017). 3D-Druck als Motivation für die Einführung in die Programmierung in der Realschule. In *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt. Lecture Notes in Informatics (LNI)* (S. 15–24). Bonn.
- Moser, A. (2015). *Personal Manufacturing und Urheberrecht - "3D Druck" im privaten Umfeld* (Recht, Technik, Wirtschaft, Bd. 109). Zugl.: Erlangen-Nürnberg, Univ., Diss., 2014. Köln: Heymanns.
- Nitz, S. (2015). *3D-Druck. Der praktische Einstieg ; [von den Grundlagen bis zum fertigen Objekt ; Druckverfahren und Filamente kennen, mehrfarbig drucken ; fertige Druckvorlagen nutzen, mit 3D-Scans und SketchUp eigene erstellen, Objekte veredeln ; kein Vorwissen erforderlich]* (Galileo Computing, 1. Aufl.). Bonn: Galileo Press.
- ProSieben. (2017, 25. Januar). *Essen aus dem 3D-Drucker - das Kochen der Zukunft gibt es schon heute*. Galileo. Zugriff am 18.09.2017. Verfügbar unter <https://www.youtube.com/watch?v=bU9YECzDKk4>
- Rattat, C. (2016). *3D-Druck für Anspruchsvolle. Mit dem Ultimaker perfekte Werkstücke erstellen* (1. Aufl.). s.l.: dpunkt.
- Romagosa, B. (2016). *Beetle Blocks - Inspirations*. Zugriff am 20.09.2017. Verfügbar unter <http://forum.beetleblocks.com/t/my-slides-and-brochures/31>
- Romagosa, B., Rosenbaum, E. & Koschitz, D. (2016). *From the Turtle to the Beetle. The Beetle Blocks programming environment*. Zugriff am 19.09.2017. Verfügbar unter <http://goo.gl/QKpu8H>
- Romeike, R. (2008). *Kreativität im Informatikunterricht*. Dissertation, University of Potsdam. Verfügbar unter <http://d-nb.info/993856977>
- Rosenbaum, E. & Koschitz, D., Romagosa, B. & Mönig, J. (Mitarbeiter). *The Beetle Blocks Primer*. Zugriff am 20.09.2017. Verfügbar unter <http://beetleblocks.com/static/bb-primer.pdf>
- Schroeders, N. von (2017, Januar). *Didaktik der Analysis*. Vorlesungsfolien.
- Sommer, W., Schlenker, A. & Lange-Schönbeck, C.-D. (2016). *Faszination 3D-Druck. Alles zum Drucken, Scannen, Modellieren*. Burgthann: Markt+Technik.
- WDR. (2017, 22. Juni). *3D-Drucker – alles ist machbar*. Planetwissen. Zugriff am 18.09.2017. Verfügbar unter <http://www.planet-wissen.de/video-d-drucker--alles-ist-machbar-100.html>

Anhang

Teil A	Arbeitsblätter der Unterrichtseinheit
Teil B	Fragebogen zur Evaluation
Teil C	Beobachtungen zu Beetle Blocks

I. Kennenlernen von Beetle Blocks

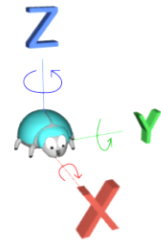
Was ist Beetle Blocks überhaupt?

Beetle Blocks ist eine blockbasierte Programmierumgebung, mit deren Hilfe sich räumliche Gegenstände erzeugen und später mithilfe eines 3D-Druckers „ausdrucken“ lassen. Beetle Blocks muss nicht installiert werden, sondern läuft direkt in gängigen Browsern: <http://beetleblocks.com/>

Wie programmiert man in Beetle Blocks?

Der Käfer (engl.: *beetle*) kann von dir gesteuert werden! Für die Bewegung im Raum ist es wichtig, sich die Rotationsachsen aus der Sicht des Käfers vorzustellen!

Zu Beginn werden wir nur mit wenigen Anweisungen (auch: „Befehle“ genannt) arbeiten:



Kategorie Steuerung

	Diese beiden Blöcke stehen stets <u>nur</u> am Beginn deines Programmes! Die Anweisung reset setzt dabei die Welt zurück.
--	---

Kategorie Bewegung

	Käfer bewegt sich um einen Schritt nach vorne.
	Käfer dreht sich um die ausgewählte Achse um 90° nach links.

Kategorie Shapes (engl. für Formen)

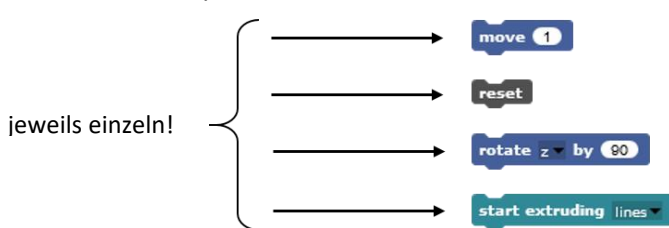
	Käfer beginnt längliche Formen hinter sich auszustoßen. Wichtig: Hier „lines“ statt „curves“ auswählen!
	Käfer hört damit auf.

Wichtige Begriffe rund ums Programmieren

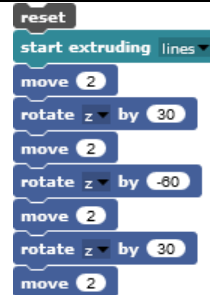
Ein einzelner Block wird **Anweisung** genannt.

Eine ganze Folge von Blöcken heißt **Sequenz**.

