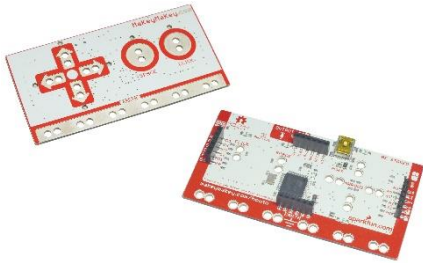
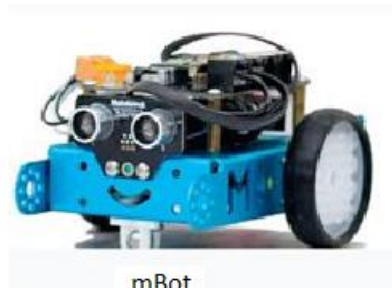


Überlegungen

zu einer
methodisch-didaktisch wohldurchdachten, standortbezogenen, zielführenden und kindgemäßen
Digitalen Grundbildung



Makey Makey



mBot



Meowbit



Ozobot



Calliope mini



Raspberry Pi



BBC Micro:bit



Codey Rocky



RCX



micro:Maqueen



Codeybot



BrainPad Arcade



Jewelbot

Didaktisches Konzept

1. Fehlentwicklung des Informatikunterrichts - Informatikunterricht ist keine Fahrschule für den Computer [1]

Unsere Verbindliche Übung "Digitale Grundbildung" soll einen Weg aufzeigen, wie **tradiierter, ECDL-gefügiger** „Informatikunterricht“ überwunden werden kann und zu **Sach-, Methoden-, Sozial-, Selbst- und Medienkompetenz** führen kann.

Eine Didaktik der Schulinformatik beschäftigt sich mit der Frage, wie der **kreative Kern des Fachs Informatik** vermittelt werden kann. Der Informatikdidaktiker Roland Mittermeier meint: "Wir brauchen mehr Konzepte, die die **kreativitätsfördernden Inhalte des Fachs Informatik** in den Vordergrund rücken - die also auf das so genannte '**computational thinking**' abzielen. Letztlich wird es wohl darum gehen, **IKT-Anwendungskompetenzen mit technischen und kreativen Fähigkeiten zu einem adäquaten Mix für den Informatikunterricht zusammenzuführen.**" [2]

„Informatik ist kein Lern- und Prüfstoff. Informatik ist ein Gebiet das durchwandert und erarbeitet werden will“, so Roland Mittermeier. [3] Mittermeier formuliert weiter, dass jene die nur Anwendung und Skills sehen, den Bildungsauftrag der Schule übersehen. „Schule soll nicht Skill-Training sein. Schule will Bildung vermitteln. Mit ungelernten Office-Hilfsarbeitern und Hilfsarbeiterinnen werden jene Büros, die zwar kurzfristig Bürohilfsarbeiterinnen benötigen, einige Zeit kostengünstig Personal einstellen können. Längerfristig werden österreichische Unternehmen aber weiterbildungsfähige Kollegen und Kolleginnen benötigen. Dazu müssen diese aber mehr mitbringen als oberflächliches Produktwissen.“ [4]

Die Europäische Union empfahl 2006 acht Schlüsselkompetenzen zum lebensbegleitenden Lernen, eine dieser acht Schlüsselkompetenzen ist die Computerkompetenz. Sie umfasst die **sichere und kritische Anwendung der Technologien der Informationsgesellschaft** (TIG) und daher die **Beherrschung der Informations- und Kommunikationstechnologien** (IKT).

