



Arbeit mit
Elektronik-
elementen
und Code.

Falsificatrix blink() – zwei Beispiele

Adriana Mikolaskova

Programmiertes Licht

Die Schülerinnen und Schüler haben einen Raum aus Karton gebaut und eine Vorstellung davon entwickelt, was sich nach Anbruch der Dunkelheit in diesem Raum abspielen könnte. Gefragt war nichts Spektakuläres, sondern eine alltägliche Situation, die sich in Lichtemissionen manifestiert. Von naheliegenden Ideen wie dem Flimmern eines Fernseherers oder dem Lodern eines Cheminéeofens bis hin zu flackernden Neonröhren eines Kühl- und Lagerraums und dem sanft gedimmten Licht eines Aquariums, in dem von Zeit zu Zeit ein grosser Fisch den grünen Schein trübt, wurden viele Ideen genannt und einige davon umgesetzt.

Mithilfe eines Mini-Computers haben die Schülerinnen und Schüler die Beleuchtung im Raum programmiert. Die Programmierung und Elektronik waren ausreichend einfach,¹ um mit wenig Programmierkenntnissen relativ rasch ein Ergebnis zu erreichen, das weiterentwickelt werden konnte.

Bald – wesentlich schneller als beim Anwenden eines Grafikprogrammes – waren die technischen Schwierigkeiten überwunden. Der Umgang mit Farbe, Rhythmus, Zufall wurde erprobt, die Überraschungen und Überarbeitungen konnten betrachtet und diskutiert werden. Auch die bis anhin noch weitgehend unerprobte additive Farbmischung war ein Thema, da für eine bestimmte Farbe die entsprechenden Rot-, Grün- und Blau-Anteile definiert werden mussten.

Im Zentrum stand nicht die Technik, sondern die Auseinandersetzung mit dem vorgestellten Innenleben des Kartongebäudes und damit, wie man sie mit den verfügbaren Gestaltungsmitteln, dem Licht, seiner Farbe und den Frequenzen zum Ausdruck bringen kann.

«Mein Haus wirkt von aussen geometrisch, streng. Innen aber stelle ich es mir ganz gemütlich vor. Ein grosser Holztisch mit bequemen Stühlen, ein kuscheliger Teppich und ein Sofa auf dem es sich eine Familie gemütlich machen und TV schauen kann. Wenn man von aussen die verschiedenen Farben, die auftauchen, betrachtet, kann man denken, dass vielleicht gerade Werbung im Fernseher läuft, während die Familie das Abendessen einnimmt. Vielleicht schauen sie gerade einen lustigen Film, bei dem die ganze Familie lachen muss.

Mein Plan war, dass man von aussen wirklich das Gefühl hat, da läuft gerade Werbung im Fernseher. Von den Farben her finde ich, dass es mir gut gelungen ist, denn als ich zu Hause die Werbung geschaut habe, wechselten die Farben auch so viel. Ich hätte nicht gedacht, dass man so viele verschiedene Farben mischen kann. Allerdings stimmen meiner Meinung nach die Übergänge zwischen den Farben nicht immer ganz. Also bei einer richtigen Werbung sind die Übergänge ein bisschen anders. Die Übergänge sind wahrscheinlich auch ein wenig zu schnell, die Farben bleiben bei der Werbung doch ein wenig länger als bei meinem Haus. Wenn ich noch ein bisschen länger daran gearbeitet hätte, hätte ich es wahrscheinlich noch ein bisschen verbessern können, aber grundsätzlich bin ich zufrieden.»

