

1 P3: Physical Computing

1.1 Einordnung

1.1.1 Modul

P3 Digitaler Wandel im Alltag

1.1.2 Kompetenzbereich

Nachempfinden der Funktion einfacher digitaler (Haushalts-)Geräte durch Programmierung von Mikrocontrollern

1.1.3 Thema/Inhalt

Physical Computing ist das kreative Gestalten und Entwickeln interaktiver Objekte und Installationen, die als programmierte, greifbare Artefakte mittels Sensoren (z.B. Lautstärkemesser, Helligkeitssensoren) und Aktoren (z.B. Servomotoren, LEDs) mit ihrer Umwelt kommunizieren. Physical Computing als gestaltungsorientierte Herangehensweise an die Programmierung von Mikrocontrollern in eingebetteten Systemen forciert die Beschäftigung mit folgenden Inhalten:

- Struktur und Eigenschaften einfacher eingebetteter Systeme (z. B. Sensorik, Aktorik, Datenerfassung)
- Praktiken der Arbeit mit eingebetteten Systemen (z. B. Prototyping, Tinkering)
- Programmierung: algorithmische Grundstrukturen, Operatoren, Parallelität, Liveness
- Anknüpfungspunkte: Ziele, Anforderungen und Herausforderungen bei der Arbeit mit eingebetteten Systemen (z. B. Systemqualität, Echtzeitanforderungen), Vernetzung eingebetteter Systeme (Internet of Things, Smart Home, Smart City)

Auf grundlegendem Kompetenzniveau bedeutet das, dass die Lernenden zwischen analogen und digitalen Ein- und Ausgängen unterscheiden können und diese zweckmäßig für analoge und digitale Sensoren (z.B. Lichtwiderstand, Temperatursensor, Potentiometer, Berührungssensor) und Aktoren (z.B. LEDs, Servomotoren und Piezo-Summer) eines vorgegebenen Physical-Computing-Baukastens verwenden. Hierfür müssen sie in der Lage sein, Operatoren und Kontrollstrukturen (Sequenzen, Schleifen, bedingte Verzweigungen, Variablen) in einer Programmiersprache sachgemäß zu verwenden, sowie sich die Bedeutung und Funktion von speziellen Blöcken zur Ansteuerung der Hardwarekomponenten mit den bereitgestellten Materialien zu erschließen.

1.1.4 Bezüge zu anderen Fächern

- Informatik (Informatiksysteme, Leben in und mit vernetzten Systemen, Information und Daten, Algorithmisches Problemlösen, Physical Computing)
- Physik (Elektronik, insb. Sensoren, Messwerterfassung)
- Kunst (Bauen und Konstruieren, Plastisches Gestalten, Inszenieren, Materialien)

1.1.5 Standards

Im Rahmen des Unterrichtsbeispiels werden gemäß Kapitel 2 des Curriculums insbesondere die folgenden abschlussorientierten Standards gefördert:

Fachwissen sachgerecht nutzen

Die Schülerinnen und Schüler ...

- ...beschreiben Funktionsweise und Anwendungsmöglichkeiten einzelner digitaler Technologien.
- ...wenden die Prinzipien Digitaler Welten (Vernetztheit, Interaktivität, Integration) an.

Methoden der Gestaltung und Nutzung von Digitalen Welten zielgerecht einsetzen

Die Schülerinnen und Schüler nutzen kreative oder systematische Verfahren (z.B. Brainstorming, Metaplan, Design Thinking, Business Modell Canvas) zur Eingrenzung von möglichen Interessengebieten und Identifikation von innovativen Geschäftsideen bei der Planung und Durchführung eigener digitaler Projekte.

1.2 Didaktische Hinweise

Es besteht weitgehend Einigkeit darüber, dass intrinsische Motivation und Kreativität wichtige Komponenten erfolgreichen Lernens darstellen. Dies lässt sich auch in der konstruktionistischen Lerntheorie erkennen, welche das Erschaffen persönlich relevanter Artefakte als Kernidee fokussiert. Die hier vorgestellte Unterrichtsreihe wurde mit dem Ziel entwickelt, diese Aspekte zu fördern und zu verbinden. Damit einhergehend ist auch die Idee des *informatischen Töpferns* maßgebend: Physical Computing ermöglicht Lernenden, selbst hergestellte und programmierte digitale, interaktive Artefakte aus der Schule mit nach Hause zu bringen, die ihrer eigenen Vorstellung entstammen und die sie mit Stolz anderen vorführen und mit ihnen diskutieren können.