Diese so definierte Schlüsselkompetenz ließ und lässt die Schulverantwortlichen der österreichischen Pflichtschule einen nicht sehr effizienten Weg von Informatikunterricht gehen. Der derzeit noch immer in der Pflichtschule **vorherrschend propagierte Bildungswert der Informatik**, nämlich das **Benutzenkönnen von Software** (*Computer Literacy, also Word, Excel, Powerpoint etc*) ist zu hinterfragen. Wenn wir die äußerst akute Kurzlebigkeit von Hardware und Software betrachten, erscheint das sich Beschränken auf das Benutzenkönnen von Software äußerst bedenklich. Zukunftssicherheit der Kenntnisse ist also garantiert nicht garantiert. "Es ist nicht erkennbar, welche grundlegenden und prinzipiell wichtigen Inhalte und Fähigkeiten damit gelernt werden können, worin also letztlich der 'Bildungswert' dieser Systeme (der oben erwähnten Software) besteht." [5] **Dieses oberflächliche und kurzlebige Produktwissen über Computer Literacy liegt höchstens auf der Ebene des Erlernens des Maschinschreibens und vermittelt fast kein langlebiges zukunftssicheres Wissen.**

Vielmehr sollen die SchülerInnen vor allem die Möglichkeit bekommen, in altersadäquater Form subjektiv relevante Problemstellungen zu definieren, zu bearbeiten und ihren Erfolg dabei zu kontrollieren, d.h. im Sinne konstruktivistischer Pädagogik die Überprüfung der Brauchbarkeit der Lösung ihrer Konstrukte durchzuführen.

2. Computational Thinking - eine grundlegende Fähigkeit für das 21. Jahrhunderts

In den USA denkt man derzeit laut über „Computational Thinking“ („Informatisches Denken“) im Rahmen der Allgemeinbildung nach.

"Computational Thinking" entwickelt eine Vielzahl von Fähigkeiten (Logik, Kreativität, algorithmisches Denken, Gliederung komplexer Probleme in leichter lösbare Teilprobleme, Modellbildung), beinhaltet die Verwendung von wissenschaftlichen Methoden und hilft sowohl Erfindergeist als auch innovatives Denken zu entwickeln. Es hat seine Wurzeln in der Mathematik, in Technik, Technologie und Wissenschaft, ist letzten Endes eine Synthese von Ideen aus all diesen Bereichen.

Jeanette M. Wing hat den Begriff „Computational Thinking“ geprägt [6]. Sie geht davon aus, dass Computational Thinking eine fundamentale Fertigkeit werden wird, die von jedem Menschen in der Mitte des 21. Jahrhundert genutzt werden wird. Wer analytisch und präzise wie ein Computerwissenschaftler denken lernt, kann komplexe Probleme effizienter lösen. Von einem solchen mentalen Hilfsmittel wie dem "Computational Thinking" können Menschen in allen Lebensbereichen profitieren. Wichtig dabei sind die informatischen Konzepte, also jene Inhalte der Informatik, die zum Unterschied vom nicht zukunftsicheren Beherrschen von Informatikprodukten langfristige Gültigkeit und Bedeutung haben.

Solche informatischen Konzepte machen einen Informatikunterricht sehr attraktiv, da sie für das Leben in unserer Gesellschaft relevant sind, sie mit der Zeit nicht an Bedeutung verlieren, sie für alle Altersstufen verständlich gemacht werden können und auch in anderen Gegenständen angewendet werden können. [7] Solche informatischen Konzepte sind nach Andreas Schwill [8] Algorithmmierung (Entwurfspadigmen, Programmierkonzepte, Ablauf, Evaluation), strukturierte Zerlegung (Modularisierung, Hierarchisierung) und Sprache (Syntax und Semantik).

Diese informatischen Konzepte müssen natürlich für unsere Schüler altersgerecht und kindgemäß aufbereitet werden. Spezielle Software hat diese Aufgabe der altersgerechten und kindgemäßen Aufbereitung bereits übernommen und ausgeführt. (z. B. Micro Worlds, Turtle Art, Scratch, ByoB)

Kompetenzen entwickelnder Unterricht ist derzeit das Um und Auf der österreichischen Pflichtschule. Ein Kompetenzen entwickelnder Informatikunterricht, der Schlüsselkompetenzen wie beispielsweise Denken in komplexen Zusammenhängen, Problemlösen, Abstrahieren, Analysieren, Zielstrebigkeit aber auch Teamfähigkeit aufbauen soll, wird besonders den informatischen Disziplinen "Programmieren" und "Robotik" aber auch "Physical Computing" breiten Raum geben müssen und das Benutzenkönnen nicht zukunftsicherer Anwendersoftware recht deutlich in den Hintergrund rücken müssen.

