Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

Emerging Social Engineering in the Wireless Classroom

Shelley Goldman, Roy Pea, Heidy Maldonado
and the WILD Team @ Stanford University
Tel: (650) 723-8422, Fax: (650) 725-7412
Email: sgoldman@stanford.edu, roypea@stanford.edu, heidym@cs.stanford.edu

Abstract: *Code It!* fosters mathematics learning environments where pre-algebra students use handheld technologies to confidently and enjoyably explore and learn about functions. The resources we developed—server-based and handheld software and paper-based student and teacher texts—were packaged as a 20-session unit on code making and breaking and designed to boost students' understanding of mathematical functions and their facility with the multiple representations of tables, graphs and symbols. We field tested the wireless system with two teachers and 120 middle school students during a five week summer school, and conducted studies on the features and function of the technology as a learning and teaching resource. We report on project development and research, focusing on what we learned about the strengths and difficulties of wireless handheld technology in the mathematic classroom, and the emerging social engineering that helped students overcome the technological constraints.

Keywords: wireless handhelds, collaborative learning, algebra, multiple linked representations, teacher support.

1. Introduction: Addressing a Need

Global concern to raise lukewarm achievement results on standardized tests and international comparisons across schools, particularly within math and science realms increases the demands upon unprepared and already overburdened teachers. The *Code It!* Materials — technology and real world based —were designed to address this challenge, capitalizing on developing technology for stimulating high performance while providing a satisfying, successful learning experience on the topic of functions. *Code It!* features multiple representations of functions to aid learners' development of rigorous algebra skills and is inspired by the curriculum unit, Codes, Inc, developed by the Middle-school Mathematics through Applications Project (MMAP), which teachers found helpful for transitioning to the use of technology and more applications-based curriculum materials (Lichtenstein et al, 1998).

Technology should be providing tools to support the activities of mathematics classrooms. Data gathering, pattern finding, spatial thinking, working and linking multiple representations, are all possible with computers, yet many problems have accompanied classroom computer use. Even when researchers developed compelling exemplars (Institute for Research on Learning, 1998; Roschelle et al, 2001), changes in organizational, social, and pedagogical practices with computers produced high hurdles (Goldman, 2001; Means et al, 2001). Recently, the combination of increased access, increasing teacher knowledge, decreasing technology costs, and new portable devices is creating a synergy for new possibilities. With a price tag lower than even the graphing calculators math classes require, greater portability than the slimmest laptops due to their low weight and wireless networking, and computing power comparable to a 1995 desktop or 1997 laptop(1), handhelds thrive across school campuses, often integrated with cellular phones, and with many of the same common productivity applications that appear on desktops. Early studies of handheld technology use are modest yet suggestive of how they might best be applied (Soloway et al, 1999; Curtis et al, 2002; Luchini et al, 2002; Vahey & Crawford, 2002; Roschelle and Pea, 2002; Roschelle, Patton and Pea, 2002; Tatar et al, in press), and the handhelds' beaming ability lets users exchange information easily by direct communication through infrared beaming, and has been used to control audiovisual equipment in the classroom, to share applications, text messages, audio clips, contact information, drawings, and data (Batista, 2001; Pownell and Bailey, 2001) through a simple physical gesture, even without typing or selecting the receiver's username or identifier.

The combination of versatile modular hardware, integrated with many of the applications of task-specific devices (such as graphing calculators), and full-blown computers (including participatory simulation software, e-mail readers, web browsers, compatible or reduced versions of Microsoft's productivity applications), plus their calendar and organizational functions, make these devices particularly appealing and many thousands of classrooms

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

across the United States are acquiring them. Because of their cost and capabilities, handhelds are perceived as holding the potential for democratizing access to the technology for all students, and of allowing educational researchers unprecedented access to students' records and interactions: from studying social networks at school through the students' beaming logs, to comparisons between the intended curriculum, the taught curriculum, and the learned curriculum, and analysis of the effectiveness of "probeware" applications (where students record real-world data on their handhelds through physical probes for later analysis) and participatory simulations in the classrooms. Instead of four or more students sharing a desktop machine, students can collaborate by fluidly sharing information across devices, or even comparing results by looking at each other's screens.

These are promising trends, and to tap the potential in *Code It!* we developed representations that can be easily manipulated in an engaging collaborative learning context for deepening understanding of algebraic formalisms (ITEA, 2000; PCAST 1997).

