

# Higher Education—Educating for Higher Order Skills

Rachel Or-Bach

The Max Stern Yezreel Valley College, Yezreel Valley, Israel

Email: [orbach@yvc.ac.il](mailto:orbach@yvc.ac.il)

Received June 3<sup>rd</sup>, 2013; revised July 4<sup>th</sup>, 2013; accepted July 11<sup>th</sup>, 2013

Copyright © 2013 Rachel Or-Bach. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preparing college students for a knowledge-based economy is a challenge that requires curriculum design that puts more emphasis on learning skills than on content to be taught. Cognitive skills should be practiced in a context of some content, but the choice of content, the choice of the learning environment, and the choice of the assessment procedures can enhance the development of such skills. In this paper we present these choices for a course that was specially designed to provide a motivating and engaging context that requires the use of higher order cognitive skills. The title of the course is “Design of computer-based games and interactive stories” and it is provided to students with no prior exposure to computer programming. At the end of the course students are required to submit an interactive artifact (a game or a story) implemented in Scratch, which is a visual programming environment. In this qualitative study we present the results from a thematic analysis of students’ post-course reflection reports.

*Keywords:* Programming; Higher Education; Higher-Order Skills; Games

## Introduction

Preparing college students for a knowledge-based economy is a challenge that requires curriculum design that puts more emphasis on learning skills than on content to be taught. Fal-lows et al. (2000) claim that today’s challenging economic situation means that it is no longer sufficient for a new graduate to have knowledge of an academic subject; increasingly it is necessary for students to gain those skills which will enhance their prospects of employment. Employability skills include the following abilities: the retrieval and handling of information; communication and presentation; planning and problem solving; and social development and interaction. Current expectations from college graduates include skills for effective use of information technologies (Kaminski et al., 2009) and skills of self-learning required for lifelong learning (Candy, 2000; Dunlap & Grabinger, 2009). The amount of knowledge and the pace of change make it necessary to prepare graduates to be long life learners. Critics of college education in the US say that US colleges put low priority on student learning (Keeling & Hersh, 2011; Arum & Roksa, 2011).

Black et al. (2006) discuss the concept of “learning how to learn” (LHTL), characterizing it as a “collection of good learning practices... in both individual and collaborative contexts that seem to have the most potential to promote pupils’ autonomy in learning” (p. 130). Learner autonomy is highly important because “it implies that the learner can not only give meaning to the learning, but that she can also create new learning tools” (p. 129). Black et al. (2006) observe that successful learners use knowledge of cognition and self-regulating mechanisms, both aspects of meta-cognition, to monitor their own understanding.

Cognitive skills should be learned in a context of some content, but the choice of content, the choice of the learning envi-

ronment, and the choice of assessment procedures can enhance the development of such skills. In the study we present in this paper we made these choices with the aim to provide a motivating and engaging context that requires the use of higher order cognitive skills. We wanted to provide a powerful learning environment (Smeets, 2005). To achieve this aim we designed a programming course for the Behavioral Sciences students in our college. During this course, titled “Design of computer-based games and interactive stories”, students learn to program using a visual programming environment. The visual programming environment that we used, along with the special activity of game design, provide a context for effective self-assessment, analysis, hypothesizing, testing, debugging, experimenting and reflecting. All these higher-order cognitive skills are essential for self-learning. The game design provides a context for exercising the iterative processes that are required from any project design and implementation. It is especially important when the resulted artifact is expected to be used by others. Such skills of project design and implementation, sometimes called design thinking, are appreciated by employers and should be part of higher education. Game design is a suitable context for developing these skills (Hayes & Games, 2008).

The following section presents related work pertaining to potential benefits of computer literacy in its broader interpretation including programming, and the potential benefits of games to higher education and to the development of higher order cognitive skills. The third section of this paper describes the main characteristics of the whole learning environment, including the main characteristics of the course and the main characteristics of Scratch, the visual programming environment. The fourth section describes the study, including goals, setting, tools and the findings divided to affective issues and to cognitive issues. The last section presents conclusions and a discus-

sion of the findings.

