

Entdeckendes Lernen im Informatik-Unterricht

Ruedi Arnold	Werner Hartmann	Raimond Reichert
Dept. Informatik, ETH Zürich	infoSense	SCIL, Universität St. Gallen
CH-8092 Zürich	CH-5430 Wettingen	CH-9000 St. Gallen
rarnold@inf.ethz.ch	hartmann@infosense.ch	raimond.reichert@unisg.ch

Abstract: Informatik-Unterricht ist häufig geprägt durch ein Vermitteln von Theorie mit anschliessenden Übungen. Wichtige Aspekte wie etwa selbstständiges Arbeiten, Kreativität und kritische Reflexion werden dabei zu wenig berücksichtigt. Eine Alternative stellt die Unterrichtsmethode des Entdeckenden Lernen dar. Allerdings gibt es bisher wenig Publikationen, die sich mit dem Thema des Entdeckenden Lernen im Informatik-Unterricht beschäftigen. Wir skizzieren in diesem Artikel die Methode des Entdeckenden Lernen allgemein und zeigen anhand von drei konkreten und sehr unterschiedlichen Beispielen exemplarisch auf, wie Entdeckendes Lernen im Informatik-Unterricht eingesetzt werden kann. Diese Beispiele umfassen die Themen Gestaltung behindertengerechter Websites, diskrete Mathematik anhand eines Kartentricks sowie die Funktionsweise anspruchsvoller Graphenalgorithmien. Die Beispiele illustrieren, dass Entdeckendes Lernen eine spannende Methode und eine Bereicherung für den Informatik-Unterricht darstellt.

1 Einleitung

Viele Themen der Informatik zeichnen sich durch einen hohen Grad an Abstraktion aus. Das zeigt sich nur schon daran, dass man Bits und Bytes oder Algorithmen und Datenstrukturen nicht „anfassen“ kann. Sichtbar wird immer nur die Spitze des Eisbergs, die Benutzeroberfläche einer konkreten Software. Das abstrakte Wesen der Informatik zeigt sich auch in *Computing as a Discipline* von Denning et. al. [Den89]. Dieser Bericht charakterisiert Informatik als eigenständige Disziplin und betont, dass Informatik eine Technologieorientierte Disziplin sei, die ihre Wurzeln sowohl in der Mathematik als auch im Ingenieurwesen habe. Informatik wird über drei Paradigmen definiert: Theorie, Abstraktion, und Entwurf – zwei dieser drei Paradigmen weisen wiederum auf das abstrakte Wesen der Informatik hin. Die Abstraktion in der Informatik findet zudem häufig auf mehreren, aufeinander aufbauenden Ebenen statt. So wies Knuth darauf hin, Informatiker seien “individuals who can rapidly change levels of abstraction, simultaneously seeing things ‘in the large’ and ‘in the small’”.

Für den Unterricht stellt das abstrakte Wesen der Informatik eine Herausforderung dar: Die Fähigkeit zur Abstraktion müssen sich Schülerinnen und Schüler erarbeiten, insbesondere auch die Fähigkeit, Dinge auf verschiedenen Abstraktionsebenen gleichzeitig zu betrachten. Eine weitere Schwierigkeit für den Unterricht ist, dass Schülerinnen und Schüler Al-

gorithmen wie effizientes Sortieren mit Quicksort oder Heapsort oder Verschlüsselungsverfahren wie RSA oder Diffie-Hellman nicht selber herausfinden können. Es ist schon anspruchsvoll, solche Algorithmen zu verstehen. Diese Schwierigkeiten tragen mit dazu bei, dass Informatik-Unterricht häufig geprägt ist durch ein Vermitteln von Theorie mit anschließenden Übungen.

Auch im Informatik-Unterricht haben aber offene Unterrichtssituationen ihren Platz und geben den Schülerinnen die Gelegenheit, selber aktiv neue Inhalte zu erarbeiten. Eine mögliche Unterrichtsmethode dafür ist das Entdeckende Lernen. Entdeckendes Lernen fördert Kreativität, Selbstständigkeit, kritisches Denken und den gegenseitigen Austausch. Entdeckendes Lernen bedeutet zudem ein hohes Mass an Individualisierung, was es der Lehrerin erlaubt, besser auf die Bedürfnisse der einzelnen Schüler einzugehen. Wir stellen im folgenden kurz den Ansatz des Entdeckenden Lernen vor. Drei konkrete und unterschiedliche Beispiele illustrieren, wie Entdeckendes Lernen im Informatik-Unterricht umgesetzt werden kann. Die Beispiele basieren auf einsatzbereiten Unterrichtsmaterialien, die im Rahmen der Informatik-Lehrerausbildung an der ETH Zürich entwickelt wurden.

