Developing Math Fluency With Math Drill Software for Classroom Response Systems

A Thesis Presented to

The Faculty of the Computer Science Program

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

Michael Weingarden

April 2011

APPROVED FOR THE COMPUTER SCIENCE PROGRAM

12/15/11

Dr Andrzej Bieszczad Date

Advisor: Dr. William Wolfe Date

12/15/11

Dr. Jorge Garcia Date

APPROVED FOR THE UNIVERSITY

12-16-11

Dr. Gary A. Berg Date

Developing Fluency in Mathematics By Drilling Using Classroom Response Systems

by

Michael Weingarden

Computer Science Program

California State University Channel Islands

Abstract

The hypothesis presented here is that remedial math students could increase their math fluency through drilling or repetitive practice using classroom response systems. While there is already drilling software available for math students, there is no system for drilling an entire class as a group. Also, few schools can support one computer per student for every math class. The sort of mechanism described here opens up a wide variety of options for improving math fluency for low functioning math students.

The experiment involved specially designed software that allowed students in a remedial math classroom to compete against one another as they practiced their basic math facts. The software collected data that was then analyzed, to determine whether math fluency increased.

Acknowledgements

Dr. Wolfe, Dr. Garcia, Dr. AJ, Dr. Ivona, and others.

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1: Introduction

There are two aspects to this thesis: developing fluency in mathematics and using a clicker system to assist in that goal. The first part of this chapter discusses the pursuit of math fluency and the second part explains the technical details of developing software for the clicker system. This way, the reader can pick and choose the parts that are most pertinent to them.

1.1 Automaticity and Fluency in Math

1.1.1 Research

Fluency is typically a concept related to language and vocabulary. However, fluency is important in mathematics as well. If you had to read a book one word at a time, you would lost track of the big picture, the story, very quickly. Or, imagine if you had to read a book one letter at a time? There is a similar problem if you must add using your fingers or if you have to pull multiplication facts from an incomplete or inaccurate table in your head. How can a student master the simplification of longer, more complicated expressions if they have to crawl painstakingly through every basic operation?

During the 1960's and 1970's, in this country, the study of basic arithmetic was deemphasized in lieu of a concept called, "New Math[1]," which emphasized more abstract aspects of math such as set theory. In 1983, a report called, "A Nation At Risk[2]," was published by the National Commission on Excellence in Education. The report chastised the nation's school system for dropping scores in English and in math on SAT exams.

During the 1980's, a reform[3] movement in mathematics began an increased emphasis on explorations, projects, written and verbal communication, working in cooperative groups, making connections between concepts, and connections between representations. The National Council of Teachers of Mathematics (NCTM) summarized the state of current research with the publication of Curriculum and Evaluation Standards in 1989 and Principles and Standards for School Mathematics in 2000.

By contrast, "traditional" textbooks emphasize procedural mathematics and provide step-by-step examples with skill exercises. Traditional mathematics focuses on teaching algorithms that will lead to the correct answer. Because of this focus on application of algorithms, the traditional math student must always use the specific method that is being taught. This kind of algorithmic dependence is de-emphasized in reform mathematics.

In 2008, the National Mathematics Advisory Panel called for a balance between reform and traditional mathematics teaching styles. This is reflected in more recent versions of the NCTM standards. Even so, districts, schools, teachers and teaching colleges still tend to emphasize problem solving and critical thinking and de-emphasize standard algorithms and rote memorization of facts, especially in secondary school.

1.1.2 Design

Unfortunately, too many students come to secondary school with a lack of fluency in basic math facts. This puts students at a disadvantage when learning almost all other topics. A strong grasp of number sense is critical to success in almost all other high school math topics. So, the objective of this thesis was to find a way to help students improve their fluency.

After observing remedial, secondary math students for the past 5 years, I have come to the conclusion that these students need an emphasis on basic math facts before they ever have a chance of simplifying multi-step expressions or solving multi-step equations. After coming to this conclusion, I began to print out worksheets to drill individual students outside of class. Unfortunately, I never felt like I had the necessary time to really bring these students up to speed in the basics. So, I began to think about how computers and software might help.

My school had a computer lab, but the lab was not always available and it was a bit run down (and this is in an affluent district). The science teachers at my school were using clickers (classroom response systems which are described thoroughly in section 1.2) and I was interested in trying them out. I learned about a Student10 and I wrote a proposal to get a clicker system. I got the Student10 and received the clicker system. At first, I merely used it to display PowerPoints and multiple choice questions. Then, I came up with the idea of using the clicker system as a drilling system. I had no idea whether it was possible. I contacted the vendor, PRS Interwrite, and requested an SDK (software development kit)/API (application programmers interface). After signing a non-disclosure agreement, they provided it.

I initially planned out my idea with some bullet points. Here's what the system should do:

- Allow students to log in so that their individual results could be tracked
- Present a screen full of basic math fact problems using the projector
- Accept answers from students using their clickers
- Stop the session in a way that ensure that all students are working until the session ends
- Provide students with feedback on how they did
- Store the information for analysis later

My idea was to use the clicker system to regularly drill my students on basic math facts as a supplement to our daily lessons on mathematical concepts. However, I felt that I should have some way to measure the effectiveness of the system and that's where this thesis comes in. In Chapter 3: Technical Details of the Work, I provide more details.

1.2 Introduction to Classroom Response Systems

Classroom response systems are known by a variety of names including audience response systems, personal response systems, student response systems and clicker systems. They usually consist of a receiver and a class set of remote controls- one for each student- software and a computer. Classroom Response Systems allow a teacher to gather input from a classroom full of remote toting students.

Classroom Response Systems are typically used in the following manner:

- a question is displayed using a computer projector
- multiple choice answers are presented
- a countdown timer is displayed
- students use their clickers to submit answers
- the question response period ends when the countdown timer gets to zero or when all students have responded
- a bar graph is displayed which shows how many students voted for each multiple choice answer

A more extensive description regarding the field of Classroom Response Systems is presented in Chapter 2: Field Overview.

Anne Cleary of Colorado State University believes that Classroom Response Systems are uniquely well suited for the teaching of psychology and other courses that emphasize behavioral research methods. Specifically, instructors can use the clicker system to engage students in an in-class replication of a known empirical phenomenon[4]. In other words, the clicker system can be used for conducting social experiments. Ms. Cleary used the clicker system in the traditional way, but then exported the results data to Excel for further analysis of the experimental data.