3. Programmieren, Robotik, Physical Computing

Programmieren ("Coding") als Basisteil der Informatik wird in mehreren Ländern immer mehr ein obligatorischer Teil des Informatikunterrichts. Der Tradition der osteuropäischen Länder und

Russlands schlossen sich mehrere asiatische Länder an, und in den USA überlegt man laut über „Computational Thinking“ im Rahmen der Allgemeinbildung nach. Diese Art von Denken ist für die Forschung und Entwicklung in den naturwissenschaftlichen und technischen Disziplinen unvermeidbar.

Programmieren in der Schule fördert die Bildung der Kinder auf vielen Ebenen. Die Schule soll zur Entwicklung zu eigenständigen Persönlichkeiten beitragen und zu lebenslangem Lernen befähigen. Dabei steht nicht umfangreiches Faktenwissen im Mittelpunkt, sondern der Aufbau von Schlüsselkompetenzen wie beispielsweise Problemlösen, Abstrahieren, Analysieren oder im Team arbeiten.

Durch das Programmieren erhalten Kinder nicht nur Einblicke in die Geheimnisse moderner Technologien, sie können so auch diese Schlüsselkompetenzen aufbauen. Der Programmierunterricht fördert diese Kompetenzen auf spielerische Art.

Natürlich wollen wir keine Programmierer ausbilden. Aber: Wir lernen z.B. ja auch Musik in der Schule, aber fast keiner wird professioneller Musiker. Programmieren bedeutet vor allem Schulung für das logische Denken. Auch die Kreativität wird gefordert und gefördert.

Es ist derzeit (leider noch nicht in Österreich) eine "Renaissance der Schildkröte" zu beobachten, also jener schon in den 60er Jahren des 20. Jahrhunderts entwickelten Computer-Schildkröte, die zeichnen kann, wenn man ihr nur die richtigen Befehle gibt. Programmiersprachen wie FMS-LOGO, MicroWorlds, Python, Scratch, Turtle Art, SmallBasic, RoamerWorld, Tynker sind heute ganz hoch im Kurs und bieten solch eine **Schildkröte** oder ein **ähnliches Reptil** an, mit der man das Programmieren erlernen kann.

Manche dieser Programmiersprachen gehen weit über das Zeichnen mit der Schildkröte ("Turtlegrafik") hinaus und wurden konsequent um vor allem multimediale Komponenten wie Import oder auch Erstellung von Sound, Grafik, Video und Animationen erweitert. Der Erstellung von Multimediaprojekten, -spielen und -simulationen steht somit nichts mehr im Wege. Auch die Kindgemäßheit bei der Handhabung dieser Programmiersprachen wurde erheblich weiterbetrieben. Manche haben auch das Internet erobert, man kann mit ihnen HTML-Seiten erstellen, die mit diesen modernsten Versionen kreierte interaktiven Projekte können mit Hilfe von Plugins lauffähig im Internet veröffentlicht werden.

Der programmierende Schüler wird somit vom passiven Web-Benutzer zum aktiven Web-Designer.

Manche dieser Programmiersprachen haben sich auch verstärkt den Kapiteln Robotik und Physical Computing zugewandt, Kapitel, die in der österreichischen Pflichtschule noch wenig bis keinen Stellenwert haben.

Denn die Verwendung eines **Microcomputers** und die Möglichkeit ihn zu **programmieren** eröffnet neue Möglichkeiten im Informatikunterricht.

Microcomputer sind selbständig in der Lage, Daten zu koordinieren und auszuwerten. Wir brauchen dazu neben dem Microcomputer einen Computer und eine geeignete Programmiersprache. Die mit

dieser Programmiersprache entwickelten Programme werden dann über eine Infrarotschnittstelle oder über Bluetooth oder WLAN in Form von Befehlen an den Microcomputer weitergeleitet.

