Impressum

Chefredakteur (V.i.S.d.P.): StD Dr. Ludger Humbert
Redaktion: StD Christian F. Görlich & Fachseminare Informatik Hamm und Arnsberg
Layout: Ludger Humbert (Vorlage von Torsten Bronger)
Copyright: Für namentlich gekennzeichnete Artikel übernimmt die Autorin die Verantwortung.

SOME RIGHTS RESERVED

Der Inhalt unterliegt der ↘ creativecommons.org/licenses/by-nc-sa/2.0/de-Lizenz …
If Fase … auch im Netz (↘ humbert.in.hagen.de/iffase)
ISSN 1861-0498

If Fase

IF FASE: INFORMATIKFACHSEMINARE HAMM, ARNSBERG

Bildung
Fachseminare
Veranstaltungen
Informatiksysteme
Ausbildungsschulen

If Fase

Nr. 13 — HERAUSGEGEBEN VON DER IF FASE-REDAKTION — 1. November 2006

Termine

Freitag, 3. November 2006

22. FIfF-Jahrestagung

Bremen – Haus der Wissenschaft / ZIMT

Auftaktveranstaltung
↘ fiff.informatik.uni-bremen.de/2006/index.html

KurzNotiert

(von Dr. Ludger Humbert)

Informatikfachleitertreffen NW 2006

Wie in der If Fase Ausgabe 12 mitgeteilt, wurde am Montag, 16. Oktober 2006 im Landesinstitut in Soest das zweite Informatikfachleitertreffen NW (== Nordrhein-Westfalen) durchgeführt. Die Beteiligung war überwältigend.
Neben den Fachleiter/innen für die Ausbildung für das Lehramt für Gymnasien und Gesamtschulen nahm eine grosse Zahl von Fachleiter/innen aus dem Bereich der Ausbildung für das Lehramt Informatik für Haupt-, Real- und Gesamtschulen (ehemals Sek I) teil. Die Frage nach einer Fortführung der Treffen und einer Erweiterung um die Kolleg/innen aus dem Bereich der Ausbildung für die Berufskollegs wurde positiv beantwortet, so dass wir im nächsten Jahr voraussichtlich wieder einen erheblich erweiterten Kreis begrüßen können.
Die allgemeine Situation der Ausbildung wurde in den vergangenen Jahren durch fortwährende Änderungen, die nicht ausbildungsdidaktisch, sondern monetär bedingt waren, nicht eben im Sinne einer zukunftsweisenden Ausbildung optimiert. Dies wird von der aktuellen Landesregierung offenbar fortgeführt – Kontinuität der zweifelhaften Sorte.
Der Stellenwert des Schulfachs Informatik ist für die Sek I Ausbildung essenziell: da Informatik nicht in der Stundentafel aufgeführt ist, wird es bei höherer Belastung einer Schule nicht mehr unterrichtet. Damit wird einerseits um Seiteneinsteiger geworben, die nach erfolgreicher Qualifikation am Ende ihrer Ausbildung Schulen zugewiesen werden, an denen Informatik nicht mehr unterrichtet werden kann.

* * *

Zur Zeit wird ein offener Brief der Fachleiter/innen Informatik an die Landesregierung vorbereitet, der in nächster Zeit veröffentlicht werden soll.

LaTeX – Teil 13: Kleine Geräte

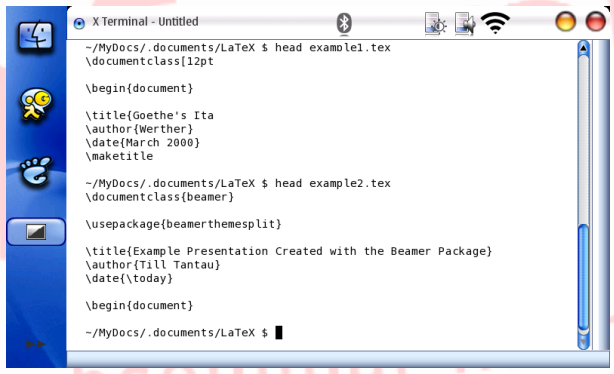
In einer Reihe von Artikeln in der If Fase werden nützliche Elemente von LaTeX vorgestellt, die erprobt sind und bei der Arbeit der Informatiklehrerin eingesetzt werden.