## Related Work

### Computer Literacy and Learning

Current expectations from college graduates include skills for effective use of information technologies (Kaminski et al., 2009) and skills of self-learning required for lifelong learning (Candy, 2000; Dunlap & Grabinger, 2009). The National Research Council report (1999) titled “Being Fluent with Information Technology”, defines “fluency” with information technologies as “the ability to reformulate knowledge, to express oneself creatively and appropriately, and to produce and generate information (rather than simply to comprehend it).” Fluency, according to the report, “goes beyond traditional notions of computer literacy... [It] requires a deeper, more essential understanding and mastery of information technology for information processing, communication, and problem solving than does computer literacy as traditionally defined.” According to this NRC report, skills associated with programming play a “central role” in fluency. The algorithmic thinking inherent in programming, writes the NRC, “is essential to comprehending how and why information technology systems work as they do.” In addition, the report argues that “the continual use of abstract thinking in programming can guide and discipline one’s approach to problems in a way that has value well beyond the information technology-programming setting.” As students create computer programs, they not only learn important mathematical concepts, they also gain a deeper understanding of the process of design: how to create a working prototype, experiment with it, debug it when things go wrong, get feedback from others, then revise and redesign it. Through this process, students develop many of the 21st century learning skills that will be critical to success in the future: thinking creatively, communicating clearly, analyzing systematically, collaborating effectively, designing iteratively, learning continuously (Resnick, 2007). Many others (e.g., Papert, 1980; Kay, 1991; diSessa, 2000) have made similar arguments on the benefits of learning to program. Graphical interfaces and multimedia capabilities makes it even more appealing and provide more opportunities for having meaningful feedback and clear demonstration of programming constructs. This was implemented in several programming environments designed for novice programmers (Kelleher & Pausch, 2005; Guzdial, 2004). The programming environment that we used in our course is Scratch, which will be discussed in a following section.

### Games and Learning

The use of computer games for learning is widely advocated. games constitute potentially powerful learning environments for a number of reasons (Oblinger, 2004): 1) they can support multi-sensory, active, experiential, problem-based learning; 2) they favor activation of prior knowledge given that players must use previously learned information in order to advance; 3) they provide immediate feedback enabling players to test hypotheses and learn from their actions; 4) they encompass opportunities for self-assessment through the mechanisms of scoring and reaching different levels; and 5) they increasingly become social environments involving communities of players. Gee (2003) stated that games are very suited to the development of inquiry skills; children learn by formulating hypotheses,

and testing them.

Educational games can provide a higher level of intrinsic motivation than a traditional school context (Tüzün et al., 2008). Players are confronted with problems they must overcome if they want to reach their goals. They are faced with a stream of both long and short term decisions, and must plan problem solving strategies which involve monitoring a series of complex tasks and nested sub-tasks (Johnson, 2005). A study to assess the learning effectiveness and motivational appeal of a computer game for learning computer memory concepts, showed that the gaming approach was both more effective in promoting students’ knowledge of computer memory concepts and more motivational than the non-gaming approach (Papastergiou, 2009).

Computerized games can provide a context for developing various skills, but constructing such games seems to entail even wider educational benefits. Robertson and Howells (2008) argue that authoring a game can engage students in authentic rich tasks offering a good degree of learner autonomy. Vos et al. (2011) report on study where they compared using a game versus constructing a game. The results suggest that constructing a game might be a better way to enhance student motivation and deep learning than playing an existing game.

Game design can be implemented in different types of development environment. The choice of the environment is an influential factor regarding any educational outcomes. Programming a game within an appropriate programming environments might combine the expected educational benefits of both programming and game design. This is what we tried to achieve in our course design.

## Main Characteristics of the Learning Environment

We refer in this section to the broader interpretation of a learning environment and describe here the course and its pedagogical approach along with the computer-based multimedia programming environment used by the students for achieving the course goals.

