Teaching Computer Science by Playing

W. Hämäläinen
Department of Computer Science, P.O. Box 111, FIN-80101 Joensuu, Finland
whamalai@cs.joensuu.fi

1 Introduction

In this paper I introduce new experimental methods in teaching theory of computer science in the university. Although I have experiences only in university teaching the same methods can be applied as well in other levels of studying. Role of programming exercises, simulation and other computer-aided methods has always been emphasized in the short history of computer science teaching, but physical methods has been nearly ignored. Still it has been proved that use of all senses in demonstrations and learning by doing give the most comprehensive learning experience (Carnevale, 2000; Gibbs, 1988). In this paper I shall call all these methods in general as experimental learning/teaching.

This kind of teaching connects so called Perinetic School¹, which emphasizes use of all kind of new creative and revolutionary methods in teaching/learning. As the Perinetic School is still quite unknown I shall first describe its main ideas.

The name of the Perinetic School comes from Greek (per=through, nein=to swim) and means “learning by swimming”. In the Perinetic School learning process happens by swimming like in the ancient Peripatetic School by walking. The Perinetic School was founded in the beginning of 21st century in Mekrijärvi, Eastern part of Finland. It is told that originally the students and teachers really went for swimming in shallow water to have educational discussions. Since that the Perinetic teachers may still take their students to the water, but the Perinetic School has a larger metaphorical meaning. The knowledge is referred as "Mare Nostre", "our ocean", mother of all life, creativity and imagination – something still unknown, into which the learners have to jump bravely and learn to deal with it. Like in water the beginner may be confused and afraid of sinking into huge amount of information unless fighting against it by all means, but after learning to relax one notices that the water itself bears and there is no hurry.

The main tasks of the teacher in the Perinetic School are to courage the students to jump into something new and find the existing curious and playing child in oneself. Music, playing, games and arts can be used as tools also in teaching very theoretical and abstract things e.g. datastructures and algorithms in computer science. Creative teachers have always used this kind of methods in their teaching, but there are no public reports and evaluation of them.

I shall now introduce a few experimental methods I have used in teaching computer science in the university. After that I shall try to evaluate their role in teaching and report students’ feedback. Finally I shall give some ideas, how to apply the ideas into new areas in computer science.

2 Some games about computer science

2.1 Drawing class diagrams from music

Learning the conceptual modelling of entities as objects of classes, which have their own attributes, methods and relations is quite hard for some students. That kind of object-oriented thinking should first be practised with common things surrounding us in real world. The students are not motivated to read and mecanically analyze several application descriptions, nor do they learn to understand the underlying modelling paradigm. Instead the teacher

¹The school was founded in An International Summer School on educational technology, University of Joensuu, Aug 2002.
may give the task in the form of music: the students listen to music and try to draw a class diagram of the story in lyrics. The first step can be to draw an object diagram, which is then 
generalized to the class diagram.

For example when I was lecturing "Introduction to Application Design" (Hämäläinen, 2000a) I let my students listen to a song "Viimeinen kylähullu" ("The last village nut") by 
Juice Leskinen. The story tells about a benevolent but simple man, who helps everybody in 
the village but is considered as a useless nut by the society. In the song the state tries to 
pull such people into mental hospital. Students experienced this very surprising but fascinating 
beginning of the lecture and I encouraged them to do the same at home, when they are practising 
for the exam.

The lyrics and resulting class diagram are shown in the appendix.

2.2 Simulating object collaboration diagrams

Object collaborating diagrams show, how and in which order the objects cooperate to carry 
out some task. Every object has a set of methods, which it can execute based on its data 
contents or return values of service calls to other objects.

This kind of collaboration can be played as a game, in which individual students or groups 
of students play objects. Each object is given one or more methods – usually very simple 
algorithms – and some data contents. To execute the service at least some of the methods 
require service calls to other objects. Each student or group is given a name card so that the 
objects can be easily recognized. Service calls are simulated by running to the right object, 
asking the service and waiting for response. If there are enough students, the whole group 
can be divided in parts, which compete against each other.