(von Dr. Ludger Humbert) In den bisher vorgelegten dreizehn Teilen der Artikelserie – Ausgaben 0 … 12: ↘ humbert.in.hagen.de/iffase/Archiv – finden Sie Hinweise und Anmerkungen zu den Themen: Installation, grundlegende Arbeitsweisen, Quellen zu Dokumentationen, Arbeit mit KOMA-Script, PSTricks, Erstellung von Arbeitsblättern, Struktogrammen, Automatengrafen, Elemente von UML, Barcodes, Formularerstellung, Zitieren, Abbildungen, ER-Diagramme. Von Ausgabe 9 (Zitieren – normgerecht) bis zur Ausgabe 12 (Quelltexte von Programmen) wurden die Themen von Fragen bestimmt, die von den Referendarinnen gestellt werden.

Allgegenwärtige Informatik

Zunehmend beschäftigt sich die Informatik mit der Durchdringung des Alltags mit Informatiksystemen (ubiquitous computing). Nicht nur mobile Telefone, auch PDAs, RFIDs, intelligente Kleidung (wearable computing) u. v. a. m. zeigen, dass der Informatisierung des Alltags keine Grenzen gesetzt zu sein scheinen.

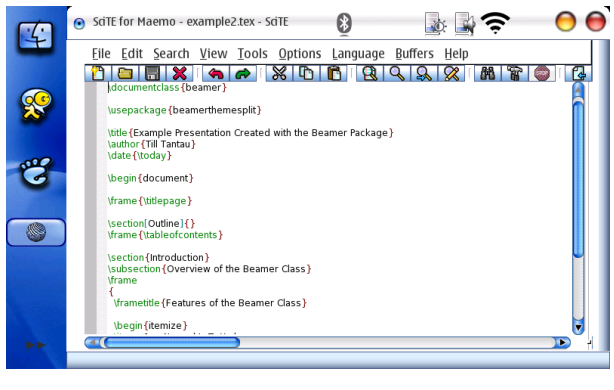
Was hat das mit einer Serie über LaTeX zu tun?
Vor einigen Jahren bestand die Herausforderung noch darin, auf einem Apple PowerBook mit OS 8.6 LaTeX (kostengünstig) zur Arbeit zu überreden – heute ist es ein Nokia 770 (N770). Aufmerksame Leserinnen werden in meinem Blog ↘ haspe.homeip.net:8080/cgi-bin/pyblosxom.cgi/LaTeX/2006-10-18_MaemoTeX_0.1.0-7.html oder in der Ausgabe 12 der If Fase Hinweise auf ein Debianpaket gefunden haben, dass es auf dem N770 ermöglicht, mit LaTeX zu arbeiten.



xterm *mit LaTeX-Quellcode*

Ohne Kommandozeile auf einem Informatiksystem fühle ich mich unwohl. Der Hintergrund: ich habe viele Jahre mit einem Betriebssystem auf dem Apple gearbeitet, das standardmäßig keine Konsole anbot. Damit wird die Benutzung auf »Klicken« reduziert. Funktionierte etwas nicht zufriedenstellend, wurde mit *ResEdit* (einer Anwendung, mit der die Ressourcen bearbeitet werden können) und einer guten Portion Versuch und Irrtum rumprobiert.

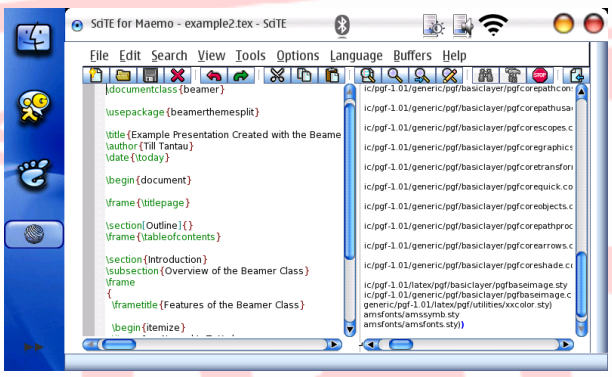
So ist verständlich, dass ich zunächst exploriere, welche kommandozeilenbasierte Schnittstelle zu einem Informatiksystem existiert und wie benutzbar sie konkret ist. Hier liegt mit *xterm* ein nutzbares Werkzeug bereit. Es ist möglich, mit *Tabs* zu arbeiten, um so gleichzeitig mehrere *Shells* zu öffnen, …