2 Entdeckendes Lernen

Die Grundidee beim Entdeckenden Lernen besteht darin, dass die Lernenden persönliche Erfahrungen machen, Dinge hinterfragen und so neues Wissen „herausfinden“. Lernende entwickeln durch Staunen, sich Wundern und Zweifeln ihre eigenen „naiven“ Theorien und müssen dabei mitunter alte Vorstellungen aufgeben und zuvor aufgestellte Hypothesen verwerfen. Für den Lernerfolg ist es oft wirksamer, wenn die Schülerinnen aktiv Erklärungen generieren, um beobachtete Phänomene zu begründen, als wenn sie Merksätze und Theorien vorgetragen bekommen und auswendig lernen müssen.

Entdeckendes Lernen braucht Zeit und Freiräume. Es gibt a priori kein Richtig oder Falsch auf der Suche nach Neuem. Wichtig ist, dass alle Erkenntnisse festgehalten und geordnet werden. Dokumentation und Präsentation von (vorläufigen) Ergebnissen helfen bei der Ordnung der eigenen Gedanken und sind ein essentielles Mittel zur (Weiter-)Entwicklung und kritischen Reflektion eigener Ideen. Piaget bringt es auf den Punkt: „Indem sich das Denken den Dingen anpasst, strukturiert es sich selbst, und indem es sich selbst strukturiert, strukturiert es auch die Dinge.“

Für eine ausführlichere Diskussion der Unterrichtsmethode des Entdeckenden Lernen inklusive pädagogischer und kognitiv-psychologischer Aspekte verweisen wir auf [Ham00] und [Neb73]. Zum Thema Entdeckendes Lernen im Informatik-Unterricht gibt es nur sehr wenig Publikationen. Ein Beispiel dafür ist das Paper *Discovery Learning in Computer Science* von Baldwin, in dem der Autor seine Erfahrungen mit Entdeckendem Lernen in einer Vorlesung zu Computer-Grafik und einer Veranstaltung zu Unix-Programmierung mit C beschreibt [Bal96]. Eine lange Tradition mit entsprechend vielen Publikationen hat Entdeckendes Lernen im naturwissenschaftlichen Unterricht, im englischen Sprachraum als „Scientific Discovery Learning“ bekannt. Stellvertretend sei hier die renommierte Literaturanalyse *Scientific Discovery Learning with Computer Simulations of Conceptual*

Domains von de Jong & van Joolingen erwähnt [dJvJ98]. Der Artikel fasst die typischen Schwierigkeiten von Lernenden beim Entdeckenden Lernen zusammen. Für Informatiker von besonderem Interesse sind die Vorschläge, wie Computer-Simulationen beim Entdeckenden Lernen eingesetzt werden können.

2.1 Themenwahl und Vorbereitung für Entdeckendes Lernen

Der Erfolg von Entdeckendem Lernen hängt vom gewählten Thema und der Aufgabenstellung ab. Die Selektion passender Inhalte obliegt der Lehrperson, genauso wie adäquate Anleitungen. Exemplarisch geben wir hier drei zentrale Hinweise zur Umsetzung im Fach Informatik. Wir stützen uns dabei auf interne Unterlagen und persönliche Kommunikation von und mit Prof. Karl Frey im Rahmen der Lehrerausbildung an der ETH Zürich.

Offenheit des Themas

Das Gebiet muss eine gewisse Offenheit besitzen. Entdeckendes Lernen heisst, ein Thema explorativ erkunden, Hypothesen aufstellen, überprüfen und mit anderen Schülerinnen austauschen. Das Lösen einer vorgegebenen Aufgabe hingegen ist kein Entdeckendes Lernen. Der Lerngegenstand muss mehrschichtig oder vielfältig sein, verschiedene Aspekte umfassen und verschiedene Entdeckungswege zulassen. Wenn das Lernziel klar vorgegeben ist, wird der Inhalt besser direkt vermittelt. Der Fliesskommazahlen-Standard IEEE 754 lässt sich nicht entdecken, genauso wenig wie der Quicksort-Sortieralgorithmus. Jedoch können Lernende sich Gedanken machen, wie sich Fliesskommazahlen im Binärsystem darstellen lassen oder sich eigene Sortierverfahren überlegen.

