

Die Kunst Programmieren zu lernen ohne Wurzeln zu ziehen

*Code und Kunst – multimediale Programmierausbildung an der
Kunsthochschule für Medien Köln*

Von Jochen Viehoff

An der Kunsthochschule für Medien Köln haben die grundständigen und postgraduierten Studierenden die Möglichkeit, in einem zweiwöchigen Java-Kompaktkurs die Grundlagen der objektorientierten Programmierung zu erlernen. Die Inhalte dieses Kurses orientieren sich speziell an den Interessen der Lernenden: Bild und Video, Klang, Musik und die Mensch-Maschine Interaktion. Alle Lernmaterialien für diesen einführenden Kurs sind im Internet als CodeKit-Programmarchiv frei verfügbar. In betreuter Projektarbeit können über den Kurs hinaus eigenständige künstlerische Programmierarbeiten weiterentwickelt und gegebenenfalls in Ausstellungen präsentiert werden.

Die Kunsthochschule für Medien Köln (KHM) vertritt in Lehre und Forschung das breite Spannungsfeld, das aus dem Interesse an der Kunst im Zeitalter ihrer technischen Reproduzierbarkeit hervorgeht. Die Institution nahm am 15. Oktober 1990 den Lehrbetrieb auf und bietet neben dem viersemestrigen Zusatz- oder Weiterbildungsstudiengang (postgraduiert) seit dem Wintersemester 1994/95 auch einen achtsemestrigen Diplomstudiengang für Audiovisuelle Medien an. Dieses grundständige Studium ist in zwei Studienabschnitte gegliedert: Grundstudium (Semester 1 bis 4) und Hauptstudium (Semester 5 bis 8). Im Hauptstudium sind die Studierenden einer der Fächergruppen Medienkunst, Mediengestaltung oder Film/Fernsehen zugeordnet. Das 8. Studiensemester schließt mit einer Diplomprüfung ab.

Computerausbildung im Grundstudium

Im Grundstudium vermitteln künstlerisch/wissenschaftliche Mitarbeiter allen Studierenden in Grundkursen elementare Kenntnisse im Umgang mit dem Computer. Die Pflichtveranstaltung „Computerkurs 1“ gibt einen kurzen einführenden Überblick über Rechnerarchitekturen, Netzwerkstrukturen, das Internet und die Geschichte programmierbarer Apparate. Danach werden Standardprogramme als Werkzeuge für die künstlerische Produktion mit Medien vorgestellt und in praktischen Übungen deren Anwendung erlernt. Zu diesen Programmwerkzeugen gehören digitale Bildbearbeitungssoftware, Layout-Programme, Web-Editoren, interaktive Animationsprogramme und digitale Klangerzeugung. Parallel zu den Werkzeugen sind die theoretischen Grundlagen, z.B. das elektronische Bild, Farbräume, Klanggenerierung, digitale Audio-Formate, Datenkompression, Hypertextstrukturen, Typografie etc. Lerninhalte des Kurses. Im zweiten Teil wird elektronische Schriftkompetenz anhand der Script-Sprachen *html* und *JavaScript*, bzw. der plattformunabhängigen Programmiersprache *Java* vermittelt.¹ Hierbei ist das zentrale Werkzeug ein Texteditor und eine Interpreter- bzw. Compiler-Anwendung (Internet-Browser und Java-Compiler)

Die konkreten Lernziele des Grundkurses sind:

1. Der sichere Umgang mit den Computer-Werkzeugen für die Produktion und Weiterbearbeitung multimedialer Inhalte (Softwarekompetenz).
2. Ein Grundverständnis für Computer als komplexe programmierbare Maschinen unter Verwendung spezieller textbasierter Codes (elektronische Schriftkompetenz).

Darauf aufbauend können in weiterführenden Grundkursen und Fachseminaren ab dem zweiten Semester die Themenbereiche Typografie, Grafik und (Web-)Gestaltung, digitale Videobearbeitung oder objektorientierte Programmierung vertieft werden.