2. The Setting and Approach

Our research questions are aimed at uncovering the strategies that teachers use to orchestrate the use of handhelds, and identifying the obstacles and affordances they interact with in using *Code It!* materials and technologies. We partnered with the Stanford Teacher Education Program and the Santa Clara, CA Schools to use *Code It!* with all of the district middle school students in their summer school program for our 20-session field test with two teachers over a five-week period. Each teacher had 60 students split between two consecutive 100 minute class sessions: one teacher used handheld iPaq computers (PDAs); the other teacher used Compaq Tablet PCs. Approximately 40% of each 110 minute class period was dedicated to encoding and decoding activities with the *Code It!* materials, with the students separated into four-person groups that remained relatively stable throughout the intervention. The classroom activities and software implemented remained the same throughout both hardware platforms: our present analysis concentrates on only the two PDA classrooms.

We collected several data streams, including observational data, videotapes of teachers and the four-member student groups used in the classrooms, problem-solving and informational interviews with student dyads, and pre- and post-testing of mathematics knowledge. These methods, of observing and videotaping classroom field tests of materials and feeding back results into the design, enable the development team to define areas for revision based on the use of the materials under real classroom conditions (Brown, 1992; Cobb et al, 2003; DBRC, 2003). The cycle of development and field test supports our learning about how teachers and students appropriate these materials and how best to iteratively improve the design for achieving the intended teaching and learning goals (and new ones that might emerge).

The project was structured around three strands of activity: materials development, supporting the teachers, and field test research in the classroom. As we address each strand below we emphasize the lessons learned on how social engineering – a term that is often misused in both meaning and purpose by the hacker community – can overcome technological constraints and lead to learning gains for students.

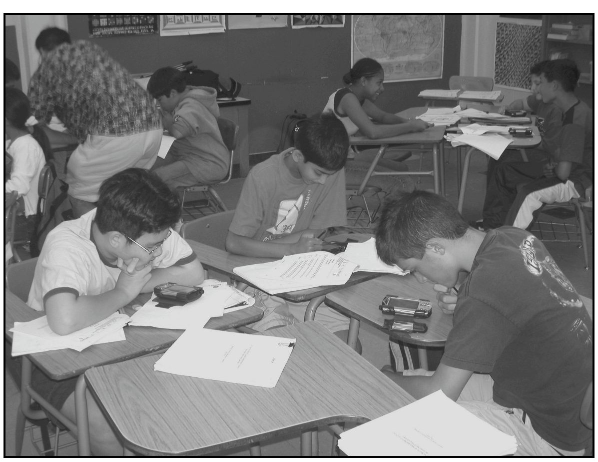

Figure 1: Student groups working on *Code It!*

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

Materials development

We created a technology-integrated 20-session curriculum unit on introduction to functions, with a code making and breaking theme. The unit was developed to bring alive in classroom activity the functionality of the *Code It!* application with pre-algebra content. The unit places algebra learning in a real-world context—making and breaking secret codes (Singh, 2000). Students can challenge others to break their codes to determine message content. The materials are structured so that students must examine properties of functions in general, learn to distinguish among "families" of functions (e.g., linear vs. quadratic) as well as learn properties of each family (such as effects of changes in constants to graph). Connections among symbols, tables and graphs are emphasized in both the software and text materials.

Exploiting the handhelds' wireless connectivity, students can create and exchange codes easily, and records of these transactions can be captured on the server for future analysis, so that the PDA display for any student at any particular time can be recreated. The server also acts as a repository for text and codes created by the teacher and students, providing an easy way to share codes among groups and to distribute practice problems. It is this promise of engaging technology that promotes increased interactions with mathematics and embedded assessment activities that we hope to promote. *Code It!* software is built to exploit an in-room wireless network allowing students to work on codes together in groups and to share codes between groups. Each PDA is connected wirelessly to a teacher's workstation. The teacher's station, shown in *Figure 3*, runs administrative software so that the teacher can control and monitor which students are logged into the server, assign them to groups, create and distribute practice problems, and open an observer window to display any group's current state. The observer window is particularly important for teachers because the small display size of the handhelds makes it more difficult to monitor group work in the traditional "over the shoulder" manner. In addition, the server logs each group's activity, allowing a researcher or teacher to reconstruct the precise ways in which students were using the software and solving the math problems.

