Blöcke, Blumen, Mikrocontroller und das Internet of Things

Ein Konzept zum kontextorientierten Einsatz von Parametric Design und Physical Computing in Mädchen-Förderworkshops

Nils Pancratz[1], Anatolij Fandrich[1], Christos Chytas[1], Mareike Daeglau[1], Ira Diethelm[1]

Abstract: Die Art und Weise, mit der Informatik über die letzten Jahrzehnte von der Industrie, den Medien und teils sogar (öffentlichen) Bildungseinrichtungen vermarktet wurde, führte in westlichen Kulturen zu einer deutlichen Geschlechterkluft in informatischen Berufen wie auch Studiengängen. Studien zu diesem aus diversen Gründen anzugehenden Missstand lieferten in den letzten Jahren gehäuft Belege dafür, dass sich Mädchen und junge Frauen eher für Informatik begeistern lassen, wenn ihnen diese faszinierende Wissenschaft in Kreativität, Kollaboration und Kommunikation fordernden und fördernden Kontexten präsentiert wird. So führten diese Erkenntnisse zuletzt vermehrt zu Bemühungen, Mädchen über spezielle außerschulische Workshopformate anzusprechen. Um in solchen Formaten Programmieranfängerinnen und -anfängern zugleich grundlegende Programmierkenntnisse (bspw. über Schleifen, Variablen, Verzweigungen und Methodenaufrufe) zu vermitteln und trotzdem schnell motivierende Ergebnisse zu erzielen, wird zuletzt vermehrt auf das Konzept des *Computational Design* zurückgegriffen. Dies ermöglicht über die Idee der *Digital Fabrication* ebenso wie das in außerschulischen Workshops bereits etablierte *Physical Computing*, Informatik im wahrsten Sinne des Wortes begreifbar zu machen. Dieser Praxisbeitrag stellt ein Workshopkonzept vor, in dem über den Kontext des *Internet of Things* die Ideen hinter *Parametric Design* und *Physical Computing* miteinander kombiniert werden – und dabei beide Geschlechter gleichermaßen ansprechen kann und soll! Das Ziel ist, am Ende des Workshops, der bereits mehrmals erfolgreich sowohl in außerschulischen wie auch schulischen Formaten durchgeführt wurde, eine selbstgestaltete Lampe – bspw. in Blumenform – fabriziert zu haben, die in Abhängigkeit der über das Internet bezogenen Wetterdaten in eigens kreierten Lichteffekten strahlt.

Keywords: Gendersensitive Informatikdidaktik; Parametric Design; Physical Computing; Internet of Things; Informatik im Kontext

1 Einleitung

„Informatik für alle“ – bei einem Blick auf aktuelle Statistiken zu Studierendenzahlen wirkt der Titel dieses Tagungsbandes nahezu utopisch und unerreichbar: Schließlich war der ohnehin schon beklagenswert niedrige Frauen-Anteil unter denjenigen, die sich deutschlandweit für eine Neueinschreibung in einen Informatikstudiengang entschieden haben, zuletzt wieder rückläufig und lag zu Beginn des Wintersemesters 2018/2019 bei nur

[1] Carl von Ossietzky Universität Oldenburg, Didaktik der Informatik, Uhlhornsweg 84, 26129 Oldenburg, vorname.nachname@uni-oldenburg.de

22,5%[2]. Die wenigen Frauen, die sich für ein Informatikstudium entscheiden, wählen dabei häufig „Bindestrich-Studiengänge“ wie bspw. Bioinformatik, Medizinische Informatik oder auch Medieninformatik [Kl16], was als Indiz dafür gelten könnte, dass ein stärkerer Kontextbezug (vgl. Abs. 2.2) besonders förderlich für das Interesse und die Begeisterung von Mädchen und jungen Frauen an der Informatik ist. Auch bei den geschlossenen Ausbildungsverträgen liegt der Frauenanteil je nach Ausbildungsberuf aktuell nur zwischen knapp 5% (Fachinformatiker/-in Systemintegration) und 15% (Informatikkaufleute)[3].

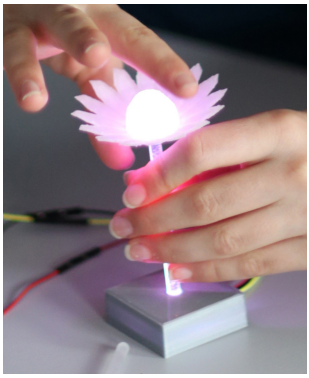

Abb. 1: Die Teilnehmerin eines dieser Workshops beim Zusammenstecken der Komponenten ihrer eigenständig gestalteten internetfähigen Lampe.