Beispiel:



Beispiel:

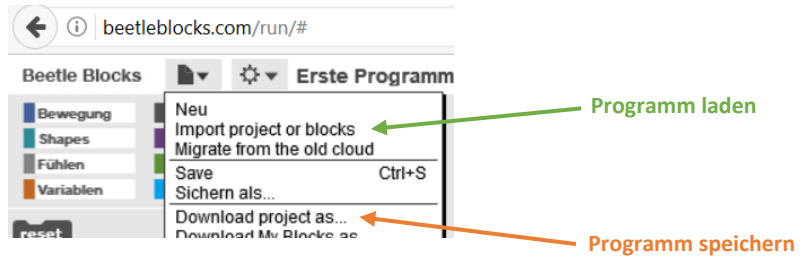


Name: Michael Müller

Datum: 29. Okt. 2017

Arbeitsaufträge

1. Lade dir unsere Vorlage „Erstes Programm“ in deine Beetle-Blocks-Programmierungsumgebung:



2. Probiere die Anweisungen der Kategorie **Bewegung** aus:

- Ändere den Zahlenwert bei **move** bzw. den Winkel und die Achse bei **rotate!**
- Was setzt **reset** zurück?

Tipp: Schau die Informationen unterhalb des 3D-Fensters an!

Die Anweisung „reset“ setzt sowohl die Position als auch die Rotation des Käfers

auf den Anfangswert (von jeweils 0) zurück.

3. Schreibe ein Programm, in dem der Beetle...

- ... eine lange „Wurst“ legt.
- ... eine Initiale deines Vor- oder Nachnamens erzeugt. Speichere deinen Code (wie in obigen Screenshot gezeigt) ab!

Tipp: Einfacher ist es „eckige“ Buchstaben wie bei den Segment-Anzeigen von Elektrogeräten zu bilden:

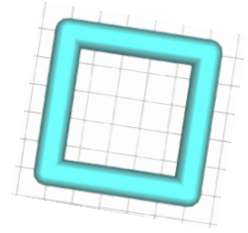


4. *-Aufgabe für Experten:

- Sieh dir den Code Seite 1 rechts unten an. Was könnte er bezwecken?
- Programmiere das Beispiel nach und führe es aus. Deckt sich deine Vermutung aus Teilaufgabe a) mit dem Ergebnis?
- Schreibe das Programm so weiter, dass ein interessantes Muster entsteht!

Name: Michael MüllerDatum: 29. Okt. 2017

II. Wiederholungen (auch: Schleifen)



Wozu braucht man überhaupt Wiederholungen?

Gemeinsam haben wir festgestellt, dass in einem Programm, in dem der Beetle ein Quadrat legen soll, manche Anweisungen wiederholen. Um den Code übersichtlicher und kürzer zu gestalten, gibt es unter der Kategorie **Steuerung** eine so genannte „Wiederholung mit fester Anzahl“:

Quadrat ohne Wiederholungs-Anweisung

```

reset
start extruding lines
move 5
rotate z by 90
move 5
rotate z by 90
move 5
rotate z by 90
move 5
rotate z by 90
stop extruding
  
```

Quadrat mit Wiederholung-Anweisung

```

reset
start extruding lines
wiederhole 4 mal
  move 5
  rotate z by 90
stop extruding
  
```

Anzahl der Wiederholungen

Nur die „umrahmten“ Anweisungen werden wiederholt!

Statt „Wiederholung“ wird oft auch von einer Schleife (engl. *loop*) gesprochen.

Arbeitsaufträge

1. Du hast ein Programm wie oben links geschrieben.

- Programmiere jetzt das rechte nach. Beide Programme sollten das gleiche Quadrat erzeugen.
- Ändere nun die Größe des Quadrates im
 - Quadrat ohne Wiederholung und
 - Quadrat mit Wiederholung.

Notiere, worin die Unterschiede zwischen i) und ii) bestehen:

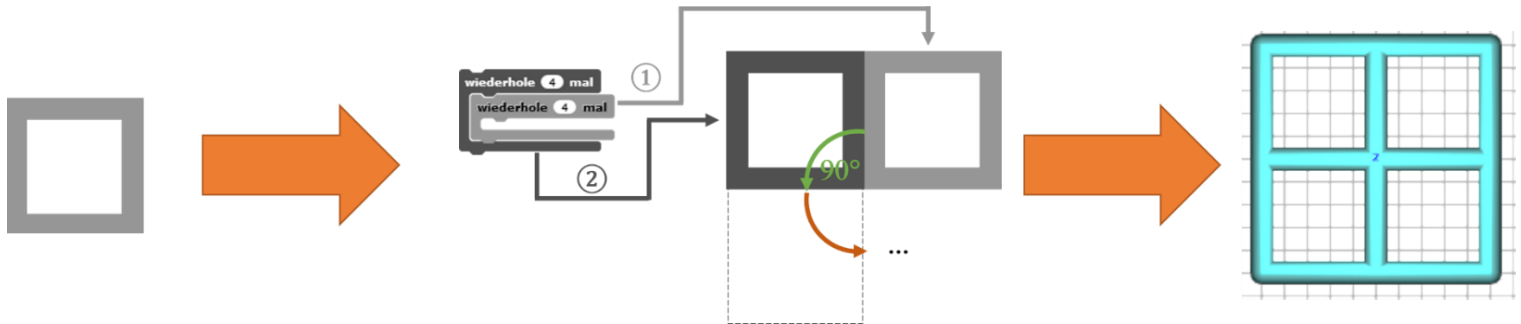
Im Quadrat mit Wiederholung genügt es, einen einzigen Wert zu ändern – ohne Wiederholung müssen vier Werte geändert werden, weshalb leicht Fehler passieren können!