Michael Salemi at the University of North Carolina has found another practical use for the clicker system. In his paper on clickers, he describes the disconnect between examples that are typically presented in an economics class and applications of economics in real-world settings[5]. The paper describes an online auction with students bidding for a real item using their clickers. It's a real auction for a real item and the resulting data illustrates supply and demand in a real situation.

The traditional use of clickers has several disadvantages. For individuals that are visually challenged, formatting PowerPoints can be daunting. It would be nice if question text could be entered in a simple way and then formatted automatically like with content management systems for web site development. Also, clickers tend to target individuals only. Michael Gebauer of MIT designed a project that allowed users to enter questions in simple XML documents with the software doing all the formatting[6]. In addition, his software allowed for the designating and scoring of teams.

These projects and others indicate that there are some uses for Classroom Response Systems in addition to the traditional PowerPoint use.

1.2.1 One More Innovative Approach

There are already several projects that demonstrate that Classroom Response Systems can be used in non-traditional ways. Each is innovative and encourages new uses for clickers, but each still follows the "present one question and wait for the last person" model of interaction. None of the papers or projects described above have been adopted on a widespread basis. For Classroom Response Systems to achieve large scale acceptance, they need a killer application. The intent of this paper is to describe a system which could open up Classroom Response Systems to wider adoption.

1.2.2 Key Terms Regarding Clicker Systems

Answer – the response to a question sent by a clicker user

Class – when students sign in to a clicker system, they become a member of a class

Classroom Response System - A set of hardware and software that includes multiple remote controls (known as clickers), a receiver, software, computer and projector.

Clicker – A remote control device which usually has some kind of keypad and sometimes has a display.

Clicker System – this term will be used interchangeably with Classroom Response System.

Events – When a receiver receives an answer from a clicker, the clicker software is made aware that new data is available. This notification is called an event. It is similar to an interrupt when talking about microprocessors.

IR – infrared. One vehicle for communicating between clickers and receivers.

Join – the act of signing in to a clicker system class.

Question – clicker sessions can be composed of multiple questions. However, for purposes of this project, each session will be composed of one question. The one clicker question will actually have 20 problems that require answers and therefore there will only be one clicker question per session.

Receiver – A device that receives signals from all the clickers in a classroom and passes the data to a computer via a USB port.

RF – radio frequency. A vehicle for communicating between clickers and receivers.

Session – a session begins when the clicker system allows students to begin answering questions

Chapter 2: Field Overview

Like Chapter 1, this chapter has two main sections. In section 2.1, the idea of developing math fluency is discussed. In section 2.2, the technical details of clicker systems are discussed.

2.1 Developing Fluency with Basic Math Facts Using Drilling

The following section discusses research regarding automaticity and fluency with basic math facts. There weren't many articles regarding these issues related to secondary school students, but there are a number of articles and books that discussed the issue with regard to primary school students.

Apparently, the idea that math facts can be learned by rote repetition alone may not necessarily be true.

Even so, a British study[7] specifically related to the characteristics of successful teaching of numeracy, outlined the characteristics of classroom processes where standards were low including too little fluency in mental calculation.

From the book, "Insights into Teaching Mathematics[8]:" "Children cannot continue forever using objects (a strategy for conceptualizing addition) to find the answer to an addition problem. Some of the additions which involve relatively small numbers can and should be memorized for easy recall. Addition of single digit numbers is known as addition bonds or addition facts.

From, "Adding it up: Helping children learn mathematics[9]:"Although some educators once believed that children memorize their "basic facts" as conditioned responses, research shows that children do not move from knowing nothing about the sums and differences of numbers to having the basic number combinations memorized. Instead they move through a series of progressively more advanced and abstract methods for working out answers to simple arithmetic problems. Furthermore, as children get older, they use the procedures more and more efficiently.

At one time, rote repetition was considered the only way to learn basic math facts. However, studies[10] have shown that children can derive basic math facts. It has been shown that children use a variety of strategies to "memorize" basic math facts. These strategies include memorizing "ties" first (ie. 3 + 3 = 6, 4 + 4 = 8, etc.) and then deriving other answers from there. Another common strategy is to "bridge 10." For example, if you wanted to add 8 + 5, you could change the problem into 8 + 2 + 3 or 10 + 3 which equals 13.

So, although rote repetition is one way to learn basic math facts, a decision must be made as to whether it is preferable over automatization of derived facts. One argument for rote learning is that it is more efficient in terms of time spent with a teacher. However, it could be argued that rote repetition only provides short term gains while automatization of derived facts can lead to long term gains.

One area of concern regarding rote learning is the tendency to produce interference errors. For example, if a student believes that 54 is the answer to 8 x 7, when the real asnwer is 56. It's easy to make this mistake because 54 is the memorized answer for 9 x 6. This is a problem with rote learning; there is no mechanism to correct.

There is hope that a clicker system can help students because of the following passage from "Adding It Up:"

"To many students, practice is as much a part of studying mathematics as of playing a sport or musical instrument. A procedure is practiced over and over until so-called automaticity is attained. The automatization of mathematical procedures is justifiable when those procedures are regularly required to complete other tasks. Therefore things such as basic multiplication facts need to be practiced until they can be produced quickly and effortlessly. The availability of calculators and computers raises the question of which mathematical procedures today need to be practiced to the point of automatization. Single-digit whole number addition, subtracction, multiplication and division certainly need to be automatic, since they are used in almost all other numerical procedures."

Some students come with some knowledge of basic math facts, but slow recall. Others have gaps and may require some of the strategies above. This thesis was undertaken to ascertain whether students can improve their recall of these basic math facts because of drilling with a classroom response system. In section 2.2, I will provide an overview of classroom response systems.

2.2 Classroom Response Systems

Classroom Response Systems are offered by several competing companies. Most companies sell the hardware and then give away the required software. Some companies allow third parties to have access to an API or SDK while other companies only provide proprietary software and do not encourage third party developers. This seems odd considering the great success that Apple has had with thirty party apps developed and sold for the iPhone.

Some clicker companies are now moving to purely software based systems that do allow input from Apple iPhones, iPads and Android-based smartphones and pads. Even so, the clicker based response system may be around for a while because not all students can afford smartphones and it is certainly cheaper for schools to buy clickers than smartphones. Regardless, a clicker drill system is mainly based on back-end software and would work with any input device.

2.2.1 Capabilities and Limitations of Classroom Response Systems

Classroom Response Systems have several advantages over traditional assessment options. However, Classroom Response Systems also have problems that need to be consider in system design.