### The Course

The rationale for the development of the course “Design of computer-based games and interactive stories” was to provide a motivating and engaging context for exercising various higher-order cognitive skills. The learning activities in the course were designed to require the application of such skills in a way that is expected to affect students’ further learning and thinking capabilities. The course is an elective course for the behavioral sciences departments (Behavioral Science, Education, Psychology, and Sociology) in our college. The course is offered for already several years and is accompanied by an ongoing action research (Or-Bach, 2009). The instructional approach, intermediate assignments and research tools were refined during these years. Throughout the years we got more convinced about the choice of the programming environment as well as about the final assignment. The final assignment of the course is the design and implementation of a game or an interactive story using Scratch. Scratch is a visual programming environment that will be described in the next section.

The course is a hands-on course and most of the classes are conducted in a computer laboratory. The course consists of 13 meetings, each of them lasting about 4 hours. All the program-

ming concepts and constructs, and all the other issues related to game design, interactivity etc.; are introduced during the course by respective examples. Classwork consists of examples that are presented to the students to explore by “reading” and/or running the program elements; as well as other examples where students have to modify a program in order to produce prescribed outcomes. Intermediate assignments in the course include programming assignments, peer-evaluation assignments, literature reading assignments about use of games in education and a class presentation assignment. The required presentation has to deal with the use of a game in an educational setting and should be based on a research paper.

## Scratch

Scratch is a visual programming environment that lets users create interactive, media-rich projects (Resnick et al., 2009; Maloney et al., 2010). Scratch was created by the Lifelong Kindergarten Group at the MIT Media Laboratory. Scratch builds on the constructionist ideas of Logo (Papert, 1980; Kafai & Resnick 1996). A key goal of Scratch is to introduce programming to those with no previous programming experience. This goal drove many aspects of the Scratch design. Programming is done by snapping together command blocks to control 2-D graphical objects called sprites moving on a background called the stage. The command blocks are dragged from a palette into the scripting pane and can be assembled to create “stacks” of blocks. Specific blocks can be placed on top of a stack of blocks to trigger that stack in response to some runtime event, such as program startup, a given key being pressed by a user, or a mouse click on the sprite. Multiple stacks can run at the same time to show simultaneous acts by different sprites on the stage. To encourage self-directed learning (of the programmer), the Scratch programming environment was designed to invite scripting, provide immediate feedback for script execution, and make execution and data visible (Maloney et al., 2010). The programmer can watch stacks in the scripting area highlighted when the action unfolds on the stage. These explorations are supported by having the palette, scripting area, and stage simultaneously visible, providing the programmer with a process model of how their constructed scripts are interpreted by the computer.

Scratch is used in various contexts for various audiences: for children, for youth in informal context (Maloney, 2008), for introducing computer programming to computer science students (Malan, 2007); and for college students in areas where programming is not an integral part of the curriculum (Or-Bach, 2009).

## The Study

### The Study Goals, Setting and Tools

The goal of the presented study was to elicit students’ insights for evaluating the course as a suitable context for developing higher order skills. The research question was: what are the themes that students use to describe the learning experience and learning outcomes; and especially how it relates to the development of higher order cognitive skills. The elective course “Design of computer-based games and interactive stories” is offered to students almost each year during the last ten years. The data presented in this study pertain to the course offering during the last two years. The course was offered twice and the

total number of participants was 35, 31 females and 4 male students. The main research tool was post-course reflection reports that students were asked to submit after handing the final course assignment—the implemented game with a required accompanying document. Two issues were carefully considered with regard to these reflection reports: whether they should be obligatory (required for the final score of the course) and thus not anonymous, or be voluntary; and whether the students’ reports should be structured with open and/or closed questions (a questionnaire) or not structured at all. The tradeoffs for the first decision that were considered were that an obligatory assignment is expected to supply more reports, but reports that are provided voluntarily might be more thoughtful and sincere. We decided on the voluntary option, being aware that we will probably get a smaller number of reflection reports but with the potential to present a finer picture of the experience that the course provided to the student. As for the second decision we chose a free text description with minimal guidance in order to obtain authentic reflections phrased with the student’s own terminology.