SciTE *mit Beamer-Präsentation im Quellcode*

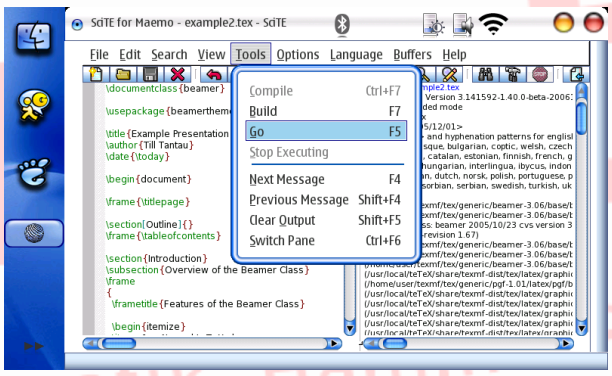
Es ist nicht das Ziel dieses Artikels, die Installation zu beschreiben, sondern einige Hinweise zu geben, wie mit LaTeX unter dem N770 effizient gearbeitet werden kann. Verlassen wir also die Kommandozeile und wenden uns dem primären Werkzeug zu, mit dem die Quelltexte erstellt werden, aus denen später die Dokumente generiert werden.

Nach dem Start stehen in dem **wissenschaftlichen Editor** *SciTE* kontextbezogene Hervorhebungen zum jeweiligen Quelltext zur Verfügung. Darüber hinaus können Routinen durch Kommandozeilen ausgeführt werden, ohne den Editor zu verlassen. Auf diese Weise kann der Quelltext eingegeben und gesetzt werden. Nach der Eingabe wird unter dem Menüpunkt *Tools* (Werkzeuge) die Auswahl *Build* aktiviert …



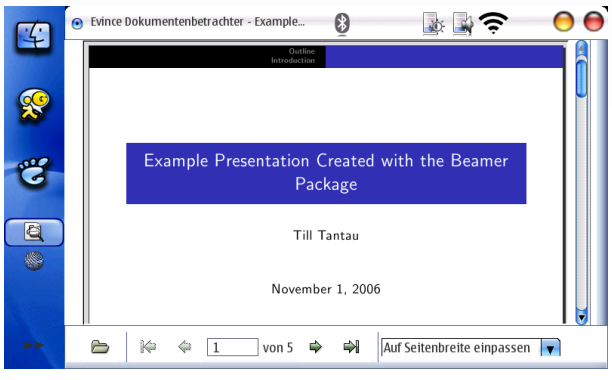
SciTE *mit Beamer-Präsentation während der Ausführung von* Build

Die Möglichkeit, Quelltexte zu kompilieren, bezieht sich selbstredend nicht nur auf LaTeX, nein auch Python (und weitere Programmiersprachen) wird unterstützt; die Kommandos sind in Textdateien abgelegt, die nach Bedarf geändert werden können.



SciTE *mit Beamer-Präsentation* – Go

Jetzt fehlt nur die Möglichkeit, das Ergebnis zu betrachten. Hier steht mit *Go* eine weitere Schnittstelle bereit, die – entsprechend konfiguriert – einen PDF-Betrachter öffnet. Allerdings wird dort [noch] nicht das Dokument direkt angezeigt; es muss explizit geöffnet werden. Die Ursache könnte in einer fehlerhaften oder unvollständigen Konfiguration zu finden sein.



Evince *mit Beamer-Präsentation nach dem Öffnen*

Quellen

Die in diesem Artikel benutzten Quellen bestehen einerseits aus Werkzeugen und Elementen, die für das hier verwendete Gerät Nokia 770 angeboten werden – ↘ maemo.org/maemowiki/ApplicationCatalog2006. Andererseits wurde nicht nur *MaemoTeX* (das vom Autor zusammengestellt und gepflegt wird) installiert, sondern auch eine Erweiterung des TDS-Baums vorgenommen, um einige zur Nutzung der LaTeX-Klasse Beamer (siehe Abbildung) notwendige Elemente auf dem Gerät nutzen zu können. LaTeX-Pakete lassen sich über die Paketsuchfunktion auf dem Dante-Server finden: ↘ www.dante.de/cgi-bin/ctan-index

Wettbewerbe

25. Bundeswettbewerb Informatik – 13. 11. 2006 ist Einsendeschluss

(von Dr. Ludger Humbert) Am 1. September 2006 wurden die Aufgaben für die erste Runde des 25. Bundeswettbewerbs Informatik bekanntgegeben: ↘ www.bwinf.de

Einsendeschluss für die 1. Runde ist der 13. November 2006 Die Anmeldung zum 25. BWINF erfolgt mit der Einsendung von Lösungen oder unter ↘ www.bwinf.de.