Einführung in die Java-Programmierung

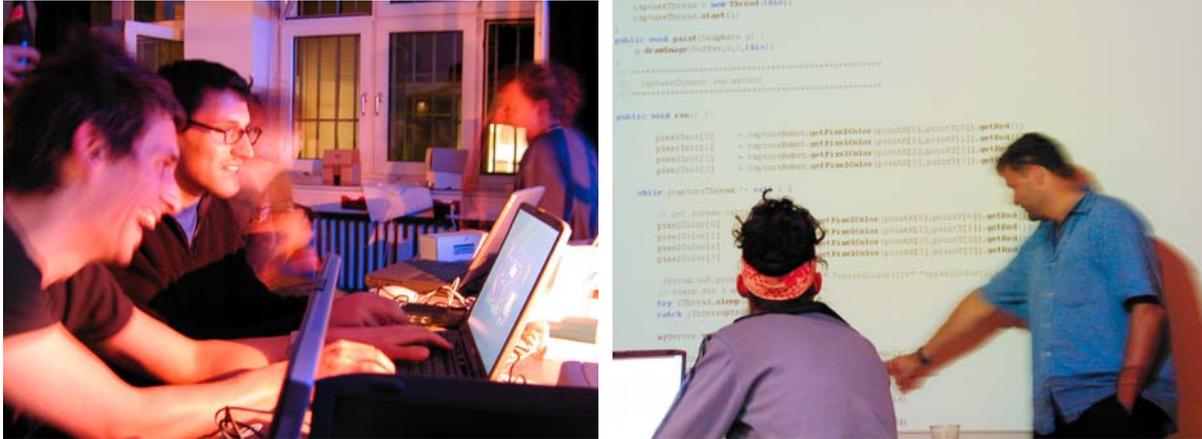


Abb. 1: In dem Java-Kompaktkurs an der KHM lernen die Studierenden die Grundlagen der Programmierung für die künstlerische Praxis kennen.

Ein zweiwöchiger Kompaktkurs „Einführung in die Programmierung“ findet während den Sommersemesterferien statt. In einer Kleingruppe (10-15 Studierende) wird ganztägig in einem mobilen Lern- und Arbeitsumfeld (PC Laptops mit WLAN-Anbindung²) Theorie und Praxis einer höheren Programmiersprache und deren Bedeutung für die künstlerische Praxis vorgestellt (Abb. 1). Thematisch wird im Gegensatz zu „klassischen“ Einführungen in die Programmierung, wie sie beispielsweise im Informatikunterricht weit verbreitet sind, weitestgehend auf technisch-wissenschaftliche Inhalte und Beispiele verzichtet (z.B. Algorithmen zum Sortieren von Zahlen oder die Berechnung einer Quadratwurzel). Das erste Beispielprogramm ist keine Textausgabe wie das weit verbreitete „Hello World!“-Programm, sondern das einfache Abspielen einer Note auf einem virtuellen Klavier. An diesem übersichtlichen Beispiel lernen die Teilnehmer, wie durch Variablen, Kontrollstrukturen in Schleifen und Verzweigungen oder Operatoren komplexere Programmabläufe für Melodien, Akkorde oder kleine Kompositionen möglich sind. Auch allgemeine Begriffe wie Eigenschaften und Methoden von Objekten, werden an Beispielen aus den Bereichen Grafik, Ton und Musik, Video oder Interaktion veranschaulicht. Die ausgewählten Beispiele im Kurs orientieren sich an den Interessensgebieten der Künstler und Künstlerinnen - Musik, Bild und Interaktion eignen sich gleichermaßen für einen Einstieg in das algorithmische Denken. Alle Programme, die in dem Java-Kurs verwendet werden, sind in dem Programmarchiv „CodeKit“ zusammengestellt und im Netz frei verfügbar.³ Ferner finden sich in dem Archiv weitere Informationen über die Software, Installationshinweise, sowie Übungen zu den Lerneinheiten.

