

Selbermachen ist Trumpf

Hilfen zum spielerischen Einstieg in die Welt der Programmierung

Was ist schon das bloße Daddeln am Smartphone gegen das schöpferische Erlebnis, selbst eine App zu gestalten? Im Rahmen des Vintage Computing Festival Berlin befasste die Tagung „Kids and Codes“ sich mit Ansätzen, Schülern Grundgedanken des Programmierens nahezubringen.

Von Pit Noack

Von der Digitalisierung des Alltags, der Berufswelt und der Industrie ist so oft die Rede, dass mancher solche Begriffe schon nicht mehr hören mag. Wo aber sollen in den nächsten Jahren diejenigen herkommen, die Systeme und Anwendungen gestalten und neue Paradigmen in der IT entwickeln?

Bereits zur Eröffnung der diesjährigen CeBIT machte Bundeskanzlerin Angela Merkel die Notwendigkeit deutlich: „Ich glaube, dass die Fähigkeit zum Programmieren eine der Basisfähigkeiten von jungen Menschen wird, neben Lesen, Schreiben, Rechnen.“ Die Informatikdidaktiker, die sich im Oktober zur Kurztagung „Kids and Codes“ im Deutschen Technikmuseum Berlin versammelten, befassten sich mit Ideen, die jungen „Digital Natives“ den Schritt zum Selbermachen erleichtern können.

Warum Programmierung?

Michael Rücker von der Humboldt-Universität zeigte, dass Kinder und Jugendliche auch über den eigentlichen Lernzweck hinaus profitieren, wenn sie Programmierkenntnisse erwerben. Die dabei

gewonnenen Fertigkeiten im logischen, strukturierten und abstrakten Denken sind auch auf nichtinformatische Bereiche übertragbar. Zudem gewinnt man ein besseres Verständnis einer Welt, die von allgegenwärtiger Digitalisierung geprägt ist. Wer programmieren lernt, gelangt leichter zu einem sicherheitsbewussten und reflektierten Umgang mit Digitaltechnik.

Im Rahmen seiner Dissertation hat Rücker Schülern Abbildungen von unterschiedlichen technischen Geräten vorgelegt. Er bat darum, die Objekte nacheinander den Kategorien „Computer“, „Programmierbar“ und „Hat was mit Informatik zu tun“ zuzuordnen. Anschließend sollten die Probanden ihre Entscheidungen begründen. Dabei kamen ziemlich eigentümliche und diffuse Vorstellungen an den Tag.

Rücker stellte fest, dass die Schüler die technischen Fähigkeiten von Geräten auf Grundlage der primären, äußerlich wahrnehmbaren Funktion bewerteten: „Eine Waschmaschine wäscht Wäsche, eine Ampel leuchtet grün, gelb oder rot, Indus-



Ein Laptop aus Papier braucht keinen Strom, kann aber beim Vermitteln von Grundeinsichten zu binärer Zeichenkodierung hilfreich sein.

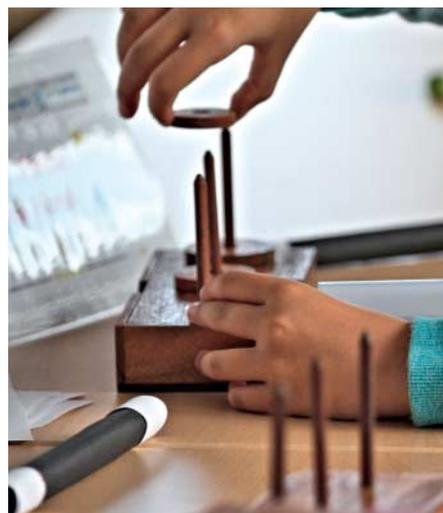


Bild: InfoSphere-Schülerlabor

Unplugged Computing: Die Türme von Hanoi verdeutlichen die Stack-Logik.

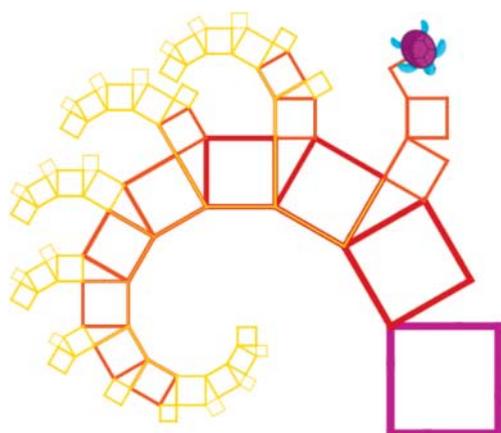
trieroboter machen immer nur dasselbe.“ Über weitere, nicht direkt wahrnehmbare Funktionen wurde in der Regel nicht einmal spekuliert. Rücker nannte es eine zentrale Aufgabe des Informatikunterrichts, „die Unabhängigkeit der äußeren Erscheinung eines Geräts von dessen digitalen Fähigkeiten erfahrbar und begreifbar zu machen.“ Denn „smarte, vernetzte“ und damit potenziell sicherheitskritische Funktionen arbeiten oft im Verborgenen.