Bundeswettwerb Informatik – Logo

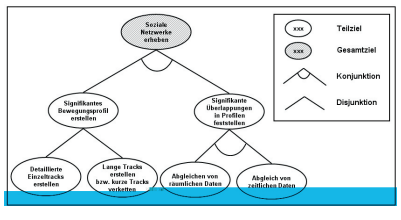
* * *

Broschüre zu RFID – Radio Frequency Identification

Im Oktober wurde die Broschüre des Forums InformatikerInnen für Frieden und gesellschaftliche Verantwortung (FIfF) zu RFID veröffentlicht. Sie soll als Auftakt zu einer Diskussion verstanden werden, die unbedingt geführt werden muss. Eine wichtige Erkenntnis: RFIDs sind nicht per se »böse« oder »gut«. Allerdings tut eine Diskussion um den poltischen Gestaltungswillen Not. Wir befinden uns mitten in der Implementierungsphase, in der zunehmend erkannt wird, welche Probleme mit dieser unsichtbar (gemachten) Technik auf uns zukommt.

Die Interessen sind relativ klar verteilt – wenn wir es jetzt nicht schaffen, eine grosse Zahl von Menschen für dieses Thema zu sensibilisieren, ist der Zug unterwegs und es gibt kaum Möglichkeiten, ihn aufzuhalten oder seine Richtung zu beeinflussen.

Lesen! ↘ www.fiff.de



Abstrakter Attack-Tree zum Erheben sozialer Netzwerkes – Broschüre, S. 31

bessere Lehrerbildung -- dank Informatik

Lesen

(von Dr. Ludger Humbert)

Informatikunterricht planen und durchführen

Werner Hartmann, Michael Näf und Raimond Reichert haben ein kleines, aber feines Büchlein mit vielen Beispielen für die Unterrichtspraxis der Schweiz vorgelegt. Sehr viele der dokumentierten Beispiele, Problemaufrisse und Lösungsideen können auch für bundesdeutsche Verhältnisse genutzt werden.

Informatikunterricht planen und durchführen

Die Kollegen schöpfen aus einem reichhaltigen Fundus. Im Unterschied zur Bundesrepublik gibt es in der Schweiz keine Informatik-EPA, kein Informatik-Zentralabitur …

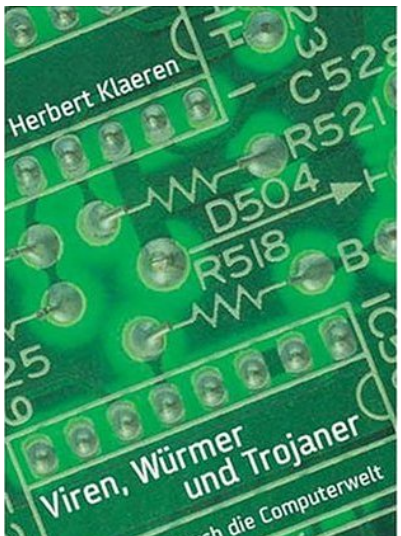
- ISBN 3-540-34484-5

Das Buch stellt ein hervorragendes Übungsbuch zur Didaktik der Informatik dar, das (von mir) jedem am Lehramt Informatik interessierten Menschen empfohlen wird.