Name: Michael Müller

Datum: 29. Okt. 2017

2. Wir erweitern unser Quadrat schrittweise zu einer spannenden Figur.

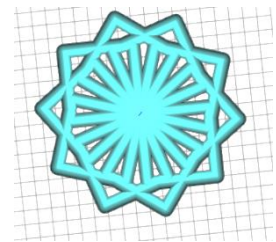
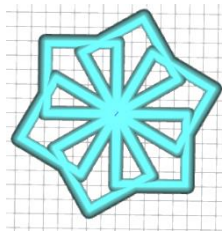
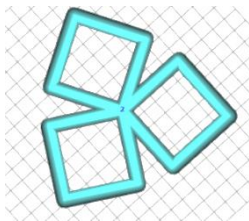
- a) Jetzt legen wir vier Quadrate. Dafür brauchen wir eine weitere Wiederholung. Du „schachtelst“ beide, d. h. du fügst die innere in die äußere Wiederholung ein:



Programmiere die Abbildung nach!

Tipp: Beachte, dass der Beetle zwischen den einzelnen Quadraten um je 90° rotieren muss, denn sonst liegen alle Quadrate auf der selben Stelle.

- b) Ändere dein Programm so ab, dass es eines der folgenden Muster erzeugt!



- c) Ergeben alle Winkel, um die du die Quadrate rotieren lassen kannst, eine schöne Figur?

Am „schönsten“, am symmetrischsten sind im Allgemeinen Figuren, bei denen die

Rotationswinkel zwischen den einzelnen Quadraten gleich groß sind.

Tipp: Es gilt:

$$\text{Rotationswinkel des Beetle} = \frac{360^\circ}{\text{Anzahl der Quadrate}}$$

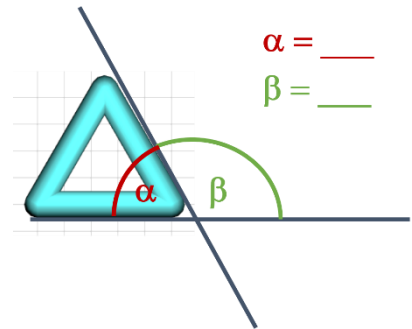
- d) *-Aufgabe für Experten: Man kann natürlich nicht nur mit Quadraten ein Drehmuster erzeugen, sondern auch zum Beispiel auch mit Rechtecken. Probiere aus!

Name: Michael Müller

Datum: 29. Okt. 2017

3. Erstellen von regelmäßigen Vielecken

- a) Betrachte die nebenstehende Abbildung und bestimme die Größe der fehlenden Winkel (ohne zu messen)!
 Tipp: Es handelt sich um ein gleichseitiges Dreieck.



- b) Programmiere den Käfer so, dass er ein solches Dreieck legt!
 Tipp: Der Beetle muss sich um den Winkel β drehen!

- c) Betrachtet man den **Rotationswinkel β** , um den sich der Käfer bei verschiedenen Vielecken drehen muss, so erhält man folgende Tabelle:

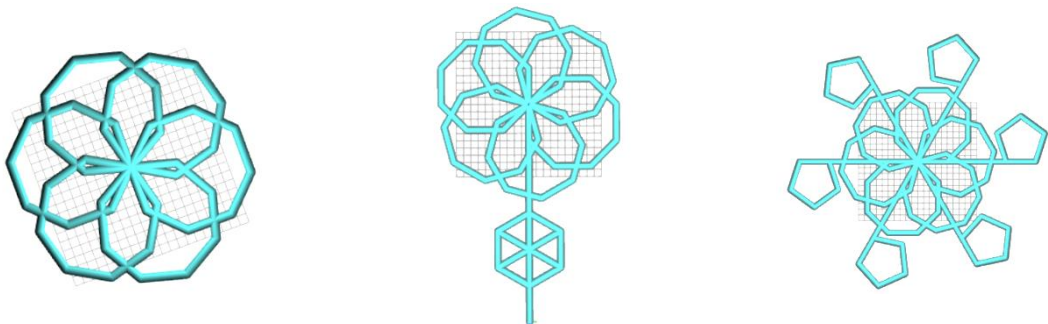
Vieleck	Rotationswinkel	Anzahl der Ecken	
Dreieck	120°	3	$360^\circ:3 = 120^\circ$ ✓
Viereck	90°	4	$360^\circ:4 = 90^\circ$ ✓
Fünfeck	72°	5	$360^\circ:5 = 72^\circ$ ✓
Sechseck	60°	6	$360^\circ:6 = 60^\circ$ ✓

Vervollständige folgende Formel:

$$\text{Rotationswinkel des Beetle} = 360^\circ / \text{Anzahl der Ecken}$$

- d) Prüfe nach, ob deine Formel aus der c) korrekt ist, indem du in der vierten Spalte der obigen Tabelle deine Formel für das Drei-, Vier-, Fünf- und Sechseck überprüfst.
- e) Mithilfe von Teilaufgabe c) können wir nun beliebige n-Ecke erzeugen. Erstelle ein 36-Eck. Unser Beetle kann also auch **Kreise** legen!
- f) Erstelle selbst ein Muster mit einem Vieleck durch Rotation (ähnlich zu Seite 4). Du kannst auch verschiedene Vielecke kombinieren. Lass deiner Kreativität freien Lauf!

