Creativity and Intrinsic Motivation in Computer Science Education: Experimenting with Robots

Mikko Apiola
Department of Teacher Education and Department of Computer Science
University of Helsinki, Finland
mikko.apiola@helsinki.fi

Matti Lattu
IT Department
University of Helsinki, Finland
matti.lattu@helsinki.fi

Tomi A. Pasanen
Tumaini University1
Iringa University College
Iringa, Tanzania
tomi.pasanen@cs.helsinki.fi

ABSTRACT
This paper describes our teaching experiment at the University of Helsinki, where the aim was to investigate a possibility to provide a learning environment supporting a combination of creativity, intrinsic motivation and the use of robots to bring constructionism into computer science (CS) education. To put the ideas into practice we developed a pilot course targeted at intermediate-level CS students. The learning objectives were not strict, but instead the attendees could participate in setting them and also take part in designing future uses of robots in CS1 and CS2 courses. There was no teaching in the traditional sense, but instead we arranged workshops based on creativity-enhancing working methods.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer and Information Science Education – Computer Science Education, Curriculum

General Terms
Experimentation, Human Factors, Design

Keywords
Robotics, Open Learning Environment, Creativity, Motivation, Constructionism.

1. INTRODUCTION
Reasons to start the activity described in this paper include our interest in investigating active, student-centered pedagogical arrangements to contrast our traditional lecture, exercise-set and exam pattern. This interest is related to development of student empowerment to increase commitment and support for creativity to bring forth new ideas. Another aspect of our interest involve the use of robots, which are often reported as being used in basic programming courses, but are also seen to offer a framework for more advanced domains, e.g. artificial intelligence (see e.g. [10-14, 16, 17]). After some preliminary planning and research [15] the University of Helsinki Department of Computer Science decided to purchase 60 LEGO Mindstorms sets.

The experiment described in this paper was carried out in spring term 2009. The task for students was to define a robotics task suitable for CS1 or CS2 students and create a robot for an example of a solution. Students were required to build and program a robot, document the solution and clarify which CS1/CS2 course objectives the task rehearses. Students were encouraged to explore, have fun, and amaze us and the other participants. The course was targeted for intermediate level studies and was optional in the curricula. All students were interviewed at the beginning and the end of the course to gain understanding of the students' viewpoint on this experiment.

2. THEORETICAL FRAMEWORK
A learning environment can be defined as the factors that define the context for studying and learning (for example [18]). These factors are many, and include for example the learner's personality, the social environment and the physical surroundings. These factors may be divided to internal (psychic) and external (physic) factors, although the distinction is often hard to make. Meisalo and Lavonen [21] define an open learning environment by using a metaphor of a market place. At the market place the learner transacts with those market stalls that best fulfill her learning needs.

The roots of the concept of an open learning environment are related to the cognitive revolution in the 1970s and the development of educational sciences in the direction of constructivistic learning theories, often seen in opposition with instructivistic learning theories. Thus, starting from the 1990s there has been great debate between the acquisitional or participational nature of learning [27], for example. Yet, as Sfard [27] points out, it is good to remember the dangers of choosing just one viewpoint. In designing a learning environment, the cleverest approach is probably not to take any single viewpoint, such as instructivism or constructivism, but to aim for a proper mix of different approaches.

2.1 Intrinsic Motivation
According to Ryan & Deci's Self-Determination Theory (SDT) [24], a person has an inborn tendency for self-determination, to use one's own capacity for achieving optimal goals. This intrinsic motivation means performing some activity for the sake of the activity itself, not for example to accomplish rewards extrinsic to the task at hand. Ryan & Deci [24, 25] define three basic needs...
which they perceive as crucial for general well-being and motivation; competence, autonomy and relatedness. In an academic context, intrinsic motivation can be supported by a) social interaction that enhances feelings of competence, b) optimal challenges, effectance-promoting feedback, and freedom from demeaning evaluations, c) supporting internal perceived locus of causality, and d) providing choice, acknowledge of feelings and opportunities for self-direction [24]. While intrinsic motivation is a favourable condition in itself, it is also often seen as one necessary component in creativity (see for example [1, 22, 28, 29]).

2.2 Creativity
Creativity is often defined as the producing of original, unexpected and useful work (for example [28]). Herrmann [8] summarizes a variety of creativity definitions: “the ability to challenge assumptions, recognize patterns, see in new ways, make connections, take risks, and seize upon change”. It seems that creativity and creative skills have become something regarded highly important in almost every field of work and education (e.g. [4]). Pioneers in creativity research have come to share somewhat similar views; that creativity requires three components; domain-relevant skills (expertise and talent in the task domain), creative processes (cognitive skills and work styles) and intrinsic motivation (for example [1, 28, 29]). The collaborative aspects of creativity is stressed by Sawyer [26], who argues that there is an important link between collaboration and creativity; creativity should not be viewed only from the viewpoint of individuals, but from the viewpoint of collaborative groups.