Classroom Response Systems help teachers and professors by allowing formative assessments to be performed without the normal labor of grading. When authoring clicker exams, content can be easily duplicated, modified and reused as desired. Classroom Response Systems do not require paper copies.

2.2.2 Hardware

There are currently two basic types of clicker system hardware: IR and RF. IR Classroom Response Systems use infrared light to communicate between the clicker and the receiver. RF Classroom Response Systems use radio frequency signals to communicate between the clicker and the receiver. While some vendors still offer IR systems, they offer several major functional disadvantages when compared to RF systems. So, RF systems will probably be the main type of clicker system in the future. Below, some of the advantages and disadvantages of IR systems and RF systems are discussed.

IR Systems

IR Classroom Response Systems use infrared light to communicate between clickers and a receiver. IR systems tend to be less expensive than RF systems, but they have a number of functional disadvantages that may eliminate their use in future Classroom Response Systems. Because of the relatively inexpensive cost of development and therefore a lower consumer cost, some clicker vendors initially offered Classroom Response Systems based on this technology. Aside from the low cost, there are mostly disadvantages to IR systems.

IR systems are only one way communication, so the user does not get any feedback whether their response was received on the clicker. Because of this, response status must appear on the projector screen. This is problematic because response status indicators for large classes can steal real estate from presented questions and answers.

There are several disadvantages to IR based Classroom Response Systems. Because they are based on light transmission, they require line of sight between the clicker and the receiver. This is a problem in a classroom setting. Clickers must be held in the air and this increases the chance of cheating. Also, this increases the chance that a signal might not be received. For some reason, IR systems move very slowly as well. There are many instances where IR clickers interfere with one another. When many students try to respond simultaneously, bottlenecks occur and some students must wait for their opportunity to submit a clicker response.

One method that vendors have used to workaround this limitation is to allow the use of multiple receivers. However, this does increase the cost of the clicker system. Also, IR based Classroom Response Systems allow relatively few students per receiver. Experience with this kind of system indicates that 40 students pushes the bounds of acceptable performance for a receiver. In universities with a hundred students or more, this would require many receivers and extra work to ensure that the approriate students are sending their signals to the appropriate receivers.

In general, the use of IR for Classroom Response Systems is discouraged.

RF Systems

RF Classroom Response Systems offer several distinct advantages over IR systems. These advantages include:

- More students per receiver (almost unlimited)

- Faster response (no bottlenecks)
- Feedback that a response was received on the clicker (for RF clickers with displays)

Although the list of advantages is not long, the advantages are significant. They allow for a very comfortable clicker experience. RF Classroom Response Systems are advertised to accommodate class sizes of 1,000 students[11]. From experience, RF systems have very fast response times with class sizes of nearly 40 students and never any bottlenecks. Since RF systems do provide acknowledgment of receipt to the clickers, there is a reduced need to provide visual feedback via the projector screen. This allows a clicker quiz to prioritize display real estate for displaying questions.

Classroom Response Systems use a variety of RF protocols. However, in the future it is likely that more Classroom Response Systems will use bluetooth or wifi.

There are a couple of disadvantages to the RF based clicker system. First, the cost is higher than for IR systems. Second, RF systems are susceptible to interference from other RF based devices like wifi access points, other Classroom Response Systems and possibly cell phones.

Smartphone-based Systems

As smartphone technology becomes ubiquitous, there has been a natural tendency for clicker companies to move in that direction. As a result, some clicker companies now allow their clicker software to receive inputs from Apple iPhones and Android smartphones. This may become more common in the future, but not all students/parents can afford a smartphone at this moment in time. Regardless, the ideas presented here would work well with any input device and the smartphone is no exception.

2.2.3 Other Hardware Considerations

Aside from communication protocol, there are other hardware concerns to consider when purchasing or designing Classroom Response Systems. Some RF clickers offer extensive keypads with many keys, while some have fewer keys. Each type has it's own advantages. Fewer keys means simpler operation and that dexterity will be less of an issue. More keys means a wider variety of possible applications. Some clickers have numeric keypads laid out like a calculator or phone keypad while others have less traditional arrangements of keys.

There are clickers that have LCD displays while others have no display. The display is important for providing users with feedback (ie. signal received) and also for more advanced applications where someone might want to verify their input before submitting.

Some clicker vendors are now also offering graphic tablets for use with their Classroom Response Systems. Some tablets have displays and others do not. In general, it can be said that there is a wide variety of hardware options to consider when designing software applications for clickers.

Another consideration is wear and tear. One thing that clicker vendors must work on is providing clickers that have keys that are more comfortable and more durable- like those found on cordless phones and cell phones. Given the current price point for clickers (around $40), they are too expensive to be a consumable. If Classroom Response Systems are to reach wide scale adoption, users must have hardware that is reliable and robust enough for regular use.

2.2.4 Software

Most clicker vendors offer some kind of plug-in in order to use their system with PowerPoint presentation software. PowerPoint is the ubuiquitous tool for making presentations and so it is the natural choice when presenting clicker questions.

PowerPoint makes it easy to author presentations and provides a visually pleasing way for viewers to view a presentation. Unfortunately, the PowerPoint paradigm is rather limiting to the type of applications that could be developed for Classroom Response Systems. Specifically, it implies a passive activity: sitting and watching a presentation.

Clickers are essentially multi-person input devices. Games systems have traditionally allowed for the use of keyboards, mice and joysticks. With the advent of the Wii, things changed and new games could be developed based on a new generation of input devices. If a virtually unlimited variety of applications can be developed for mice, keyboards and joysticks, then a wide variety of applications should be possible for Classroom Response Systems.

Most clicker vendors do allow their clicker software to work with PDFs, Microsoft Word and even web browsers. But even these applications are mostly in the form of "present and react."

Some adventurous clicker vendors do offer API's or SDK's for developers to create their own applications. With an API, a creative software developer can create a variety of interactive experiences for Classroom Response Systems. Even so, clicker vendors are slow to promote third party development for their systems. Considering Apple now has 100,000 developers and 140,000 applications[12], one might think that clicker vendors would be happy to encourage the development and adoption of applications for their systems. There are a few things that more vendors should do to encourage a greater adoption of Classroom Response Systems:

- Encourage more application development
- Offer up API's or SDK's
- Consider making the SDK open source
- Establish development sites like SourceForge that encourage developers to share their products and knowledge with one another
- Establish sites for producing and sharing subject matter content

2.2.5 Content

Classroom Response Systems involve functionality and content. Application functionality is dictated by software. Content can be static (with fixed questions and answers) or algorithmically generated (where questions and answers are changed based on some formula). Algorithmic content can be generated by software. However, clicker exams that are derived from static content are useful and are actually the dominant variety. Currently, there are thousands of teachers and professors authoring clicker exams, yet there is no vehicle for sharing content (actually this is similar to the situation regarding traditional exam development). Clicker system vendors could benefit from a system that would allow authors to share.