Die Offenheit einer Aufgabe kann im Wesentlichen auf zwei Arten auftreten. In der einfacheren Form können verschiedene Lösungen eines Problems möglich sein. In der offeneren Form des Entdeckenden Lernen wird von den Lernenden der zu explorierende Aspekt eines Themas selbstständig festgelegt. Im nächsten Kapitel zeigen wir Beispiele für beide Formen von Offenheit.

Vollständig zur Verfügung gestelltes Material

Das notwendige Material wird vollständig von der Lehrperson zur Verfügung gestellt, so aufbereitet, dass alle Lernenden die Informationen vollständig verstehen und verarbeiten können. Sie sollten dabei keine Assistenz durch die Dozentin oder den Lehrer benötigen. Beim Entdeckenden Lernen geht es darum, basierend auf dem vorgegebenen Material und eigenem Vorwissen selbstständig neue Dinge zu entwickeln. Das Ziel ist die Entwicklung eigener Ideen und nicht das Herausfiltern und Strukturieren von bestimmten Informationen, zum Beispiel der korrekten Syntax einer komplexen Anweisung in einer bestimmten Programmiersprache aus zehn dicken Fachbüchern. Die selbstständige Informationsbeschaffung kann durchaus ein Ziel sein, sprengt dann aber schnell den Rahmen von Präsenzveranstaltungen und müsste in einem grösseren Zeitrahmen stattfinden.

Aufgabe und Bewertung: Alle ernsthaften Beiträge honorieren

Aufgabenstellungen zum Entdeckenden Lernen müssen so angelegt sein, dass sie verschiedene Lösungen, Herangehensweisen und Perspektiven zulassen. Sie müssen genau wie das Thema eine gewisse Offenheit besitzen, damit die Lernenden die notwendige Freiheit haben, eigene Entdeckungen zu machen. Bei der Beurteilung von Beiträgen ist wichtig, dass alle Vorschläge und Ideen ernst genommen werden: Jeder ernsthafte Beitrag muss honoriert werden. Als Lehrerin darf man bei halbrichtigen Lösungen nicht sofort intervenieren. Es braucht eine gewisse Freiheit, ein gewisses Mass an „Laisser-faire“.

3 Beispiele für Entdeckendes Lernen im Informatik-Unterricht

Wir zeigen im Folgenden anhand dreier bewusst unterschiedlich gewählter Beispiele, wie Entdeckendes Lernen im Informatik-Unterricht eingesetzt werden kann. Wir gehen dabei insbesondere auf die Offenheit des Themas, auf das zur Verfügung gestellte Material sowie die Bewertung der Schülerbeiträge ein. Die präsentierten Beispiele sind vollständig mit Aufgabenblättern, Begleitmaterial für die Lehrperson sowie der verwendeten Software beziehungsweise Links zu den entsprechenden Programmen frei verfügbar auf dem Bildungsserver SwissEduc [Swi05].

3.1 Gestaltung behindertengerechter Websites

Wie „sieht“ ein Blinder eine Webseite? Auf was muss man achten, wenn man behindertengerechte Websites erstellen will? Können Blinde und Sehbehinderte die vielfältigen Informationen auf dem Internet überhaupt nutzen oder werden sie erst recht benachteiligt? Welche technischen Hilfsmittel stehen Blinden heute zur Verfügung? Wie sieht die Rolle von Blinden in der Informationsgesellschaft generell aus? Alle diese Fragen können auftreten, wenn man selber in die Rolle eines Blinden schlüpft, sich die Augen zubindet und versucht, im Web mit einem Webseiten-Leseprogramm zu navigieren. Hier setzt diese Unterrichtseinheit an: Lernende erleben die Rolle von Blinden mit Hilfe von Programmen, welche die Webseiten analysieren und für Behinderte zugänglich machen.

Die Unterrichtseinheit kann beispielsweise bei einer Anwenderschulung im Bereich Webpublishing eingesetzt werden. Im Anschluss sind die Lernenden sensibilisiert für die Thematik und können fundiert über sinnvolle Richtlinien hinsichtlich Accessibility beim Entwurf von Webseiten argumentieren.