Der Nährboden für diese Geschlechterkluft ist zweifelsfrei bereits in der Schulinformatik zu suchen: In den zahlreichen Bundesländern, in denen Informatik ein Schattendasein als Wahlpflichtfach fristet, entscheiden sich erfahrungsgemäß nur wenige Mädchen, es zu belegen; in den Ländern mit verpflichtendem Informatikunterricht (IU) hingegen sind es im Vergleich zu ihren Geschlechtergenossen überproportional viele junge Frauen, die sich im Laufe ihrer Schulzeit gegen eine weitere Belegung entscheiden. So ruft diese offensichtliche Geschlechterkluft seit Jahren zahlreiche Studien zu möglichen Ursachen (vgl. Abs. 2.1) auf die Programme (inter-)nationaler Informatikdidaktik-Tagungen. Der vorherrschende Fachkräfte-Mangel im IT-Sektor ist hierzu sicherlich genauso als Ansporn zu begreifen wie der Anspruch, *allen* Schülerinnen und Schülern (SuS) die Vielfalt informatischer Bildung zu bieten. Da nicht nur im schulischen Informatikunterricht, sondern auch in entsprechenden außerschulischen Lernangeboten vorwiegend Jungen und junge Männer anzutreffen sind, wird zuletzt vermehrt auf außerschulische Formate gesetzt, die gezielt Mädchen ansprechen und ihnen so die klischeebehaftete Informatik aus einer anderen Perspektive aufzeigen sollen, um langfristig mehr Schülerinnen an die faszinierende Wissenschaft der Informatik heranzuführen. In diese Angebote reiht sich auch dieser Workshop ein, wenngleich die von den Autorinnen und Autoren dieses Beitrags umgesetzten Ideen ihrer Ansicht nach ausdrücklich für beide Geschlechter gleichermaßen förderlich sind. So berücksichtigt das in diesem Praxisbeitrag vorgestellte Workshopkonzept gezielt grundlegende Ansprüche an eine gendersensitive Informatikdidaktik (Abs. 2.1) und bedient sich innerhalb des Kontextes des *Internet of Things* (Abs. 2.2) den Ideen hinter *Digital Fabrication* (Abs. 2.3) und *Physical Computing* (Abs. 2.4). Am Ende des (in Abs. 3 beschriebenen) Workshopablaufs steht dabei eine internetfähige, selbstgestaltete und mithilfe von 3D-Druckern konstruierte Lampe (Abb. 1), die in Abhängigkeit der jeweiligen Wetterlage, die von dem verwendeten Mikrocontroller über einen Webservice abgerufen wird (Abs. 3), in unterschiedlichen Lichteffekten leuchtet.

[2] *Quelle:* Statistisches Bundesamt: „Bildung und Kultur (2018): Schnellmeldungsergebnisse der Hochschulstatistik zu Studierenden und Studienanfänger/-innen (vorläufige Ergebnisse) - Wintersemester 2018/2019“

[3] *Quelle:* Deutscher Industrie- und Handelskammertag (2017): „Statistik Ausbildung 2017“

2 Didaktische Relevanz des Workshopkonzepts

Im Folgenden sollen die didaktischen Überlegungen, die bei der Konzeption dieses Workshops (WS) zum Tragen kamen, von den grundlegenden Aspekten gendersensitiver Informatikdidaktik (Abs. 2.1[4]) und Kontextorientierung (Abs. 2.2) über die gewählten Paradigmen des *Parametric Design* (Abs. 2.3) und *Physical Computing* (Abs. 2.4) dargelegt werden.

2.1 Aspekte gendersensitiver Informatikdidaktik

Auf der Suche nach Ursachen für die unverkennbare Geschlechterkluft in der Informatik (vgl. Abs. 1) begnügen sich nur wenige Studien mit der platten Erkenntnis, dass Mädchen und junge Frauen schlichtweg weniger Interesse an entsprechenden Themen hätten [An08, SM11]. Tiefgreifendere Untersuchungen hingegen kommen zu dem Ergebnis, dass sie die Informatik als attraktiveres Umfeld betrachten, sobald

- die behandelten Technologien in *soziale Kontexte* eingebettet sind, in denen sie selbst involviert sind bzw. sich selbst wiedersehen können [FMM97],
- Informationstechnologien verwendet werden, um *Menschen* in ihren *alltäglichen Situationen* zu unterstützen [ZS08],
- sie zu einem *übergeordneten Ziel* eingesetzt wird [FM02] und
- Informatik als *kreativer* [DKW14] und *kollaborativer* [ZS08, Eg16, MNS08] Prozess aufgezeigt wird, in dem den Mädchen die Möglichkeit gegeben wird, *sich miteinander auszutauschen* und *zusammenzuarbeiten* [ZS08].