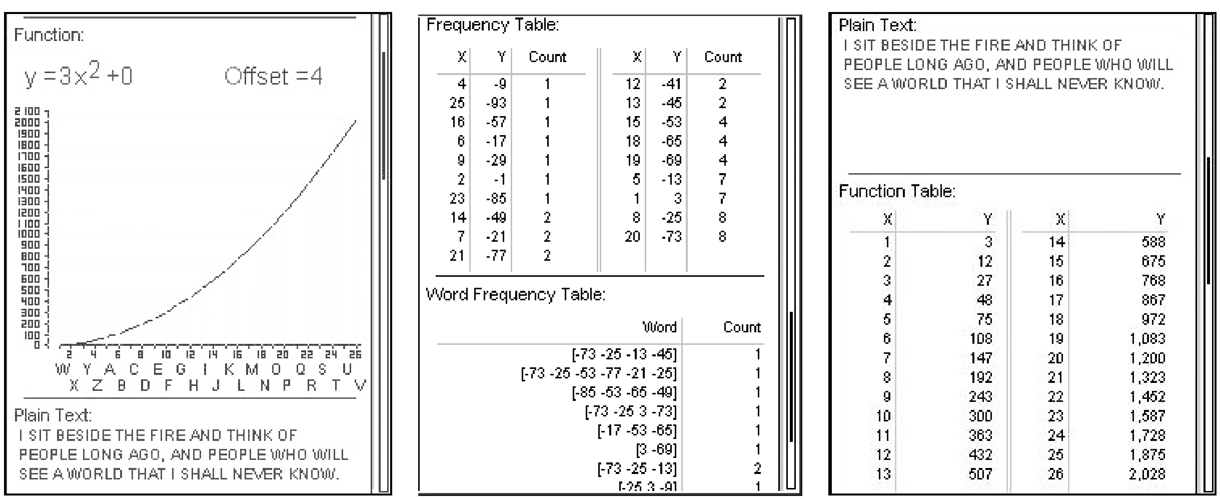

Figure 2: Three *Code It!* representations on a student's PDA: The expression and graphical screen view of a code solution, the frequency tables' view and the function table's view.

Each student PDA runs *Code It!* software for creating plaintext, for making and breaking codes, and for uploading and downloading text and codes from the server. *Code It!* software is written in Java to allow for portability across platforms and for wireless communication. When making or breaking codes, the student's screen displays the encoding function (or current guess) and the graphs and tables that represent the function and the coded text, as *Figure 2* illustrates. These representations include the function's expression itself, a graph of the function, a function table, and word and letter frequency tables. Students change the values of the function's coefficient, exponent, constant, and offset by clicking with the handheld's stylus on the small arrows that appear immediately below (to decrease) and above (to increase) each of these terms on the expression. These representations were selected to encourage students' fluency with multiple representations of functions, a central element of secondary math curriculum (NCTM 1989, 2000), as well as to foster collaboration within the student groups, rather than to create an ideal coding tool. Any change made by one student immediately propagates to the other students in his or her group through their server linkage, so the PDAs within a group always display the same function and text. As we

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

discuss in the next section, to foster communication among group members and increase the diversity of information available to the group for the coding or decoding task, students were encouraged through assigned group roles that rotated across days, so that all students took on all roles, to focus on a different representation per teammate. When a group has successfully created or broken a code, they receive a congratulatory message from the teacher's station, which also records their problem and solution with the corresponding time-stamp.

Importance of teaching to *Code It!* success

The teaching component turned out to be as important as the materials and software in ensuring that students get a well structured, engaging and deep mathematical learning experience through *Code It!* Professional development included three days of training and planning with the teacher partners as well as just in time, in-class support. The workshop included an introduction to the technology and software, as well as discussion and revisions of the curriculum unit and activities. Just-in-time daily consultations ranged from strategic advice on classroom management techniques, to math and technology specific problem solving while implementing the unit, often as a result of the teachers' lack of prior experience with the devices. Teachers were central to the curriculum's success in the classroom. They innovated by creating tasks, activities, and explanations, and by setting into motion social rules for encouraging appropriate forms of mathematical discourse and technology use.

Turning the attentions of students in the *Code It!* classroom to the mathematics took a great deal of social engineering on the part of the teachers and the development team. The team had to mitigate the effects of the widely heterogeneous groups – in terms of previous mathematics knowledge, English language dominion, socio-economical levels, race, gender, and age – and the constraints of the technology. During the teacher preparation workshop, we decided to introduce rotating social roles within the groups to ensure that all students had a chance to work with each tool and representations during the problem solving process. However, once students learned to code and decode using the PDAs, they forged ahead without clear group structures, leading to heated discussions within the group on recording and changing privileges. We worked with the teaching staff to design social roles that would foster group collaboration, rather than competition.