(Hefteintrag einer Schülerin)

«Licht in einem Raum – vom Flackern über Blinken bis hin zu ruhigem Licht. Ich wollte einerseits viel mit dem Arduino experimentieren und dieses Programm kennenlernen, aber auch auf gestalterische Ebene gewisse Ideen und Ziele erreichen. Beides liess sich nicht immer ganz aufeinander abstimmen, doch musste dennoch etwas entstehen. Eine Idee war, mit verschiedenen Farben den Zuschauer an ein Fest zu erin-

uern, wo viel getanzt wird, man sich bewegt und Spass hat. Wenn man weiterdenken wollen würde, würde man vielleicht diese Überlegungen auch aufs Leben anwenden, dass es viel zu bieten hat, dass man sich auch Auszeiten gönnen soll und Spass haben kann. Denn heutzutage stehen viele Leute unter Druck und geraten in die Gefahr, die andere Seite von Arbeit und Stress zu vergessen. Zwischen den vielen kräftigen Farben gibt es auch zwei kleinere Pausen mit etwas gelbem Dämmerlicht. Im Raum würde sich bei diesen Pausen die Menschenmasse etwas erholen, also Kraft sammeln für den nächsten Tanz und für das nächste Lied, um somit mit viel Schwung und Elan zu starten.

Wenn der Film abläuft, erkennt man das Fest mit den farbfrohen Lichtern. Doch die dämmrigen Pausen werden viele falsch verstehen oder nicht wissen, wie zu interpretieren. Denn was sollen Pausen während einem Fest? Vielleicht müsste man für das nächste Mal den Film länger machen, damit man mehr Zeit hat, diese Unterbrechungen und den Farbablauf eindeutiger darzustellen, zum Beispiel längere Dämmerpausen. Oder einfacher zu erkennen, aber nicht einfach darzustellen, wäre eine ruhigere Situation, in der man Stille findet und zum Nachdenken anregt. Doch dies wäre wieder eine ganz neue Idee und bräuchte ganz andere Mittel. Grundsätzlich mag ich die Lichter und die Farben, doch wenn ich meine Arbeit ansehe, sehe ich auch die Arbeit und meine Überlegungen dahinter, wie auch die Freude, die ich hatte, als ich verstand, wie das Arduino aufgebaut wird und funktioniert.»

(Hefteintrag einer Schülerin, L. R.)

«[Z]udem soll das Haus aus der Menge herausstechen. Weil ich dachte, dass die meisten farbige, sehr flackernde, fröhliche Lichter auswählen würden, wählte ich eine eher triste, unbe-

lebte Lichtgestaltung aus. Das weisse Licht stärkt zudem die Lagerhausstimmung [...]»

