"Age of Computers:" An Innovative Combination of History and Computer Game Elements for Teaching Computer Fundamentals

Lasse Natvig, Steinar Line and Asbjørn Djupdal
Department of Computer and Information Science (IDI), Norwegian University of Science and Technology (NTNU),
Sem Sælands vei 7-9, N-7491 Trondheim, NORWAY.
Lasse.Natvig@idi.ntnu.no, Steinar.Line@idi.ntnu.no, Asbjorn.Djupdal@idi.ntnu.no

***Abstract* – Age of Computers (AoC) is an online multiplayer game used in teaching of computer fundamentals in a M.Sc. study in computer science. It supplements traditional auditorium lectures by a rich set of problems in a computer game like environment. The story of the game is linked to the historical periods (epochs) of computers, and the content is organized in rooms or places on a map. A chat window for each historical period is used for communication with other students and teaching assistants. The first version of AoC was used in a class with 250 students. Almost all students agreed that "AoC is more motivating than traditional exercises", and a majority claimed that they learn more through AoC than by traditional exercises. We are currently working on an improved version of AoC for the 2004 fall semester based on the feedback from an extensive questionnaire.**

Index Terms – Learning by doing, Collaborative learning, Educational Computer Game, Edutainment, Distance learning.

INTRODUCTION

As part of the development of a new 2nd year course in computer fundamentals for students following a 5-year M.Sc. program in computer science or telecommunications, we faced *the challenge of a large and very inhomogeneous class*. The class was expected to have 200-300 students. Some students have had computer hardware as hobby for 6-8 years, while many had no or very little knowledge of computer fundamentals.

Many of the students are interested in using computers or in aspects of its software, but the interest for important, fundamental and "invisible" concepts inside the computer is generally declining. We know that students want a *flexible learning environment* and believe that a variation from traditional exercises might be stimulating. Organizing the exercises as a computer game might give this flexibility and variety. Also, by adopting the style of a multiplayer on-line game, we hope to be able to exploit some of all the "instructional power" among the computer hobbyists by engaging them in "peer-to-peer" teaching of the "computer novices". Such *collaborative learning* will also have positive social effects. We also believe that computer history, besides being interesting in itself, might give a deeper perspective of the impressive development in computer technology.

In this context, we decided to develop Age of Computers (AoC) as a radical supplement to ordinary auditorium lectures. AoC is a web-based exercise environment that presents problems from the syllabus to the students. It is also a computer game where the player travels through the computer history from the early mechanical computers, visits the vacuum tube age, the transistor age etc. and ends up solving problems related to embedded systems of the present age.

The AoC project is grounded in NTNU and our department's priority area "ICT in learning". The development work is done by faculty, PhD students and graduate students. AoC was used during the fall semester 2003. A questionnaire showed that we met many of our goals.

The paper is organized as follows: We start with a short discussion of computer gaming, how it has been inspirational for the AoC project, and related work. Then we give a short presentation of AoC and its user interface. Following is a description of how computer history is central in AoC and how it is linked to the course contents. We then discuss AoC as a computer game, student feedback obtained by an extensive questionnaire, and observations from analyzing player data. A short presentation of the technical solutions and tools underlying AoC and our plans for future versions concludes the paper.

COMPUTER GAMES AND LEARNING

The Norwegian society has in general a very large density of personal computers and other modern IT equipment. Computer games are very popular among young people, and most students have their own PC at home. According to a recent survey, nine out of ten children aged 9-16 in the Nordic Countries play online games [1]. The Interactive Digital Software Association (ISDA) reports that 60% of all Americans play video games, 43% are women and the average age is 28 [2]. Young people have, to a large extent, moved away from watching television. Instead, they spend their time on a computer playing games or chatting with friends. As a consequence, students are used to advanced multi-media user interfaces. This background makes it more challenging for textbooks and traditional classroom teaching to fetch the students' interest. AoC is an attempt to meet this challenge.

0-7803-8552-7/04/$20.00 © 2004 IEEE

One of the main sources of inspiration for Age of Computers has been real-time strategy games such as Age of Empires II, which was the top-selling PC game worldwide in 1999 [3]. In addition to being a fun game, it provides a lot of textual, multimedia and interactive learning experiences from many disciplines, not only history.