Viren, Würmer und Trojaner. Streifzüge in der Computerwelt

Das Herbert Klaeren nicht irgendwer ist, wird Lesern meiner »Didaktik der Informatik« bekannt sein. Mit dem Buch wird eine von vielen Anekdoten, Geschichten, Hintergründen subjektiv gefärbte Sicht auf die Informatik vorgestellt. Ich habe das Buch noch nicht gelesen, aber bald ist ja Weihnachten …

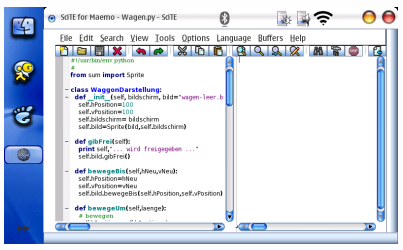
- ISBN 3-937667-72-5
- Verlag: Klöpfer und Meyer



Schüler brauchen keine Entwicklungsumgebung – ein Editor reicht aus

(von Dr. Ludger Humbert)

Ich kann mich erinnern, dass wir uns gefreut haben, als erste integrierte Systeme verbreitet wurden (UCSD-Pascal war, soweit ich erinnere, das erste System, das allgemein verbreitet wurde – lange vor Turbo), die es gestatteten, mit einem einfachen Editor, der viele Sprachelemente hervorhob und es ermöglichten, direkt aus dem Editor das Programm übersetzen zu lassen, an fehlerhafte Stellen geführt zu werden, …



SciTE *mit Pythonprogramm*

Bildung hinter den K[n]öpfchen?

Zunehmend jedoch wurden diese Ansätze immer komplexer: hier eine Möglichkeit, den Programmablauf zu beobachten, dort eine Variante, um Tipparbeit zu sparen, um die **professionelle Programmiererin** zu unterstützen. Und dann noch: Mal eben eine Benutzungsoberfläche zusammenzuklicken. Featurities machte sich breit. Viele Kollegen glauben heute, ohne eine Entwicklungsumgebung wäre Programmieren nicht möglich. Wenn ich die Argumente für Entwicklungsumgebungen höre, habe ich häufig den Eindruck, dass es Schülerinnen im Unterricht ermöglicht werden soll, professionelle Softwareentwicklung zu betreiben.
Das ist aber nicht unser Ziel.

Nach wie vor gilt:

- einfache, übersichtliche, konfigurierbare Texteditoren
- einfache Werkzeuge, um notwendige Arbeiten zu erledigen

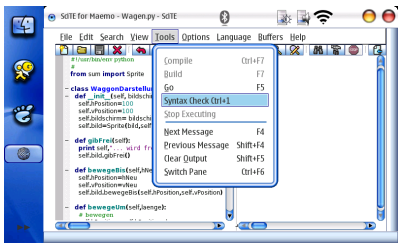
sind die Voraussetzungen für die erfolgreiche Arbeit bei der Umsetzung von modellierten Lösungen.

Kleine Geräte – leistungsfähige Umgebungen für den Informatikunterricht in der Schule?

Dass der Einsatz von komplexen Entwicklungsumgebungen auf einem kleinen mobilen Gerät kaum sinnvoll ist, bedarf kaum der weiteren Erläuterung. Damit kommt ein solches Gerät meinem didaktischen Wunsch sehr entgegen.

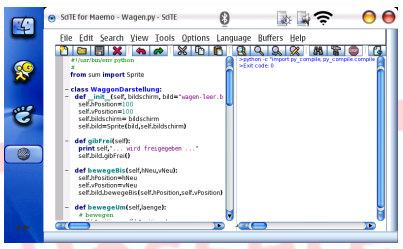
Nützliche Werkzeuge

Es sollte möglich sein, dass der Programmcode syntaktisch geprüft wird, bevor er zum Ablauf gebracht wird. Der bereits in dem LaTeX-Artikel dieser Ausgabe vorgestellte Editor bietet (voreingestellt) diese Möglichkeit.



SciTE *Pythonprogramm* Syntaxcheck

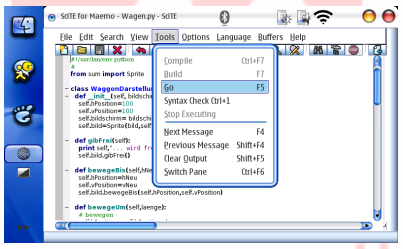
Um zu prüfen, ob die Syntax des eingegebenen Programms korrekt ist, wird von SciTE das Werkzeug *Syntax Check* zur Verfügung gestellt.