Man kann mit Hilfe dieser geeigneten Programmiersprache den Microcomputer in der Art programmieren, dass dieser unter Verwendung unterschiedlicher Sensoren (wie etwa Licht-, Wärme, Bewegungs- und Berührungssensoren) seine Umgebung passiv beobachtet und physisch wahrnehmbare Aktionen registriert und mittels sogenannter Aktoren (wie Motoren, Lampen oder Tönen) Reaktionen und Aktionen auslöst und somit seine Umwelt verändert. So ist es möglich, einen beweglichen, selbstgesteuerten Roboter zu entwerfen.

Eingebettet wird die Nutzung der Microcomputer in eine handlungs- und problemorientierte Lernumgebung, in der das "Lernen mit allen Sinnen" praktiziert und eine positive emotionale Bindung an den Informatikunterricht gefördert wird. In solch einem Informatikunterricht besteht die Verzahnung von Theorie und Praxis auf besondere Weise: Einen Roboter kann man in die Hand nehmen und seine Fähigkeiten bereits erahnen, wenn man ihn "morphologisch" untersucht: Ein Roboter mit "Füßen" oder Rädern kann sich potenziell fortbewegen. Ein Roboter mit Tastsensoren kann registrieren, wenn er an ein Hindernis stößt. Es existiert somit eine problemhaltige Ausgangssituation und damit ein guter Einstieg für genetisches, entdeckendes, forschendes und selbständiges Lernen.

Roboter bieten eine gute Grundlage, um Schülerinnen und Schüler in einem handlungsorientierten Unterricht an das technikbasierte Thema Informatik heran zu führen. Dies gilt für alle Altersgruppen und jede Vorbildung der Schülerinnen und Schüler. Beim Entwerfen, Konstruieren, Programmieren und Testen von mobilen autonomen Robotern erfahren sie, dass Technik Spaß macht, lernen wie technische Systeme entwickelt werden und erwerben Fachkenntnisse. Der Bau und die Verwendung von Robotern vereint in idealer Weise viele Elemente technischen Wissens, die für ein Verständnis technologischer Probleme bis hin zu philosophischen Fragen – wie etwa der der Intelligenz und Autonomie von Gegenständen, die ihre Form und Funktion durch menschliche Einwirkung und Interpretation erhielten – hilfreich sind.

Roboter bieten einen auf Technik bezogenen Zugang zu Informationstechnik. Neben vielen fachlichen Kenntnissen und Methodenwissen werden vor allem nicht-fachliche Kompetenzen, sogenannte Soft Skills wie zum Beispiel Teamfähigkeit, gefördert und Interaktion, Kommunikation, Präsentation und Dokumentation geübt. Die Schülerinnen und Schüler erledigen die Arbeitsaufträge in Gruppenarbeit mit 2-3 Lernenden pro Gruppe. Idealerweise verfügt dabei jede Gruppe über einen eigenen Roboter-Bausatz.

Der **Raspberry Pi** wiederum macht es uns sehr einfach, in die Welt des „Physical Computing“ einzusteigen, indem man eigene Projektideen mit Hilfe von computergesteuerten Sensoren, Tasten oder Leuchtdioden realisiert.

Er besitzt nämlich eine so genannten „General Purpose Input / Output“- Schnittstelle (GPIO), mit deren Hilfe sich weitere elektronische Bauteile an den Raspberry Pi anschließen und steuern lassen.

Add-on-Boards für den Raspberry Pi wie die **LedBorg-Platine** oder auch **Pibrella**, das **Makey Makey** oder das **PicoBoard** ermöglichen zusammen mit der Programmiersprache Scratch recht interessante ja manchmal sogar recht verrückte Projekte.

Neuere Boards wie der **CodeBug**, der **BBC Micro:bit**, der **Calliope mini**, das **Sense Hat** oder das **Pixel Kit** müssen erst auf ihre Brauchbarkeit für den Informatikunterricht in der Sekundarstufe 1 getestet werden.