The possibilities given by the ever improving computer technology has spurred a large interest in e-learning. The combination of education and entertainment (*edutainment*) has become a hot topic, even though several researches [2,4] report that computer games have had an image problem in that it has not been accepted as an important serious topic by academic circles. Recently, several reports on the use of computer games in education have become available [5,6]. Much of the reported activity has been at the primary and junior high school levels, but there seems to be a growing interest at the college and university level. Today, several gaming related courses are given [4], and computer games are now a topic in higher education. Best known is perhaps the MIT Games-to-Teach Project [7] where they have designed prototypes of games to support learning in advanced mathematics, science, and engineering. The project is now part of *The Education Arcade* [8] – a consortium for exploring the new frontiers of educational media that have been opened by computer and video games.

Although the integration of historical subjects in undergraduate learning of computer science has been proposed [9], we are not aware of other projects using history as part of a computer game used in teaching computer science at the university level. The goal of the AoC project is to contribute to this new and very exciting area of educational research by developing prototypes for use in large classes with several hundred students.

A Short Presentation of AoC

The goal of the student playing AoC is to walk around in different epochs of the computer history and solve problems to earn points. The journey starts outside the NTNU main building, but after a short introduction to the game, the player travels backwards in history meeting Charles Babbage in the "Age of Mechanical Computers" working on his Difference Engine. As soon as enough points have been earned, the player is allowed to proceed to the next age.

I. User Interface

The graphical user interface (GUI) of AoC is shown in Figure 1. (The text is in Norwegian). AoC is presented to the player as dynamic web-pages with the content organized in maps containing a number of rooms (or places). Every room typically contains text and images giving the right background for presenting a set of problems. The current room is displayed in the main window, which also shows the map in its upper right corner. On the map a yellow square shows the location of the current room, and red squares give the location of other rooms with problems. The player moves around by clicking on the map or by following links in the text. On top of the screen, player name, earned points and current map is shown. There is also a help button and a log-out button.

FIGURE 1
GRAPHICAL USER INTERFACE IN AOC.

The availability of rooms depends on the player's progress in the game. This is to ensure a pedagogical order of introduced topics that fits with the pace of the lectures. When entering a new room, the player typically reads through the text and tries to solve the given problems. Often this requires a lookup and a bit of reading in the textbook or information search on the Internet. *Immediate response* is given when the player enters an answer.

At the bottom of the screen there is a chat window. There is one chat channel for each epoch in the game. Typically it is used by the students to ask for help when they encounter difficulties. To the right of the chat window all the players active in this map are listed. Often students help each other, but also teaching assistants "walk around in AoC" and provide help.

Originally we wanted to offer an advanced GUI in AoC. However, strong limitations of both time and budget forced us to choose simple technical solutions for the first version.

II. Problem Classes

There are currently more than 500 different problems stored in AoC. They can be categorized into four groups according to the format of the answer given by the student:

- **Multiple choice**: The majority of the problems fall into this category. An important aspect of the mechanism used, is that after entering a wrong answer, you have to wait for a certain amount of time before a new attempt

can be made. The students dislike this the first time they experience it. However, they accept it when they recognize that it is deliberately made so to motivate for a "read, think and try approach" instead of a "trial and failure approach".

- **"Number problems"**: These are problems where the answer is a number. The mechanism is very versatile and a significant enhancement from the simple multiple choice mechanism. As an example, we have made a set of problems for checking the understanding of the addressing modes used in computers. By displaying a small set of memory locations with addresses and contents together with a few named registers with contents, it is straightforward to generate many different problems testing the students' knowledge of this topic.
- **ALU (processor) control signals**: From earlier classes we have learned that a detailed understanding of the organization and behavior of a simple arithmetic/logic unit (ALU) and data path can be tested by asking for the exact control word for a given micro operation. This has been implemented with the number problem mechanism where the answer is simply a short binary string.
- **DARK assembler**: Although *assembler coding* is used less and less frequently in the industry, we still believe it is relevant to give the students a deeper understanding of the computer. Among several simulators freely available we decided to use DARK developed by Ola Ågren at Umeå University. It offers four simple processor architectures with the same user interface [10].

To motivate the students to use AoC during the time scheduled lab-hours where teaching assistants are available, we developed a mechanism called "*Happy Hour*". During Happy Hour AoC presents a multiple choice question in the chat window every other minute (adjustable) giving the students a chance to earn extra points.