Sowohl *schulische* als auch *außerschulische* Lernumgebungen bieten ausreichend Spielraum, diesen „besonderen“ Anforderungen an eine gendersensitive Informatikdidaktik gerecht zu werden und so situatives Interesse zu wecken, das sich bestenfalls zu langfristig anhaltender Motivation entwickeln kann [Ma17]. Vor allem letztere bieten die Möglichkeit, den zahlreichen vorherrschenden Klischees und Stereotypen, die noch immer über den Berufsalltag von Informatikern[5] existieren, entgegenzuwirken. Doch dazu sollten sich diese Angebote explixit und ausschließlich an Mädchen und junge Frauen richten, denn außerschulische Angebote rund um die Informatik werden ansonsten leider immer noch grundsätzlich von Jungen wahrgenommen [SM11, Wi16]. So richtet sich auch das BMBF-geförderte Projekt *smile* (`https://www.smile-smart-it.de`), im Rahmen dessen dieses WS-Konzept entwickelt und durchgeführt wurde und wird, über ein Angebot an (außerschulischen) WS explixit an Mädchen und junge Frauen, um das übergeordnete Ziel zu verfolgen, langfristig und nachhaltig den Anteil an Frauen in informatischen Studiengängen und Berufen zu erhöhen.

[4] Ausschnitte aus diesem Abs. 2.1 wurden bereits in [PD18] veröffentlicht.

[5] Hier wird bewusst nur die männliche Form verwendet. Schließlich ist das Bild eines typischen Informatikers männlich, unattraktiv und sozial wenig kompetent [Ch13].

2.2 Informatik im Kontext und das „Internet of Things“

Im *Internet of Things (IoT)* unterstützen immer kleiner werdende Computer, die in immer mehr alltägliche Gegenstände wie Lampen, Kühlschränke, Rasierapparate, Staubsauger oder Autos eingebettet werden, den Menschen in unterschiedlichsten Situationen. Um genau diese „Bewältigung lebensweltlicher Herausforderungen in Verbindung mit Informatiksystemen“ [DKW11, 97] geht es in dem Konzept *Informatik im Kontext (IniK)* zur Planung, Durchführung und Auswertung von IU, in dem über eine Ausrichtung an der Lebenswelt der SuS eine zusätzliche Motivation verfolgt wird. Das IoT erfüllt zweifelsohne die von [DKW11] gestellten Anforderungen an einen guten Kontext im Sinne einer IniK-Reihe. So bewegen sich die Schülerinnen in dem in diesem Beitrag vorgestellten WS im Rahmen dieses Kontextes – wenn auch nicht stringent in den einzelnen Phasen, für die [DKW11] plädieren. Sie begreifen die Technologien, an die sie herangeführt werden, so im Sinne einer Gendersensitivität (vgl. Abs. 2.1) in soziale, lebensweltliche Kontexte eingebettet.

2.3 Parametric Design

Computational Design (CD) ist ein Prozess, der Merkmale der Programmierung zur Erzeugung und Modifikation von 2D- und 3D-Visualisierungen beinhaltet. Ein typisches Konstruktionsparadigma im Sinne von CD ist *Parametric Design* (PD), das oft mit der Architektur und der Verwendung von Parametern zur Steuerung der Eigenschaften von 2D- oder 3D-Modellen zusammenhängt. Obwohl es einen anhaltenden Trend zum Einsatz von *Digital Fabrication (DF)* und insbesondere 3D-Druck in Bildungskontexten gibt, ist ihr Potenzial für den Anfangsprogrammierunterricht für viele Lehrerinnen und Lehrer immer noch etwas Neues. Schließlich lassen sich sowohl schulische als auch außerschulische programmierlastige Lernaktivitäten über CD und DF auflockern und so Lernprozesse über das Herstellen von eigens kreierten Artefakten unterstützen [CDT18]. Zusätzlich schärft die Verbindung der Gestaltung von 2D- und 3D-Formen mit grundlegenden Programmierkonzepten wie Schleifen, (konditionalen) Verzweigungen und Methodenaufrufen Fähigkeiten des *Computational Thinking* wie Zerlegung und Mustererkennung [CTD19] . Parametrische Entwurfswerkzeuge gehen also deutlich über die Möglichkeiten von nicht programmierbaren CAD-Werkzeugen hinaus und ermöglichen die Erstellung komplexer Formen und Geometrien wie Fraktale oder Bäume mit Zweigen, die rekursive oder repetitive Elemente beinhalten können [CDT18]. CD findet vereinzelt bereits Einsatz in Lernaktivitäten: So konzentrieren sich bspw. [DKW14] und [JB13] auf die Verwendung der Programmiersprache *Processing* zur Erzeugung von 2D-Formen, die mit Lasercuttern hergestellt werden können, und ihre Auswirkungen; in dem Ansatz von [KKR17] werden *Turtle Graphics* erzeugt und mithilfe von 3D-Druckern als motivierende Unterstützung genutzt. Der Einsatz von PD-Tools wie *BlocksCAD* (vgl. Abs. 3) ist bislang jedoch noch deutlich seltener. So darf und soll dieser Praxisbeitrag auch als Plädoyer für einen verstärkten Einsatz von PD verstanden werden.