Even though we don’t consider the final assignment as a research tool, it provides a control mechanism. This control mechanism is important from two reasons: to assure that students based their reflection reports on a meaningful activity, and that students who submitted reflection reports are not a subset of the course participants that succeeded in the course and thus had the opportunity to practice higher order skills; but rather a subset of students that were willing to invest time in going through a reflection process and in documenting it.

## Findings

All the final assignments of computerized (Scratch based) games and interactive stories were of good quality, some were of excellent quality. The assignments exhibited programming capabilities, creativity, awareness of interactivity options for users; and also use of some evaluation techniques with actual users as was required by the final assignment. Some of the projects were quite complex and it was apparent that the students invested a lot of thought and a lot of time in designing and implementing the project.

We received eleven reflection reports. We do not present it as a representative sample, but as a way to discover students’ subjective perceptions of the whole experience.

The reflection reports, maybe due to fact that it was voluntary, were thoughtful, sounded sincere and dealt with several aspects of the course and its impact. The length of the reports ranged from 145 to 460 words, with an average of 217 words.

All students mentioned the fact that the course was different from other courses they took. The following adjectives (translated from Hebrew) were used by student when explicitly stating that the course was different from other courses: “hands-on”, “practical”, “enjoying”, “mind-opening”, “encourages creativity”, “an opportunity to enter the world of computers”, “an opportunity to create something”, “challenging”. Some of these aspects will be elaborated through the following themes that we defined for presenting the data.

Thematic analysis was used to identify the themes that students related to in the reflection reports. Thematic analysis is a useful and flexible method for qualitative research (Braun and Clarke, 2006). Thematic analysis is approached here in a semantic way—coding and theme development reflect the ex-

explicit content of the data. We differentiate between cognitive and meta-cognitive themes versus affective themes. The choice to illuminate the affective themes is based on the assumption that positive affect influence cognition (Ashby et al., 1999). We present here a sample of citations exhibiting the different themes. The citations were translated from Hebrew.

## Affective Themes

### 1) Concerns and worries at the beginning of the course

Part of the students mentioned in the reflection reports that they were nervous at the beginning of the course, doubting whether they will be able to learn to use Scratch and whether they will be able to construct a working computer program. "In the beginning I felt I will not be able to do the assignments, it seemed to me too difficult." "I am using the computer a lot, and it is easy for me to use it for personal uses like Facebook or Google. But I've never studied programming; it seemed far away from my capabilities."

### 2) Fun and challenge during the course (and during the preparation of the final assignment)

"I have to say that it was so much fun! First of all any creation is blessed, any subject where there is an opportunity to create something it should be done. In addition it is fun to try and construct a game for children to use."

"During my work on the final assignment I had fun, learned, thought a lot, and dealt with a topic that is of interest to me—Music."

### 3) Confidence and pride after submitting the final assignment

"My feelings and confidence with regard to using the computer improved, if I succeeded to construct an interactive game there is no reason why I would not be able to master other computer applications."

"The final assignment made me understand that I'm capable of developing basic nice educational games without being a professional programmer..."

"Constructing a whole game fills me with sincere and big feeling of pride."

"The course strengthened me in front of my children. They are proud of me that I do not only look at what they are doing but able to do. I learned that I can do many things that I thought only children can do. I was wrong, and I can do more than they can."

## Cognitive and Meta-Cognitive Themes

The themes for the cognitive and meta-cognitive skills include: Planning, Problem-solving, Learning and creativity. These are not mutual-inclusive categories, as planning might need some problem-solving when re-planning is required, problem-solving might require learning of something etc. But still these categories seem useful for presenting students' perceptions as stated in their reflection reports.

### 1) Planning

"I improved my long term planning with the understanding that each action influences others, like a domino effect. When I had to write a script I imagined chains of actions and chose the one that will bring the desired effect."

### 2) Problem-solving

"No doubt that constructing the game was very challenging! The truth is that sometimes it was a bit frustrating; things were

not going as we planned, but I learned during the course to go back and check step by step what the problem is or at least where it can be and try to solve it or at least bypass it."