Die Studierenden arbeiten und lernen im Kurs auf der Ebene des Codes, ohne weitere grafische Oberflächen. Sie verwenden zusätzliche *Medienobjekte* (z.B. ein „VideoPlayer“-Objekt) in ihren Applikationen, um komplexere Funktionalitäten einzubinden (hier: die Wiedergabe und Steuerung von Videos). Einige Beispielprogramme aus dem Archiv unterstützen auch zusätzliche Hardware. Für eine Web-Kamera stehen Bildanalyse-

Algorithmen zur Verfügung, und ein Messcomputer⁴ mit einer Vielzahl von physikalischen, chemischen und biologischen Sensoren erweitert das Interaktionspotential zwischen Computer und Benutzer.

Im zweiten Teil des Kurses, nachdem die Grundlagen eingeführt und praktisch eingeübt worden sind, arbeiten die Studierenden bereits an konkreten Projekten, für die sie schrittweise tiefer in die Programmierung einsteigen müssen, z.B. eine temperaturabhängige Videosteuerung oder ein interaktiver Audio-Mixer kamen zum Einsatz. Auch über den Zeitrahmen des Java-Kurses hinaus können die Teilnehmer im Rahmen eines „künstlerischen Experiments“ weiter an ihren Programmen arbeiten. Dafür stehen in einem Computer-Labor der KHM Rechner zur Verfügung. Wissenschaftliches und technisches Personal betreuen die Projekte. Dass daraus eigenständige künstlerische Arbeiten erwachsen können, zeigen die kreativen Beispiele „Bitmirror“ von Tobias Grewenig und „Update“ von Vera Doerk.

Bitmirror – Partikelanimation für Ascii-Zeichen



Abb. 2: Bitmirror: In der Animation wird das Selbstbild des Betrachters, in einer Repräsentation als ASCII-Zeichen, gemäß den physikalischen Gesetzen der Newtonschen Mechanik gesteuert.

„Bitmirror“ ist eine Projektarbeit des postgraduierten Studenten Tobias Grewenig. Die Arbeit verwertet Pixel-Informationen aus Bilddateien oder von einer Web-Kamera, um sie in ASCII-Zeichen zu übersetzen und gemäß den Regeln eines Partikel-Systems durch physikalische Kräfte zu animieren. Abbildung 2 zeigt zuerst als Ausgangsbild das Selbstbildnis des Künstlers und dann ein Schwarmverhalten der einzelnen Zeichen, initiiert und gesteuert von den aktuellen Mouse-Koordinaten oder den Bewegungen des Betrachters vor dem Bildschirm. Die Arbeit wurde von dem Studenten selbst in Java programmiert. Ein Applet des „Bitmirrors“ (ohne Web-Kamera) kann aus dem Netz geladen und getestet werden.⁵ Als Grundlage für die komplexe Teilchenanimation wurde im Java-Kurs der Mechanismus einfacher physikalischer Animationen erklärt und an Beispielen demonstriert. Die Applets und Applikationen sind im CodeKit-Archiv unter der Rubrik „Animation“ abgespeichert. Über die Parameter Dämpfung und Federkonstante (lineare Rückstellkraft) ist es möglich, die Dynamik der Bewegung in weiten Bereichen zu variieren. Die Newtonschen Bewegungsgleichungen werden durch Gauss-Integration numerisch gelöst. Daraus resultieren „natürlich“ wirkende Bewegungen, wie man sie aus der realen Welt kennt (z.B. Wellen und Schwarmverhalten). Alternativ zur Interaktion mit der Mouse wertet ein Algorithmus die Bewegungsinformationen aus, die von einer Kamera aufgezeichnet werden. „Bitmirror“ ist einerseits ein gutes Beispiel für eine computerbasierte künstlerische Arbeit, die nur auf Ebene der Programmierung und bei weiterführender Betreuung auch über einen Programmierkurs hinaus realisiert werden konnte. Andererseits nutzen die neuen Programmteile, die von dem Studenten geschrieben worden sind, im CodeKit-Archiv anderen Lernenden bei ihrem Einstieg in die kreative Programmierarbeit mit dem Computer.