III. Integration into a Complete Course

Age of Computers was used in the course "Computer Fundamentals" fall 2003 and replaced traditional paper exercises. Since this was a completely new experience to us, we made sure we had enough assistants available. We divided the 246 students into two groups and offered each group six hours of lab each week assisted by at least three student assistants. In addition we offered a Frequently Asked Questions service on our web page where students could ask the staff when stuck in AoC. To supplement the regular lectures, we gave two hours of exercise lectures which were aimed more directly at the exercises in AoC.

To assure the progression in AoC during the semester we introduced three obligatory milestones in weeks 9, 13 and 15. One interesting thing we discovered was that many students preferred to work outside lab-hours. This made us focus some of our staff resources into improving the contents of AoC instead.

The possibility of closing and opening rooms within the game during the semester made it possible to create many of the rooms after the semester had started. Small changes to active rooms could also be done dynamically. It was an overall goal when we filled the rooms that they should be independent of the book currently used in the course [11]. This is because textbooks typically are changed every third or fourth year, while we hope that AoC will have a much longer lifespan.

LINKING COURSE CONTENTS TO HISTORY

One of the main thoughts behind AoC is to teach computer fundamentals using history as a governing idea. The game is divided into several levels that correspond to historical periods (epochs), and the student travels through computer history during the game. Information and questions from the course contents are mixed with historical information.

Using history this way is believed to be a motivational factor because the historical material is quite different from the otherwise theoretical content. Learning how modern computers evolved from the early days of computing may also give the student a broader perspective and a more humble view on the subject. AoC divides computer history into the following periods, each being a level in the game:

- **Mechanical age (1642-1945)**: The main historical events presented in this period are the inventions of mechanical computers, like Babbage's Analytical Engine.
- **Vacuum tube age (1945-1953)**: This period is about the first electronic computers that used vacuum tubes as basis for its computations. The ENIAC computer is one example.
- **Transistor age (1954-1965)**: The next breakthrough in digital logic came with the invention of the transistor.
- **Integrated Circuit age (1965-1980)**: The first computers using integrated circuits are shown here.
- **Present time (1980-today):** Recent history is presented at this level like the introduction of home computers.
- **Future (today-):** A look at future technologies, e.g. evolvable hardware.

The course material is divided into small sections of text. One such section typically explains one concept or idea from the syllabus, or it gives the student a look into a historical event. Historical sections are placed according to the corresponding game levels.

One way of linking the syllabus to the historical frame given in the game is to present the course material chronologically correct, i.e. presenting a concept at the point in the game timeline where it was invented. This would give a strong link between course contents and history, but was considered too strict to result in a pedagogically good presentation. Instead, the course material is presented in pace with the lectures, but with links to the history where applicable. Historical links to the fundamental theory (e.g. numbers and digital design) could easily be given in the mechanical age. Effects of Moore's law and introduction to current research in future technologies have a natural fit in the future. Most of the other content is evenly spread across the levels.

AoC as a Computer Game

I. Computer Game Elements in AoC

The student using AoC experiences a story that is typical for many computer games. It starts at present time by a well known building at the student's own university. Then, by traveling backwards in time the player arrives in the mechanical age and begins on a journey through the time periods (epochs) of computer history. The player's mission is to return to the starting point before the course's final examination. By exploring a map the student finds problems from the course syllabus and meets persons from the computer history. By solving problems the player earns points and advances to higher levels (later epochs). AoC maintains a high-score list that evidently has motivated many students to extra efforts.

II. Collaborative learning and the Chat Window

On-line chatting or instant messaging through services like MSN messenger has become very popular among young people, in everyday life, computer games and project work. In most on-line multi-player games it is a central part for collaboration when solving problems. AoC has one chat room for every epoch for this purpose. During the first few days AoC was running, parts of the chatting were about the interface and use of the AoC prototype. By observing the discussions the AoC design team could rapidly see what should be changed in the interface and what should be told the students about AoC in the lectures.

Throughout the semester most of the communication was students helping other students, or teaching assistants helping the students. We learned from the questionnaire that many students would rather have assistants available through the chat at several hours throughout the day than physically present in the lab at scheduled hours.

III. "To play or not to play"

The game property of AoC is meant to be motivational for the students. There is however some students that do not like computer games and would rather not use extra time on something not strictly part of the course. As a consequence, AoC is designed such that students may ignore the game property and instead concentrate on course contents and problems. By providing links that guide students through AoC, it is possible to view AoC as an on-line textbook with interactive problems.