Als Anregung kann folgendes dienen:



- g) ***-Aufgabe für Mathe-Fans:** Sei n die Anzahl der Ecken. Leite aus der dir bekannten Innenwinkelformel für regelmäßige n -Ecke ($\text{Innenwinkel} = (n - 2) \cdot 180^\circ$) und Winkelbetrachtungen (Scheitelwinkel, etc.) die Formel aus c) her.

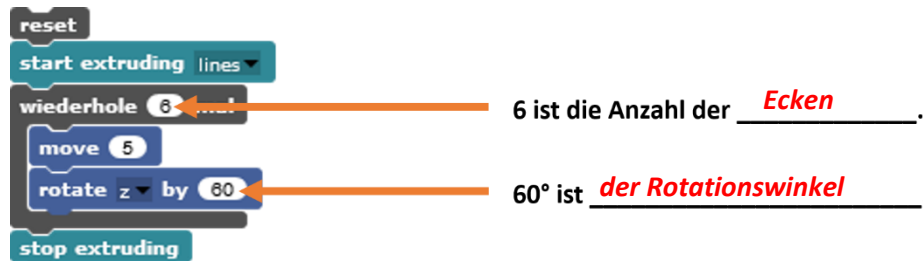
Name: Michael Müller

Datum: 29. Okt. 2017

III. Variablen (Teil 1) und die dritte Dimension

Beispielcode für n-Ecke

In der Aufgabe zu regelmäßigen n-Ecken auf Seite 5 hast du ein Programm geschrieben, das ähnlich ist wie das untenstehende:



Arbeitsaufträge (Teil 1)

1. Vervollständige die Beschriftung obiger Abbildung!

Das Programm legt also ein 6-Eck.

2. Schreibe obiges Programm zu einem Fünfeck um. Was fällt dir dabei auf?

Es müssen zwei Werte angepasst werden: Sowohl die Eckenanzahl als auch der Winkel.

Wofür sind Variablen gut?

Variablen sind eine Art Platzhalter bzw. Speicher für eine bestimmte Größe, die wir später einfach ändern können werden. In unserem Beispiel kann dies z. B. die Anzahl der Ecken sein.

Kurzanleitung für Variablen (Teil 1)



1. Deklaration einer neuen Variable

Eine neue Variable muss erst angelegt und benannt werden.



2. Initialisieren einer Variable

Vor der ersten Verwendung benötigen Variablen einen Startwert.



3. Verwenden einer Variable

Da wir die Variable **Ecken** mit der Zahl 6 initialisieren, kann die Variable die 6 in der Schleife ersetzen.

Name: Michael Müller

Datum: 29. Okt. 2017

Arbeitsaufträge (Teil 2)

3. Füge wie in der Kurzanleitung beschrieben Variablen in dein n-Eck ein!
4. Wir wollen nun auch den Rotationswinkel (siehe Abb. auf Seite 6) nur noch mithilfe der Variablen berechnen lassen.

Tipps:

- a) Betrachte die Formel für den Rotationswinkel auf Seite 5!
 - b) Programmiere diese Formel mithilfe der Variable **Ecken** und den neuen Anweisungen unter der Kategorie **Operatoren**!
 - c) Du kannst dein fertiges Programm testen, indem du beim Initialisieren der Variable **Ecken** verschiedene Werte ausprobierst.
5. *-Aufgabe für Experten: Es ist möglich, statt aus der Anzahl der Ecken den Rotationswinkel berechnen zu lassen (wie bisher), auch aus dem Rotationswinkel die Anzahl der Ecken berechnen zu lassen. Stelle zunächst eine entsprechende Formel auf und arbeite mit an Stelle einer Variable **Ecken** mit einer Variable **Winkel**.

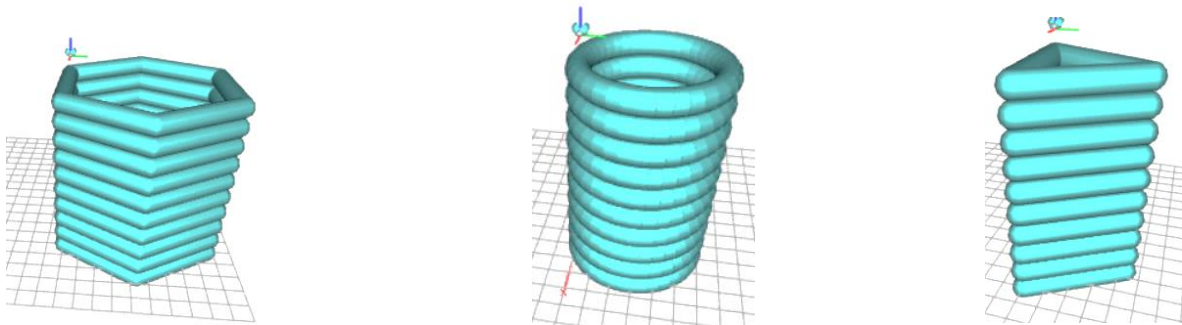
Ein Schritt in die dritte Dimension

Bisher haben wir nur mit ebenen Mustern gearbeitet und wollen diese nun in Richtung der z-Achse erweitern. Durch den folgenden Befehl aus der Kategorie **Bewegung** wird dies nun einfach möglich:

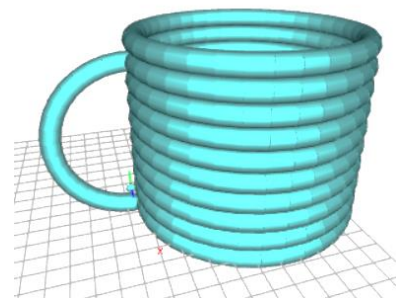
`change absolute z by 1` | Käfer bewegt sich um eine Einheit in z-Richtung nach oben.