Software algorithms could be used to generate content or even to provide static content in a dynamic fashion. One example of this is a tree-like presentation where some branches of data are presented, from a database, due to student choices and other branches are ignored. The content is static, but the choices that students are given is dynamic.

If developers offered algorithmically generated content, the burden on authors to produce clicker exams would be greatly reduced.

2.2.6 Modes

Classroom Response Systems typically offer two modes: group and self-paced. In group mode, the entire group of students views one question and responds to that one question together. There is usually a timer, but the instructor has the option of extending the time if desired. So, in group mode, a particular question answering period is ended when the timer counts down to zero or the last person answers the question. Typically, most students finish well before the end of the question period while a few students take longer- sometimes much longer. This is one problem with traditional PowerPoint clicker exams.

In self-paced mode, students are typically given a printed test of which the students can answer the questions at their own pace. This mode could possibly replace the use of scantrons. Like group mode, students that finish first must wait for other students to finish.

Like self-paced mode, some Classroom Response Systems also offer a homework mode which allows students to answer their questions at their own pace and then at some point just have the clicker rattle off answers to the receiver. The difference between self-paced mode and homework mode is primarily in the design of the clicker itself and not in the clicker system software on the PC.

2.2.7 Answers

Classroom Response Systems allow students to answer questions in a variety of ways. The most common type of question is multiple choice. However, for clickers that have more extensive keypads and displays, it is possible to have short answers as well.

Chapter 3: Technical Details of the Work

Like Chapter 1 and 2, this chapter has two main sections. In section 3.1, the math fluency experiment is discussed. In section 3.2, the technical details of implementing the clicker software are discussed.

3.1 Developing Fluency in Basic Math Facts using Drilling

The experiment involves a system that would allow drilling students on basic math facts in a way that would motivate them to try and keep trying and also hold students accountable with minimal effort on the part of the teacher. Here are several features of the drill system:

- Present many questions at once
- End the session when the fastest student is finished
- Generate the questions and answers algorithmically
- Allow students to enter actual answers instead of just choosing an answer
- Display a leaderboard at the end of the session
- Allow multiple sessions to be conducted without having students have to login repeatedly
- Allow students to answer multiple questions without having to specify question number

The premise is that if students are drilled regularly, the better their recall of basic math facts will be. The goal is to get improved recall. Competing for speed is also a part of the application. Basic math problems and their answers are generated algorithmically. The leaderboard is not an original idea, but for basic math drills, the leaderboard encourages competition and therfore provides and incentive to participate.

Below several technical aspects of the system design are discussed. In section 3.2, technical aspects of the clicker system and software design tools are discussed.

Data Entry Issues

When designing a system that tests for response speed, the ease in which data is entered is a critical element. Clickers are not quite as easy to use as a computer keyboard or even a well designed cell phone. For this reason, software must anticipate and compensate for the inadequacies of the input device. Below is a picture of the RF clicker used in the experiments.

Figure 1. Interwrite PRS RF Clicker

For data entry, an assumption was made that the self-paced mode- of the chosen clicker system- would be required in order to process multiple questions where students work at their own pace. Prototyping (described in section 3.2.4) showed that self-paced mode was not required. That was fortunate because using self-paced mode had a negative impact on user data entry. In order to answer questions in self-paced mode, students had to use their clicker cursor to move to question 1 (Q01 on the clicker), enter the answer, press the Enter key, then press the down arrow cursor to move to Q02, enter the answer, press the Enter key, etc.

This requirement to continually have to move to the next question number made data entry very cumbersome. Because of this realization, self-paced mode was replaced in order to improve speed and accuracy of data entry. The software API allowed users to send clicker data by pressing an answer and then hitting enter. The receiver and computer received the data regardless of the current mode.

Unfortunately, outside of self-paced mode, the system does not keep track of which question number is being answered. So, there was a need to create an algorithm to track which answers were coming from which students and how many had been received. The algorithm takes in the first response from a particular user and treats that response as the answer to the first math problem. Then, when it receives another response from the same user, it assumes that this next response was the answer to the second math problem and so on and so forth. This drastically improved the entry speed, but also introduced the possibility that some students would get "lost" at some point and then be out of the race. A teacher that trialed the system expressed concern about students "getting lost." So, a question that might be pursued in the future is: which is more important, that the students have the ability to enter data quickly or that there is a low chance of getting "lost" occasionally?

Scoring and Fairness

A speed based application can be unfair to good students that just process information slowly. For this reason, scoring and fairness must be considered when designing clicker based applications. There are several ways to deal with this issue. First, a modified scoring mechanism could be used. Student accuracy could be measured generating a percentage based on the number of correct answers compared to the number of attempted questions. Another measure would be to compare the number of correct answers to the total number of questions. If the accuracy percentage were averaged with the absolute or raw percentage, then scoring would appear to be more fair.

User Gratification

Students are motivated to compete because of score, but another aspect of user satisfaction is ability to compete. Giving a student a higher score for relatively low performance may still be unsatisfying and may discourage the student from continued participation. One way to improve user gratification for slower students is to include a deceleration function that increases the complexity of the problems. If the last few problems increased greatly in difficulty, then slower students would be able to finish more problems and would have a stronger feeling of inclusion.

A tournament system might also allow for improved student satisfaction by pairing students of matched ability against one another. With fewer students participating at any one time, it would be easier to have an onscreen display for visual feedback. The visual feedback may or may not be useful to the participating students, but it would definitely make things more interesting for the students waiting on the sidelines.

Section 3.2 describes the technical aspects of the software design.

3.2 The Classroom Response System

3.2.1 Hardware

This project will be impemented using an RF based clicker system. The clickers are PRS Interwrite RF clickers. These clickers have an LCD display, a numeric keypad style keypads with extra keys for navigation and a negative sign.

The software will be developed using Visual Basic .NET 2008 Express edition. VB was chosen because a VB project and ActiveX control was provided as part of the SDK for the clicker system. Microsoft Access will be used for the database because of the ease of prototyping with VB and the fact that the database is not expected to be very large. If the database was expected to be large, Microsoft SQLServer or MySQL would have been a better choice for database development.