Linking the process roles of (1) recorder, (2) presenter, (3) publisher and (4) equipment manager to the different code representations in the software, and rotating these roles daily, had several positive effects in addition to resolving much of the inter-group conflict. Role based pedagogy and "group-worthy" tasks have been shown to serve as powerful supports for heterogeneous grouped students (Cohen & Lotan, 1997), and with *Code It!* they contributed to increased engagement for more students in code analysis and breaking activities, provided a vehicle for teachers to engage students in content-based conversations, and overcame some of the technological constraints that are discussed in detail below. The student in charge of taking notes (the recorder) one day would also be responsible for monitoring the function graph as the student publisher altered the coefficient, exponent and constant of the function, while the equipment manager and presenter monitored the frequency and function tables, respectively. The next day, roles would rotate so that all students gained experience at interpreting and communicating key indicators for each representation that signaled a closer match between the encoding and decoding functions.

In addition to group roles, another rule was introduced stating that: "all students in a group must be able to explain the strategies and routes to solutions in order to receive credit," which lead to increased team cooperation in heterogeneous groups. Enforcement of these rules, both during in-class presentations and during mathematical activities, often involved the teacher and student teacher querying any student from the group about their strategy and solution, and was facilitated by wall posters reminding students of the rules and rotating assignments. After instituting these social rules new patterns emerged, and we observed multiple instances of the students who took problem-solving approaches talking with those who usually guessed or seemed uninvolved. In addition to this knowledge sharing we recorded an increase in consistency of students' performance over time. This teacher-led social engineering improved group relations and engaged students more deeply with the software and the mathematics under study.

Teachers appropriated the technology into their teaching practices increasingly throughout the summer, as their familiarity and confidence with the tools increased. They assessed students' work and needs, developed activities and code breaking problems to scaffold students' understanding, bridging the gap between simple and complex code breaking problems in the curriculum. Teachers relied on the server screen to keep track at a distance of which students were on-line and which groups were most active. As *Figure 3* illustrates, a circle represented each

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

student in a group, and its color changed when the student connected to the server. Teachers monitored group activity through the observer windows, which presented information in exactly the same way group members viewed it, as if the teacher was another teammate, albeit virtual. These multiple views into the groups' activities enabled the teacher to troubleshoot for students who were having log-on issues, as well as to visit groups that were lagging behind in the decoding problems. Furthermore, both teachers used projection equipment with their PDAs, so that they and their students could discuss with visual aids the various strategies for code breaking, as well as demonstrate how the different *Code It!* tools could be used. These activities illustrate the robustness of the software package, and how teachers made it compatible with their teaching practices.

Technology design and performance

Constraints imposed by the technology in everyday teaching practice were many and ranged from overload delays during whole class log-on, to the lengthy scrolling screen with multiple representations of data, to the fact that the system could log only group work. Specifically, three features of *Code It!* were shown to need reworking during the classroom test: (1) how mathematical representations and tools are organized; (2) the choice to base the system on group activities and interaction over individual acts; and, (3) the design and usability of record keeping functions—potentially so promising with wireless technology. Given that our developing time for this first version was only twelve weeks, we are looking forward to correcting these three shortcoming in the next software release, but until then let us consider each of these three issues in greater depth:

The *Code It!* student PDA screen features a vertically lengthy set of representations for the function under consideration, which students scrolled through to focus on their assigned view, as *Figure 2* illustrates. These representations are: the encoded and decoded text, the manipulable mathematical function expression, a graph of the function, a function table, and word and letter frequency tables. The word frequency analysis data representation suffered from the system's delay when handling longer codes than three or four sentences, and until student roles were established, the vertical scrolling layout preferenced the graphical representation over others, since it was directly below the manipulable function. The group process and daily rotation of assigned roles for students, attaching each teammate to one of the representations per day, was a compromise between the constraints of the system and the ways students could best benefit from the representations available. Yet although all students gained experience with each representation and the stylus wars for control of the encoding process were resolved in favor of strict, daily roles, some students became alienated, discouraged, and in some instances bored, particularly if they had a less-than-critical role for the day. Through this field test, we were able to deduce that the multiple representation capabilities of the software are a strength and real resource for mathematical work and learning. A question that now remains for investigation is how structured or unstructured students' access to those representations should be.