SciTE *mit Pythonprogramm* Syntaxcheck *Ergebnis*

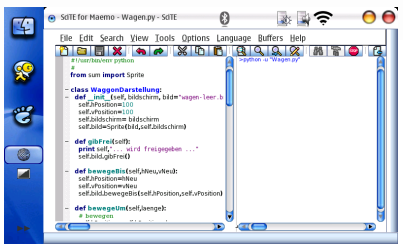
Das Ergebnis der Prüfung wird in einem eigenen Textfenster dargestellt. So hat man gleichzeitig beide Sichten im Blick und kann auf diese Weise nach und nach die Syntaxfehler beseitigen.

Eine weitere Anforderung betrifft die Möglichkeit, aus dem Editor heraus das Programm zu starten. Dies bereitet modernen Editoren keine Schwierigkeit – sind erst einmal alle Pfade richtig eingestellt. Zu meiner Überraschung waren die Einstellungen bei *SciTE* bereits so vorgenommen, dass direkt damit gearbeitet werden konnte.



SciTE *mit Pythonprogramm* – Go

Im konkreten Fall muss aus der Werkzeugkiste (Tools) das Werkzeug *Go* ausgewählt werden. Damit werden – transparent in einem gesonderten Textfenster – die Ausgaben protokolliert, die beim Ablauf auf der Konsole ausgegeben werden.



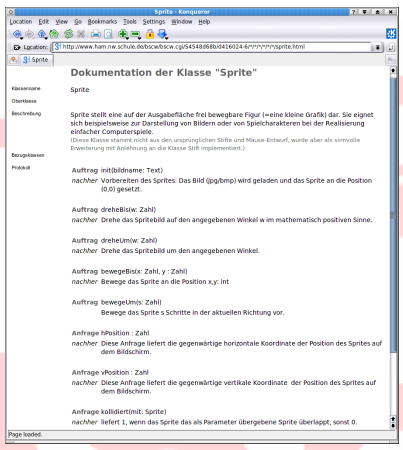
SciTE *Start des Pythonprogramms*

Darüber hinaus werden die Aktionen durchgeführt, die durch das Programm erfolgen.

Im vorliegenden Programm wird mit der Klassenbibliothek Stifte und Mäuse (SuM) gearbeitet, um die Darstellung eines aus verschiedenen Waggons bestehenden Zuges zu erstellen. Dazu wurde die von Ingo Linkweiler im Zusammenhang mit seiner Diplomarbeit ↘ingo-linkweiler.de/diplom vorgelegte Implementierung von SuM verwendet, die gegenüber dem Original um die Klasse *Sprite* erweitert wurde.

Aus SciTE *heraus laufendes Pythonprogramm*

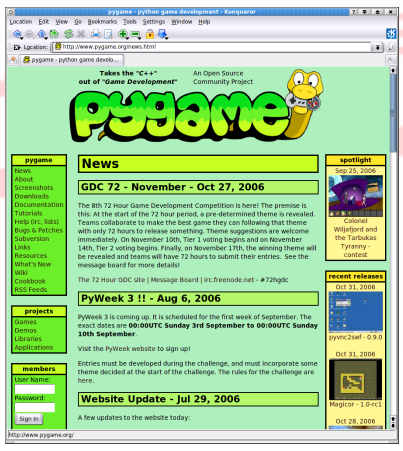
Diese kommt in dem dargestellten Beispiel zum Einsatz und zeigt, dass es gut möglich ist, diese erweiterte Klassenbibliothek auf einem kleinen Gerät zu nutzen.



Sprite Klassendokumentation – Bestandteil der Materialien SuM für Python: ↘ingo-linkweiler.de/diplom/index.htm

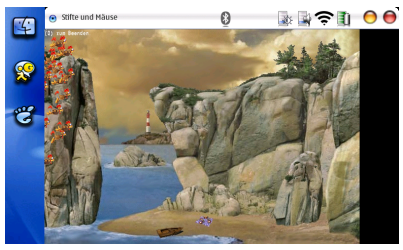
Endlich – Spiele

Die im Zusammenhang mit der Diplomarbeit von Ingo Linkweiler vorgenommenen Erweiterungen der Klassenbibliothek *SuM* in der Python-Implementierung betreffen nicht nur die neuen Möglichkeiten, die mit der Klasse *Sprite* in performanter Weise gegeben sind, sondern darüber hinaus beispielsweise die Möglichkeit der Ausgabe von Tondokumenten (realisiert - siehe Quellcode) oder Videofilmen (nicht realisiert). Die erweiterten Möglichkeiten sind darauf zurückzuführen, dass die Implementierung mit der plattformunabhängigen Schnittstelle SDL (Simple DirectMedia Layer) vorgenommen wurde, für die eine Python-Schnittstelle existiert.