Student Feedback

To assess the educational value of using AoC in the new Computer Fundamentals course, the students were asked to complete an extensive questionnaire. About 55% of the 246 enrolled students filled in the web-based survey, all anonymously. In our first question *"Give a rough estimate of the number of hours you are using through the week on different computer games (not AoC)"* the students were given 5 alternatives as shown along the x-axis of Figure 2. The distributions of answers for the two groups (101 male and 33 female students) show clearly that in our class computer games are significantly more popular among the male students.

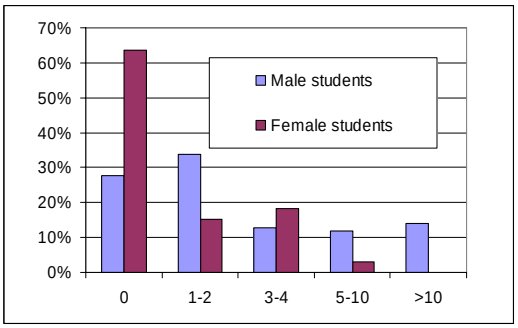

FIGURE 2
HOURS/WEEK SPENT BY STUDENTS ON COMPUTER GAMES.

Wherever it was natural, we used a Likert scale by posing a statement and asking the students whether they 'Strongly Agree (A)', 'Somewhat Agree (a), 'Undecided (?)', 'Somewhat Disagree (d)' or 'Strongly Disagree (D)'. On the statement *"AoC is more motivating than traditional exercises"* close to 95% strongly or somewhat agreed! This very positive feedback on the motivational effect of AoC combined with the fact that more than one third of the students do *not* play computer games regularly, is interpreted as a strong indication that the AoC GUI and playing rules are simple enough for *all* students. One student clarified what probably is the most important reason for the increased motivation in only two words; "*instant gratification*".

A crucial aspect that is difficult to measure is how much the students learn from using AoC. On the statement *"I perceive the learning effect by using AoC as good compared to traditional exercises"* we got the percentage distribution shown in Figure 3. This is very promising since another question showed that the students use slightly less time on AoC compared with exercises in other courses. Also, the female students are slightly more positive to the learning effect in spite of being less used to computer games!

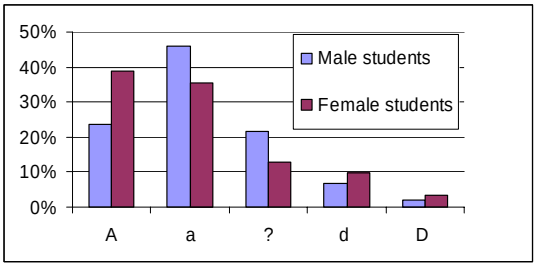

FIGURE 3
STUDENT'S OPINION ON LEARNING EFFECT (SEE TEXT.)

As expected, the DARK problems are the most time-consuming and difficult problems in AoC. We included some really difficult but optional problems to challenge the most eager students. An interesting, but perhaps not surprising result was that the DARK problems both are judged as the

most popular by some students and the most disliked by other students!

We also asked the students to suggest technical improvements of AoC. They gave highest priority to better possibilities for cooperation among players, more figures and images and better playing rules. Fancier layout or incorporation of sound was given low priority.

Analysis of Player Data

Overall the female students worked a bit more than the male students – they answered on average 45 questions each week while male students answered 35. In percentage of correct answers females got 65,5 and males got 70,1. Figure 4 shows in percentage how much work the average student puts in every week relative to their effort through the whole semester. This data is retrieved from each student's gathered points. Different exercises give different output in points depending on their difficulty.