Offenheit des Themas. Das Thema besitzt die notwendige Offenheit, da eine breite Palette von Aspekten entdeckt werden kann. So können beispielsweise praktische Dinge wie „Eine Homepage ohne Text lässt sich nicht vorlesen“ betrachtet werden. Aber auch technische Details wie die Problematik von Flash-Animationen oder Frames können von Interesse sein. Vielleicht erheben die Schülerinnen die Design-Maxime, dass eine Website nur dann eine gute Website ist, wenn auch Blinde und Sehbehinderte Zugang zur

Website finden. Oder sie schlagen die Brücke zu Suchmaschinen, die Websites auch nur wie Blinde „sehen“. Daraus lassen sich wichtige Erkenntnisse für die Optimierung von Websites für Suchmaschinen ableiten. Das Thema kann auch im grösseren, gesellschaftlichen Rahmen betrachtet werden: Websites von staatlichen Institutionen sind per Gesetz zu Sehbehinderten-Tauglichkeit verpflichtet. Werden diese Richtlinien eingehalten? Sind die Richtlinien überhaupt umsetzbar? Können Sehbehinderte gegebenenfalls Schadenersatz geltend machen? Wie diese Beispiele zeigen, ist das Spektrum möglicher Entdeckungen bei diesem Thema breit.

Material: Zwei Software Tools. Für die Analyse von Webseiten kann zum Beispiel der IBM Home Page Reader verwendet werden. Diese Software liest Webseiten vor und ermöglicht Tastatur-gesteuerte Navigation im Web. Andererseits kommt Bobby zum Einsatz. Dieses Programm zeigt Barrieren der Erreichbarkeit auf, basierend auf den Accessibility-Richtlinien des W3C.

```
<script language="JavaScript" type="text/javascript">
document.writeln('<a href="/information.html?sub">Informationen</a><br><br>');
document.writeln('<a href="/weiter.html">Hier weiter</a><br><br>');
function preload(imgObj,imgSrc) {
    eval (imgObj + ' = new Image()')
    eval (imgObj + '.src = "' +imgSrc+'"')}
</script>
<a href="index2.html"></a>

```

Abbildung 1: Was soll der Home Page Reader bei diesem HTML-Codefragment vorlesen?

Aufgabe und Bewertung: Erkenntnisse festhalten. Die Bewertung hängt davon ab, wie offen die Aufgabenstellung formuliert wurde. Stehen nur technische Fragen zur Implementierung blindengerechter Websites im Vordergrund, oder wird das Spektrum geöffnet bis hin zu rechtlichen oder soziologischen Aspekten?

Eine Schüler-Arbeit wird mit „sehr gut“ bewertet, wenn fünf oder mehr Erkenntnisse genannt werden. Eine Erkenntnis besteht aus einer Aussage mit zugehöriger Begründung, wobei die Aussagen und Begründungen so zu formulieren sind, dass die Mitschüler sie verstehen. Ein Beispiel einer technischen Erkenntnis: „Eine Richtlinie für Web-Entwickler sollte sein, die verwendete Sprache über das lang-Attribute zu kennzeichnen. Nur so ist es einem Tool wie IBM Home Page Reader möglich, ein Wort korrekt auszusprechen.“

3.2 Diskrete Mathematik anhand eines Kartentricks

Viele Kartentricks basieren auf einem geschickt genutzten mathematischen Sachverhalt, meistens aus dem Gebiet der Stochastik oder der diskreten Mathematik. Wie wäre es mit einer vertieften mathematischen Auseinandersetzung anhand von Kartentricks? Kann man faszinierenden Kartentricks systematisch selber auf die Spur kommen? Funktioniert der Kartentrick immer oder nur zufällig in vielen Fällen? Kann ich den Kartentrick abändern, erweitern oder selber neue Zaubertricks erfinden? Um solche Fragen geht es in dieser Unterrichtseinheit.