Creativity-enhancing processes are based on the thought that creativity requires an environment which encourages risk taking and self-initiated projects and provides help and time for developing ideas and individual effort. The literature suggests several methods for enhancing creativity, of which methods especially intended for concrete problem-solving situations include for example; brainstorming [23], verbal check-lists [5, 23], picture stimulation and mind mapping (for example [3]). Higgins [9] has also provided many methods for enhancing creative problem solving. A general idea with many of these methods is the purpose of generating ideas by suppressing the common tendency to criticize or reject ideas using different types of games or tasks. Collaborative methods include for example Open Spaces Technology (OST) [7] and a method developed at the University of Helsinki Teacher Education Department, named $3+ [19]$.

2.3 LEGO Mindstorms and Constructionism
LEGO Mindstorms is a robotics kit based on a small microcontroller unit, a set of input sensors, motors and parts for building robots. The robot can be programmed natively with NXT-G visual programming software which comes bundled in the micro-controller. However, there are a range of third-party programming languages to allow significantly more sophisticated programming. NXT-G is adequate for very basic programming and learning simplified programming structures and flow control whereas e.g. libraries for Java [20] and C/C++ [2] enable to utilize the full potential of more advanced programming languages. There are reports of many positive teaching experiments using Lego Mindstorms in CS1 courses and also at more advanced domains (see e.g. [10-14, 16, 17]).

Constructionism is a learning theory developed by Seymour Papert [6], and is based on constructivist learning. Papert has worked closely with LEGO Mindstorms and also for example, with the LOGO programming language. Constructionism holds that learning happens most effectively in situations where people actively make tangible objects in the real world. Constructionism is connected with experiential learning and builds on similar ideas of Jean Piaget on constructivism. Papert himself defines constructionism as follows: "From constructivist theories of psychology we take a view of learning as a reconstruction rather than as a transmission of knowledge. Then we extend the idea of manipulative materials to the idea that learning is most effective when part of an activity the learner experiences as constructing a meaningful product [6]."

2.4 Guidelines for the Teaching Experiment
The learning environment of the teaching experiment was designed to offer maximum support for intrinsic motivation (as it is defined in Ryan & Deci’s SDT Theory), creativity, and constructionism. The components of intrinsic motivation are competence, autonomy and relatedness, and the components of creativity are intrinsic motivation, domain relevant skills, and creative processes and working styles. Since intrinsic motivation is defined as one component of creativity, creativity inherits all the components of intrinsic motivation. Thus, in attempting to support both intrinsic motivation and creativity, we are left with five components; competence, autonomy, relatedness, domain-relevant skills, and creative processes and working styles. Table 1 shows all our components and literature-derived ideas customized to our environment on supporting each component.

Table 1. Components of our Learning Environment

<table>
<thead>
<tr>
<th>Component</th>
<th>Method of support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competence</td>
<td>Encouraging teamwork, use of creativity enhancing methods, providing effectance promoting feedback</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Providing choice and opportunity for self-direction</td>
</tr>
<tr>
<td>Relatedness</td>
<td>Promoting social interaction and creative working methods</td>
</tr>
<tr>
<td>Domain relevant skills</td>
<td>Requiring good computing skills from all attendees.</td>
</tr>
<tr>
<td>Creative processes and working styles</td>
<td>Use of creativity-enhancing methods in course sessions: brainstorming, 3+ [19], and open space workshops [7].</td>
</tr>
<tr>
<td>Constructionism</td>
<td>Use of LEGO-mindstorms robots. Lend each student their own robotics-kit.</td>
</tr>
</tbody>
</table>

3. RESEARCH QUESTIONS AND DATA ANALYSIS
The purpose of this study was to find out how our approach to support intrinsic motivation, creativity and constructionism would succeed. We were also interested in gaining information about the possibility of using robots in computer science education. The main research questions (RQ) for this study were.

I. Teachers perspective: (how) is it possible to support intrinsic motivation, creativity and constructionism in CS education by following the principles listed in table 1.

II. Students perspective: how will the students experience our learning environment.
To answer these questions, we 1) collected detailed observation notes from every learning session and 2) interviewed all course attendees at the beginning and the end of the course. The interviews were transcribed and analyzed by finding common patterns in the transcribed notes using ATLAS.TI – qualitative analyzing tool.

4. RESULTS
In section 4.1 we will answer research question I by describing our course Robotics Programming Project, its student population, course meetings, and outcomes. Section 4.2 will answer research question II by explaining our student interview data.