Informatische Sphären

Nadine Bergner leitet an der RWTH Aachen das „Schülerlabor Informatik – InfoSphere“. Dort werden altersspezifische Module zur Vermittlung eines fundierten Umgangs mit Digitaltechnik entwickelt und mit besuchenden Klassen praktisch durchgeführt. Bergner kritisiert, dass viele Bundesländer Informatikunterricht erst ab der Mittelstufe und teilweise nur im Wahlbereich anbieten. Das dreistündige Modul „Zauberschule Informatik“ wendet sich bereits an Grundschüler. Es verfolgt den Ansatz des „Unplugged Computing“ – hierbei wird etwa die binäre Kodierung von Zahlen und Bildern oder die bitweise Datenübertragung zunächst ohne Computer spielerisch mit Murmeln, Stiften und Papier erprobt. Dieses Vorgehen kommt laut Bergner bei den Kindern gut an.

Fortgeschrittene Module demonstrieren die Funktionsweise des Internet mittels eines Brettspiels oder führen in die App-Programmierung mit dem App Inventor ein. Die Entwicklung von vorzeigbaren Ergebnissen wie selbst program-

Bild: InfoSphere-Schülerlabor



TurtleCoder: Die emsige Schildkröte zeichnet mittels Rekursion einen pythagoräischen Baum.

mierten Apps und die Einbindung von Eltern – „Mensch, Papa, wieso verstehst du das denn nicht?“ – nannte Bergner wichtige Beiträge zur Steigerung der Nachhaltigkeit von Informatik-Lehrprojekten. Für den Einsatz von Sensorik und Steuerung in den unteren Klassenstufen empfiehlt sie Plattformen wie Calliope oder Makey Makey, die bereits die dafür notwendige Grundausstattung mitbringen, denn: „Wenn es am Ende nicht läuft, liegt es oft an der Verkabelung.“

Schildkrötenprogrammierung

Thomas Schmidt von der Initiative „Code your Life“ (einer Kooperation mit Microsoft YouthSpark) befasst sich ähnlich wie Bergner mit der Entwicklung von Unterrichtskonzepten. Diese wenden sich vorwiegend an Mädchen und Jungen im Alter von 10 bis 14 Jahren.

Ein zentrales Element ist die Browser-Anwendung TurtleCoder, ein optisch ansprechendes und funktional überzeugendes Remake von LOGO: Gesteuert von Funktionsaufrufen für Drehung, Vorwärts- und Rückwärtsbewegungen zeichnet eine Schildkröte Linien auf den Bildschirm. Durch Verwendung von Schleifen,

selbst definierten Funktionen und rekursiven Funktionsaufrufen lassen sich zunehmend komplexe Strukturen auf den Bildschirm bringen. Außer in einer übersichtlichen Darstellung mit farbig markierten Code-Elementen lassen sich die Programme auch als TypeScript-Code anzeigen. Man kann sie zudem auf den mit einer Stifthalterung versehenen Miniroboter „Mirobot“ transferieren, der die gemalten Muster direkt auf Papier überträgt. Eine Schnittstelle für Stickmaschinen etwa zur Gestaltung von T-Shirts ist in Arbeit.

Rote Fäden

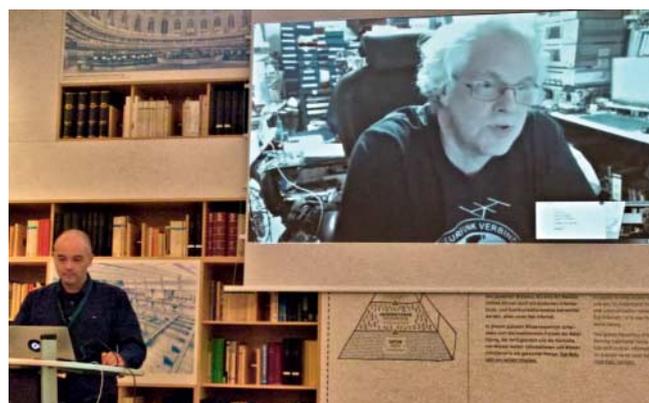
Einigkeit herrschte unter den Tagungsteilnehmern in dem Punkt, dass angesichts der gesellschaftlichen Relevanz der Digitaltechnik mehr Vermittlungsarbeit an den Schulen geleistet werden müsse. Neben der öffentlich immer wieder geforderten besseren technischen Ausstattung sind vor allem mehr und besser ausgebildete Lehrkräfte erforderlich. Außerdem ist es dringend nötig, tragfähige Unterrichtskonzepte zu entwickeln.

Informatikunterricht muss nicht unbedingt mit Programmierübungen am PC beginnen. Reflexionen über die Präsenz von Digitalität und spielerisches „Unplugged-Computing“ können durchaus solide Fundamente für ein tiefgehendes Verständnis schaffen. Wenn es um algorithmische Steuerung geht, braucht man nicht sofort von autonomen Fahrzeugen oder Bank-Scoring zu reden. Im Unterricht empfiehlt es sich vielmehr, Bezüge zu Erfahrungen zu nutzen, die den Schülern präsent sind. Eine algorithmische Bearbeitung von Aufgabenstellungen findet sich etwa schon bei Bastelanleitungen, Kochrezepten und bei der schriftlichen Multiplikation im Matheunterricht.

Mitschnitte der gelungenen und anregenden Tagung sollen bis zum Jahresende online zur Verfügung stehen – das haben die Veranstalter in Aussicht gestellt. (psz@ct.de) **ct**

Material und Konzepte: ct.de/yfhr

Anzeige



Stefan Höltgen von der Humboldt-Uni (links) hatte per Skype den Hardware-Didaktiker Rolf-Dieter Klein hinzugeschaltet, der das modulare Konzept seines 1984 als Selbstbauprojekt erschienenen NDR-Klein-Computers (NKC) erläuterte.