

Using Interactive Technology in a Short Java Course: An Experience Report

Roberto E. Lopez-Herrejon
Department of Computer Sciences
The University of Texas At Austin
Austin, Texas, 78712 U.S.A
+ 1 512 232-7859
rlopez@cs.utexas.edu

Morrie Schulman
Center for Instructional Technologies
The University of Texas At Austin
Austin, Texas, 78712 U.S.A
+1 512 475-6057
schulman@mail.utexas.edu

ABSTRACT

Keeping students alert and responsive during lectures is a challenge even for experienced teachers in small group settings. Research has shown the importance of student participation and involvement in the learning process. Many ideas and strategies have been proposed to promote these two vital education elements [5]. Among them is the use of interactive technology where the instructor asks a question to the class and each student answers individually. These answers are tallied and the professor can get immediate, quantitative, and real-time feedback information that can be used to detect and address comprehension problems and to adapt the lecture plan accordingly. In this paper we report our experiences using a wireless interactive system named the Classroom Performance System (CPS) [9] in a fast-paced, short but comprehensive Java programming course. We present the challenges we faced and the lessons we learned in designing and delivering lectures using this type of technology.

Categories and Subject Descriptors

K.3.1 [Computer Uses in Education]: Computer-Assisted Instruction, Computer-Managed Instruction.

General Terms

Languages, Design, Human Factors

Keywords

Interactive Systems, Classroom Performance System, CPS, Java Programming, Educational Technology, Information Technology, Delivery Systems, Instructional Development, Performance Support Systems

1. INTRODUCTION

The Department of Computer Sciences at The University of Texas at Austin started using Java as the main programming language for instruction in the Fall 2001 semester. Previously, the main

languages used were C++ and Scheme. The department services more than 2,000 undergraduate students, some of whom have not taken courses with Java, even by their junior or senior years. It also receives students from other majors and universities; all of them bringing different backgrounds and programming experiences. In an effort to help the students in all those situations, the department offers a five-week, fast-paced and comprehensive course (referred to as CS105 in the rest of the paper) in which the main goal is to cover Java basics and a few advanced topics, so that students can apply what they learn in the courses that they take concurrently or in their future professional practice.

Being a comprehensive course, a lot of material, ranging from simple programs to GUI design, has to be covered in a short period of time. As a result, we needed to proceed as quickly as possible without losing students along the way. This fast-paced environment is what motivated us to experiment with interactive technology for teaching basic Java concepts. In this paper we share our experiences using this media in the described setting; further, we present the challenges faced and the lessons learned in designing and planning for it.

Section 2 presents an overview of interactive technology and its pedagogical underpinnings. Section 3 describes the CS105 course, the lecture plan, sample questions, lessons learned, and an overview of students' perceptions. Section 4 summarizes our experiences and future plans.

2. INTERACTIVE TECHNOLOGY OVERVIEW

Electronic classroom communications systems, first introduced in 1992, were quickly embraced by physics education researchers committed to educational reform. These researchers and other science and engineering educators devoted themselves to the measurement and improvement of student learning [11] [12]. They realized that traditional higher education instruction—which consists of instructors attempting to pass on knowledge to their note-taking students—was not effective in teaching difficult concepts, especially in large introductory courses. Instead, they found that interactive engagement (student engagement and interactivity), which yields immediate feedback through class discussion, promoted conceptual understanding more effectively [11].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE '04, June 28-30, 2004, Leeds, UK.

Copyright 2004 ACM 1-58113-000-0/00/0000...\$5.00.

2.1 Pedagogical Underpinnings

Interactive engagement is nothing new. Instructors using this teaching style are simply adhering to a form of the Socratic Method in which a series of questions leads the learner to grasp a complex concept or new perception. What is new is using technology to effectively incorporate questioning strategies into classroom instruction to supplement the teaching of factual content.

More recently, Jean Piaget introduced the notion that children discover and construct knowledge through activities rather than by receiving it passively. Contemporary educational researchers [14] advanced the premise that active learning improves concept formation and knowledge acquisition regardless of age.