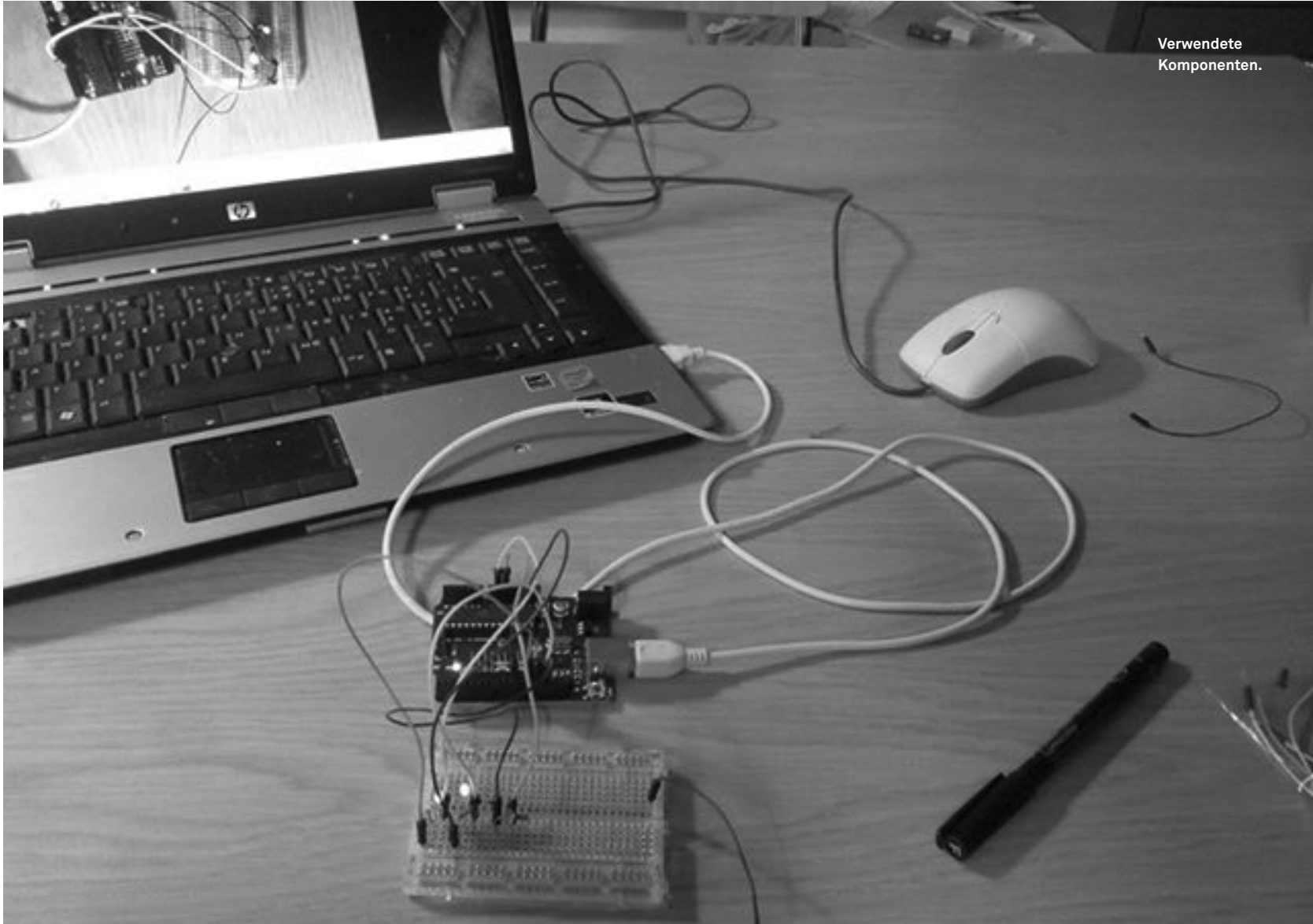
(Hefteintrag eines Schülers, Y. S.)

«Am Anfang sieht man ein Flackern danach wird das weisse Licht greller und flackert nicht mehr. Dann wird das Licht ganz ausgeschaltet und es bleibt einige Zeit dunkel. Ein kurzes Aufleuchten des grünen Lichts unterbricht die Dunkelheit. Mit dem Anfang der Frequenz wollte ich eine Neonlampe mit Wackelkontakt, die plötzlich ausfällt, darstellen. Das grüne Licht stellt einen Hoffnungsschimmer dar. Die ganze Szene spielt in einem Krankenhaus. Die Stimmung ist ähnlich wie in einem Horrorfilm. Die Krankenbetten sind fast alle verlassen, nur einige wenige Betten sind besetzt, die Patienten haben keine Gesichter, sie liegen verdeckt auf ihrem Lager. Beim Auf-flackern des grünen Lichts könnte man noch kurz eine Person sehen, die vorher noch nicht da war. Danach wird das Licht völlig ausgeschaltet, die Schleife wiederholt sich [...]»

(Hefteintrag einer Schülerin, L. L.)

Generierte Malewitsch-Bilder – Falsificatrix

Bei einer später initiierten, analytisch orientierten Arbeit hat eine Gruppe dieser Klasse ebenfalls programmiert. Die Arbeit zur ungegenständlichen Malerei der Moderne enthielt den Auftrag, neue, bisher unbekannte Bilder einer ausgewählten Künstlerin oder eines ausgewählten Künstlers herzustellen, also Fälschungen. Der Schwerpunkt lag auf den formalen Eigenschaften, Gemeinsamkeiten und Unterschieden im Werk der Künstlerin oder des Künstlers. Nach einer ersten Sichtung von Malewitsch-Bildern, meinte ein Schüler, er schreibe ein Programm, das nicht nur eine, sondern unendlich viele Fälschungen generiert.