Arbeitsaufträge (Teil 3)

6. Lass den Käfer mehrere n-Ecke oder Kreise übereinanderlegen. Die Ergebnisse könnten wie folgt aussehen:



7. *-Aufgabe für Experten: Füge zu dem „Türmchen“ aus der vorherigen Aufgabe einen Henkel – ähnlich dem einer Tasse – hinzu.



IV. Variablen (Teil 2) und bedingte Anweisungen

Unsere bisherigen Kunstobjekte (Türme und Tassen) von Seite 6 haben bisher einen entscheidenden Nachteil: Sie haben keinen Boden! Dies wollen wir im Folgenden angehen:

Arbeitsaufträge

1. Dein Programm aus der vorherigen Aufgabe könnte wie das nebenstehende aussehen.



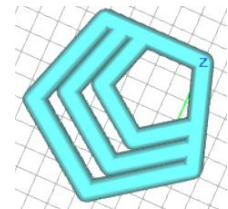
a) Führe eine zusätzliche Variable **Groesse** ein, mit der du die Größe des n-Ecks verändern kannst.

Tipp: Sieh dir erneut die Kurzanleitung zu Variablen auf Seite 6 an!

b) Nun sollen zunächst drei n-Ecke ineinandergelegt werden. Dazu musst Du die Variable **Groesse** ändern:



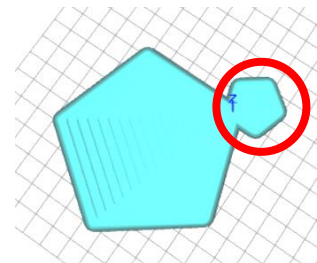
Variable **Groesse** wird um 1 verringert.



c) Die entstehende Fläche soll nun vollständig ausgefüllt werden. Dazu ist es nötig, sehr viele n-Ecke ineinander zu legen und die **Groesse** weniger stark zu verändern.

Hinweis: Das Ergebnis ist wahrscheinlich nicht optimal! (Siehe Aufgabe 2.)

2. Leider kann es schnell passieren, dass die Variable **Groesse** durch die Änderungen in 1c) schnell negativ wird. Dies erkennt man daran, dass dann mehr als das ursprüngliche Rechteck ausgefüllt wird (siehe rechts).



Um dies zu verhindern, müssen wir dafür sorgen, dass der Beetle nur unter der Bedingung läuft, dass die Variable **Groesse** positiv bleibt.

a) Formuliere die obige Bedingung mittels der Variable **Groesse** und Anweisungen aus der Kategorie **Operatoren**.

b) Damit unser Käfer auch nur bei erfüllter Bedingung weiterarbeitet, gibt es unter der Kategorie **Steuerung** die sogenannte „bedingte Anweisung“:

Bedingung (hier leer!)

Sie muss **wahr** sein, damit innere Blöcke ausgeführt werden. Sie beinhaltet oft einen **Operator**.

eingeschlossene Anweisungen

Die eingeschlossene Sequenz an Blöcken führt der Beetle nur bei erfüllter (wahrer) Bedingung aus.

Füge diese Sequenz in das Programm ein und ergänze die Bedingung aus a)!

Name: Michael Müller

Datum: 29. Okt. 2017

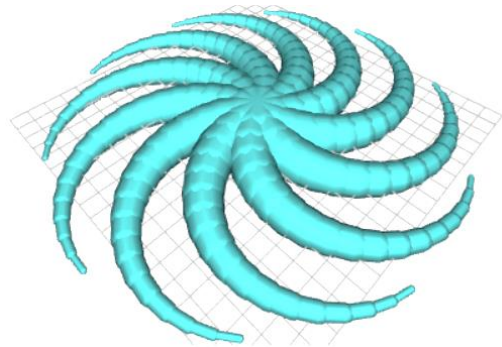
Projektidee: Swirl

(engl. für Strudel)

mittlerer Schwierigkeitsgrad

Das brauchst du dazu:

- geschachtelte Wiederholung
- bedingte Anweisung

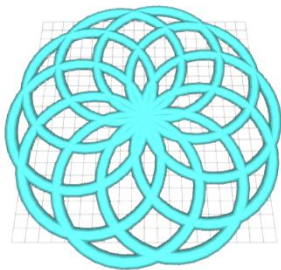
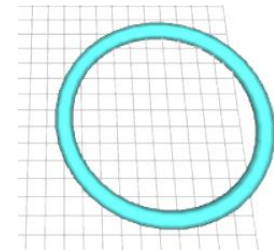


Wichtig vor dem Start: Öffne unbedingt die Vorlage „Extrusion“. Sie enthält unter der Kategorie **My Blocks** Anweisungen rund um den Extrusions-Durchmesser. Verwende diese nur hier und nicht etwa unter Variablen.¹

Schrittweise Anleitung:

1. Beginne damit, einen Kreis zu legen.

Tip: Wenn du nicht mehr weißt, wie das geht, schau Dir Seite 5 nochmals an!



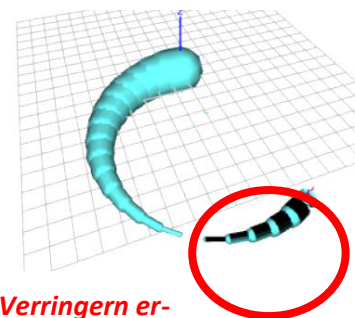
2. Lege viele Kreise übereinander. Verwende dabei eine geschachtelte Wiederholung!