The social constructivist theorist Palincsar draws from the work of Piaget to support the notion that interpersonal interactions promote cognition and learning [17]. He also points out that, though direct (traditional) instruction is an effective means of teaching factual content, there is less evidence that it "transfers to higher-order cognitive skills such as reasoning and problem solving," nor is there sufficient evidence that "it results in the flexibility necessary for students to use the targeted strategies in novel contexts" [17].

Traditional instruction addresses knowledge formation and comprehension, the first two levels of Bloom's taxonomy of educational objectives [4]. The higher-order thinking required to form concepts, addressed by the later objectives in the hierarchy, includes application, analysis, synthesis, and evaluation. Research indicates that these types of objectives can be met with interactive engagement, which includes asking questions in class and promoting discussion [6][7].

Eric Mazur [12] applied the theory of active learning and developed the Peer Instruction method to help students' basic understanding of physics concepts. He interspersed short lecture presentations with multiple-choice questions to engage his class. Mazur used the Classtalk (described below) electronic response system to ask questions of students, both before and after peer discussion..

Meltzer and Manivannan [13], who used a variant of Peer Instruction in their physics classes, found gains in student learning when measured against comparable courses. They reported greater conceptual understanding, higher scores, and higher attendance than in traditional courses. Their method employed flash cards which students displayed to respond to multiple choice questions posed during the class. The system allowed them to observe students' responses and body language and to modify the pacing and direction of instruction accordingly. While they achieved positive results, an electronic system would have provided quantitative information, a record of student responses to identify individual difficulties, and a means to keep those responses private and uninfluenced by others.

Questions and student participation through discussion assist instructors in a variety of ways: to see if students have completed assignments, to introduce new points, to assess background knowledge, to determine the level of class understanding, to see if students can incorporate past material into present material, and to correct misconceptions [5].

2.2 Interactive Systems Background

Interactive response systems have provided educational institutions a mechanism to deliver questions and record responses. The technology, currently used by nearly 1,000 colleges and universities, consists of pads resembling television remote controls, receivers to record and display student responses, and software to handle the student-instructor interaction [1].

Classtalk [3], the first system widely used was introduced in 1992. It required wiring and was expensive and time-consuming to install. In spite of these obstacles, early physics education reformers adopted the product as a mechanism to test their ideas.

The wireless Personal Response System (PRS) distributed by Educue [8] provides a more practical and less expensive means to encourage and monitor student participation and comprehension. The product was introduced in 1998, and soon it was adopted by a number of high-profile institutions doing research on interactive engagement including Harvard, Indiana University, University of Maryland, University of Massachusetts, Vanderbilt University, and the University of Strathclyde in the UK, to name a few. PRS allows students to respond to multiple-choice questions with nine possible responses plus two confidence ratings.

The Classroom Performance System (CPS), the other major wireless electronic response system, became available shortly after PRS was introduced. Developed by eInstruction [9], it has gained acceptance in primary, secondary, and higher education markets. This system, like PRS, uses infrared technology but provides faster response times. It accepts five possible responses and has a number of additional features including a student management mode and the ability to input the contents of questions into the system and to output session records.

Student responses to both systems have been overwhelmingly positive. Participation, interactivity, and attendance have increased. Though quantitative and qualitative research corroborates these outcomes, additional research is needed [19].

Recently, the Classroom Feedback System (CFS) has been introduced. This prototype system is under development at the University of Washington [2]. It provides a more sophisticated feedback mechanism where students using laptops can mark areas of the presentation slides and indicate the types of feedback they request from the selected areas or topics. The challenge of this approach is to present the feedback to the instructor in a concise way so that it can be incorporated in the ongoing lecture.

2.3 CPS in The University of Texas at Austin

In recent years, The University of Texas at Austin has been actively engaged in the use of classroom interactive systems [17]. The first interactive system, Classtalk, was introduced in 1996 driven by the efforts of the Department of Physics and the College of Natural Sciences. Soon faculty realized that the system improved student attention and provided instant feedback on student comprehension. The growth in its use and the encouraging reception by the students and faculty led to the adoption of a more cost-effective alternative. In the spring 2001 semester, the first CPS system was introduced. Since then its use has increased steadily. As of fall 2003 semester, there are more than 25 instructors and 3,000 students in courses from several departments.