2.4 Physical Computing

Eingebettete Systeme bieten durch das Zusammenspiel von Hardware- und Softwaresystemen und den damit verbundenen Möglichkeiten zur Interaktion zwischen Mensch und Maschine eine Vielzahl von Chancen, Informatik in einem praktischen Kontext zu vermitteln. *Physical Computing (PC)* beschreibt dabei den Prozess, ein solches System zu erschaffen. Neben den bekannten, positiven Effekten der Kontextualisierung und dem Einsatz von Fallstudien in Ingenieurwissenschaften [YAM19], wirkt sich auch die Entwicklung solcher physischer Systeme positiv auf Lerngruppen aus. Sentence et al. fassen diese Effekte folgendermaßen zusammen: Die Motivation steigt, da die Lernergebnisse nicht virtuell existieren, sondern in Form eines praktischen Produktes sichtbar werden. Zudem hilft die Greifbarkeit dieser Systeme den Lerngruppen beim Debuggen und Optimieren, Programmierkonzepte besser zu verstehen. Die Arbeit lässt sich außerdem sinnvoll auf mehrere Personen aufteilen, z. B. kümmert sich eine Person um Schnittstellen und Algorithmen, während eine andere ein Gehäuse zum 3D-Druck modelliert. So wird das Arbeiten im Team geschult. Zudem bietet das PC die Möglichkeit, sich kreativ zu entfalten [Se17]. Speziell Mädchen profitieren von dem Einsatz solcher Systeme im schulischen Kontext. So gaben Mädchen, die sich mit PC beschäftigt haben, an, dass sie Spaß am Programmieren gefunden haben und sich außerdem selbstbewusster fühlen [Se17].

3 Workshoptools und -ablauf

Um eine günstige IoT-Anwendung zu entwickeln, wurde das Entwicklungsboard *D1 mini* vom Hersteller *WEMOS Electronics* gewählt. Dieses basiert auf den 32-Bit Mikrocontroller *ESP8266EX* von *Espressif* und zeichnet sich unter anderem durch die native Unterstützung der 801.11 b/g/n Funknetzwerkstandards und der Netzwerkprotokolle TCP, UDP und HTTP aus. Darüber hinaus bietet der Hersteller zahlreiche pin-kompatible Erweiterungen für die Boards der D1-Reihe an, welche unkompliziert auf das Entwicklungsboard gesteckt werden können. Dadurch entfallen potentielle Fehlerquellen beim Anschließen neuer Sensoren bzw. Aktoren durch Wackelkontakte oder fehlerhafte Verkabelungen. Um die über das Internet gewonnene (s. u.) Außentemperatur auch optisch anzuzeigen, wurde ein 0.66 Zoll großes OLED-Display mit einer Auflösung von 64x48 Pixel, ebenfalls vom Hersteller WEMOS, verwendet. Nachdem man zwei Reihen von Stiftleisten an die OLED-Display Erweiterung gelötet hat, kann man dieses auf das Entwicklungsboard stecken. Zusätzlich wurde zur visuellen Darstellung des aktuellen Wetters eine WS2812B RGB-LED verwendet. Der Vorteil gegenüber konventionellen RGB-LEDs, welche drei Kathoden bzw. Anoden zur Darstellung der verschiedenen Farben benötigen, ist die flexible Programmierbarkeit. Während zur Darstellung der Farben an drei Pins verschiedene Spannungspegel mittels PWM gesetzt werden müssten, kann die WS2812B LED die Farben mit Hilfe der Arduino kompatiblen FastLED Bibliothek über eine einzige Datenleitung und in verschiedenen Farbräumen darstellen. Dies erleichtert auch die Erstellung von eigenen Animationen deutlich. Um die LED mit dem Entwicklungsboard zu verbinden,

wurde ein Steckbrett verwendet. Die aktuellen Wetterdaten werden über die kostenfreie *Openweathermap* (`https://openweathermap.org/`) API aus dem Internet bezogen. Mit Hilfe der HTTP-Methode GET wird das Wetter für den entsprechenden Standort angefragt und die Antwort des Servers vom Mikrocontroller empfangen, geparst und anschließend mit einer vorher festgelegten Farbe oder Animation dargestellt. Ein Plexiglasstab, welcher mittels einer Halterung aus einem 3D-Drucker unmittelbar über der LED befestigt ist, leitet das Licht in eine von den Teilnehmerinnen selbst modellierten Blume (siehe Abb. 1), welche ebenfalls mit einem 3D-Drucker gefertigt wurde. In beiden Fällen wurde PLA als Druckmaterial verwendet. Die Firmware für den Mikrocontroller wurde innerhalb der *ArduinoIDE* (`https://www.arduino.cc/en/main/software`) geschrieben. Zusätzlich haben die Workshopteilnehmerinnen ihre Animationen mit dem blockbasierten Werkzeug *mBlock3* (`http://www.mblock.cc/mblock-software/`) erstellt. Die darin verwendeten Blöcke wurden eigens für den Workshop entwickelt und zur Verfügung gestellt. Die Modellierung der Blumen erfolgte parametrisch mit dem Tool *BlocksCAD* (`https://www.blockscad3d.com/`), welches spielerisch Programmierkonzepte durch die Visualisierung von Formen vermittelt.