Anhand eines Kartentricks machen sich Lernende mit Diskreter Mathematik vertraut. Zuerst zeigt die Lehrerin zusammen mit einer eingeweihten Schülerin einen Kartentrick vor. Dann spekulieren die Schüler, wie der Trick funktionieren könnte und halten ihre Vermutungen schriftlich fest. Mit Hilfe einer Computer-Simulation können die Schülerinnen selber Zauberer und Assistent spielen und anhand selber definierter Kartenkonfigurationen versuchen, dem Trick auf die Schliche zu kommen. Im Anschluss an diese erste Entdeckungsphase erhalten alle eine Anleitung zum Trick, üben diesen mit Hilfe einer Simulation am Computer und können anhand eigener Kartenkonfigurationen wiederum ihre Vermutungen überprüfen. Anschliessend ist das Spektrum der möglichen Fortsetzungen der Entdeckungsreise offen. Warum funktioniert der Trick überhaupt? Würde der Trick mit mehr oder weniger Karten ebenfalls funktionieren? Können die Zuschauer anstatt fünf Karten auch sechs Karten auswählen?

Offenheit des Themas. Ausgangspunkt der Unterrichtseinheit ist zwar ein klar definiertes Problem. Die Schülerinnen kommen nicht darum, sich zuerst mit dem Kartentrick als solches zu befassen und zumindest die Funktionsweise des Tricks zu verstehen. Anschliessend können aber unterschiedliche Wege eingeschlagen werden. Man kann versuchen, die mathematischen Hintergründe des Kartentricks genau zu analysieren, zu verstehen oder gar einen Beweis zu führen, dass der Trick immer funktioniert. Man kann den Trick in mancherlei Hinsicht variieren. Man kann ausgehend von dem vorgestellten Trick generelle Überlegungen anstellen, was einen guten Kartentrick ausmacht. Wichtig ist dabei, dass der Kreativität und Phantasie auf der Suche nach eigenen Erkenntnissen keine Grenzen gesetzt werden!

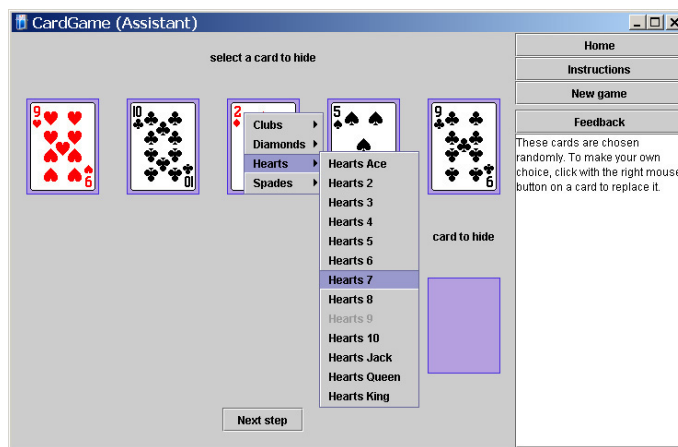


Abbildung 2: Screenshot von CardGame.

Material: Kartentrick vorbereiten. Die Lehrperson muss den Kartentrick zusammen mit einer Schülerin einüben, damit sie ihn gemeinsam überzeugend vorzeigen können. Zur Analyse des Tricks wird das Computerprogramm CardGame (Abbildung 2) eingesetzt.

Aufgabe und Bewertung: Vermutungen notieren. In einem ersten Schritt notieren die Lernenden Vermutungen und Entdeckungen zum Kartentrick aufgrund der Vorführung der Lehrerin. Solche Vermutungen könnten etwa sein: „Die Wahl der Karte, die versteckt werden soll, spielt eine Rolle“ oder „Der ganze Kartenstapel wurde im voraus speziell sortiert“. Nachdem die Lernenden den Trick mit Hilfe der Software beherrschen, überdenken sie ihre Vermutungen aus dem ersten Schritt und machen sich eigene Überlegungen zum Trick. „Was steckt alles dahinter? Lässt sich der Trick verbessern? Würde eine andere Codierung Sinn machen? Funktioniert der Trick auch mit einer anderen Anzahl und anderen Farben von Karten? Gibt es ähnlich Tricks?“ Falsche Aussagen werden dabei nicht berichtigt. Es werden Punkte vergeben für plausible Erkenntnisse und Begründungen. Originelle Ideen können zusätzlich belohnt werden.