Verwendete
Komponenten.

```
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}

void loop() {
  setColor(255,255,255);
  delay(random(100,2000));
  setColor(0,0,0);
  delay(30);
  setColor(95,100,100);
  delay(random(200,1000));
  setColor(0,0,0);
  delay(50);
  setColor(250,255,255);
  delay(random(500,3000));
  setColor(0,0,0);
  delay(random(50,70));
  setColor(115,120,120);
  delay(random(700,4000));
  setColor(0,0,0);
  delay(random(20,100));
}

void setColor(int red, int blue, int green)
{
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

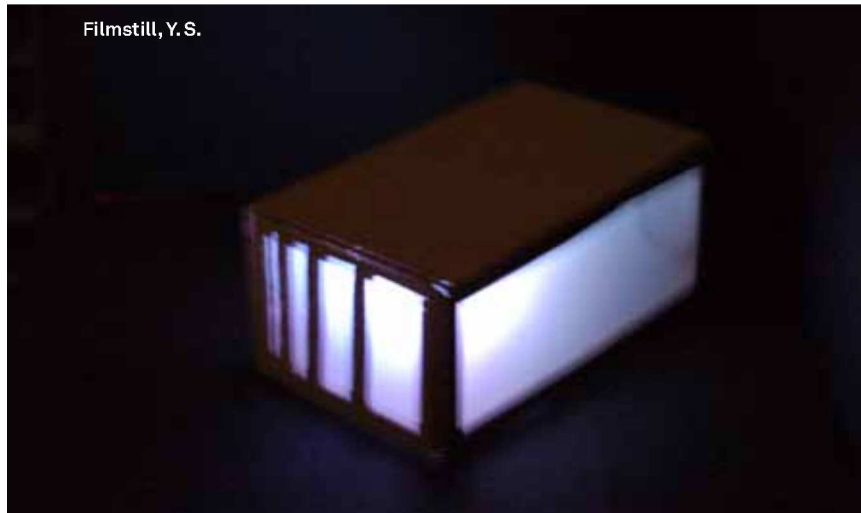
Codebeispiel, Y. S.

```
void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop()
{
  for (int i=0; i < 50; i++){
    setColor(150, 200, 250);
    delay(110);
    setColor(0, 0, 0);
    delay(20);
  }
  setColor(255,255,255);
  delay(2500);
  setColor(0,0,0);
  delay(3000);
  setColor(0,random(20,200),0);
  delay(90);
  setColor(0,0,0);
  delay(3000);
}

void setColor(int red, int green, int blue)
{
  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

Codebeispiel, L. L.



Filmstill, Y. S.



Filmstill, L. L.

Da ich den anfänglich nur halbwegs ernst gemeinten Vorschlag unterstützte, beschloss seine Gruppe, gemeinsam ein solches Programm zu entwickeln. Bald merkten die Schüler, dass dafür nicht in erster Linie Programmierkenntnisse ausschlaggebend waren, sondern hauptsächlich die genaue Analyse der Bilder. Einer von ihnen vergrösserte das Archiv ähnlicher Bilder, der zweite erprobte in einem Bildbearbeitungsprogramm verschiedene Kompositionen, der dritte setzte die Erkenntnisse in einem Programm um. Es war faszinierend, mit welcher Präzision die Gruppe über den Aufbau der Bilder und später über die noch bestehenden Unterschiede diskutierte: Relativ rasch gelangten die Schüler zu dem Schluss, dass nicht alle ungegenständlichen Bilder von Malewitsch gleich gut geeignet waren, die vordergründig einfachsten suprematistischen Kompositionen waren am schwersten zu durchschauen.

Als Bezugsquelle diente sowohl die Bildersuche grosser Suchmaschinen, aber zunehmend auch spezialisierte Datenbanken², über deren Vorteile sich die Schüler bald einig waren.

