

# Programmieren als dritte Fremdsprache



**Lehrpläne sollen die Bedeutung der digitalen Welt berücksichtigen, fordert Repenning**

Alexander Repenning ist beseelt von einem ansteckenden Enthusiasmus. Wenn er spricht, schlägt gelegentlich ein leicht amerikanischer «twang» durch, ein Echo der gut zwei Dekaden, die er an der University of Colorado in Boulder zuge-

bracht hat. In dieser Zeit hat er die preisgekrönte Programmiersprache entwickelt, mit der selbst Laien Videospiele programmieren können. Bereits Mitte der 1990er-Jahre war er zusammen mit dem renommierten Massachusetts Institute of Technology (MIT) in Boston an der Entwicklung der ersten visuellen Programmiersprache Lego-Sheets beteiligt, aus der später die erfolgreiche Lego-Mindstorms-Serie hervor ging.

Seit über einem Jahr leitet der gebürtige Schweizer die informatische Bildung an der Pädagogischen Hochschule der Fachhochschule Nordwestschweiz (FHNW) in Brugg-Windisch AG und möchte den Primarschülerinnen und -schülern das Programmieren beibringen.

Warum? – «Wir müssen von einem Schulsystem, das Auswendiglernen lehrt, zu einem kommen, das Denken vermittelt», sagt der Informatik-

professor, der mehr an einen Marathonläufer erinnert. «Kreativität ist der Rohstoff der Schweiz, nicht die Reproduktion von Wissen.»

Die Kinder sollen keine Memorysticks sein, fügt Alexander Repenning eine Metapher aus seiner Domäne an. Mit der fortschreitenden Digitalisierung und der fast überall verfügbaren Information, die auf dem Internet zum Abruf rund um die Uhr bereit steht, sollte sich auch das schulische Lernen verändern. Etwas, das mit dem Lehrplan 21 (LP21) in mancher Hinsicht angegangen wurde, aber sich in der Umsetzung als äusserst tückisch erweist.

Repenning mag nicht warten, bis die föderalistischen Mühlen gemahlen haben. Der Berater der US-amerikanischen National Academy of Science befürchtet, dass die Schweiz international den Anschluss verpasst, und weiss aus Erfahrung, wie hartnäckig sich Vorurteile halten. «Als wir vor gut 20 Jahren ein Mädchen in den USA fragten, was sie von Informatik halte, sagte sie: «Programmieren ist schwierig und langweilig.»»

### **Repräsentation einer Lösung nach dem Wenn-dann-Ansatz**

Diese nicht zu unterschätzende Hürde zu überwinden, hat sich Alexander Repenning zum Ziel gesetzt. Er avisiert mit seinem zugänglichen Programmiersystem «Scalable Game Design» in erster Linie Lehrpersonen, die kein Informatikwissen mitbringen – also die Mehrheit: «In einem Workshop zeigen wir Lehrpersonen, wie sie selber ein Videospiele entwerfen und umsetzen können», erklärt der Informatiker. Mithilfe der ebenfalls von ihm entwickelten Programme Agentsheets und Agentcubes können Laien tatsächlich binnen weniger Stunden Programmieren lernen\*, indem sie ein simples Videospiele nach dem Modell des Klassikers «Frogger» kreieren.

Entscheidend dabei ist, dass sie ihre eigenen Figuren entwerfen und selbst dreidimensional darstellen können: «Solche Erfahrungen sind ungeheuer motivierend. Ein entsprechend hoher Prozentsatz ist am Ende überzeugt und möchte weitermachen.» Weitermachen bedeutet, das Gelernte an die Schülerinnen und Schüler weiterzugeben.

Doch wie so oft ist der erste Schritt der schwierigste: die Leute dazu zu bringen, etwas Neues zu wagen und ihre Ängste abzulegen. Denn wie eine Umfrage von Google kürzlich zeigte, wird Programmieren auch heute noch für «schwierig» und «langweilig» gehalten. Kommen dann noch Ausdrücke wie «computational thinking» oder Simulation hinzu, dann winken Skeptiker erst recht ab.

Doch was hat es damit auf sich? Was bringt es, die Simulation einer Schlammlawine zu programmieren statt einfach eine solche zu beschreiben? – Beim «computational thinking» müsse man als Erstes das Problem formulieren, sagt Repenning. Dann folgt die Repräsentation einer Lösung nach dem Wenn-dann-Ansatz. Man stellt eine simple Regel auf oder anders gesagt, bricht das Problem auf das Verständnis eines Computers herunter, der dann ein Modell erstellt. «Diese Visualisierung zeigt, ob man die Problemstellung richtig erfasst hat», sagt Repenning.

Und was passiert, wenn zum Beispiel die Schwerkraft als Faktor vergessen wurde? Dann schweben die Schlammteile in der Luft, statt den Hang hinunterzurutschen, und man muss in einem nächsten Schritt die Simulation verbessern und nähert sich so der richtigen Lösung. Der Computer helfe einem dabei, die eigenen Gedanken und damit Ideen umzusetzen, sagt Repenning und vor allem: «Er nimmt einem nichts weg.»

### **Eine Gitarre aus Karton und Folie an den PC angeschlossen**

Viele Lehrpersonen befürchten, dass das Programmierenlernen und damit verbunden das «computational thinking», beispielsweise das Herunterbrechen einer Problemstellung auf einfache Wenn-dann-Regeln, auf Kosten anderer Fächer gehe.

«Am einfachsten ist der Transfer in naturwissenschaftlichen Fächern, aber auch beim Werken ist die Verbindung zum Computer möglich«, sagt Repenning und zeigt das Beispiel eines Schülers, der aus Karton und etwas Alufolie eine Gitarre gebastelt hat, die er über eine simple Schnittstelle mit dem PC verbunden hat. «Wenn man Regeln für ein Computerspiel aufstellt, so kann dies Teil des Sprachunterrichts sein», findet der Hasler Professor für Informatische Bildung an der Pädagogischen Hoch-

schule der FHNW.

Auf der Primarstufe gibt es praktisch keine Fachlehrer, was dem fachübergreifenden Konzept des «computational thinking» entspricht. Denn dabei geht es prinzipiell um analytisches Denken und Lösungsfindung mithilfe des Computers, was in allen Fächern einsetzbar ist.

Im vergangenen Dezember konnten Repenning und sein Team im Rahmen der weltweiten Computer Science Education Week mit der ersten Swiss Computer Science Education Week über 10 000 Schülerinnen und Schüler in 168 Gemeinden sowie deren Lehrpersonen für gemeinsames Programmieren begeistern. Auch in diesem Dezember wird es wieder so weit sein, doch bis dahin wird Alexander Repenning weitere Skeptiker für die Sache gewinnen, denn er weiss, ohne Informatik läuft nichts.

\* Mit der von Alexander Repenning entwickelten Software Agentcubes kann jeder selbst 3-D-Videospiele programmieren: <http://www.agentcubesonline.com>

Marc Bodmer (Text) und Michele Limina (Foto)