"The transitions in mindset between programmer and user was instrumental."

### 3) Learning

Under this theme we include insights about self-learning, using the feedback that is an integral part of the programming environment. We also include learning about considering the user when designing an artifact to be used by others.

"I presumably know what should be done, but I'm not sure of the actual results. Here whatever I plan I can get immediate feedback."

"I feel that the understanding of the importance of providing interactivity to the learner will stay with me behind this course."

"I came to understand the necessity of adopting the game and its interface to the learners that will use it, such as reading capabilities etc."

### 4) Creativity

"It is the first course in which I felt that it emphasizes the requirement from the student to be creative and think out of the box."

## Conclusions and Discussion

Post course reflection can support the student in consolidating the knowledge and practice of a course. The externalization of the reflection in a written report can produce further learning benefits due to the fact that thoughts should get more explicit. Along with the benefits to the student, a student's post-course reflection report can provide insights to the course designer about actual and potential effects of the course. In this qualitative study we used students' post-course reflection reports to get insights about the course as perceived by the students. Even though we got a small number of reflection reports, they were thoughtful and sincere. We presented students' inputs according to cognitive and affective themes. We see the affective facet as important by itself, but moreover, an enabler and supporter for the cognitive and meta-cognitive facet. Positive attitudes and confidence can raise students' motivation to learn new things, try, handle failures, be persistent and monitor progress. The findings from the thematic analysis show on one hand the change in the affective themes, and on the other hand students' perceptions regarding the cognitive themes which indicate development of cognitive and meta-cognitive skills. "I improved my long-term planning" refers to the development of planning skills, as well as to the meta-cognitive capability of monitoring the improvement. The findings also show students' awareness of the application of problem-solving techniques such as step-by-step diagnosis, looking for ways to bypass an obstacle; and for transition between developer and user mindsets. The use of the computer as a model, metaphor, and modeling tool has tended to privilege the "cognitive" over the "affective" by engendering theories in which thinking and learning are viewed as information processing and affect is ignored or marginalized. In the last decade there has been an accelerated flow of findings in multiple disciplines supporting a view of affection as complexly intertwined with cognition in guiding rational behavior, memory retrieval, decision-making, creativity, and more (Picard et al., 2004). Research has demonstrated, for example, that a slight positive mood does not just make you feel a little better but also induces a different kind of thinking, characterized by a

tendency toward greater creativity and flexibility in problem solving, as well as more efficiency and thoroughness in decision making. These effects have been found among many groups of different ages and professions (Isen, 2000).

The analysis and presentation of the different themes was important to understand the landscape, but looking at a reflection report as a whole has also its merit and was not captured by the thematic analysis presented in the study findings. The reports, that were more like a story than a set of bullets, exhibited intellectual and emotional processes throughout the course. Some reports showed clearly that the student came to understand the iterative processes needed for designing an interactive artifact to be used by others. Such an understanding is important for further learning and for work experiences.