As we consider the redesign of the system, not only will interface and display constraints – such as button-linked screens instead of a longer scrolling window – have to be addressed, but also the underlying system architecture. Although having a portion of the data displayed for interpretation on each student's PDA made for within group collaborative learning, it meant that on any day a student had the opportunity to relate to only one view of the data, and that *Code It!* activities outside the group were not enabled. That is, students could not practice encoding and code breaking on their own, nor could they draw parallels between representations by comparing them side-by-side. Excitement for exploration of the software turned to frustration – particularly in the first sessions with *Code It!* – until all teammates had experienced the different roles and representations. Schools also rely on individual accountability, and the *Code It!* system was unable to account for individual problem solving activity. This field test has proven that we must also support individual activities within *Code It!*, and we are exploring potential avenues that would maintain support for collaborative learning.

Our design decision was to bias the function of the system to support groups over individual students, and the architecture of the system made the server act as a gatekeeper, transmitting changes from one PDA to all the others within a team, and recording them for later retracing of strategies. However, after this field test we believe that the server can do much more with this information without waiting until class is over. We observed that student groups quickly developed strategies to break codes, which were shared approximately once a week with the whole classroom. Some of these strategies depended excessively on a simplistic approaches, such as curve-fitting functions through the graph that usually got the students within range of a correct code. Many of the students determined the exponent graphically, and often the lead coefficient as well by adjusting the graph. Others consistently sought out single, two, and three- letter words, as well as mappings for the most common letter in the English language ("e") as a path for breaking codes. We are discussing incorporating a prediction capability to the software, which could alert

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

the teacher or team itself when a group is in danger of depending solely on a particular pattern of code breaking. Moreover, as the students presentations of decoding attempts showed, when faced with reconstructing all steps, students needed access to records of their transactions and solutions shortly, before the end of class.