Der **Ablauf** des in diesem Praxisbeitrag vorgestellten WS[6,7] gliedert sich in drei Teile: An den Programmiereinstieg über PD (Teil 1) schließt sich die Phase zum PC (Teil 2) an, bevor die einzelnen Komponenten abschließend zur fertigen Blume zusammengesteckt werden und eine Präsentation ihres Entstehungsprozess vorbereitet wird (Teil 3). In Abhängigkeit der Vorerfahrung der SuS bzw. des gewünschten Grads an Anleitung lässt sich die Gestaltung der Teile 1 und 2 bzgl. der einzuplanenden Zeit anpassen (vgl. Abb. 2), sodass sich der WS insgesamt auf 18 bis 30 (Zeit-)Stunden beläuft und so sowohl für den außerschulischen (bspw. für die Durchführung in einwöchigen WS oder Projektwochen) als auch schulischen (bspw. als Unterrichtsreihe zum Programmier-Anfangsunterricht) Einsatz eignet.

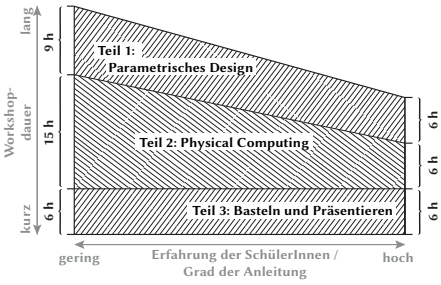

Abb. 2: Überblick über das zeitliche Verhältnis der einzelnen WS-Abschnitte

Im **ersten Teil** des WS lernen die Teilnehmenden nach einer Einführung in die Oberfläche des verwendeten Tools *BlocksCAD* (vgl. Abs. 3), wie man grundlegende geometrische Formen (Würfel, Kugeln und Zylinder) über *Parameter* oder ein Anwenden von *Operationen* wie bspw. Skalierungen anpassen kann. Im Anschluss daran begreifen sie, dass sich diese Grundkörper durch *Mengenoperationen* wie bspw. Vereinigungen oder Differenzen zu nahezu jeder vorstellbaren Gestalt formen lassen. Über eine Verkettung mehrerer solcher Operationen wird dann mit einem Arbeitsblatt dazu angeleitet, die Grundform eines

[6] (Stunden-)Verlaufspläne, Arbeitsblätter, verwendete Libraries und Code-Beispiele finden sich im Git-Repository des Projekts (`https://github.com/projekt-smile/bloecke-blumen-mikrocontroller-und-das-internet-of-things`) oder über den Download-Bereich der Projekt-Webseite (`https://www.smile-smart-it.de`).

[7] Dieser WS wurde als Anschluss an den in [PD18] beschriebenen Einstieg in Sensorik und Aktorik mithilfe von *littleBits* konzipiert und auch mehrmals im Anschluss daran durchgeführt.

Blattes zu modellieren sowie den entstehenden *Code* über die Deklaration einer *Methode* auszulagern und so an mehreren Stellen des *Programms* aufrufbar zu machen (vgl. Abb. 3, *links*). Anschließend lernen die SuS das Prinzip hinter *Schleifen* kennen und können so bspw. Blüten statt über repetitive Methodenaufrufe durch Verwendung eines einzigen zusätzlichen Blocks aus den zuvor modellierten Blättern kreieren (vgl. Abb. 3, *rechts*). Über die zusätzliche Ergänzung einer *Verzweigung* (`falls i ist gerade`) lässt sich optional an dieser Stelle ein weiteres Programmierkonzept einführen und so bspw. jedes zweite Blatt in der Reihe länglicher gestalten. Um die Blume 3D-druckbar zu gestalten, muss sie nach einer individuellen Anpassung (optional nach einer Online-Recherche) durch die S(uS) mit Hilfe der Differenz-Funktion von unten abgeflacht werden. Außerdem sollte eine Knospe, die von einem Zylinder ausgepart ist und in die sich so später ein Acrylglasstab stecken lässt (vgl. Abb. 1), ergänzt werden. Nach dieser (verhältnismäßig) angeleiteten Einstiegsphase lässt sich optionalerweise der Arbeitsauftrag geben, weitere parametrische Designs zu recherchieren und mit *BlocksCAD* zu modellieren, um sie anschließend für die personalisierten Lampen zu verwenden. Für den gesamten Prozess des Gestaltens der Blume bzw. einer anderen beliebigen Form empfehlen die Autorinnen und Autoren dieses Beitrags dabei einen zeitlichen Rahmen von 6 bis 9 Zeitstunden (s. o.). Selbstverständlich lässt sich dieser Teil des WS dabei auch isoliert von dem restlichen WS durchführen und so ein kreativer Einstieg in die Vermittlung von Programmiergrundlagen finden.