3.3 Schwierige Probleme der Graphentheorie

Diese Unterrichtseinheit richtet sich an Gymnasien und an Fachhochschulen, an denen ausgewählte Themen der theoretischen Informatik unterrichtet werden. Die Schüler entdecken mit Hilfe der Software GraphBench NP-vollständige Probleme und Lösungsalgorithmen. Je nach Interesse untersuchen sie Laufzeitaspekte, Extremfälle, Korrektheit, Anwendungen in der Realität oder entwerfen gar eigene Algorithmen. Die Problemstellungen stammen aus der Graphentheorie: Graphfärbbarkeit, Vertex Cover, Clique, Travelling Salesman, Erfüllbarkeit logischer Formeln und ähnliche Probleme.

Üblicherweise werden im Informatik-Unterricht bei diesen Problemen die Algorithmen und auf Hochschulstufe auch die Reduktion anderer NP-vollständiger Probleme auf diese Probleme vorgestellt. Bei diesem Zugang müssen Schüler und Studentinnen zuerst die Problemstellungen als solches verstehen, zum Beispiel die Definition eines Vertex Cover in einem Graphen. Ohne ein intuitives Verständnis für die Problemstellung bleiben mögliche Lösungsalgorithmen oft buchstäblich im luftleeren Raum stehen, und die Reduktion eines Problems auf ein anderes Problem verkommt zu einer rein formalen Angelegenheit ohne tiefere Einsichten.

Mit der Lernumgebung GraphBench können die Lernenden beliebige Probleminstanzen erkunden und erhalten so ein intuitives Gefühl für die Beschaffenheit der Probleme. Anschliessend können sie verschiedene Lösungsalgorithmen durch Beobachten des animierten Ablaufs entdecken. GraphBench ermöglicht es zudem, die Reduktion eines Problems auf ein anderes Problem in Echtzeit zu verfolgen. Eine Änderung der Konfiguration des Graphen bei der Problemstellung Vertex Cover lässt sich parallel dazu in der entsprechenden Probleminstanz eines Hamilton Kreises beobachten. Die abstrakten Reduktionen werden damit wesentlich fassbarer.

Offenheit des Themas. Die Lernumgebung GraphBench erlaubt Entdeckungen in vielerlei Hinsicht. Ob die Lehrperson die Entdeckung bewusst in eine bestimmte Richtung lenken will, hängt von den Zielsetzungen und Vorkenntnissen ab. Sollen die Schülerinnen sich nur je mit einer Problemstellung auseinandersetzen oder eine ganze Palette von NP-vollständigen Problemen kennen lernen? Stehen algorithmische Aspekte im Vordergrund

oder die Reduktionen? Oder es eher um Fragen der Berechenbarkeit? Die Schülerinnen können sich in Fragen vertiefen wie zum Beispiel: Wie funktionieren die Algorithmen? Kann ich selbst einen Algorithmus finden, der das Problem löst? Wie viele Schritte braucht der Algorithmus, bis er eine Lösung findet? Welcher Algorithmus ist der schnellste? Gibt es praktische Anwendungen für dieses Problem? Gibt es Grenzen in der praktischen Nutzung der Lösungsalgorithmen?

Material: GraphBench Die Software GraphBench dient den Schülern als Werkzeug für das Erkunden der verschiedenen Graphenprobleme, zum Erzeugen von Probleminstanzen sowie zum Betrachten der animierten Lösungsalgorithmen und Problem-Reduktionen. GraphBench enthält zudem eine Programmierumgebung für die Implementation eigener Lösungsalgorithmen [Gra05].

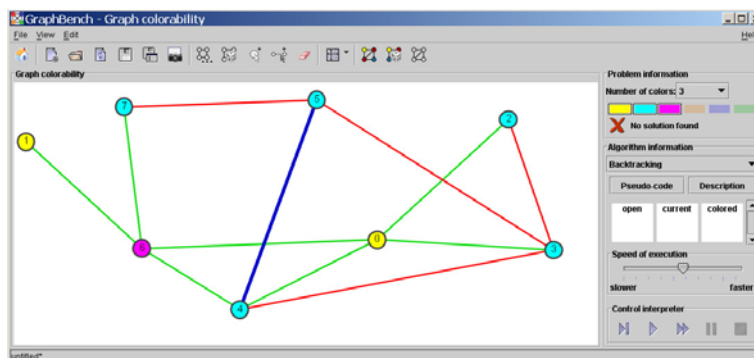


Abbildung 3: Screenshot GraphBench: Färbbarkeitsproblem von Graphen.