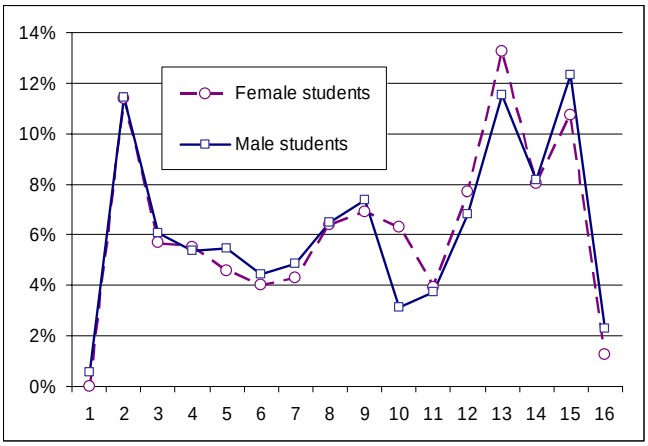

FIGURE 4
RELATIVE EFFORT PER WEEK FOR THE AVERAGE STUDENT

The curve shows that male and female students distribute their work throughout the semester in a very similar way. They are influenced by the obligatory milestones in AoC in semester weeks 9, 13 and 15. The exam was early in week 16. One experience for the next run of AoC is the need of one or two early milestones to assure more constant work by the students during the semester.

AoC Implementation and Tools

I. The AoC Software Architecture

The user interface of AoC is written in C# ASP.Net. The interactive chat window is a Java applet, and a chat server is used for coordinating the communication between the chat applet of each user. See Figure 5.

The chat server stores each chat channel's content so whenever a student logs on, the last contributions to that channel are shown. The AoC database is implemented in MySQL. It contains:

- Information about the content of each room including problems (XML presented on the web page by C#).
- Information about each epoch and the placement of rooms on the map used in that epoch.
- Information about users and user activity. This includes among other things which questions each student has answered correctly or wrong. It makes the system able to behave differently for each student depending on the progress in AoC.

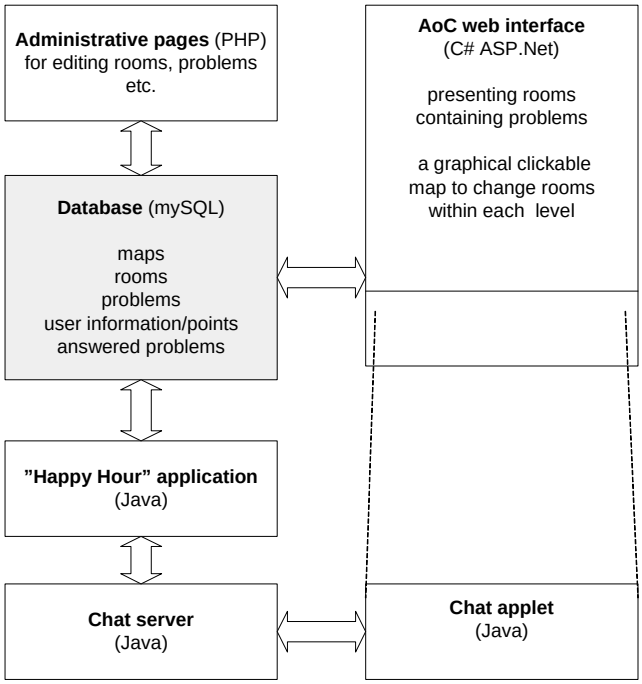

FIGURE 5
THE AOC SOFTWARE FRAMEWORK.

II. The AoC Tools

We have identified the needs for various tools to support both the development and use of AoC during the semester. These can be divided into three groups.

- Tools for *entering course contents* including a room editor, a question editor and tools to inspect which question (problem) is used in a given room. A simple map editor facilitates placement of rooms onto maps.
- Tools for *helping the course staff* to run the course and monitoring the student progress during the semester.
- Tools for *helping the students navigate* throughout the whole game. This includes a graphical presentation of the progression of a student from the first room in the game giving links to every room he has been to, and showing the remaining path to the next deadline. It is also possible to list the rooms in alphabetical order to ease the search for a specific room. These tools are important for those students that want to use AoC more as a database of exercises than a computer game.

It is also a goal that the AoC framework should be possible to adopt by other developers of new course material, not necessarily within computer science.

The AoC software framework is partly language-independent. The comments in the code are in English. The course contents, computer history and problems are written in Norwegian. The main effort involved in adopting AoC to a course in another language would be to rewrite this text. Parts of the user interface are based on buttons and standard GUI techniques such as a scroll-bar. There is some other user dialogue written in Norwegian, but it would be quite straightforward to change to another language.