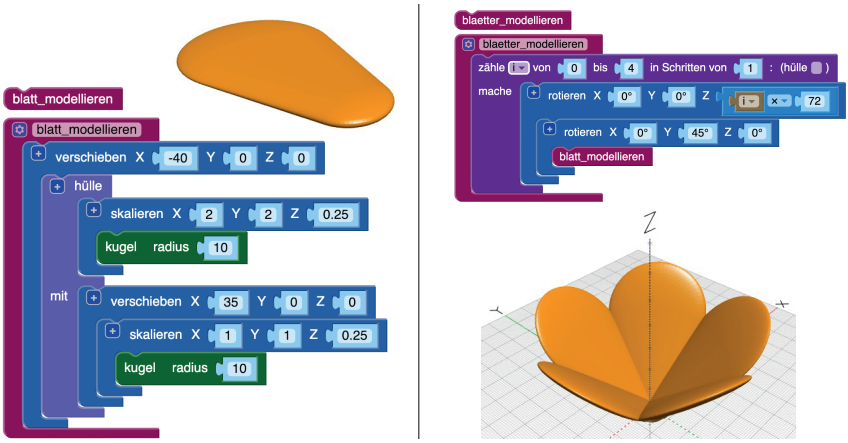

Abb. 3: Methode (*links unten*) zur Modellierung eines Blattes (*links oben*) und Methode (*rechts oben*) zur Modellierung einer Blüte (*rechts unten*) über die Verwendung einer Schleife in *BlocksCAD*

Der **zweite Teil** des WS behandelt die Programmierung des Mikrocontrollers. Dazu erschließen sich die Teilnehmenden nach einem Einstieg in die verwendete blockbasierte Programmierumgebung *mBlock* (vgl. Abs. 3) zunächst aus den ihnen aus *BlocksCAD* bekannten Schleifen die Bedeutung der `loop`-Funktion in Programmen, indem sie die onboard LED des verwendeten Mikrocontrollers (vgl. Abs. 3) über die `delay`-Funktion in einem beliebigen Intervall zum Blinken bringen. Der Bogen zum textuellen Programmieren wird kontinuierlich dadurch gezogen, dass *mBlock* den blockbasiert einzugebenden Code

on the fly als textuelle Beschreibung in Arduino-Code bereitstellt. Das Anschließen und Ansteuern des OLED-Displays[8] (vgl. Abs. 3) stellt den nächsten Abschnitt im WS dar: So sollen die S(uS) zunächst ihren Namen auf dem Display anzeigen lassen, bevor sie (optional) kleine Bitmap Grafiken erstellen können, die in Hinblick auf das übergeordnete Ziel, eine smarte Wetterlampe zu kreieren, verschiedene Wetterlagen (Sonne, Wolken, Schnee...) symbolisieren. Dieses übergeordnete Ziel findet sich auch im nächsten Abschnitt zur Gestaltung von Lichteffekten (bspw. in Farbübergängen von Gelb- zu Rottönen für *sonnig* oder Farbverläufe in Blautönen mit kurzen gelben *Blitzen* bei *Gewitter*) wieder, in denen die anzuschließende externe RGB-LED leuchten soll[9]. Dazu bietet es sich an, die Gruppe in Zweier- oder Dreierteams aufzuteilen und jeder Gruppe einen Effekt zuzuordnen. So entsteht gleichzeitig der Nebeneffekt, dass die S(uS) (Software-)Entwicklung als kollaborativen Prozess (vgl. Abs. 2.1 und 4) begreifen, indem alle etwas zum Endprodukt beitragen. Auch dieser zweite Teil des WS lässt sich zeitlich variieren: Die Autorinnen und Autoren veranschlagen je nach Vorerfahrung der Teilnehmenden, Grad der Anleitung und je nachdem, ob das OLED-Display behandelt wird oder nicht, für diesen Teil 6 bis 15 Zeitstunden.