4. Roboter mit Smartphones oder Tablets steuern

*Roboter der neuesten Generation können neben dem Computer auch bereits mit Tablet oder Smartphone programmiert und gesteuert werden. Die etwas kostspieligen Roboter **mBot**, **Ozobot**, **Sphero**, **Ollie**, **BB-8**, **Dash and Dot**, **Cubelets** und die österreichische Entwicklung **Robo Wunderkind** können mit speziellen Apps über Smartphone oder Tablet programmiert und gesteuert werden.*

5. Making - kreatives digitales Gestalten und Experimentieren [9]

Making, das kreative Gestalten und Selbermachen insbesondere mit digitalen Technologien, soll in den Schulalltag auch der Pflichtschule Einzug halten. Die Schüler sollen zu „Makern“ werden. Technisches Verständnis, kreatives Problemlösen, soziales Miteinander und handwerkliches Geschick werden dabei trainiert.

Makers sind Menschen, die basteln, etwas Neues gestalten. Sie verwenden dazu einerseits digitale Technologien wie z.B. eine Programmiersprache, andererseits Materialien, mit denen sie experimentieren und plötzlich etwas ganz Interessantes aber auch manchmal etwas ganz Verrücktes herstellen. Makers wissen, dass z.B. ein Bleistift nicht nur zum Schreiben zu gebrauchen ist, oder dass man leere Küchenrollen für eine interessante Musikinstallation verwenden kann.

Weil es darum geht, dass etwas „gemacht“ (engl. „to make“) wird, also etwas Neues entwickelt und produziert wird, wird diese Entwicklung des „digitalen Do-It-Yourself“ auch als Maker-Bewegung bezeichnet.

Wenn man davon ausgeht, das Making in den Schulen Einzug halten soll und muss, dann fällt das zu großen Teilen in den Verantwortungsbereich des Informatikunterrichtes. Auch das Technische Werken wird hier einen großen Beitrag leisten müssen. Die Abschaffung des Technischen Werkens in unseren 8. Schulstufen ist somit zu überdenken.

Making ist nicht nur für den Informatikunterricht sondern für alle anderen MINT-Fächer interessant. Aber es werden in den Making-Projekten nicht nur MINT-Interessen geweckt und MINT-Kompetenzen entwickelt, sondern auch Soft Skills wie Teamfähigkeit, analytische Kompetenz, Problemlösekompetenz, Kreativität, eigenständiges Arbeiten, Flexibilität und vor allem Neugierde bei der Suche nach Lösungen.

Making mit Schülern stellt das aktive Handeln und die damit verbundenen Wissenszuwächse in den Mittelpunkt und fördert diese im späteren Berufsleben unserer Schüler von ihnen erwarteten Soft Skills.

Making baut auf den Ideen des Konstruktivismus des Mathematikers und Vater der Computerpädagogik Seymour Papert auf. Der Erfinder der Programmiersprache LOGO geht davon aus, dass Menschen besonders dann etwas lernen, wenn sie etwas selbst konstruieren und sich im Konstruktionsprozess das dafür nötige Wissen aneignen und anwenden. Durch das eigene aktive Handeln werden schwierige Inhalte leichter durchschaubar und begreifbar.

Literatur

[1] Alpen-Adria-Universität Klagenfurt: Uni News

[Http://www.uni-klu.ac.at/main/inhalt/41394_41280.htm](http://www.uni-klu.ac.at/main/inhalt/41394_41280.htm)

[2] ebenda

[3] Roland Mittermeir in Reiter, Scheidl, Strohmmer, Tittler, Weissenböck (Hrsg.): Schulinformatik in Österreich. Erfahrungen und Beispiele aus dem Unterricht. Ueberreuter Wien. 2003. S. 10

[4] ebenda, S. 11

[5] H. LÖTHER, in: Mindstorms: Kinder, Computer und Neues Lernen

Birkhäuser, Basel 1982

[6] Jeanette M. Wing: Computational Thinking. In: CACM Viewpoint, March 2006

[7] Gerald Futschek: Österreich braucht gut ausgebildete InformatiklehrerInnen

http://publik.tuwien.ac.at/files/PubDat_200228.pdf

[8] Andreas Schwill: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25 (1993) Nr. 1, S. 20-31.

[9] Sandra Schön, Martin Ebner und Kristin Narr (Hrsg.): Making-Aktivitäten mit Kindern und Jugendlichen. Handbuch zum kreativen digitalen Gestalten, 2016