Tip: Erinnerung dich an die Formel auf Seite 5!

3. Verändere nun schrittweise den Extrusions-Durchmesser.

- a) Welches Problem könnte dabei auftreten?

Tip: Ähnliches ist uns schon auf Seite 8 passiert.



Der Extrusionsdurchmesser wird durch das sukzessive Verringern er-

neut negativ, weshalb „seltsame“ Formen entstehen.

- b) Löse das Problem wie auf Seite 8.
-

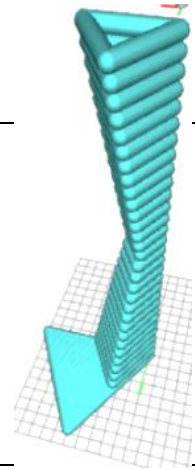
¹ Dies hat mit einer Einschränkung in der aktuellen Version (Stand: Juli 2017) von Beetle Blocks zu tun: Es ist unter der Kategorie Shapes zwar möglich den Durchmesser zu ändern, aber nicht abzu prüfen.

Projektidee: Vasen, Türme und Becher

geringer Schwierigkeitsgrad

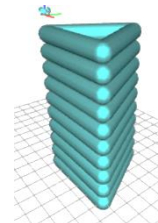
Das brauchst du dazu:

- geschachtelte Wiederholung
- Variablen
- optional: bedingte Anweisung für die Bodenfläche



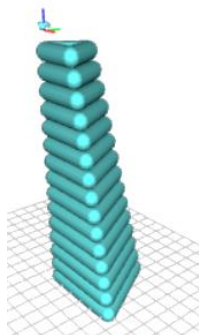
1. Wir beginnen mit einem allgemeinen n-Eck als Grundfläche. Den Code dazu findest du auf **Seite 8** abgedruckt. Programmiere nach!

2. Lege dieses n-Eck mittels einer weiteren Wiederholung übereinander.



```
wiederhole 10 mal
  Code für n-Eck
  change absolute z by 1
```

3. Jetzt sollen die einzelnen Schichten des Turms eine unterschiedliche Größe erhalten. **Tipp:** Lese dir die Kurzanleitung zu Variablen **auf Seite 6 und 8** erneut durch!

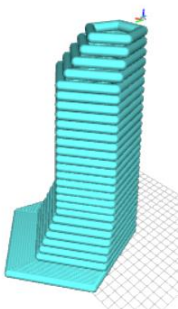


- a) Lege eine neue Variable **Groesse** an!
- b) Initialisiere diese mit einem sinnvollen Wert, wie 5.
- c) Setze für den Zahlwert bei **move** nun die Variable **Groesse** ein: `move Groesse`

```
wiederhole 10 mal
  Code für n-Eck
  change absolute z by 1
  ändere Groesse um 0.3
```

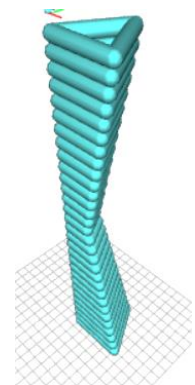
d) Die **Groesse** muss Schicht für Schicht verändert werden.

4. Es können weitere Schichten erzeugt werden, z. B. kann die **Groesse** zunächst ansteigen und dann wieder geringer werden.



- a) Erstelle selbst eine schöne Form!
- b) Falls dir deine Grundfläche nicht gefällt, kannst du diese noch immer einfach über die Anzahl der Ecken verändern. Probiere aus!

```
wiederhole 10 mal
  Code für n-Eck
  change absolute z by 1
  ändere Groesse um 0.3
wiederhole 7 mal
  Code für n-Eck
  change absolute z by 1
  ändere Groesse um -0.2
```



5. *-Aufgabe: Erstelle für die Figur noch eine Standfläche; das Vorgehen ist ähnlich zu Seite 8.

Name: Michael Müller

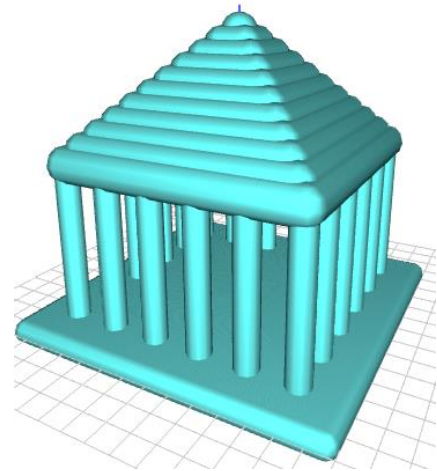
Datum: 29. Okt. 2017

Projektidee: Tempel-Haus

hoher Schwierigkeitsgrad

Das brauchst du dazu:

- geschachtelte Wiederholung
- Variablen
- bedingte Anweisung für die Bodenfläche
- weitere Operatoren

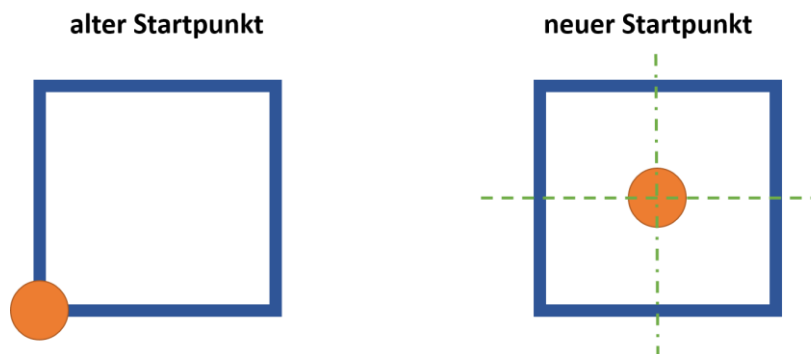


-
1. Der Tempel braucht eine Standfläche! Erstelle sie wie auf Seite 8 beschrieben.
 2. Die Säulen werden durch **change absolute z by** und Extrudieren erzeugt.
Tipps für alle: Verwende auch hier am besten eine Variable **Hoehe**.