## REFERENCES

- Arum, R., Roksa, J., & Cho, E. (2011). *Improving undergraduate learning: Findings and policy recommendations from the SSRC-CLA Longitudinal Project*. [http://www.ssrc.org/workspace/images/crm/new\\_publication\\_3/%7Bd06178be-3823-e011-ade5-001cc477ec84%7D.pdf](http://www.ssrc.org/workspace/images/crm/new_publication_3/%7Bd06178be-3823-e011-ade5-001cc477ec84%7D.pdf)
- Ashby, F. G., Isen, A. M., & Turken, U. (1999). A neuropsychological theory of positive affect and its influence on cognition. *Psychological Review*, *106*, 529-550. doi:10.1037/0033-295X.106.3.529
- Black, P., McCormick, R., James, M., & Pedder, D. (2006). Learning how to learn and Assessment for Learning: A theoretical inquiry. *Research Papers in Education*, *21*, 119-132. doi:10.1080/02671520600615612
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*, 77-101. doi:10.1191/1478088706qp0630a
- Candy, P. C. (2000). Knowledge navigators and lifelong learners: Producing graduates for the information society. *Higher Education Research & Development*, *19*, 261-277. doi:10.1080/758484346
- Clear, T. et al. (2008). The Teaching of Novice Computer Programmers: Bringing the scholarly-research approach to Australia. *Proceedings of the Tenth Australasian Computing Education Conference (ACE 2008)*, Wollongong, Australia.
- diSessa, A. (2000). *Changing minds: Computers, learning, and literacy*. Cambridge, MA: MIT Press.
- Dunlap, J. C., & Grabinger, S. (2009). Preparing students for lifelong learning: A review of instructional features and teaching methodologies. *Performance Improvement Quarterly*, *16*, 6-25. doi:10.1111/j.1937-8327.2003.tb00276.x
- Fallows, S., & Steven, C. (2000). Building employability skills into the higher education curriculum: A university-wide initiative. *Education and Training*, *42*, 75-83. doi:10.1108/00400910010331620
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.
- Guzdial, M. (2004). Programming environments for novices. In S. Fincher, & M. Petre (Eds.), *Computer science education research* (pp. 127-154). Lisse, The Netherlands: Taylor & Francis.
- Johnson, S. (2005). *Everything bad is good for you: How today's popular culture is actually making us smarter*. London: Allen Lane.
- Hayes, E. R., & Games, I. E. (2008). Making computer games and design thinking: A review of current software and strategies. *Games and Culture*, *3*, 309-332. doi:10.1177/1555412008317312
- Isen, A. M. (2000). Positive affect and decision making. In M. Lewis, & J. Haviland (Eds.), *Handbook of emotions* (pp. 417-435) (2nd ed.). New York: Guilford Press.
- Kafai, Y., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Kaminski, K. et al. (2009). Workforce readiness: A study of university students' fluency with information technology. *Computers & Education*, *53*, 228-233. doi:10.1016/j.compedu.2009.01.017
- Kay, A. (1991). Computers, networks and education. *Scientific American*, 138-148. doi:10.1038/scientificamerican0991-138
- Keeling, R. P., & Hersh, R. H. (2011). *We're losing our minds: Re-thinking American higher education*. Palgrave Macmillan. doi:10.1057/9781137001764
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, *37*, 88-137.
- Malan, D. (2007). Scratch for budding computer scientists. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 223-227). Covington. doi:10.1145/1227310.1227388
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, *10*, 1-15. doi:10.1145/1868358.1868363
- Oblinger, D. (2004). The next generation of educational engagement. *Journal of Interactive Media in Education*, *8*, 1-18.
- Or-Bach, R. (2009). A programming course for behavioral sciences students. *Proceedings of the IADIS International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2009)*, Rome, 20-22 November 2009.
- Papastergiou, M. (2009). Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*, *52*, 1-12. doi:10.1016/j.compedu.2008.06.004
- Papert, S. (1980). *Mindstorms*. New York: Basic Books.
- Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., Machover, T., Resnick, M., Roy, D., & Strohecker, C. (2004). Affective learning—A manifesto. *BT Technology Journal*, *22*, 253-269. doi:10.1023/B:BTTJ.0000047603.37042.33
- Resnick, M. (2007). *Learning from scratch*. Microsoft Faculty Connection.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*, 60-67. doi:10.1145/1592761.1592779
- Robertson, J., & Howells, C. (2008). Computer game design: Opportunities for successful learning. *Computers and Education*, *50*, 559-578. doi:10.1016/j.compedu.2007.09.020
- Smeets, E. (2005). Does ICT contribute to powerful learning environments in primary education. *Computers and Education*, *44*, 343-355. doi:10.1016/j.compedu.2004.04.003
- Tüzün, H., Yılmaz-Sollu, M., Karakus, T., Yinal, Y., & Kizilkaya, G. (2008). The effects of computer games on primary school student's achievement and motivation in geography learning. *Computers & Education*, *52*, 68-78. doi:10.1016/j.compedu.2008.06.008
- Vos, N., van der Meijden, H., & Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, *56*, 127-137. doi:10.1016/j.compedu.2010.08.013