Aufgabe und Bewertung: Populärwissenschaftlicher Kurzvortrag. Die Aufgaben sind in eine Rahmengeschichte eingebettet, bei der einige logistische Probleme bei der Organisation eines Stadtfests gelöst werden sollen. Ein Beispiel: „Die Polizei interessiert, wie viele Leute sie einsetzen muss. Die Polizisten werden auf Kreuzungen positioniert. Du denkst, dass es ausreicht, wenn jeder Strassenabschnitt von einem Polizisten überwacht wird. Durch Ausprobieren findest du die Lösung nach einer Stunde. Geht das effizienter?“ Von den Schülern wird erwartet, dass sie einer Person, die sich nicht speziell mit Informatik beschäftigt, ihre Erkenntnisse zur Lösung von solchen Problemen in einfachen Worten anhand von Beispielen erläutern können.

4 Schlussfolgerungen

Entdeckendes Lernen kann den Unterricht bereichern und die Lernenden dazu bringen, vermehrt selbstständig zu arbeiten und eigene Ideen zu entwickeln. Wir haben drei unterschiedliche Informatik-Themen (Gestaltung behindertengerechter Websites, diskrete Ma-

thematik anhand eines Kartentricks und NP-vollständige Graphenprobleme) und deren didaktische Konzeption für die Umsetzung im Informatik-Unterricht skizziert.

Beim Einsatz von Entdeckendem Lernen sind die folgenden Punkte besonders wichtig:

- Die Problemstellung muss offen sein: Es darf keine einzig richtige Lösung geben.
- Die Studierenden müssen genügend Zeit haben zum Bearbeiten der Aufgabe. Entdeckendes Lernen in fünf Minuten macht wenig Sinn.
- Die Studierenden brauchen Freiraum. Die Lehrperson darf nicht zu früh eingreifen, sie muss (als Zwischenschritte) halbrichtige Lösungen akzeptieren.
- Alle (ernsthaften) Beiträge und Erkenntnisse werden von der Lehrperson honoriert.

Natürlich lässt sich nicht jedes Thema im Informatik-Unterricht mit Entdeckendem Lernen umsetzen. Herauszufinden, welche Themen sich für Entdeckendes Lernen besonders eignen, und die benötigten Materialien zusammen zu stellen, stellt einen nicht zu unterschätzenden Aufwand dar. Allerdings muss dieser Aufwand nur einmal geleistet werden. Wenn Entdeckendes Lernen vermehrt im Informatik-Unterricht Einzug halten soll, müssen die Ideen und Materialien unter den Lehrpersonen ausgetauscht werden. Die von uns vorgestellten Beispiele finden sich deshalb neben einem guten Dutzend weiterer Beispiele samt ausführlicher Dokumentation frei verfügbar auf dem Bildungsserver SwissEduc [Swi05]. Die Materialien erfreuen sich zunehmender Beliebtheit: Wöchentlich über 2'500 Zugriffe total, mit über 800 Downloads von Office- und PDF-Dokumenten (Suchmaschinen ausgeschlossen; Stand Februar 2005). Ein entdeckender Blick in diese noch bescheidene Beispielsammlung lohnt sich!

Literatur

- [Bal96] Doug Baldwin. Discovery learning in computer science. *SIGCSE*, Seiten 222–226, 1996.
- [Den89] Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., und Young, P. R. Computing as a Discipline. *Communications of the ACM*, 32(1):9–23, 1989.
- [dJvJ98] Tom de Jong und Wouter R. van Joolingen. Scientific Discovery Learning with Computer Simulations of Conceptual Domains. *Review of Educational Research*, 68:179–201, 1998.
- [Gra05] GraphBench: A System for Prototyping and Animating Graph Algorithms. <http://www.inf.ethz.ch/personal/braendle/graphbench/>, Februar 2005.
- [Ham00] Uwe Hameyer. Entdeckendes Lernen. In Jürgen Wiechmann, Hrsg., *Zwölf Unterrichtsmethoden*. Beltz Verlag, Weinheim, 2. Auflage, 2000.
- [Neb73] Heinz Neber, Hrsg. *Entdeckendes Lernen*. Beltz Verlag, Weinheim, 1973.
- [Swi05] Entdeckendes Lernen in der Informatik auf dem Bildungsserver SwissEduc. <http://www.swisseduc.ch/informatik/entdecken/>, Februar 2005.