Während die vorangegangenen WS-Abschnitte hinsichtlich ihrer Länge relativ variabel sind, hat die Erfahrung gezeigt (vgl. Abs. 4), dass der **dritte Teil**, in dem die Schülerinnen ihre Ergebnisse präsentieren sollen, vergleichsweise konstant 6 Zeitstunden beansprucht. Die Präsentation sollte dabei den Anspruch erfüllen, multimedial (ob in Form eines VLOGs, eines kurzen „Werbe"-Videos o. Ä.) und ansprechend gestaltet zu sein. Projektseitig (vgl. Abs. 2.1) wird über diesen Teil das Ziel verfolgt, die Artefakte der WS-Teilnehmerinnen auf Veranstaltungen und Social-Media-Kanälen adressatengerecht zu präsentieren, um so über einen Multiplikatoreffekt weitere WS-Teilnehmerinnen zu akquirieren.

4 Darstellung der Erfahrungen, Reflexion und mögliche Abweichungen

Wie bereits erwähnt, steht im Fokus dieses Workshops, die Zielgruppe junger Mädchen und Frauen für informatische Inhalte nicht nur temporär zu begeistern, sondern auch nachhaltiges Interesse zu festigen. Während der zweite Aspekt zu diesem Zeitpunkt noch nicht hinreichend beantwortet werden kann, soll abschließend eine erste Evaluation hinsichtlich der motivierend gestalteten Inhalte und insbesondere deren ansprechender Vermittlung gegeben werden. Ein Kernanliegen dieses Konzepts ist es, die Artefakte möglichst preisgünstig zu realisieren, damit die Teilnehmerinnen diese nach Ende des Workshops ohne Zuzahlung mit nach Hause nehmen können. Hintergrund dafür ist die Annahme, dass die Schülerinnen sich stärker mit ihren Projekten identifizieren und somit das persönliche Engagement adressiert wird. Die Wetterblume soll hierbei einen kreativ gestalteten Alltagsgegenstand darstellen, der zudem noch den praktischen Anwendungsfall der Wettervorhersage miteinbezieht. Ein weiterer Effekt der Artefaktbildung ist die Möglichkeit, informatische Inhalte eng an kreative

[8] Um die Kosten der einzelnen Lampen zu reduzieren, lässt sich alternativ auch auf das Display verzichten.

[9] *mBlock*-Libraries, die die Ansteuerung der in Abs. 3 beschriebenen Hardwarekomponenten ermöglichen und u. a. die Effektgestaltung stark vereinfachen, finden sich in dem Git-Repository des Projekts (s. Fußnote 6).

Prozesse zu binden und somit den Einstieg in diese zu erleichtern. Das blockbasierte Design von *BlocksCAD* führt die Schülerinnen bereits an die spätere Programmierung heran und die Funktionsweise der Blöcke wird direkt sichtbar im Editor. In einer früheren Version des Workshopangebots wurde die Programmierung der Lichteffekte textuell mit der *Arduino IDE* realisiert. Insbesondere bei jüngeren Schülerinnen stieß diese auf Ablehnung oder war nur schwer umsetzbar. Inspiriert von den deutlich kostenintensiveren, aber attraktiven Systemen wie *Calliope* und *microbit* wurde folglich auf eine blockbasierte Form der Programmierung mit *mBlock* umgestiegen. Diese Art der Programmierung löste mehr Begeisterung aus und führte dazu, dass die Schülerinnen deutlich mehr von sich aus ausprobierten. Die parallele Übersetzung der gesetzten Blöcke in textuelle Prorammiersprache und die Möglichkeit, diese direkt auf den Mikrocontroller hochzuladen, ermöglicht gleichzeitig eine Binnendifferenzierung in einer Gruppe mit heterogenen Informatikkenntnissen. Das spätere gemeinsame Zusammenführen der individuell gestalteten Lichteffekte für die gewählten Wetterphänomene kennzeichnet einen zentralen Punkt für das Gemeinschaftsgefühl der Gruppe; jede einzelne Schülerin trägt einen Teil zum späteren Effektrepertoire bei und ist doch unabhängig in ihren Entscheidungen. Zusammenfassend lässt sich schlussfolgern, dass die beschriebenen Kriterien an ein gendersensitives Workshopkonzept von diesem Ansatz erfüllt werden. Ob diese Einordnung den gewünschten Motivationseffekt herbeiführen wird, bedarf weiterer Evaluation und Erprobung. In diesem Zusammenhang sei darauf verwiesen, dass der hier beschriebene Ansatz Spielraum für zahlreiche Variationen – sowohl für eine Verwendung in außerschulischen als auch schulischen Kontexten – lässt: denkbar wäre die Anbindung anderer APIs wie *Instagram* anstatt der *openweathermap* und andere parametrische Formen, wie bspw. Regenschirme, gestalten zu lassen.

Abschließende Anmerkung: Dieser Workshop wurde im Rahmen des Projekts *smile*, das vom Bundesministerium für Bildung und Forschung unter den Förderkennzeichen 01FP1611 bis 01FP1615 gefördert wird, konzipiert und mehrmals durchgeführt.