Um den Schülerinnen und Schülern einen Einblick in gängige Arbeitsweise der Informatik zu geben, mündet die Unterrichtsreihe in ein Projekt, in dem die Lernenden mit einzelnen Elementen agiler Methoden der Softwareentwicklung arbeiten (u. a. Ideensammlung, Nutzung eines Projektboards, zyklische Iterationen in der Projektarbeit, Pair-Programming, Erstellung von Prototypen). Auf diese Weise werden Praktiken des Fachs mit didaktischen Zielen kombiniert, den Lernenden Methoden bereitzustellen, mit denen sie zielgerichtet, selbstverantwortlich und selbstorganisiert eine größere Aufgabe bewältigen können. In der Konzeption der Unterrichtsreihe wurden darüber hinaus die folgenden Prinzipien für Physical-Computing-Projekte im Unterricht verfolgt, die fachliche und didaktische Überlegungen zusammenführen: Integration von Tinkering-Aktivitäten in dedizierte Lernphasen, Herstellung eigener interaktiver Objekte, Entwicklung funktionsfähiger Prototypen, Angebot eines interessanten Rahmens zur Anregung von Fantasie und Kreativität, Integration von Methoden kreativen Lernens, Verbindung technischer Aspekte mit Kunst/Basteln, Strukturierung der Arbeitsprozesse, Auswahl zielgruppenadäquater und für die Projekte geeigneter Werkzeuge und Materialien, kollaborative Arbeit an einer gemeinsamen Ausstellung und Präsentation der finalen Produkte.

1.3 Überblick über die Unterrichtsreihe Physical Computing

1.3.1 Ablauf

Die Unterrichtsreihe „Physical Computing“ gliedert sich in eine Lern- und eine Projektphase. Die Lernphase ist allgemein für Physical-Computing-Projekte nutzbar und kann der eigentlichen Projektarbeit vorgeschaltet oder aber in diese integriert werden.

- *Einführung und Motivation (15 Min.):* Den Schülerinnen und Schülern wird das Themengebiet vorgestellt. Sie erhalten anhand realweltlicher Beispiele eine kurze Einführung in eingebettete Systeme, Mikrocontrollerprogrammierung und Physical Computing und es werden Fotos von Schülerprojekten gezeigt (M1).