Diskutiert wurden Farbtöne und deren Verteilung, Größenverhältnisse, aber auch die Notwendigkeit, die allzu klaren Ränder der generierten Formen sowie den Untergrund in seiner Struktur und Farbe zu brechen. Gesetzmässigkeiten wurden benannt, Spielräume des Zufalls abgesteckt und erprobt.

Bemerkungen

Die Beispiele wurden im BG-Unterricht einer zweiten Klasse des Mathematisch-Naturwissenschaftlichen Gymnasiums Rämibühl (Kurzgymnasium) in Zürich durchgeführt. Die Schülerinnen und Schüler dieser Klassen hatten sich zuvor bereits im Informatikunterricht mit der Programmierung auseinandergesetzt – die erworbenen Kenntnisse und Fähigkeiten wur-

den aber bisher nicht für eigene Projekte eingesetzt. Sowohl die Tatsache, dass man mit Programmieren ästhetisch interessante Situationen und bildnerische Probleme bearbeiten kann, als auch die prozessorientierte Arbeitsweise waren für die meisten neu.

Im ersten Fall war die Programmierung ein fester Bestandteil der Aufgabenstellung, d. h. alle Schülerinnen und Schüler haben programmiert. Im zweiten Fall entschied sich eine Schülergruppe, ihren Auftrag mithilfe der Programmierung zu bearbeiten, wobei sie sich die fortgeschrittenen Kenntnisse eines einzelnen Schüler zunutze machten.

Beide Formen, sowohl selber programmieren in einem technisch niederschweligen Rahmen als auch die Beteiligung an der Entwicklung eines Programms, halte ich für lehrreiche Arbeitsweisen, bei denen der Einsatz des Computers nicht auf das Bedienen von Anwendungen beschränkt wird.

Da diese Sequenzen im BG-Unterricht durchgeführt wurden, bestand das Ziel und Zentrum der beiden Aufgaben in beiden Fällen in der Auseinandersetzung mit Vorstellungswelten sowie dem Einsatz und der Analyse bildnerischer Mittel. Die programmierten Bilder wurden immer wieder revidiert, das Programm aufgrund genauer Betrachtung der Zwischenergebnisse weiterentwickelt und somit eine prozessorientierte Arbeitsweise angestrebt, wie sie im BG-Unterricht oft anzutreffen ist.

Trotzdem bin ich unsicher, inwiefern solche Aufgaben im BG-Unterricht wirklich Platz haben. Das Entwickeln solcher bildnerisch orientierter Aufträge und das Nachdenken darüber, inwiefern die Programmierung für bildnerische Zwecke eingesetzt werden könnte, scheint mir aber auf jeden Fall sinnvoll: Im Rahmen des zukünftigen Grundlagenfaches Informatik könnten solche Aufgaben eine attraktive Alternative zu

oft beliebigen Anwendungsbeispielen des Programmierunterrichts bilden und so die Auseinandersetzung mit bildnerischen Themen über den BG-Unterricht hinaus fördern (mit oder ohne Einbezug einer BG-Lehrperson), insbesondere auch, weil eigentlich in der Informatik und deren Anwendungsbereichen Bilder und bildnerische Themen einen grossen Raum einnehmen.

Adriana Mikolaskova unterrichtet am MNG Rämibühl Bildnerische Gestaltung.

Kontakt

Adriana Mikolaskova: adriana.mikolaskova@mng.ch

Mit bestem Dank an Lorenz Brun, der das Programm auf seinem Server hostet: <https://mng.dolansoft.org/bg/malewitsch/g.php>

Anmerkungen

1 Verwendet wurde ein Arduino-UNO-Controller, eine Steckplatine, eine RGB-LED, Widerstände und Verbindungsdrähte. Das Programm wurde am PC geschrieben und über ein USB-Kabel auf den Controller geladen, der durch die programmierten Stromimpulse das LED-Lämpchen in einer bestimmten Farbe und Frequenz zum Leuchten brachte.

2 Dazu zählten sowohl die Archive grosser Museen als auch vor allem das Prometheus Bildarchiv (<http://www.prometheus-bildarchiv.de>).