Tipps für Experten: Folgendes könnte hilfreich sein:

falls `0 = Counter modulo 2`

3. Das Dach besteht aus mehreren gestapelten Quadraten (auch andere Vierecke sind möglich). Damit diese mittig übereinandergelegt werden, ist es wichtig im Mittelpunkt (des Umkreises) zu beginnen, nicht in der Ecke:



4. Lege nun diese Quadrate übereinander, ...
 - a) ... indem du die Variable **Groesse** in einer Schleife verwendest.
Tipps: Lese dir bei Bedarf die Kurzanleitung zu Variablen **auf Seite 6 und 8** erneut durch!
 - b) Verwende außerdem eine bedingte Anweisung, damit die Variable **Groesse** nicht negativ wird.
5. Tempel besitzen stets eine gewisse Ornamentik. Anregungen könnten sein:
 - a) Oben abgebildeter Tempel besitzt als Dachspitze eine kleine Kugel.
 - b) Als alternative Grundfläche wäre außerdem ein Sechseck möglich.

Fragebogen zum 3D-Druck im Unterricht mit Beetle Blocks

Der vorliegende Fragebogen bezieht sich **nur** auf die Unterrichtsreihe zum 3D-Druck.

Er ist vollkommen **anonym**.

Bitte antworte so **ehrlich** wie möglich, denn nur dadurch können wir uns verbessern.

Vielen Dank für deine Mitarbeit!

1. Fragen zu deiner Person

Ich bin männlich. weiblich. keine Angabe.

Mein Alter beträgt _____ Jahre.

Bisher waren meine Leistungen im NT-Unterricht: 😊 😐 ☹️

So viele Unterrichtsstunden zum 3D-Druck habe ich versäumt: _____




2. Allgemeine Fragen zur Unterrichtsreihe mit dem Beetle

	😊 Ja, klar!	😐 Geht so.	☹️ Gar nicht!	Weiß nicht.
Der Unterricht hat mir Spaß gemacht!	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich konnte selbst kreativ sein.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Meine Freunde oder Familie kann ich mit ausgedruckten oder virtuellen 3D-Modellen beeindrucken.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt verschiedene Lösungswege in Beetle Blocks.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe beim Unterricht mitgemacht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Was hat dir besonders am Unterricht Spaß gemacht, was nicht?




Gut gefallen hat mir 😊:	Nicht so gut war ☹️:

Bitte wenden!

	 Ja, klar!	 Geht so.	 Gar nicht!	Weiß nicht.
Ich habe den Stoff verstanden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Meine Kenntnisse aus der Mathematik haben mir im Unterricht weitergeholfen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich habe viel gelernt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Für 3D-Druck muss ich sehr räumlich denken.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich kann jetzt selbständig 3D-Objekte entwerfen!	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Was ist dir leichtgefallen und wo hast du dich schwergetan?

Das war für mich leicht 😊:	Das war echt schwer ☹️:

	 Ja, klar!	 Geht so.	 Gar nicht!	Weiß nicht.
Das Thema 3D-Druck ist spannend!	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Das Gelernte wird auch in Zukunft für mich wichtig sein.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Arbeitsblätter sind hilfreich.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ich hätte gerne mehr über 3D-Druck in Industrie, Medizin und Online-Communities erfahren.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dass mein Entwurf dann auch ausgedruckt wird, ist mir wichtig.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Die Arbeitsblätter habe ich genau gelesen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Schon vor dem Unterricht hatte ich viel über 3D-Druck gehört.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Konkretere Fragen zur Unterrichtsgestaltung zum 3D-Druck

Der Umgang mit **Sequenz, Wiederholung** und **bedingter Anweisung** („falls“) war für mich:



Der Umgang mit **Rotationen, Winkeln, Formeln** und dem **3D-Koordinatensystem** war für mich:



Mir wären **mehr Erklärungen** durch den Lehrer lieb.

Das Verhältnis von eigener Arbeit und Lehrererklärung war **passend**.

Ich würde lieber mehr **selbst arbeiten**.

z. B. an Tafel oder Beamer vor der ganzen Klasse

z. B. am AB oder am eigenen Projekt

Wenn ich selbst mit **Beetle Blocks** arbeite, dann ...

- ... am liebsten an einem eigenen Projekt ohne Arbeitsauftrag.
- ... am liebsten an einer klaren Aufgabe mit Arbeitsauftrag (z. B. Arbeitsblatt).
- Weiß ich nicht.

Bitte wenden!

4. Freie Beurteilung

Hier kannst du alles loswerden, was dir noch wichtig ist.

Anregungen könnten folgende Aspekte sein:

- Du möchtest zu den vorherigen Fragen noch etwas **ergänzen**.
- Gab es ein **Thema**, das näher vertieft hätte werden sollen?
- Hat dir etwas in der Unterrichtseinheit **gefehlt**?
- Wie sehr haben dir die **Arbeitsblätter** geholfen?
- ...