- *Lernphase 1 (2 Doppelstunden)*: Im Stationenlernen findet eine detaillierte Einführung in Physical Computing, die genutzten Werkzeuge und Bauteiltypen sowie die entsprechenden Programmelemente statt. Die Lernenden arbeiten zu zweit an den Stationen, die Lehrkraft unterstützt, wenn nötig (M2).
- *Optional: Lernphase 2 (3 Doppelstunden)*: In vertiefenden Aufgaben wenden die Schülerinnen und Schüler im Pair-Programming die neuen Kenntnisse in Miniprojekten an und erarbeiten weitere Grundlagen zur Programmierung (Grundstrukturen, Operationen, Variablen, ...) oder rufen diese wieder in Erinnerung (M3).
- *Planungsphase (45 Minuten)*: Im Brainstorming werden Ideen für mögliche Teilprojekte gesammelt, ggf. kann auch die Lehrkraft Ideen beisteuern. Anschließend finden sich die Schülerinnen und Schüler in kleinen Gruppen von zwei bis vier Personen zu Teams zusammen, wählen zu realisierende Teilprojekte und formulieren hierzu aus Nutzersicht User-Stories, in denen sie die gewünschten Funktionen beschreiben. Anschließend identifizieren sie die zur Bearbeitung notwendigen Tasks aus Perspektive der Entwickler, priorisieren diese und heften die User-Stories mit den Tasks an das Projektboard. In einem Stand-Up-Meeting präsentieren alle Teams ihre Ideen und vorläufigen Arbeitspläne und diskutieren diese mit ihren Mitschülern und der Lehrkraft (vgl. (Brichzin, Kastl, & Romeike, 2018)).
- *Projektarbeit und Reflexion (mindestens vier Doppelstunden)*: Die Schülerinnen und Schüler arbeiten in kleinen Gruppen von zwei bis vier Personen eigenständig und entsprechend ihrer Tasks an der Umsetzung der Teilprojekte. Hier können zusätzliche methodische Elemente wie Pair-Programming genutzt werden, um Programmierarbeiten gleichmäßig auf die Schülerinnen und Schüler zu verteilen und sicherzustellen, dass alle im Detail über die entstehenden Programme Bescheid wissen. Während der Arbeitsphase stehen eine Anleitung (M4) und Aufzeichnungen bzw. Poster aus der Lernphase als Nachschlagewerke zur Verfügung, die Lehrkraft wird hier vor allem beobachtend tätig und greift nur ein, wenn unbedingt nötig. In regelmäßigen Abständen (z. B. zu Beginn und Ende jeder Unterrichtsstunde sowie nach Bedarf) reflektieren die Teams in kurzen Stand-Up-Meetings ihren Fortschritt, diskutieren auftretende Probleme und mögliche Lösungsansätze und definieren die nächsten Arbeitsschritte. Der Fortschritt wird fortlaufend am Projektboard dokumentiert. In regelmäßigen Intervallen (z. B. am Ende jeder Doppelstunde) präsentieren die Teams ihren Mitschülern und der Lehrkraft ihre Prototypen, die mit jeder Iteration um Zusatzfunktionalität erweitert werden und schließlich im fertigen Produkt resultieren. Die Lehrkraft hat jederzeit die Möglichkeit, Wünsche und Prioritäten einzubringen und so aus didaktischer Sicht Einfluss auf das Projektgeschehen zu nehmen.
- *Ausstellung und Präsentation (30 Min. bis zu mehreren Stunden, z. B. Tag der offenen Tür)*: Die Schülerinnen und Schüler präsentieren ihr Projektergebnis, erklären ihre Erfindungen und diskutieren mit Besuchern. Hierfür erstellen sie geeignete Medien und beschreiben anekdotisch die Funktion und den Zweck ihres interaktiven Objekts.

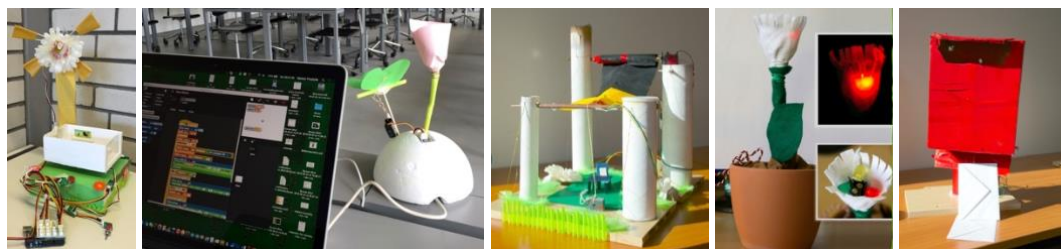
1.3.2 Wahl des Projektkontexts

Im Rahmen des Physical-Computing-Projektes werden breite Themen anstatt konkreter Aufgaben oder Problemstellungen angeboten. Auf diese Weise arbeiten die Teams kollaborativ an einem gemeinsamen Projekt und können innerhalb des vorgegebenen Rahmens ein Teilprojekt wählen, dass ihren Interessen entspricht und an ihre Erfahrungen anknüpft. Solche Kontexte sollten breit genug gewählt werden, um einerseits viele verschiedene Projektideen zu ermöglichen, aber auch spezifisch genug, um Ideen hervorzubringen und den Lernenden einen sinnvollen Austausch untereinander zu ermöglichen. Das Thema sollte darüber hinaus die

Fantasie der Lernenden anregen und auch erlauben, dass künstlerische Produkte entstehen, die nicht unbedingt in der Realität existierende Gegenstände repräsentieren müssen.