2.4 CPS Usage Description

CPS provides several multiple-choice question formats and performance reports. It can be used in different modes for particular teaching exercises; for further details see [9]. Learning how to use CPS was simple for both students and instructors.

The general process, illustrated in Figure 1, starts with the instructor posing a question with its corresponding possible answers. Students then respond to the question individually, and their answers are recorded. Once the answers have been gathered, a histogram is displayed on the teaching screen with one column for each possible answer. The value of a column is the number of students who chose the corresponding answer. The instructor can then select and highlight the correct answer so students can see how each of them and the entire class performed. With this information the instructor can adapt or modify the lecture to emphasize unclear topics or to move on, knowing that the students have satisfactorily assimilated the material and that they have reinforced the correct answer.

Arthur Woods and Charles Chiu, early adopters of CPS at The University of Texas at Austin, have identified three types of questions: fact or process questions, problem-solving or concept-probing questions, and opinion or belief questions [19]. In the next section we show examples of typical questions asked in CS105. Not surprisingly, given the nature of computer programming, all the questions fell in the first two categories.

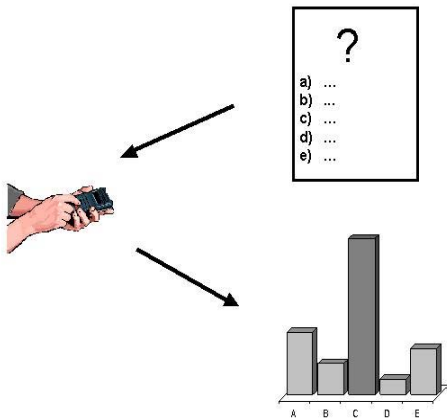


Figure 1. Illustration of CPS use

3. CS105 COURSE DESCRIPTION

The course consisted of 13 lectures, each 50 minutes long, and two sessions of student presentations of the same length. For this class 15 students were enrolled. The goal of the course was to cover the following topics, ranging from basic to advanced:

- ◆ Expressions, keywords, arrays, variables
- ◆ Control flow, methods
- ◆ Classes, inheritance, and interfaces
- ◆ Exceptions
- ◆ String processing
- ◆ Collections

- ◆ I/O, serialization
- ◆ Swing API and graphics
- ◆ Processes, threads
- ◆ Reflection
- ◆ Ant and unit testing
- ◆ Networking

Any Java programmer can agree that the material is extensive, and we had to cover it at a fast pace. Since we did not want to bewilder students along the way, we decided to experiment with interactive technology. Because it was the first time we had tried to incorporate this strategy, we decided to use it only during the first part of the course, which covered topics from basic concepts to the collections classes, and left experimentation in the advanced topics for future courses.

When developing the lecture notes we had to consider the time it took to pose the questions and to tally results. This included the time spent to set up the multimedia system in our classrooms, and to switch back and forth between the presentation and the display of questions with their answers. Our rough estimate is that this process took about 1 minute and an additional 2—4 minutes were needed for the students to provide an answer. This gave us around 3—5 minutes per question. With these estimates we determined that three questions per lecture was an appropriate number, with a question posed about every 15 minutes.

First we focused on notes for each lecture, leaving the design of corresponding questions to a later period. It turned out that question design took, on average, at least half the time it took to develop lecture notes. We attribute that to the following factors:

- Framing good questions in multiple-choice format was a challenge in itself. Special care should be taken to avoid “given away” answers.
- Effective questions should be self-contained. All the context needed to answer a question should fit in a single page or slide, so that students are not distracted by back and forth paging. The limited amount of information that can be effectively displayed can be problematic.

The following examples illustrate the types of questions posed during a typical lecture. The first one tests students’ knowledge of Java collections.

```
List list = new LinkedList( );
list.add("Once");
list.add("upon");
list.add("a");
list.add("time");
Collections.sort(list);
System.out.println("Sorted List");
iterator = list.iterator( );
while(iterator.hasNext( )) {
    String value = (String) iterator.next( );
    System.out.println(value);
}
```

What is the order in which strings are displayed?