Literaturverzeichnis

[An08] Anderson, N.; Lankshear, C.; Timms, C.; Courtney, L.: ”Because it’s boring, irrelevant and I don’t like computers”: Why high school girls avoid professionally-oriented ICT subjects. Computers & Education, 50(4):1304 – 1318, 2008.

[CDT18] Chytas, C.; Diethelm, I.; Tsilingiris, A.: Learning programming through design: An analysis of parametric design projects in digital fabrication labs and an online makerspace. In: IEEE Global Engineering Education Conference (EDUCON). S. 1978–1987, 2018.

[Ch13] Cheryan, S.; Plaut, V. C.; Handron, C.; Hudson, L.: The Stereotypical Computer Scientist: Gendered Media Representations as a Barrier to Inclusion for Women. Sex Roles, 69(1):58–71, 2013.

[CTD19] Chytas, C.; Tsilingiris, A.; Diethelm, I.: Exploring Computational Thinking Skills in 3D Printing: A Data Analysis of an Online Makerspace. In: 2019 IEEE Global Engineering Education Conference (EDUCON). IEEE, S. 1173–1179, 2019.

[DKW11] Diethelm, I.; Koubek, J.; Witten, H.: IniK — Informatik im Kontext. LOG IN, 31(1):97–104, 2011.

[DKW14] Dittert, N.; Katterfeldt, E.; Wilske, S.: Programming Jewelry: Revealing Models behind Digital Fabrication. In: FabLearn Europe. June 2014.

[Eg16] Eguchi, A.: RoboCupJunior for Promoting STEM Education, 21st Century Skills, and Technological Advancement Through Robotics Competition. Robot. Auton. Syst., 75(PB):692–699, 2016.

[FM02] Fisher, A.; Margolis, J.: Unlocking the Clubhouse: The Carnegie Mellon Experience. SIGCSE Bull., 34(2):79–83, 2002.

[FMM97] Fisher, A.; Margolis, J.; Miller, F.: Undergraduate Women in Computer Science: Experience, Motivation and Culture. In: Proc. of the Twenty-eighth SIGCSE Technical Symposium on Computer Science Education. SIGCSE '97, ACM, New York, S. 106–110, 1997.

[JB13] Jacobs, J.; Buechley, L.: Codeable objects: computational design and digital fabrication for novice programmers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, S. 1589–1598, 2013.

[KKR17] Krisch, O.; Kastl, P.; Romeike, R.: 3D-Druck als Motivation für die Einführung in die Programmierung in der Realschule. Informatische Bildung zum Verstehen und Gestalten der digitalen Welt, 2017.

[Kl16] Klan, F.: Informatik – reine Männersache? LOG IN, 36(1):35–38, 2016.

[Ma17] Master, A.; Cheryan, S.; Moscatelli, A.; Meltzoff, A. N.: Programming experience promotes higher STEM motivation among first-grade girls. Journal of Experimental Child Psychology, 160:92 – 106, 2017.

[MNS08] Mitnik, R.; Nussbaum, M.; Soto, A.: An autonomous educational mobile robot mediator. Autonomous Robots, 25(4):367–382, Nov 2008.

[PD18] Pancratz, N.; Diethelm, I.: Including Part-Whole-Thinking in a Girls' Engineering Course through the Use of littleBits: A Practical Report on Including Part-Whole-Thinking into the Content of Computer Science Education. In: IEEE Global Engineering Education Conference (EDUCON). S. 1134–1138, 2018.

[Se17] Sentance, S.; Waite, J.; Hodges, S.; MacLeod, E.; Yeomans, L. E.: "Creating Cool Stuff" - Pupils' experience of the BBC micro:bit. In: Proceedings of the 48th ACM Technical Symposium on Computer Science Education. 2017.

[SM11] Stoilescu, D.; McDougall, D.: Gender digital divide and challenges in undergraduate computer science programs. Canadian Journal of Education, 34(1):308–333, 2011.

[Wi16] Witherspoon, E. B.; Schunn, C. D.; Higashi, R. M.; Baehr, E. C.: Gender, interest, and prior experience shape opportunities to learn programming in robotics competitions. International Journal of STEM Education, 3(1):18, 2016.

[YAM19] Yadav, A.; Alexander, V.; Mehta, S.: Case-based Instruction in Undergraduate Engineering: Does Student Confidence Predict Learning? International Journal of Engineering Education, 35(1):25–34, 2019.

[ZS08] Zimmermann, L.; Sprung, G.: Technology is Female: How Girls Can Be Motivated to Learn Programming and Take up Technical Studies through Adaptations of the Curriculum, Changes in Didactics, and Optimized Interface Design. In: Proc. of ICEE 08, Pécs. 2008.