3.2.2 SDK

The vendor, eInstruction, does not normally allow third parties to have access to their SDK. The SDK was provided after a non-disclosure agreement was signed. Although a Visual Basic project was provided with the SDK, the project was very limited in it's capabilities. The VB project simply provided examples of the events generated by the ActiveX control provided with the SDK. Even so, a modified version of the SDK form and several of the display features of the provided VB project were retained and included in this project.

3.2.3 Software Design Considerations

Hardware Limitations

With PowerPoint clicker questions, one question is presented at a time and students typically send one answer per question over a period of time that could be anywhere between 30 seconds and several minutes. The idea behind this new clicker application is to have all students enter as many answers as they can as fast as they can in a short period of time. There were no guarantees that the clicker system could handle this increased demand on bandwidth.

Fortunately, the specifications for the system that will be used in the experiments indicate that the system can be used for as many as 1,000 students. Internet searches indicated that the practical limit of students per receiver is closer to 400. Regardless, this new application will only be used for at most 40 students at a time. So, it is possible that the system might be able to handle many responses in a short period of time.

Realtime Data Storage

What can be done to ensure that all responses are captured reliably? If you miss a response, it's gone forever. Storing directly to a database is time consuming. Storing in memory is faster.

The initial design involved saving directly to the database for every response. In the second version, data was saved in arrays and then the instructor is given the option to store the session to the database on the hard drive.

Separating Storage and Processing Code From Event Code

The SDK is proprietary, but the idea of this software application is generic. The first prototype had processing code all mixed up with the proprietary SDK event code. In the second version, processing and display code was separated and some classes were defined to provide more structure to the design process.

Display Options

Several choices present themselves when deciding how to display questions, answers and results. What is the best way to display the problems? How do you display a table in Visual Basic? What is the best way to display results after a match? A question to pursue in the future is: is there an effective way to display realtime results for as many as 40 students at a time? In order to get something done relatively quickly, a decision was made to display the questions using a Visual Basic Rich Text Box with text laid out in tabular format. The text of the questions are color coded using a concept called syntax coloring. The problem number is one color, the problem itself is a different color. When a match is completed, the answers appear next to the problems and they are yet another color for easy distinction.

One issue with computer projectors is brightness. In the experiment classroom, large curtains are used to dim the outside lighting, so it was too cumbersome to have to do that all the time. As a result, contrasting colors were an issue. You can see a first attempt at question display and text coloring in Figure 3: Prototype Problem Display Window. In later attempts, the background was black and the text was colored white. The experiments discussed in this paper were all performed with a black background and white text.

Report Generation

Visual Basic Express puts some limitations on report generation capabilities. The professional version of Visual Basic Studio does include Crystal Reports which makes it simple to generate reports. Microsoft Access does provide report generation capability, but it may require some research to determine whether that would have an impact on application distribution. For this experiment, the software did not include any report generation capabilities. Any reports produced were a result of pulling the data from the database using queries and analyzing the data using a spreadsheet program.

Non-disclosure Agreement

Because of the non-disclosure agreement, permission must be obtained to distribute this application. If distribution is a consideration, this does have an effect on design. The less that this application is tied to a specific clicker system, the greater the opportunity and freedom of distribution. Several vendors have been contacted regarding the further development and possible distribution of the application used in this experiment.

3.2.4 Prototyping and Algorithms

Prototyping had to be done before the application could be delivered to the classroom.

Storing the data seemed was a big concern and some preliminary trials were required to ensure that a particular method would work well. The real concern here was regarding whether the hardware could support many students responding at once.

Also, system response time was a concern. Since the system was capable of dealing with more than a 1,000 users responding in a few minutes, it was hypothesized that the system might be capable of handling 40 users responding in a few seconds. However, this needed to be proved before beginning design or implementation of many, time consuming application features.

So, a prototype was designed and implemented that was based on the sample application provided by the clicker vendor in the SDK. A screen shot of the initial prototype appears below.

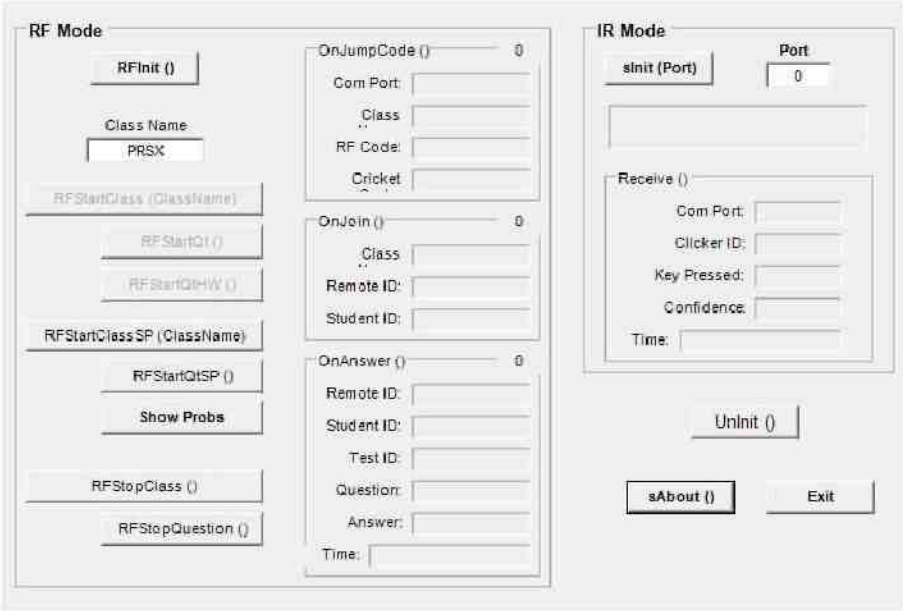

Figure 2. Prototype Software Status Console

Even a prototype required some kind of display mechanism for the problems. To quickly produce a display, several options were considered: labels on Visual Basic forms, buttons on Visual Basic forms, a grid (table) or a richtextbox. Any one of these options was viable, but the richtextbox appeared to be the most straightforward. So, here is the initial look of the display:

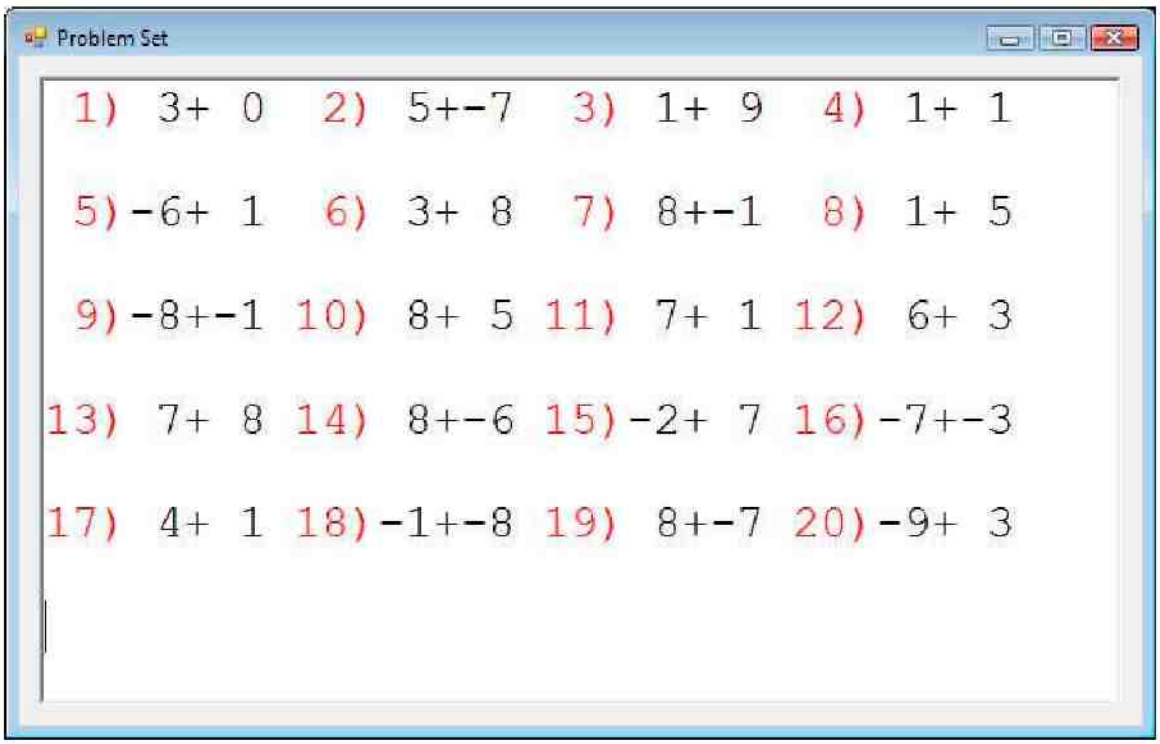

Figure 3. Prototype Problem Display Window

Next a decision had to be made regarding how to store the data. The first approach was to use Visual Basic to store the data in a Microsoft Access relational database. This is a common rapid prototyping approach and it seemed appropriate for this project. Eventually, this mechanism was changed in order to improve processing speed and reduce the chance of lost data. The method used for storage in this experiment involved saving data to arrays in memory and then allowing the instructor to optionally transfer the information in memory to non-volatile storage on the Microsoft Access database on the hard drive.

The SDK provided a sample application that provided some insight into the events that are exposed to a programmer by the vendor's ActiveX control. The ActiveX controls (PrsX) serves as a realtime monitor for the clicker system receiver plugged into the USB port of a PC. The ActiveX offered a number of events that the programmer can tap into in order to keep track of user inputs.

These events included an OnJoin event and an OnAnswer event. These are the two primary events required in order to process inputs from the clickers. Each event also had it's own set of associated parameters.

The OnJoin event is triggered when a user attempts to join the clicker class. When the OnJoin event is triggered, the user's clicker number and student ID is stored, so that all session responses can be matched with the appropriate student. When an OnAnswer event is triggered, several things happen:

- the clicker ID is matched with the student ID
- a question number counter is created for the particular student ID
- the counter is incremented from zero to one
- the question number, answer and student ID are stored in an array
- when the same student answers another question, the counter is incremented and the question number, answer and student ID are stored in the array
- when some student answers question 20, the match stops and a leader board is displayed with the results of the match with students sorted by most correct answers

This algorithm is significant because it allows the system to keep track of question numbers without the need for students to enter the question number. This greatly speeds data entry, but does add the possibility that a student might enter answers out of order. There was a calculated risk in going with this option, but the desire to allow for faster entry was the stronger motivator for this experiment.

As a result of earlier experimentation, it was determined that students could intentionally end the match early by just repeatedly hitting the Enter button on the clicker. As a result, a heuristic was used to prevent this from happening. In the experiment, the software would not stop the match until a student with a score of 13 or higher answered question 20. The students were not told of this criteria and none were able to stop the system prematurely.

That concludes the summary of software implementation details. Experimental results are described in Chapter 4: Experiments.

Chapter 4: Experiments

As in previous chapters, this chapter has two main sections. In section 4.1, the math fluency experiment is discussed. In section 4.2, the technical details of the experiment are described.

4.1 Hypothesis Testing

The conjecture of this thesis was that using a clicker-based drill system would help improve students fluency and automaticity with basic math facts. So, an experiment was designed to test this assumption. In an ideal world, the experiment would entail a treatment group and a separate control group, but there was no practical way to implement such an experiment with existing resources. So, instead a matched pair design was used where the same group of students were used for the treatment and then again later as a control group. The scheme that was decided upon worked in the following manner:

4.1.1 Experiment Design

A benchmark test was chosen that would test students' current speed and accuracy with a pencil and paper speed drill for adding integers. The benchmark test had 54 problems arranged in 3 columns of 18 problems and students were given 90 seconds to complete the test. Limiting the testing to one particular operation would reduce the number of variables or factors to consider.

Initially, the idea was that students would be tested for just adding whole numbers, but that idea was dismissed after an initial test was given and most students were able to finish most of the speed test with almost perfect accuracy. So, the next level of interest was adding integers, but no subtracting, multiplying or dividing.

Math worksheets (see Figure 4 below) from the internet (math-drills.com) were used to establish the baseline performance before treatment and after treatment. Since, no control was possible, the following methodology was used:

- Provide initial baseline test with pencil and paper
- Administer a week of treatment (students using the drill system)
- Provide a similar, but different, benchmark test
- Go a week with no further treatment (this week is intended to serve as the control)
- Provide a final benchmark test

About 20 students participated in 3 benchmark tests. Some students did not participate in all tests.