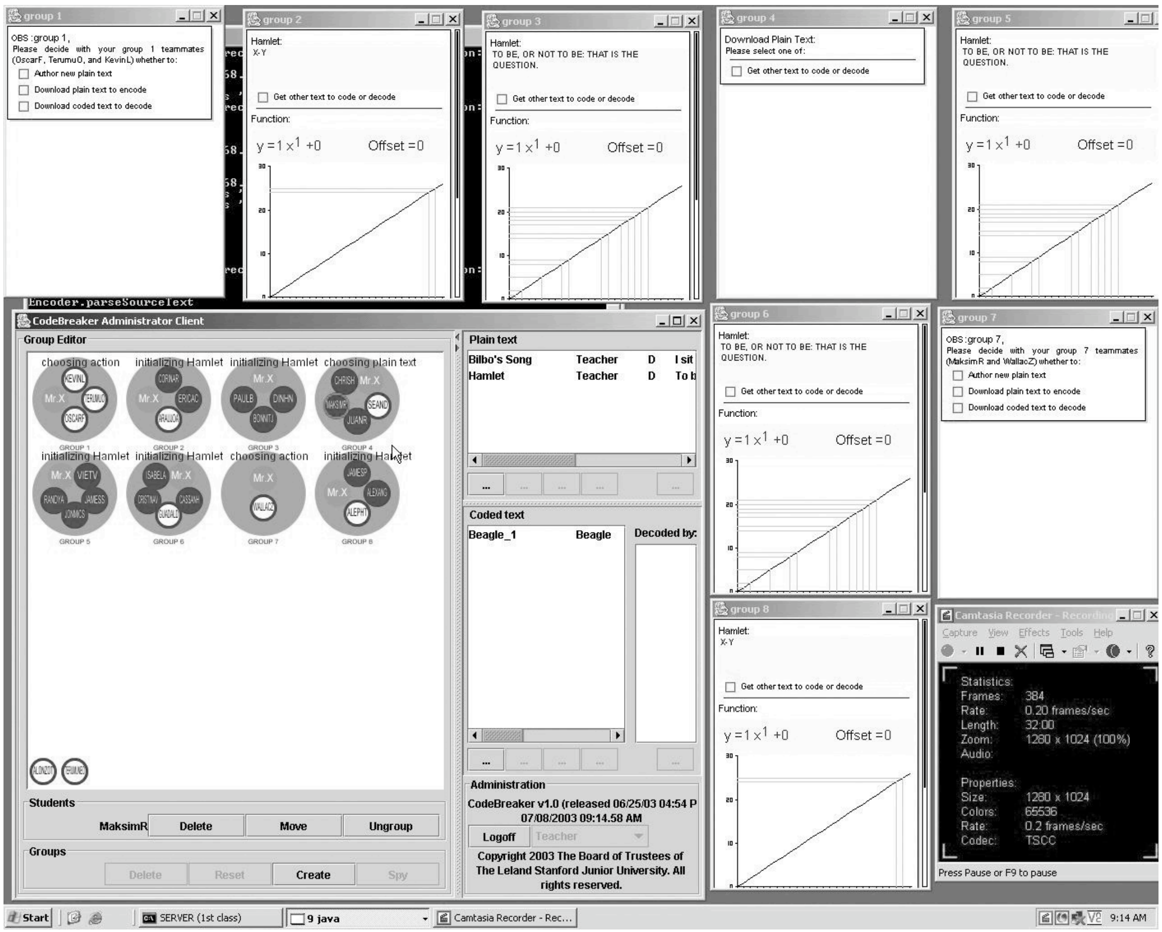

Figure 3: Teacher's view on the server, with observing windows for every group in the classroom

3. Findings to Date

Preliminary findings include analysis of pre- and post-tests, and analysis of observations, videotapes, and interviews. We report on progress on math test items, and video analysis that indicates increased sharing of mathematics and use of software representations in the groups. Results of the pre- and post-tests in the PDA class (N=45) with our targeted student groups are very promising. The mean increase from pre- to post tests was eight percentage points, with great variability in whether groups functioned well (a fertile area for subsequent studies, as in Barron, 2003). In four of six focus groups students made significant gains, in some cases raising their scores by 15-30%. The PDA students showed significant gains on test items relating to evaluating exponents and the graphs of functions. On one graphical item, 44% of students answered correctly on the post-test, as compared to only 13% on the pre-test.

These trends are particularly exciting because these results were for students in grades 6 through 8 who were placed in heterogeneous groups regardless of their previous school math course achievement. Many students in the class had done poorly in sixth or seventh grade math, and were required to attend the summer academy to advance to the next grade, while some students had completed the first year of algebra. The teacher's recommendation and innovative social structures encouraged the more skilled algebra students to help others, and the pre- and post-test results showed that these interactions may have had an impact on achievement. Currently, we are analyzing the activities and progress of four groups of students in the two handheld classrooms, which we

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

captured through 120 hours of classroom interaction video, the server's records, and code-breaking strategy interviews with twelve pairs (N=24) of students in these chosen groups at the conclusion of the program.

Future plans for the project include a significant redesign of the software for version 2, such that no representation is privileged over others, as the graph view was preferred in this version over the function table, letter frequency and word frequency table representation. Depending on the more familiar graph led some observed groups to follow simplistic decoding strategies that comparatively did not improve their dominion of functions as much as their peers', such as predicting the range and domain of the encoding function by methodically changing values in the function until the axis labels came within the range of the traditional alphabet.

We seek to address the ideal balance between scaffolding for collaboration and individual exploration of the encoded functions, within the *Code It!* software environment, so as to lessen opportunities for disruptive stylus-wars. Our upcoming iteration will incorporate a flexible grouping mechanism, while maintaining both the appeal of collaboration and software robustness that made this first experience with *Code It!* successful. We are also considering ways to decouple the individual PDAs from the system. This would allow for students to explore and problem-solve individually as well as in groups, and as well as provide opportunities for "anytime, anywhere computing", home and family connections.

Endnote

(1) The comparison is based on the reported clock speeds of the Toshiba Libretto (70CT : Pentium MMX 120 MHz), a PC laptop introduced in 1997; the Apple Macintosh 7200, a desktop computer introduced in 1995, which could go 120 MHz, and the latest Palm Tungsten T handheld introduced in 2003, which on average can reach 120Mhz as well. Information for this comparison was obtained from: http://www.priceminister.com/offer/buy/1192850/PR030, http://www.apple-history.com/7200.html, http://www.palm.com/products/handhelds/tungsten-t/specs.html, March 2003.

References

Batista, E. "Debating the merits of palms in class." Wired News, Available online: <http://www.wired.com/news/school/0,1383,45863,00.html> August 23, 2001

Barron, B. (2003). When smart groups fail. The Journal of the Learning Sciences, 12, 307-359.