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Stand: 13. Juli 2017 *mit ergänzenden Fußnoten zum aktuellen Stand (Okt. 2017)*

Vorschläge, Wünsche, Anregungen & Bugs zu Beetle Blocks

Zusammengestellt von Manuel Riel, Rudolf Pausenberger und Sven Jatzlau zum Vorstellen auf der Scratch2017BDX gegenüber den BB-Hauptentwicklern Bernat Romagosa und Eric Rosenbaum.¹

Bei Fragen gerne an manuel.riel@outlook.de wenden.

- **Open Project** klappt oftmals nicht auf Beetle-Webseite
Betrifft Öffnen von vorgefertigten Beispielen; Fehler ist nicht deterministisch reproduzierbar, tritt aber nach einiger Zeit meines Erachtens nach immer auf.
- **Reporter der z-Rotation:** nicht gerundet, nicht wohldefiniert
Neben fehlender Rundung ist problematisch: Unterhalb des 3D-Views wird z. B. 180° angezeigt, intern aber mit -180° gerechnet.
- **Konzept der Koordinatensysteme:** Ortsfestes (allgemeines) vs. bewegtes (individuelles) KoSy
 - Achsendimensionen nicht unabhängig: Dies wird unangenehm dadurch deutlich, dass sich beim Rotieren um die z-Achse auch die Rotationswerte unterhalb des 3D-View für die anderen beiden Achsen ändern. Unklar ist, ob sich dies „besser“ lösen lässt, denn zunächst sind die BB-KoSy durchaus intuitiv
 - Problem „High Ceiling“: Rotationen um mehr als eine Achse kompliziert, da insbesondere nicht kommutativ!
 - Die beiden Koordinatensysteme werden **auch bei den Befehlen** der Kategorie Bewegung nicht unterschieden → zusätzliche Verwirrung; z. B. „change absolute z by“ (individuelles KoSy) vs „set x to“ (globales KoSy)
 - Vermutlich wäre ein (stärkeres) Explizitmachen der beiden KoSy schon hilfreich: Trennen entsprechender Befehle, etc.
- **Extrusion-Diameter**
 - nicht änderbar zwischen start/stop
 - fehlt als Reporter!
 - Negativ als Radiergummi → „negative geometry“
 - Momentan als Bug mit Lines: Wird wieder größer (vgl. Codebeispiel)
- **Performance:** Mathe-Berechnungen & besonders **push/pop**
 - Abstürze direkt beim Programmablauf oder beim Exportieren der STL (vgl. Codebsp.)
 - noch problematischer als extruding „lines“ sind die „curves“ (vgl. unten)
- **Das Erbe von Snap:**
 - Nebenläufigkeits-Aspekt, Ausführung zweier Code-Abschnitte parallel unsinnig?!
 - Mehrwert von broadcast/receive
 - wie oben: KoSy → zwei Befehle für das Gleiche (Change absolute by // set z to)
 - Manche Formen sind vorgegeben, andere nicht...
 - Break-Block?
 - Variablentypen sorgen für Verwirrung, da nur ein Objekt (Beetle!)
 - Skriptvariablen
 - Globale „normale“ Variablen
 - Input für Funktionen ← Usability außerdem problematisch

¹ Illustriert wurden die Fehler durch entsprechende Screenshots und/oder Codebeispiele.

Stand: 13. Juli 2017 *mit ergänzenden Fußnoten zum aktuellen Stand (Okt. 2017)*

- **Sensing/Touching**
 - eigentliches Argument: Kreismittelpunkt & -Radius so besser berechenbar
 - evtl. weitere Anwendungsmöglichkeiten à la Robot Karol (Suchen von Steinen, etc.)

 - **push/pop als toller Ansatz**
 - Stack visualisieren bzw. überhaupt erst sichtbar machen
 - Performance viel schlechter als normales „move“ & „rotate“ (s. o.)

 - **Verbindung von Code & Objekt** → Bezug schaffen;
Kommentare & eigene Blocks nur bedingt geeignet; C-Block (Notiz für Sven)

 - **Negative Geometrie (CNC-Maschine)**
 - Negativer Extrusions-Durchmesser
 - funktioniert momentan auch nicht richtig mit Standardformen beim Ausdrucken (Erkenntnis von Petra)

 - **curves vs lines** → Codebeispiel
Generell unklar, was Curves machen... offenbar komplizierte mathematische Berechnungen?!

 - **Block to Text-Conversion**
Zur informationszentrierten Betrachtung des 3D-Druck-Prozesses im Unterricht
wünschenswert: Von XML zu STL zu GCODE ☺

 - **Bezugspunkt der Kamera²** ändern: (0|0|0) oder neuer Mittelpunkt des darstellenden 3D-Views mittels „zoom to fit“, vgl. Codebeispiel
Wäre gut, wenn auch generell änderbar, nicht nur über „zoom to fit“.

 - **lokales Speichern** auf Rechner
 - Download für SuS bisher relativ kompliziert...
 - gemeinsamer Dialog für XML & STL wünschenswert
 - Öffnen-Dialog für lokal gespeicherte XML könnte zentraler sein.
 - Umbenennen der Projekte momentan nur umständlich über Browser-Speicherung
- Speichern/Laden als nicht-unerhebliches Problem im Unterricht!

² Dies ist mit einem Rechtsklick auf das Vorschaufenster und einem „Ziehen“ der Maus bereits länger möglich; wegen des Mangels an einem Handbuch (oder einem „Wikia“) war uns dies aber im Juli 2017 noch unbekannt.