Integer Addition; Range (-9) to (+9) (A)

(-7) + (+2)	=	(-2) + (-4)	=	(-1) + (+8)	=
(+5) + (-7)	=	(+9) + (+3)	=	(+2) + (-7)	=
(-2) + (+9)	=	(-2) + (+1)	=	(+6) + (-8)	=
(+2) + (-6)	=	(-3) + (-8)	=	(-6) + (-8)	=
(0) + (-2)	=	(-3) + (+6)	=	(-8) + (+4)	=
(+6) + (+6)	=	(+6) + (+9)	=	(-6) + (-3)	=
(-7) + (-2)	=	(-5) + (+1)	=	(+2) + (+6)	=
(+9) + (-8)	=	(-5) + (-4)	=	(-4) + (-5)	=
(0) + (-7)	=	(-8) + (-1)	=	(+5) + (+1)	=
(+1) + (-3)	=	(-5) + (+9)	=	(+6) + (-4)	=

Figure 4. Drill worksheet from math-drills.com

4.1.2 Results

The first benchmark test was given on Monday, February 28th at the end of the 95 minute class period. This benchmark was given after a week of lessons regarding operations on integers. Most of the students had already been exposed to integers in prior years, but high school pre-algebra classes always include a refresher on integer operations. The test was administered at the end of the period after a math lesson and practice. There are pros and cons for giving the test on a Monday and at the end of the period. It is possible that students brains and minds are "warmer" after a week of practice and they may be "cold" after a weekend of rest. There are similar concerns regarding giving the benchmark test at the end of a class period.

There were two performance indicators considered: percent correct out of problems attempted and percent correct out of all possible problems. Looking at all possible problems is the more absolute indicator. If only attempted problems are considered, a student could get away with getting 100% by only doing 1 or 2 problems. However, if a student is truly putting forth their best effort, the problems attempted criteria could be used to determine whether the student has a good grasp of the basic math facts, but processes information more slowly or is just not speedy at entering data into the clicker.

The first benchmark test resulted in a class average (20 students) of 78% correct out of problems attempted and 35% correct out of 54 total problems. Of course, the 35% figure was attributable to the timed nature of the test.

After the first benchmark test, the class began a week of treatment. Every day, near the end of the period, we spent as much time as possible using the clicker drilling system. Students already had familiarity with the use of the clicker drilling system from previous weeks, but this was the first week where integer operations were introduced.

Each clicker session started with whole number operations and then progressed to integer operations. This was partially due to the requests from the students who preferred operations without negative signs.

Sessions involved several basic integer operations including addition, multiplication, subtraction and division. The number of sessions involving each type of integer operation (addition, multiplication, subtraction or division) varied. Anywhere from 15 to 30 minutes were spent each day with clicker drills.

At the end of the first week of treatment, on Friday, March 4th, a second benchmark test was given. Again, 54 questions were used and 90 seconds was the time alotted. The class scored 81% correct out of problems attempted and 42% out of 54 problems. This did indicate some improvement, although not necessarily statistically significant.

The following week, no treatment was given. The hypothesis here was that if treatment were no longer applied, student performance on the benchmark test would go down. So, on Friday, March 11th, another benchmark test was given. The class scored 90% correct out of problems attempted and 58% correct out of 54 total problems.

So there was an increase in performance after a week of treatment, but there was an even larger surge in performance after a week with no treatment. If students improved more during a week with no treatment, than they did during a week with treatment, this seems to require the rejection of the hypothesis that the treatment improved students fluency in basic math facts. However, there are a number of factors involved and there certainly could be lurking variables. These factors will be discussed in Chapter 5: Analysis of Results.

4.1.3 Alternative Results

Althoug the benchmarks did not show statistically significant improvements in scores, some data mining was performed using the data produced by the daily sessions during the week of treatment. The clicker software collected and stored data for every session (or match) for every student. So, Microsoft Access queries were created to compute the average score of all addition sessions for each student for each of the 5 days of treatment. A top scores query was also created (just in case the average score was not a good indicator of improvement).

Of course, in any experiment such as this, some students put in more effort than others and some students put in more effort on some days than on others. However, after looking at the average scores for individuals, it can be seen that there was an upward trend in average scores for some students. For more details, see Chapter 5: Analysis of Results.

4.2 Experiment Design from Technology Perspective

The initial introduction to the clicker system went very well. For clarification, here is an explanation of the scenario of usage in the first classroom experiment:

- Student names and student ID's were manually entered into the Microsoft Access database before class. Information was stored in a table called, "roster."
- The software was run, the InitRF button was pressed and the eInstruction ActiveX control eventually presented the class login code.
- Students were instructed to turn their clickers on, enter the login code, their student ID and then wait for the session to begin.
- The software displayed each student ID as they joined providing some feedback that the this part of the scenario was working. A button and code were added to "Check Role" and that was used to determine whether all students had joined the current class. For those students that were in the database roster but had not signed in, their names were displayed when the Check Role button was pressed. This really helped to ensure that all students were participating.
- Student clicker displays had a prompt, "ANS:", that indicated the clickers were ready to send answers. No answer could be submitted until the session was initiated by the teacher.
- After all present students were accounted for, the session was initiated with addition of integers as the first activity. A set of 20 integer addition problems was displayed as can be seen in Figure 3: Prototype Problem Display Window.
- Students raced to enter the answers. Entering the answers was a multiple step process. Students had to look up at the problem display, look down at the clicker, press the numbers for the answer (and possibly a negative sign), press the Enter key to submit the answer and then press the down arrow to move the clicker cursor to the next question number.
- When the first person reached the 20th question, the session was ended and the leaderboard was displayed.

Students asked to repeat the activity and this occurred several times. Eventually, the novelty wore thin and the activity came to a halt. The entire activity was captured on video tape for later analysis and presentation. Several lessons were learned during this first experiment:

- Students that initially did not make it to the leaderboard, eventually did make it to the leaderboard. This was a surprising result. The anticipation was that fast students would always dominate and slower students would be left behind all the time. However, it's possible that the competitive nature of the program drove some students to improve their performance. It could be conjectured that what really happened was that faster students just slowed down due to diminished interest except that the change of standing occurred fairly quickly-after only two or three sessions.
- Eventually, faster students did lose interest and drop out of the activity.

In general, students were interested in using the software and the software and hardware seemed to keep pace with the relatively high bandwidth activity. Students were eager to compete against one another and they were very excited when the leaderboard popped up. This seemed to motivate students to want to use the system more.