4.1 Description of our course (RQ I)
To explore our ideas, we set up a voluntary programming course. The recruitment for the course started from a message on the student mailing list and an announcement on the web pages of our department. Despite the short notice, the course turned out to be highly popular with an enrolment rate of 42 students. Of these 42 students, 36 showed up at the first course session. There were approximately 25 active students in the course. Considering that the announcement for our course came late, only two weeks before starting, the course can be considered highly popular. The attending student population (n=36) ranged from 1st year to postgraduate students, with an average study year of 4.2. The age of the attendees ranged from 18 to 40, average 25.76. All students had relatively good computing skills, since we required that all students must have passed Data Structures, which is regarded as one of our hardest intermediate level courses. Thus, we had a very diverse but able population to work with. A total of 18 students completed the course. The drop-out rate is very normal at our department, where students are allowed to register for courses and drop out at any time they wish without consequences; many register for a lot of courses but complete only some of them in a study term. The main reason given by students dropping out was the optional status of this course in the curricula; the mandatory courses were preferred as more important.

There were three of us teachers working as instructors. We had six course meetings. At the first meeting we explained the general idea behind the course and the practical arrangements. The Mindstorms tool was presented, and every student was lent his own Mindstorms kit for personal use during the course. At the second session we experimented with a creativity method called 3+ [19]: all students were divided into groups, set in circles, where everyone in turn had to present some idea regarding building the robot. Then the next student in the circle had to come up with three positive things about the idea and after that give critique in a positive form. The idea was to create a creativity-enhancing, psychologically safe environment for ideas. At the third meeting we experimented with a method based on open space workshops [7], where groups of students would prepare all their ideas, possible solutions and problems in posters. After preparing the posters half of the group would stay to present the posters half of the group would explore the posters of the other groups. All our meetings followed a similar pattern aiming for a psychologically safe atmosphere for ideation and reduction of a common tendency for criticism. A short synopsis of our learning environment. There probably was quite good support for autonomy: there was plenty of opportunity for self-direction, probably even too much for some students. The provision for relatedness seemed successful; although some students did question the need for the course sessions in the interviews, the same students described the value of comparing their work to peer groups, how seeing what others were doing made them not feel isolated. The component of domain-relevant skills was supported by requiring good computing skills. Creative processes and working styles were supported by creativity-enhancing working methods at group sessions, where we used 3+ and open space workshop methods. This turned out to be a success, as previously

There were a few extraordinarily stable end products resulting as student coursework. A lot of products did have a lot of effort, but most end products were a bit unstable. Table 3 shows a few of the robots; a checkers-playing robot, a tic-tac-toe robot and a guard robot shooting rubber bands. There were also many other subjects and ideas that can probably be used later for example as problem sets in CS1 or CS2 courses.

<table>
<thead>
<tr>
<th>#</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Information and instructions. Borrowing of robotics kits.</td>
</tr>
<tr>
<td>2.</td>
<td>Creativity meeting. Short lecture on creativity methods. Divide students into groups. Ideation exercises with the 3+ [19] method.</td>
</tr>
<tr>
<td>3.</td>
<td>Creativity-meeting. Divide students into groups. Continue on creative working methods. Prepare posters. Use open-space workshop-method [7].</td>
</tr>
<tr>
<td>4.</td>
<td>Creativity-meeting. Generate ideas using creativity methods in one large group. Discuss the phase and ideas of every group’s project. Use open space workshop [7].</td>
</tr>
<tr>
<td>5.</td>
<td>Creativity-meeting. Generate ideas using creativity methods in one large group. Discuss the phase and ideas of every group’s project. Use open space workshop [7].</td>
</tr>
<tr>
<td>6.</td>
<td>Final demo. Every group presents their final project.</td>
</tr>
</tbody>
</table>

Let us now consider the results contrasted to each component of our learning environment. There probably was quite good support for autonomy: there was plenty of opportunity for self-direction, probably even too much for some students. The provision for relatedness seemed successful; although some students did question the need for the course sessions in the interviews, the same students described the value of comparing their work to peer groups, how seeing what others were doing made them not feel isolated. The component of domain-relevant skills was supported by requiring good computing skills. Creative processes and working styles were supported by creativity-enhancing working methods at group sessions, where we used 3+ and open space workshop methods. This turned out to be a success, as previously
described. It is quite clear that the role of the robots in supporting constructionism was essential in influencing the students’ curiosity throughout the course. They provided the students with a lot of technological challenges and enjoyment of problem solving. It seems that our learning environment indeed was supportive for creativity, intrinsic motivation and constructionism.

We argue that we did indeed succeed in producing a learning environment supportive of intrinsic motivation, creativity and constructionism. As an output of the course we also gained a lot of practical material and information to be used in CS1/CS2 courses.

4.2 Student experience (RQ II)

The interviews were personal and they were recorded and transcribed. At the first interview, which was held at the beginning of the course we had asked students to describe the process of deciding to attend the robotics programming project. At the second interview, which was held at the end of the course, students were asked to describe in detail the process of building the robot. The students who dropped out were especially asked to describe the conditions at the moment of deciding to drop out. Students were also asked for suggestions to improve the course.