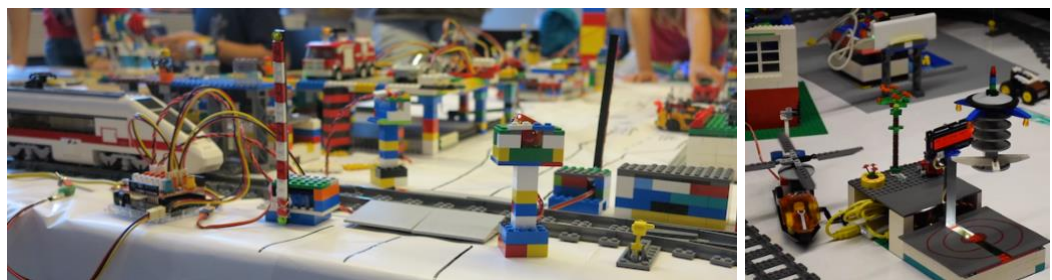
My Interactive Garden

Das Thema „My Interactive Garden“ hat zum Ziel, die Lernenden kollaborativ an der Erstellung einer Ausstellung interaktiver Objekte arbeiten zu lassen, wie sie in einem futuristischen interaktiven Garten zu finden sein können. In einem solchen Garten können bekannte Gegenstände des Alltags stehen, aber auch futuristische Objekte, die es bisher nicht gibt. Dieser Rahmen ermöglicht vielfältige Projekte und regt die Kreativität der Schüler an. Mögliche Teilprojekte sind Gartenlaternen, die entsprechend der Wetterbedingungen unterschiedlich leuchten, Futteranlagen für Vögel, Alarmsysteme für Haus und Garten, Vogelscheuchen im Gemüsebeet, automatische Markisen oder magische Blumen, die mit Besuchern interagieren.



Smart City

Der Kontext „Smart City“ bietet einen interessanten Rahmen mit Anknüpfungspunkten zur Erfahrungswelt der Schülerinnen und Schüler, der durch seine Offenheit kreative Ideen und Lösungswege ermöglicht und Phänomene erwarten lässt, die für die Arbeit mit eingebetteten Systemen typisch sind. Ziel des Projektes ist es, eine interaktive Modellstadt zu gestalten, in der eingebettete Systeme an verschiedenen Stellen ihre Umwelt erfassen (z. B. Helligkeit, Verkehr) und Objekte steuern (z. B. Beleuchtung aktivieren, Ampeln steuern). Dazu können die Lernenden eigene interaktive Objekte entwerfen und gestalten sowie diese mit Sensorik und Aktorik anreichern und miteinander vernetzen.



Alternative Kontexte

Weitere Kontexte, die sich für Physical-Computing-Projekte gut eignen, sind unter anderem: Smart Home, interaktive Spiele oder Enhanced Living.

1.3.3 Unterrichtswerkzeuge

Für die Arbeit in diesem Projekt werden keine Vorkenntnisse vorausgesetzt, weder im Bereich der Arbeit mit eingebetteten Systemen noch in der Programmierung. Daher werden Unterrichtswerkzeuge genutzt, die von Beginn an effizientes Arbeiten ermöglichen, ohne die Möglichkeiten zu stark einzuschränken. Aus diesem Grund fiel die *Hardwareentscheidung* auf die Kombination eines Mikrocontrollers mit einem Baukastensystem: Anstatt mit Steckplatinen

Sensoren und Aktoren aufwendig und fehleranfällig zu verkabeln, nutzen solche Systeme übliche Steckverbinder. Im Rahmen der hier vorgestellten Projekte wird eine Kombination der verbreitet eingesetzten und vielfältigen Plattformen Arduino Uno und Grove genutzt. Als Programmierumgebung wird die blockbasierte Programmiersprache Snap4Arduino genutzt, die flexibel erweiterbar ist und somit auch im fortschreitenden Unterricht genutzt werden kann.

1.3.4 Arbeitsmaterialien

Die hier bereitgestellten Arbeitsmaterialien (siehe digitaler Anhang) wurden für die ausgewählten, oben beschriebenen Werkzeuge erstellt, die neben didaktischen Überlegungen auch deswegen gewählt wurden, da sie sich an vielen Schulen im Informatikunterricht bewährt haben. Bei der Nutzung anderer Werkzeuge können die Materialien entsprechend angepasst werden.

M1: Foliensatz „Einführung in Physical Computing mit Arduino, Grove und Snap4Arduino“

Mit diesem Foliensatz wird zunächst die Idee des Physical Computing vorgestellt. Anschließend wird ein Überblick über die im weiteren Verlauf verwendeten Mikrocontroller, Sensoren und Aktoren sowie die Programmierumgebung „Snap4Arduino“ gegeben. Anhand von Beispielprojekten wird dann der Rahmen der Möglichkeiten gezeigt. Schließlich werden unterstützt durch Folien die notwendigen Vorbereitungen zur Stationenarbeit getroffen und der Ablauf derselben erklärt.