pygame ↘www.pygame.org die Pythonschnittstelle zu SDL (Simple DirectMedia Layer) ↘www.libsdl.org

Die Schnittstelle erlaubt auch auf nicht performanten Geräten, wie dem hier benutzen Nokia 770, den zügigen Ablauf grafikorientierter Programme.



Der Klassiker – das Huhnspiel – Bestandteil der Materialien SuM für Python: ↘ingo-linkweiler.de/diplom/index.htm

Glossar: E-Learning

(von Dr. Ludger Humbert)

E-Learning

»Unter E-Learning (englisch electronic learning – elektronisch unterstütztes Lernen), auch E-Lernen genannt, werden – nach einer Definition von Michael Kerres – alle Formen von Lernen verstanden, bei denen digitale Medien für die Präsentation und Distribution von Lernmaterialien und/oder zur Unterstützung zwischenmenschlicher Kommunikation zum Einsatz kommen.«
[↘de.wikipedia.org/wiki/E-Learning]

Der Begriff umfasst in meiner Sicht die Nutzung von Informatiksystemen in Lernkontexten, da digitale Medien immer diesen Zusammenhang implizieren. Wer von uns hätte vor zehn Jahren prognostiziert, dass auf Informatiksystemen, die in normalen Haushalten benutzt werden, komplette Filme aus dem Internet gespeichert und geschnitten werden können. Wer hätte vorhergesagt, dass mit einem normalen Mobiltelefon kleine Filme aufgenommen werden, …

Allerdings hat die Qualität der Arbeit mit Informatiksystemen mit der Erweiterung der technischen Möglichkeiten nicht Schritt gehalten. Um mit Unterstützungssystemen des E-Learning angemessen, sach- und fachgerecht umgehen zu können, Chancen und Grenzen dieser Systeme einzuschätzen, zu bewerten und sie problemangemessen einzusetzen, ist eine mehr als fundierte Informatische Bildung nötig.

Persönliche Erfahrungen mit dem Elektrischen Lernen

In Gesprächen verwende ich häufig den Begriff »Elektrisches Lernen«, um mich gegenüber den Heilsversprechen des E-Learning deutlich abzugrenzen.
Mein persönlicher Hintergrund: als Schüler sollte ich Teile des Unterrichtsstoffes mit Hilfe der sogenannten Programmierten Unterweisung lernen. Die Umsetzung erfolgte mit Hilfe von Büchern, die nichtlinear aufgebaut waren. Nach kurzen Wissenshäppchen wurden Fragen gestellt, zu denen Antwortmöglichkeiten vorgegeben waren, die mit Verweisen zu den weiteren Arbeit führen sollten. Ich fand das nicht sehr interessant, sondern langweilig, in diesem Buch zu blättern.

Später arbeitete ich als studentische Hilfskraft im Forschungs- und Entwicklungszentrum objektivierter Lehr- und Lernverfahren (FEoLL, Paderborn). Dort wurden die Reste des rechnergestützten Unterrichts verwaltet: Lernmaschinen.

Ich habe mit dem Testen von computerbasierten Lernprogrammen Geld verdient und Scheine erworben – die Qualität der Systeme sind in vielen Bereichen nicht über das Niveau der 60er und 70er Jahre hinausgekommen. In dieser Zeit habe ich erlebt, welche Hoffnungen an die Unterstützung des Lernens durch kybernetischen Modellierung des Lehr-/Lernprozesses geknüpft wurden. Letzlich ging es schlicht darum, Lehrer zu ersetzen. Bereits damals schienen mir die Hoffnungen maßlos überzogen zu sein. In diese Zeit fällt auch die Veröffentlichung zahlloser Modelle zur Unterrichtsplanung, die die vollständige Planbarkeit zum Ziel hatten. Dem mochte ich weder als Schüler, noch als Student zustimmen und ich habe mir diese Auffassung bis heute bewahrt.