For example I have used this game after the previous one in "Introduction to Application 
Design". First we considered, what kind of methods we should add to the class diagram 
to implement the task "Find all village nuts and put them into mental hospital". Then we 
completed the collaboration diagram with order of execution. After that we shared the roles 
and began the game, in which two competing states were hunting their village nuts. After 
some confusion the game succeeded quite well and afterwards even the sleeping students were 
awaken and motivated to know, what was the idea of the game.

PhD Harri Laine in the University of Helsinki has also tried similar play, in which the 
students simulated the bottle recycling machine (Laine, 1996).

2.3 Simulating abstract datastructures

Also the abstract datastructures and their algorithms can be simulated by playing. The 
students may construct linked lists, binary trees, graphs or heaps. Each student is playing a 
node in structure and one student may be a pointer. The students are given some data, for 
example numbers or letters, which they should order by the algorithm or search the biggest 
one. Another interesting game is playing a balanced binary tree, in which the inner nodes 
have arithmetic operations and the leaf nodes have integer numbers, and the main task is to 
evaluate the expression.

I have tried this game only in the Christmas party of Department of Computer science, in 
which the participants varied from undergraduate 2nd year students to professors. The game 
(on in fact a competition between several groups) succeeded well and it was believed to be 
suitable also in real teaching.

2.4 Simulating finite automata

In this game the group of students is given a transition diagram of a finite automaton. They 
may look at it for a while and then they have to construct the same automaton themselves. 
Each student corresponds a state of the automaton and knows only, how the automaton
behaves in one's state i.e. is s/he the final state or to whom one should transfer the control. The string is given to the initial state as a stack of paper cards, one symbol in each. Every state reads one symbol, takes off the card and gives the rest to the following state.

The final state reports, if the string was accepted or not. This game I have only tried in the Christmas party, but this time the idea of finite automata was forgotten or new for more participants. Especially those participants gave good feedback about the game: they found it really enlightening demonstration.

2.5 Some related experiments

I have also tried some other experiments, which don't require concrete playing, but still attract imagination an use of all senses. For example the Finnish salt liquorice sweets are usually black and diamond-shaped like the composition symbol in class diagrams. Sweeties can be used as counters when completing the class diagram with composition relations. Afterwards the salt liquories can be eaten by meditating the meaning of the composition: "A salt liquorice addict is as existentially dependent on salt liquorice as it were a composition relationship."

Writing poems seems also variety to the scientific texts. One Finnish teacher in physics has given his students a task to write poems about physical phenomena and laws. I developed the idea further and gave an exercise to program a poet generator, which should produce poems by given grammar (Hämäläinen, 2000b). The idea was to use the technique of implementing a finite automaton, but in this case the automaton didn't read strings but generated strings by selecting a random transition from the current state. Some students owned themselves very ambitiously for this task and the resulting poems about the Pumping lemma, automata and grammars were also usable in the later teaching (Hämäläinen, 2001).

3 Evaluation

I don't have any statistics to evaluate the use of playing and other experimental methods in computer science teaching. The course official feedback about innovativity of teaching covers other things, too. But I have got verbal feedback about playing immediately and afterwards in feedback forms.

The students seem to be eager to try everything new, if the teacher encourages them from the beginning. That's why the first year students are easiest group for such experiments: they haven't got used to the traditional university teaching yet. Still all the students can be lured to try new methods, if the atmosphere in the course is open and the teacher is oneself innovating. For example in my course "Introduction to Application Design" the students varied from first to third year students and the oldest ones were middle-aged. Still nearly all (except one, who didn't give any reason) took a part the object collaboration game.

The feedback of students (and also exercise teachers, who hear only the stories the students tell) has been good. The traditional teaching has been considered quite boring and there might really be a danger of falling asleep. Concrete playing gives wanted change and stimulate the students to study also the theory more carefully.

The games also work as problems, which wake the interest to understand the secrets of diagrams, structures and algorithms. Even if the game itself proves to be a confusion, the students are afterwards motivated to learn, what was the idea, what they should have done, and the play may be tried again. The students may also be asked to invent new games in the context of the course, which serves as learning by teaching.