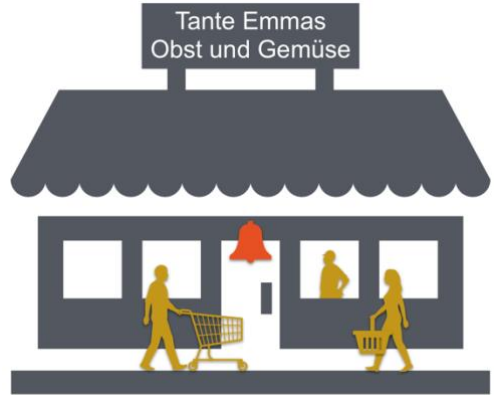
M2: Stationsarbeitsblätter und Laufzettel

Jede Schülerin und jeder Schüler erhält einen eigenen Laufzettel zum Abarbeiten der Stationen, auf dem die Ergebnisse der Stationenarbeit in Stichpunkten festgehalten werden und der eigene Lernfortschritt eingeschätzt werden. Nach jeder bearbeiteten Station berichten die Schülerinnen und Schüler der Lehrkraft und führen ggf. Programmelemente vor. Wenn alles zufriedenstellend erledigt wurde, wird die Station auch von der Lehrperson als abgeschlossen gegengezeichnet.



M3: Foliensatz „Aufgaben zur Anwendung und Vertiefung“

Mit diesem Foliensatz wird zunächst methodisch das Pair-Programming eingeführt, bevor verschiedene Aufgaben die Erkenntnisse aus dem Stationenlernen zur Anwendung bringen:

<p style="text-align: center;">Dynamic Desktop</p> <p>Apples Betriebssystem bietet mit <i>Dynamic Desktop</i> eine Funktion, mittels derer das Desktop-Bild automatisch an die Tageszeit angepasst wird.</p> <p>Aufgabe: Entwerft im Team eine Dynamic-Desktop-Simulation, die sich jedoch nicht an der Tageszeit, sondern an Sensordaten orientiert und beispielsweise entsprechend der Umgebungshelligkeit, der Temperatur oder des Geräuschpegels das Hintergrundbild der Bühne anpasst.</p> <p>Zeit: 30 Minuten</p>	
<p style="text-align: center;">Etch A Sketch</p> <p><i>Etch A Sketch</i> ist ein "Magisches Zeichenbrett" aus den 1960/70er Jahren, das mit zwei Drehknöpfen bedient wird. Die Kunst beim Zeichnen mit Etch A Sketch liegt darin, das Bild in einer durchgängigen Linie zu zeichnen.</p> <p>Aufgabe: Entwerft im Team eine Etch-A-Sketch-Simulation, die für die Eingabe zwei Drehwiderstände nutzt und das Bild während seiner Entstehung auf der Bühne anzeigt.</p> <p>Zeit: 45 Minuten</p>	
<p style="text-align: center;">Palim-Palim</p> <p>Viele kleinere Geschäfte besitzen einen Türgong, der den Ladenbesitzer auf Kunden aufmerksam macht, wenn er sich beispielsweise gerade im Lagerraum befindet.</p> <p>Aufgabe: Entwerft im Team einen Türgong für den Informatikraum, der jeden zehnten Schüler auf besondere Weise begrüßt.</p> <p>Zeit: 20 Minuten</p>	

Auf den folgenden Folien wird – je nach Kenntnisstand der Schülerinnen und Schüler – das Konzept von *Variablen eingeführt oder wiederholt*, deren Verwendung in Snap4Arduino

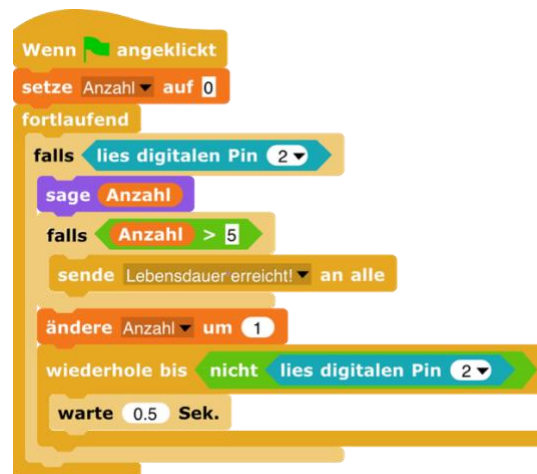
vorgestellt und mit Übungsaufgaben gefestigt, Musterlösungen finden sich dann auf den anschließenden Folien:

Schaltzyklen einer LED-Lampe

Die Anzahl der Schaltzyklen bestimmt, wie oft eine LED-Lampe an- und ausgeschaltet wird und hat Einfluss auf ihre Lebensdauer.