- a) Once upon a time
- b) Once a time upon

- c) a time upon Once
- d) a Once time upon
- e) none of the above

The correct answer for this example is “b” because the default string order is lexicographic. To our surprise around 90% of the class missed this question. Our first reaction was to assume that the problem was in the understanding of the collections API just presented. However, when we questioned the students about the rationale of their answers, and by looking at how they were spread in the histogram, we could pinpoint the problem. It turned out that the students did not know that uppercase letters appear before lowercase ones in default string ordering. This situation prompted us to provide a short explanation about the topic. It dawned on us that without this immediate feedback we would not have been able to identify and address the exact cause and magnitude of the problem.

We think this kind of situation is the quintessential illustration of the value of interactive technology. It encourages instructors to ask key questions, and provides accurate information on students’ comprehension that can be incorporated in the lecture plan.

The second example, presented in the Appendix, illustrates a review question that consists of detecting and counting errors in a code sample. After a short inspection the reader can detect that there are six errors (the errors are underlined and italicized). This type of question was very useful to review the topics covered in a single lecture. In our surveys students agreed that this was the most effective type of question for learning to program.

The last two sessions of the course were devoted to team presentations of short topics: using certain GUI components, Javadoc, applets, etc. The presentations were graded for their content and their delivery. Each presentation was evaluated by the rest of the students who were not part of the team being graded. We used CPS for this peer evaluation. Each student was asked to select a pad that was not his or hers, to preserve anonymity. Students expressed a very positive reaction and evaluation of the exercise. On one hand, they got the opportunity to practice public speaking (something they rarely have), and on the other hand they felt that being an anonymous grader relieved them from external factors and enabled them to provide a more accurate and objective evaluation. From the instructor’s point of view, using CPS was a very convenient way to gather the grades.

Because it was the first time CPS was used for the course, we decided to count its results as extra credit toward the final grade, so that students could feel more confident and not afraid of the exercise. A percentage of extra credit was assigned depending on the number of correct responses students got during the course. In contrast, other instructors count CPS use as 5%—10% of the final grade.

Overall, our course surveys indicated that students perceive that effective use of interactive technology improved their attention, participation, attendance, and ultimately their learning.

All lecture notes and their questions are available upon request.

4. CONCLUSIONS

Using interactive technology for our class was a rich experience. The system is a good tool for reinforcing basic concepts, though it requires more involvement from the instructor to craft meaningful questions. However, since questions can be saved and incorporated in future lessons, in the long run this approach could end up being a time-saver.

Our informal course survey indicated that students perceived that using this technology motivates them to attend the lectures and to remain alert and engaged in them. On several occasions we were able to experience the positive contribution of interactive systems to the efficacy of the lectures. With the questions and the immediate feedback we got from them, we were able to spot several problems in students’ understanding and we addressed them.

We are encouraged by our experiences so we will experiment with CPS in the future for the advanced topics of the second part of the course. We also plan to use the system for larger introductory courses and compare the experiences in computer sciences with other departments that also make use of the technology in large classes. One interesting venue to explore is the use of interactive technology in more advanced courses (e.g. the programming languages course), which are more abstract and require more context for the questions.

5. ACKNOWLEDGEMENTS

The first author thanks his advisor Don Batory for letting him take virtually two months off from his research duties, Jacob Neal Sarvela for the experienced comments about the lecture topics and review, and especially his wife Lupita Hernandez for her support and encouragement during lectures preparation. We both thank Charles Chiu for leading by example and his strong support throughout the entire enterprise. We are also grateful to our reviewers Roger Priebe, Judythe Wilbur, Karron Lewis, and Lisa Kaczmarczyk for the insightful comments and suggestions on early drafts of the paper. We are deeply thankful to all the staff of the Division of Instructional Innovation and Assessment because they are the ones who make all the technical wonders happen. Finally we thank Marty Abrahamson and the eInstruction staff for their prompt and supportive response to all our technical and educational queries.