Update – interaktive Zeitobjekte

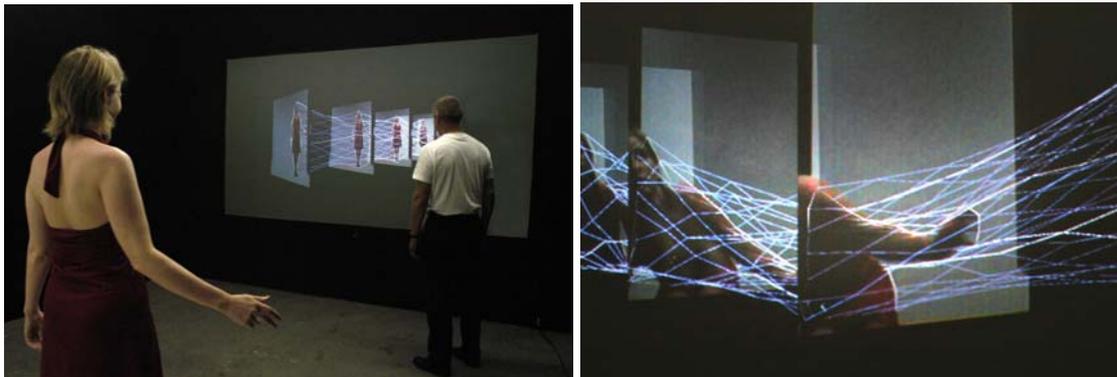


Abb. 3: Die Konturen des Betrachters werden auf jeder Bildebene einzeln erfasst und durch komplexe Netzstrukturen zu sogenannten Zeitobjekten verwoben.

„Update“ ist die Diplomarbeit der postgraduierten Studentin Vera Doerk. Im Java-Kurs machte sie sich mit den Grundlagen der Programmierung vertraut. Daran schlossen sich Experimente mit Videotexturen und dreidimensionalen Objekten an. In ihrer interaktiven Installation schließlich werden in Echtzeit die Konturen der Besucher erfasst, ihre Bilder, aufgezeichnet von einer Kamera, auf zeitlich und räumlich getrennten Bildebenen wiedergegeben und Konturpunkte durch komplexe Netzstrukturen verwoben: Es entstehen sogenannte „Zeitobjekte“ (Abb. 3). Durch die dreidimensionalen Netze kann der Betrachter navigieren und verschiedene Blickwinkel des Objekts wählen, indem er oder sie die Raumposition verändern. Die Komplexität der Netze ist korreliert mit der Bewegung des Betrachters.

„Update“ wurde von der Künstlerin in Java3D programmiert⁶, nachdem sie bereits während des Java-Kurses erste Module, die ursprünglich aus dem CodeKit-Programmarchiv stammen (Rubriken: „Darstellung in Java3D“ und „Kamera Tracking“), für ihre Arbeit adaptierte. Die Künstlerin stellte ihre Diplomarbeit bereits mehrfach aus, zuletzt auf der Popkomm-Messe Köln (2003) und während des Festivals „tanzmedial“ (2004). Sie zeigt – ähnlich wie der „Bitmirror“ von Tobias Grewenig – welche thematischen Ausrichtungen für einen Programmierkurs sinnvoll, vielleicht notwendig sind, um die Studierenden in eigenständiges kreatives Arbeiten mit dem Computer einzubinden. Zuletzt ist dieses selbstständige Arbeiten unverzichtbar für das Erlernen einer höheren Programmiersprache.