Aufgabe: Implementiere ein Programm, das die Schaltzyklen einer Lampe zählt, den aktuellen Wert anzeigt und bei Erreichen der Lebensdauer einen entsprechenden Hinweis sendet.

Mögliche Lösung:



Countdown

Ein Countdown ist eine regelmäßige Bekanntgabe einer noch verbleibenden Zeitspanne bis zum Eintreten eines Ereignisses und wurde nicht, wie häufig vermutet, zuerst in der Raumfahrt verwendet, sondern in einem Film über die Raumfahrt (Fritz Lang, *Frau im Mond*, 1929; vgl. Wikipedia: [https:// de.wikipedia.org/wiki/Countdown](https://de.wikipedia.org/wiki/Countdown)).

Aufgabe: Implementiere einen Countdown, der die letzten Sekunden bis zum Event spannend macht! Der Startwert soll zu Programmbeginn vom Benutzer eingegeben werden.

Mögliche Lösung:



Fenster-Auf-Erkennung

Die "Fenster-Auf-Erkennung" moderner Heizungsthermostate ermöglicht die Erkennung von plötzlichen, starken Temperaturabfällen, wie sie beim Öffnen eines Fensters entstehen und verhindert, dass während das Fenster geöffnet ist, stark geheizt wird.

Aufgabe: Implementiere ein Programm, das den Durchschnitt der letzten zehn sekundlich erfassten Messwerte eines Temperatursensors berechnet und dann, wenn der Durchschnittswert einen vorab festgelegten Schwellwert deutlich unterschreitet, das Thermostat schließt, also einen Servomotor auf einen bestimmten Winkel einstellt.

Mögliche Lösung:



Abschließend werden noch zwei komplexere Aufgaben zur Vertiefung der Verwendung von Variablen und schließlich zur Anwendung des Gelernten in der gesamten Unterrichtsreihe bereitgestellt. Anstelle des Spiels „Pong“ könnte auch ein beliebiges anderes komplexes Miniprojekt stehen, beispielsweise kann man Pong als Anregung verstehen, ein beliebiges interaktives Spiel implementieren zu lassen.

Anzeigetafel

Bei vielen Sportveranstaltungen wird der aktuelle Spielstand und die verbleibende Zeit über ein digitales Display angezeigt.

Aufgabe: Entwerft im Team eine Anzeigetafel für eine beliebige Sportart, die auf Knopfdruck den Spielstand des jeweiligen Teams erhöht und die verbleibende Spielzeit anzeigt.

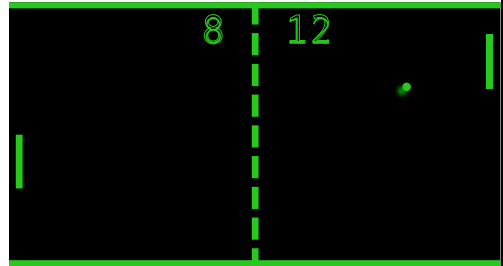
Zusatz: Auszeiten sollen live erfasst werden, indem die Zeit gestoppt und nach der Auszeit wieder gestartet wird.

Zeit: 45 Minuten



Pong

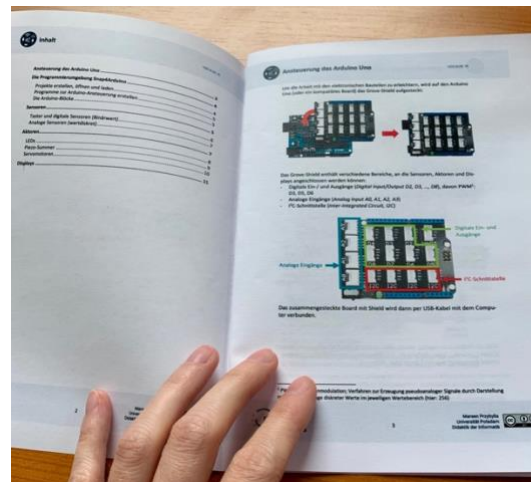
Pong wurde im Jahr 1972 entwickelt und zählt zu den ersten weltweit beliebten Videospielen in Spielhallen, später auch auf Heimgeräten. Zwei Spieler spielen mit Joysticks gegeneinander und versuchen, den Ball am Gegner vorbei zu schlagen. Jeder erfolgreiche Versuch bringt einen Punkt.