CONCLUDING REMARKS

I. Experience so far

Age of Computers has been very well received by the students as a new and more motivating way to organize the exercises involved in a course. It is too early to claim that it increases learning, but a follow-up questionnaire is very promising in this respect. A few months after the examination, the students were asked to evaluate both the total and relative learning effect of AoC compared to the auditorium lectures and text-book reading. 67% rated AoC on top for "total learning", 16% rated auditorium-lectures and 14% rated text-book reading on top. With *relative learning effect* we mean the perceived learning compared to the time used on the activity. On this question AoC was clearly best, and attending the lectures was judged to be slightly more efficient than text-book reading.

We have good experience in conducting HW lab assignments for big classes with more than 400 students in a previous version of the computer fundamentals course [12]. In comparison, AoC is simpler to administrate, but we do not think that AoC should replace such hands-on experience. However, as an alternative to traditional paper based exercises we believe that AoC exemplifies a new learning approach that will become more widespread in the future.

II. Further Work

Based on the experience from the first version of AoC and the student feedback, we are currently working on the next version, AoC-2, that will be used in fall semester 2004.

By adding more course contents with related problems we will increase the learning potential of AoC. The link between the computer history and course topics will be strengthened. Both these extensions will make the "AoC world" richer, more diverse and less predictable. We believe it will make AoC more fun as a computer game.

AoC will be reorganized, both the underlying technology and the presentation to the user. An improved map and redesigned levels will make AoC much more game like. The student will move around the map, meeting historical persons and picking up items useful for completing the game (e.g. keys for opening doors). Course contents will likewise be found in the map as chapters for the student's textbook, history book and problem book; three dynamic books that will grow as the student makes progress. Using LaTeX as the underlying document format it is easy to present the books nicely both in the browser as HTML and as a printable PDF document.

The tool for helping students navigate mentioned earlier will be implemented inside AoC appearing as a "time machine" with links to all epochs giving possibilities to navigate easily.

It will be interesting to follow the future developments in e-learning and edutainment. The technology offers a plethora of possibilities and computer games may be an important factor in making e-learning more enchanting. We believe Age of Computers is a small but early step in this new direction.

ACKNOWLEDGMENT

We would like to thank Ola Ågren for letting us use the DARK source code, Bård Kjos for strong administrative support, NTNU for financial support and finally but not least the many contributors to AoC (in alphabetical order); Tanveer Hussain Awan, Bård Terje Fallan, Morten Hartmann, Ole Andreas Hegle, Ole Kristian Hoel, Lars Ivar Igesund, Per Kristian Lehre, Bjørg Peggy Sveen Lyngstad, Simon Thoresen, Håvar Valeur, Jeppe Andreas Berg Weinreich and Olaug Kyoko Namba Østhus.

REFERENCES

[1] SAFT (Safety, Awareness, Facts and Tools), "Nine out of ten children play online games", Press release 2004.01.15, http://www.saftonline.org

[2] Kirriemuir, J., "Video Gaming, Education and Digital Learning Technologies: Relevance and Opportunities", *D-Lib Magazine*, Volume 8, Number 2, February 2002.

[3] Microsoft game studios, "Age of Empires II: The Age of Kings", http://www.microsoft.com/games/age2/

[4] Squire, K, "Gaming in Higher Education part I", *Joystick101.org*, http://www.joystick101.org/story/2001/2/18/111322/124

[5] BECTA (British Educational Communications and Technology Agency), "Computer Games in Education Project", http://www.becta.org.uk/

[6] TEEM (Teachers Evaluating Educational Multimedia) , "Report on the educational use of games", http://www.teem.org.uk/publications/teem_gamesined_full.pdf

[7] MIT Games-toTeach Project, http://www.educationarcade.org/gtt/

[8] The Education Arcade http://www.educationarcade.org/

[9] Impagliazzo, J., Lee, J.A.N. and Williams, M.R., "Using History to Teach Computing", Workshop Session W2C, ASEE/IEEE *Frontiers in Education Conference*, Boulder, Nov. 2003.

[10] Ågren, O., "Virtual Machines as an Aid in Teaching Computer Concepts", IEEE Computer Architecture Technical Committee (TCCA) Newsletter, Sept. 2000, pages 72 - 76.

[11] Stallings, W., *Computer Organization & Architecture*, 6th edition, Prentice Hall, 2003.

[12] Njølstad, T. and Natvig, L., "Experience from a 450 Students/Year Course on Digital Logic and Computer Fundamentals using FPGAs and microcontrollers", Proc. of 2001 *Int'l conf. on Microelectronic Systems Education*, 2001, pp.18-19.