Zusammenfassung und Ausblick

Ein künstlerischer Umgang mit computerbasierten Multimedia-Anwendungen setzt insbesondere dann fundierte Programmierkenntnisse voraus, wenn der Rechner nicht nur als standardisiertes Werkzeug eingesetzt werden soll, sondern die Komplexität programmierbarer Maschinen für einen eigenständigen kreativen Prozess relevant ist. Hierfür werden die Grundlagen der Programmierung den Studierenden an der Kunsthochschule für Medien Köln in einem kompakten Java-Kurs vermittelt. Das CodeKit-Programmarchiv dient dabei als Materialsammlung für die Veranschaulichung essentieller Programmierstrukturen. Es ist ein Werkzeugkasten mit ausführlich dokumentierten Java-Programmen, in denen unterschiedliche Standardapplikationen für Bild, Video, Klang und Interaktion aus- und umgebaut werden können. Die Beispiele reichen von einfachen Applikationen, die zum Einstieg in die Programmierung geeignet sind, bis hin zu Modulen, welche in aufwendigeren Programmen zum Einsatz kommen. Das CodeKit-Archiv wird regelmäßig erweitert. Wie die Programme für weiterführende künstlerische Projekte eingesetzt werden können, wurde hier an zwei

konkreten Beispielen skizziert. In diesem Falle sollten im Sinne der „open source“-Bewegung die resultierenden Codes anderen Studierenden wieder zur Verfügung stehen.

Programmieren lernen wird dann einfacher, wenn die Lernenden von der möglichen Umsetzung ihrer individuellen Ideen und Interessen motiviert werden. Mathematische Probleme beispielsweise interessieren Künstler und Künstlerinnen primär wenig, effektvolle Animationen dagegen umso mehr. Diese Tatsache muss von einem Ausbildungskonzept an einer Kunsthochschule reflektiert werden. Abschließend stellt sich die Frage, ob das Erlernen einer textbasierten höheren Programmiersprache anhand adäquater multimedialer Inhalte in einem kreativen Prozess nicht ohnehin zugänglicher ist, als neuerlich das klassische Beispiel der Berechnung von Quadratwurzeln zu bemühen.

Anmerkungen

1. Die Programmiersprache *Java* ist von *Sun Microsystems* entwickelt worden, und kann kostenlos aus dem Netz geladen werden (<http://java.sun.com>).
- html* (hyper text markup language). Script-Sprache, beschreibt Inhalt und Aufbau einer Web-Seite.
- JavaScript* - Interpreter-Sprache für Internet-Browser.
2. WLAN - Wireless Local Area Network. Eine kabellose Anbindung (Funk) der Computer an das Internet.
3. Siehe auch den Workspace-Beitrag „CodeKit“ in dieses Ausgabe.
4. LabPro ist ein Messcomputer (Analog/Digital-Wandler) von der Firma Vernier, der ursprünglich für den Physikunterricht an Schulen entwickelt wurde.
5. Ein Java-Applet kann aus dem Netz geladen und lokal auf dem Rechner ausgeführt werden. (Bitmirror-Applet: <http://java.khm.de/projekte>)
6. Java3D ist ein Erweiterungspaket (API) für die Konstruktion und Darstellung dreidimensionaler Objekte.

Links & Literatur

Kunsthochschule für Medien Köln:<http://www.khm.de>
CodeKit-Programmarchiv: <http://java.khm.de>
Interface-Labor der KHM: <http://interface.khm.de>
Tobias Grewenig Homepage: <http://www.xi-bot.de>
Vera Doerk Homepage: <http://www.khm.de/~veradoerk>

Georg Trogemann, Jochen Viehoff: CodeArt – Grundlagen der Programmierung für künstlerische und medienzentrierte Praktiken. Springer Verlag, Wien, erscheint September 2004.

Donald .E. Knuth: The Art of Computer Programming. Addison-Wesley, Reading Massachusetts, 1973.

Zum Autor

Dr. Jochen Viehoff
Künstlerisch-wissenschaftlicher Mitarbeiter
Kunsthochschule für Medien Köln

Peter-Welter-Platz 2
50676 Köln
E-Mail: viehoff@khm.de