Brown, A. (1992). Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. Journal of the Learning Sciences, 2(2). 141-178.

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design Experiments in Educational Research. Educational Researcher, 32 (1), 9-13.

Cohen, E. G., & Lotan, R. A. (1997). Working for equality in heterogeneous classrooms: Sociological theory in practice. New York: Teachers College Press.

Curtis, M., Luchini, K., Bobrowsky, W., Quintana, C., & Soloway, E. (2002). Handheld Use in K-12: A Descriptive Account. Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp. 22-30). New York: IEEE Press. .

Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. Educational Researcher, 32 (1), 5-8.

Goldman, S. (2001). Technology in the Mathematics Classroom: Guidelines from the Field. ERIC Update. ERIC Clearinghouse on Information and Technology, 22 (2). <http://ericit.org/newsletter/Volume22-2/goldman.shtml>

International Technology Education Association (2000). Standards for Technological Literacy: Content for the Study of Technology. Reston, VA: ITEA.

Lichtenstein, G. Weisglass, J., & Erickan-Alper, K. (1998). Final Evaluation Report: Middle-school Mathematics through Applications Project. Denver, CO: Quality Evaluation Design.

Luchini, K., Bobrowsky, W., Curtis, M., Quintana, C., & Soloway, E. (2002). Supporting learning in context: Extending learner-centered design to the development of handheld educational software. In Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp. 107-111). New York: IEEE Press.

Means, B., Penuel, W. R., & Padilla, C. (2001). The Connected School: Technology and Learning in High School. San Francisco, CA: Jossey Bass.

Middle-school Mathematics through Application Project Final Report to the NSF (1998). Menlo Park, CA: Institute for Research on Learning.

Copyright 2004, the International Society of the Learning Sciences (ISLS). Presented at the International Conference on the Learning Sciences 2004. Copies can be distributed with ISLS permission.

National Council of Teachers of Mathematics. (1989). Curriculum and evaluation standards for school mathematics. Reston, VA: Author.

National Council of Teachers of Mathematics: (2000). Principles and standards for school mathematics. Reston, VA

Pownell, D., Bailey, G. "Getting a handle on handhelds." Electronic School. Available online: <http://www.electronic-school.com>. June 2001

President's Committee of Advisors on Science and Technology and Panel on Educational Technology (1997, March). Report to the President on the Use of Technology to Strengthen K-12 Education in the United States.

Roschelle, J., Pea, R., Hoadley, C., Gordin, D., & Means, B. (2001). Changing how and what children learn in school with collaborative cognitive technologies. In M. Shields (Ed.), The Future of Children (Special issue on Children and Computer Technology, published by the David and Lucille Packard Foundation, Los Altos, CA), Volume 10, Issue 2, pp. 76-101.

Roschelle, J., & Pea, R. D. (2002). A walk on the WILD side: How wireless handhelds may change computer-supported collaborative learning (CSCL). The International Journal of Cognition and Technology, 1(1), 145-168.

Roschelle, J., Patton, C., & Pea, R. (2002). To unlock the learning value of wireless mobile devices, understand coupling. In Proceedings of The First IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02, pp. 2-7). New York: IEEE Press.

Singh, S. (2000). The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. New York: Anchor Books.

Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R., & Eisenberg, M. (1999). Science in the palm of their hands. Communications of the ACM, 42(8), 21-26.

Tatar, D., Roschelle, J., Vahey, P., & Penuel, W. R. (in press). Handhelds go to school. IEEE Computer.

Vahey, P., & Crawford, V. (2002, September). Palm Education Pioneers Program: Final Evaluation Report. Menlo Park, CA: SRI International. (http://www.palmgrants.sri.com).

Acknowledgments

The WILD Team @ Stanford members contributed to the *Code It!* work reported herein. Thanks to Toby White, Lee Martin, Sarah Walter, Gloria Miller, Mike Mills, John Murray, from Stanford University, and to Nicolai Scheele and Wolfgang Effelsberg, from University of Mannheim (Germany). Special thanks to our partnering teachers and their students, the Stanford Teacher Education Program and the Stanford student teachers, and the Santa Clara School District. This project was supported by Stanford Center for Innovation in Learning (SCIL) and the Wallenberg Global Learning Network. We thankfully acknowledge an HP mobile wireless computing equipment grant to SCIL.