4 Final remarks

The atmosphere in university teaching at least in computer science has been rigid for new teaching methods. Yet all abilities of human being should be utilized in learning. Playing is part of human being and it should be encouraged in all learning and investigating. Playing has
been shown to be inspiring both for students and teachers. The games and plays concretize abstract things, courage in understanding the underlying theory and offer wanted change in monotonic teacher-centered lectures.

The previous examples are free to be utilized and developed further. The same kind of plays can be applied to several areas in computer science. It is hard to give any generic recipe for generating new plays, but the following ideas seem promising:

- Message passing/routing in the network
- Graph algorithms, like finding the shortest path or Hamiltonian cycle in the network or solving the Travelling Salesman Problem. This is especially suitable for simulating parallel algorithms, in which the group of students travels the graph and labels the nodes.
- Resurse sharing protocols, like dining philosophers, readers and writers, time-stamp method.
- The idea of recursion in algorithms. The algorithm itself should be interesting like Hanoi towers.
- Turing machines.
- Finding the pair, who has the equivalent expression. The students can be given e.g. relation algebra, row calculus and column calculus expressions (or equivalent SQL-queries) and they have to find the corresponding expressions.
- Sorting algorithms.
- Pigeon-hole principle and its applications.
- Listening music and constructing a relational model of the song or answering the given SQL-query on it (the song should be selected carefully so that the lyrics really describe meaningfull entities and relations).
- Simulating SQL-queries. Each student is given a data record and they have to answer the query consisting group by expression (e.g. counting the number of members in the group and calculating the average/min/max value of selected data field.)
- Coding and decoding. This game is easy to implement in the class room: one of the students invents a message and codes it. The coded message is transferred to another student, who has to decode it. For example professor Pasi Fränti has simulated Huffman coding with this game. Also suitable for encrypting methods.

5 Thanks
I thank professor Timo Alanko, professor Pasi Fränti and FM Reijo Siven for their play ideas.

References
URL http://chronicle.com/free/2000/05/2000053001u.htm

URL http://www.chelt.ac.uk/gdn/gibbs/
Hämäläinen, W., 2000a. Johdatus sovellussuunnitteluum/introduction to application design (course homepage).


URL http://www.cs.helsinki.fi/u/laine/syme/syme_h5.txt
6 Appendix

VIIMEINEN KYLÄHULLU
Hänest auton tuo uuden kodin luo,
taivas vettä valuttaa.
Kapuloidalin suin, ilmeen masennetuin,
häntä pitää talutaa.
Väkivalloon viedään noihin
ihmiskunnan arkistoihin
kylähullu viimeinen.
Hän oli tärkeä mies, hän arvonsa ties,
mies täynnä tarmoa.
Aina ketä vaan riensi auttamaan
eikä kerjänyt armoa.
Aina kaikkensa hän antoi,
lumet loi ja postit kantoi
kylähullu viimeinen.
Nyt on valtio kylään tullu
ja on siellä kuin kotonaan
ja viimeinen kylähullu
pois höidetään.
Kylän kahvilaan iloin istumaan
hän saapui kaskuineen
ja yhtenään, hyvä hyvyyttään,
koitotta kiitos laskuineen.
Ehkä oli hän liian viisas,
se ihmisille püsas.
Hän on
kylähullu viimeinen.
Mutta kylän uus saapui arvokkuus
viran sosiaalisen.
Otti kohteenkseen hulun vanheneen,
sittä sekosi kaali sen.
Kun maailma tasapäättyy
niin persoonat sivun väästyy.
Hän on
kylähullu viimeinen.
On valtio kylään tullu...
On valtio kylään tullu...
Joskus paikka tää tulvi elämää,
nyt se kaatui paperisotien.
Se autoitoi, kuolee puistossakaupu,
väki viedään hoitokotiin.
Missä syy on moiseen vainoon?
Nämä päätään tasapainoon,
Pois viedään
kylähullu viimeinen.
Kylähullu viimeinen...
Kylähullu viimeinen...
**Figure 1:** The lyrics and the corresponding class diagram