As their reasons for attending the robotics programming project, students mention visions of playfulness, fun and practical work to contrast with more theoretical studies. Often the novelty value was also mentioned; the course with its new technologies (robots) and a new kind of pedagogical arrangement was seen as new and amazing. Often the Lego trademark was mentioned to make a mental association to free playing experienced in childhood, where one could build creatively without restrictions. Other reasons mentioned include the physicality of the apparatus; the causality of a program could be perceived with ones senses, the program being not something just running in the depths of the computer, but instead actualizing in real life, moving and communicating with its environment. Working in pairs was also mentioned as motivating. Some students mention an interest in robotics and artificial intelligence and sometimes previous experiences in programming robots.

The descriptions of the ones who completed the course have many similarities. The descriptions include phases such as ideation, implementation, designing, testing, documentation. The working processes of students could roughly be divided to top-down and bottom-up types of approaches. In the top-down approach students did choose some topic at the beginning of the course, and then the development of the project was approached from around the selected subject. The bottom-up approach could be described as follows: students carried out a lot of research and experiments aiming to discover the limits of this new technology before starting any planning; for example they made measures about the accuracy of the sensors and motors. It seems that the bottom-up approach gained better end products than did the top-down approach.

Students reported a lot of unpredictable challenges during the project. Students reported technology-related challenges, and in many cases the students’ viewpoint was that there was a problem with the technology; the data from the sensors was inaccurate and there were for example inconsistencies in the documentation of the programming environments, for example. In some cases, the data structures provided by the programming language did not work properly. In many reported cases of technology problems one could ask if the problem is really in the technology or could the students be able to do better groundwork for example in finding out how do the sensors really work? In many cases the students also report a lot of challenges related to ideation and to general uncertainty related to less support structures in the course than they were used to.

Students report a lot of positive emotions, joy and excitement during the project. There were also times of frustration. Long detailed reports of attempts at technical solutions, some of which could be described as strange or even crazy, certainly seem to approach creativity. One team, for example, used remarkable effort trying to fix a bad signal from a color sensor, in aiming for recognizing the color of a checkers piece on the checkers board. Another student reports the most original ways of debugging his program. In general, the projects certainly went in very original directions, and probably much learning happened in a trial-error way.

The meetings and methods practiced were reported as useful. Although some students stated that for them the meetings were mostly useless, when asked to elaborate, they described how they for example felt relief when realizing that many students were struggling with the same kind of problems as themselves. Some students did hope for more support structures. Many hoped for an introductory section into the building and programming, and to get to know the technology and sensors better. Students also hoped for better technology, mostly more accurate sensors.

From the process descriptions it can be noted that students faced many challenges and used great effort on finding solutions. Many of the challenges were related to incorrect presumptions of the performance of the technology; the sensors and motors, and inexperience in the problems of robotics; i.e. working in an unstable environment. Also the ideation and the fact that there were no traditional support structures, such as technical help, instructions or traditional teaching, was seen as challenging for many. We argue that the amount of support structures is an issue worth considering. Since most of the teaching is quite tightly structured, should the future computer science programme include training where students would be required to cope in an environment with less support structures?

As a conclusion for research question II we argue that the robots seemed to work as a powerful trigger of the initial curiosity and motivation of students. Students report that working as pairs was motivating, and that the working methods practiced at group sessions were also motivating and useful. Most students report a lot of challenges relating to the technology and to less support structures than they are used to. A lot of information regarding the technology and ideas for further development of this activity was also gained.

5. CONCLUSIONS AND FUTURE

This paper described our teaching experiment at the University of Helsinki, where the aim was to create an open learning environment to support creativity, intrinsic motivation and constructionism. This experiment is related to ideas of investigating more student-centered pedagogies in contrast with our traditional lecture, exercise set and exam pattern. Based on this study we gained important knowledge for developing more student-centered pedagogies at our university, and also a lot of useful practical information for using robots in CS1 and CS2 courses in our curricula.
After the spring 2009 course we held a successful small teaching experiment in autumn 2009 involving four first year students with strong computing backgrounds. At the beginning of autumn season 2010 we are starting another round of the teaching experiment described here improved with a short, structured introductory period, and technical tips and all the wisdom from spring 2009. We are also setting social media structures, so the students can better report their problems to each other better, teach and learn together.

6. ACKNOWLEDGEMENTS
The authors would like to thank the students at the Robotics Programming Project during spring 2009 and University of Helsinki Computer Science Department head of studies Jaakko Kurhila for inspiration and broad support of this activity.

7. REFERENCES

Proceedings of the 38th SIGCSE technical symposium on Computer science education (Covington, Kentucky, March 2007).