6. REFERENCES

- [1] Abrahamson, Marty. Interactive Class Consulting. http://www.ph.utexas.edu/~ctalk/bulletin/UTBrownBag8_03.ppt.
- [2] Anderson, R.J., Anderson, R., VanDeGrift, T., Wolfman, S., Yasuhara K. Promoting Interaction in Large Classes with Computer-Mediated Feedback. To appear in CSCL 2003.
- [3] Better Education Inc. website <http://www.bedu.com/>
- [4] Bloom, D. Taxonomy of Educational Objectives, the classification of educational goals - Handbook I: Cognitive Domain, New York: McKay, 1956.
- [5] Boyle, J. Listening to Learners. <http://www.strath.ac.uk/Departments/CAP/courses/interactiv/e/powerpoint/>, 2001.

- [6] Crouch, C.H., Mazur, E. Peer Instruction: Ten years of experience and results. American Journal of Physics, 69, 970-977, 2001.
- [7] Dufresne, R.J., Gerace, W.J., Leonard, W.J., Mestre, J.P. and Wenk, L. Classtalk: A classroom communication system for active learning. Journal of Computing in Higher Education, 7, 3-47, 1996.
- [8] Educue website <http://www.educue.com>
- [9] eInstruction website <http://www.einstruction.com>
- [10] Hake, R. Interactive engagement versus traditional methods: a six thousand student survey of mechanics test data for introductory physics courses. American Journal of Physics, 66, 64-74, 1998.
- [11] Hake, R. Lessons from the Physics Education Reform Effort. Conservation Ecology 5(2): 28. <http://www.consecol.org/vol5/iss2/art28>
- [12] Mazur, E. Peer Instruction: A User's Manual. Upper Saddle River, NJ. Prentice-Hall, 1997.
- [13] Meltzer, D., Manivannan, K. Transforming the lecture-hall environment: The fully interactive physics lecture. American Journal of Physics, 70 (6), 639-654, 2002.
- [14] Meyers, C., Jones, T. Promoting Active Learning. San Francisco, CA. Jossey-Bass Publishers, 1993.
- [15] McDermott, L.C, Redish, E.F. RL-PER1: resource letter on physics education research. American Journal of Physics 67(9): 755-767. <http://www.physics.umd.edu/rgroups/ripe/perg/cpt.html>
- [16] Nicol, D.J., Boyle, J.T. Peer Instruction versus Class-wide Discussion: a comparison of two interaction methods in the wired classroom. Studies in Higher Education. 28(4), 457-473, 2003.
- [17] Palinscar, A.S. Social constructivist perspectives on teaching and learning. Annual Review of Psychology, 49, 345-375, 1998.
- [18] University of Texas at Austin-CPS-support website: <http://www.ph.utexas.edu/~ctalk/sectionIII.htm>.
- [19] Woods, A., and Chiu, C. Wireless Response Technology in College Classrooms. The Technology Source, <http://64.124.14.173>, September/October 2003.

7. APPENDIX

```
public final interface Set {
```

```
    private int size;
    public int getSize() { return size; }
    public void put(int value, int index);
} // of Set
```

```
public class Bunch extends Set {
    private String name;
    private int[] values;
```

```
    public Bunch(int[] values) {
        this.values = values;
        size = values.length;
    }
```

```
    public static int accum(Set set) {
        int result = 0;
        for(int i=0; i < set.getSize(); i++)
            result = result + set.get(i);
        return result;
    }
```

```
    public int get(int index) {
        return values[index];
    }
```

```
    public void put(int val, int ind) {
        values[ind] = val;
    }
```

```
    public static void main(String[] args) {
        Bunch b1 = new Bunch(new int[] {3,6,2});
        Bunch b2 = new Bunch(new int[] {4});
        System.out.println("B1 size" + b1.getSize());
        System.out.println("B1[3] " + b1.get(3));
        b2.get(2);
        b2.put(5,0);
        System.out.println("Accum B1 " + accum(b1));
    }
} // of Bunch
```

How many semantic or syntactic errors can you find?

- Less than 4
- 4
- 5
- 6
- More than 6