Aufgabe: Entwerft im Team eine Pong-Umsetzung mit Zweispielermodus und Punktestandsanzeige.

Zeit: 90 Minuten

M4: Anleitung



Alle Foliensätze und sonstige Materialien werden als separate Dokumente und Dateien digital bereitgestellt.

1.3.5 Werkmaterialien

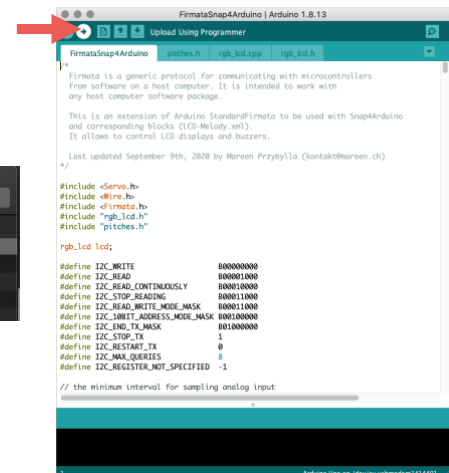
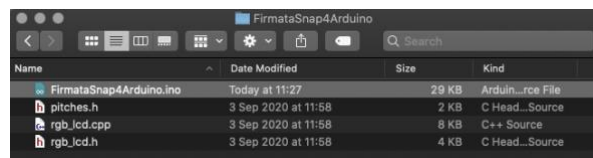
Neben den Arbeitsmaterialien wird auch Bastel- und Modellbaumaterial benötigt, um den kreativen Aspekt auch in der Gestaltung der äußeren Hüllen der interaktiven Objekte zu betonen. Viele Projekte lassen sich mit Pappschachteln, Styropor, Balsaholz und ähnlich leichten Materialien erstellen, mit denen es sich einfach arbeiten lässt und die keine Spezialwerkzeuge erfordern. Außerdem sollten Materialien zum Herstellen von Gelenken und Scharnieren bereitgestellt werden, hierfür eignen sich beispielsweise Bauteile aus Metallbaukästen mit Schrauben und Muttern. Als Befestigungsmittel sind Modellbaukleber, Heißkleber, Panzertape, doppelseitiges Klebeband, Isolierklebeband, Draht und Schnur gut zu gebrauchen. Oft benötigte Werkzeuge sind Scheren, Cutter- oder Taschenmesser, Bohrer, Schraubenzieher, Zangen und Seitenschneider. Zur Dekoration der Projekte eignen sich beispielsweise farbige Stoffe, Malfarben, Federn, Buntpapier und vieles mehr. Hier ist die Kreativität der Lernenden gefragt, die auch eigene Materialien beisteuern können. Als alternativer Ansatz hat sich die Nutzung von LEGO-Bausteine bewährt, die den Vorteil bieten, dass die Jugendlichen problemlos damit umgehen können, kreativ gestalterisch tätig werden und nur wenige weitere Hilfsmittel benötigen. Für die Ausstellung des Gesamtprojektes

empfiehlt es sich, Tische mit beschreibbarer Folie auszulegen, auf der die Schülerinnen und Schüler den Grundriss ihrer Projekte planen können oder große Pappen oder Modellbauplatten als Unterlage zu nutzen. Sollte der Platz es hergeben, kann die Platte während der gesamten Projektarbeitsphase im Raum stehenbleiben und man kann dem Projekt beim Wachsen zusehen. Mit Zeitrafferaufnahmen davon lassen sich schöne Dokumentationsvideos erstellen.



1.3.6 Vorbereitungen:

- Arduino-Uno-kompatiblen Mikrocontroller mit modifizierter Firmata bespielen: Datei FirmataSnap4Arduino.ino in Arduino-IDE öffnen und auf das Board hochladen



- In Snap4Arduino Bibliothek für LCD-Ansteuerung laden: LCD-Buzzer.xml per Drag&Drop auf den Skriptbereich ziehen