4.2.1 Subsequent Classroom Experiments

RF Interference

One very high performing student began to get very low scores consistently. His clicker was replaced with a different one, but his scores continued to be low. He became very frustrated. It turned out that his cell phone was on the desk near his clicker. This student was asked to turn off the cell phone and put it away. His scores went right back up to where they previously were. The clickers use RF for communication and so do cell phones and it's possible the cell phone was interfering with the operation of the clicker.

Variety of Activities

The other major problem with the software was the limited number of operations and difficulty led to diminishing interest in the activity. For this reason, additional operations were added including multiplication, subtraction and division. Also, some students grew frustrated with trying to solve problems with integers. For this reason, an option was added to switch between integers and whole numbers (negatives or no negatives).

Chapter 5: Analysis of Results

5.1 Hypothesis Testing

The goal of this thesis was to determine whether a clicker based drilling system would improve fluency with basic math facts for remedial math students. Unfortunately, the results of the benchmark tests seemed to reject that hypothesis. Even so, some interesting data did show up in an alternative analysis.

5.1.1 Analysis of Treatment Data

Even though the paper and pencil benchmark tests did not indicate that the treatment helped improve student fluency, there was several days worth of data from the days of treatment. So, although the initial intent of the experiment was to use the benchmark tests to evaluate the hypothesis, it certainly would be reasonable to look for evidence of performance gains in the daily treatment data as well.

The data could be filtered based by student, session, session score and operation (addition, multiplication, etc.). Several database queries were created to find the following two performance metrics: 1) average daily scores for addition sessions per student 2) top daily scores for addition sessions per student. This would have been a great application for data mining (in order to search for patterns that could not be anticipated) and if there was more time, that would have been done. However, it took some effort to get the queries to produce the correct data to analyze just these two metrics.

In section 5.2, all of the data from the benchmark tests, the average daily treatment scores and the top daily treatment scores are available. After plotting the data, it became apparent that there was an positive growth trend for 8 of the participating students in average daily score.

In Figure 11, a graph is included to illustrate the positive growth trend shown for 8 of the participating students. There are only 17 students shown in Figure 10 because three of the 20 students in the experiment missed 2 or more of the 5 days of treatment. The positive growth trend for these 8 students is reinforced numerically in Figure 12. Figure 12 shows the slopes of trendlines for each student's daily average scores.

This is very interesting data. The reason why it's interesting is because it's difficult to "fake" the average daily score or accidentally show an upward (or downward) trend. Below is a statistical analysis regarding the likelihood that 8 out of 17 students would show consistent improvement.

One way to look at the data is to say that 8 out of 17 students is 47% and to have 47% of a sample show consistent improvement is statistically significant. It is also possible, however, to use a One Sample t-Test as a hypothesis test. Unfortunately, 17 students is not a large enough sample to assume a normal distribution and so another method must be used to determine whether the population distribution is Normal.

Here are some simple statistics related to the data in Figure 12:

Min	Median	Mean	Max
-1.95	-0.8333	-0.01814	2.45

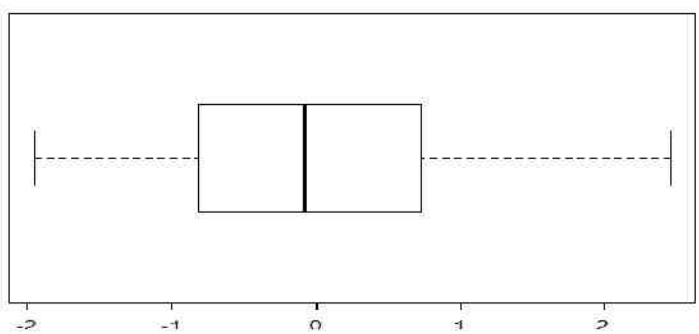

Figure 5. Boxplot of slope trendlines for daily average score

The boxplot seems to indicate that the sample distribution might be Normal, but a better test would be the Shapiro-Wilk normality test available via R- shapiro.test(x). The test calculated a p-value of 0.8724. In the R Cookbook by O'Reilly, it is stated that, "A high p-value, suggests that the underlying population could be normally distributed."

In order to use a One Sample t-Test, other conditions must be met. The sample must be a simple random sample. That is fair assumption in this case. The classroom is at a school that is in a middle class area. The only thing that the students have in common is that they have performed poorly in math prior to this class- Algebra Readiness. Otherwise, there is a mix of ethnicities, income level and gender. This is not a random sample with regard to the general population. However, it is a random sample among remedial math students.

Another condition that must be met is independence. Are the individual observations independent? The students do interact during the matches, but the data is based on daily averages over a 5 day period. The slope of the trendline takes into account many sessions over a 5 day period and while it's possible that a student might degrade in performance based on interaction with other students, it's highly unlikely that they would improve in performance. So, there is a good case for independence between the slopes of trendlines for average daily scores.

With the conditions for a One Sample t-Test being met, the calculations were performed using R. The null hypothesis is that the slope of the trendline for the students daily average during treatment will be zero- which means that students show no improvement. The null hypothesis will be rejected if the slopes of the trendlines for daily average scores are positive- which means that students do show improvement. R produced the following results:

```
t = -0.0699, df = 16, p-value = 0.5274
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 -0.4712089        Inf
sample estimates:
  mean of x
-0.01814265
```

The main thing here is the p-value. It's a rather high p-value which leads to the conclusion of statistically non-significant. However, it's still possible to use a confidence interval for another perspective. In this case, R produced the wrong confidence interval. This happened because a one-sided test was specified. R will produce the correct confidence interval when a two-sided test is specified as seen here:

```
t = -0.0699, df = 16, p-value = 0.9451
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.5682696  0.5319843
sample estimates:
  mean of x
-0.01814265
```

Here the p-value is incorrect, but the confidence interval is correct. What this means is that based on this sample, there is a 95% confidence that the true mean of slopes of trendlines for students in the greater population would fall between -0.5682696 and 0.5319843. Again, this is not a clear indication of whether students in the greater population would likely improve in their math fluency through use of the clicker system. However, it does not wholly discount the possibility either.

5.1.2 Student Effort and Motivation

Another factor that may have skewed the outcome has to do with student motivation. Some students are indifferent to various classroom activities and, in fact, are not motivated by grades or disciplinary actions. So, it's possible that some students had varying levels of effort throughout and even possibly declining levels of effort.

5.1.3 Additional Math Lessons

Although there was no treatment applied during the week prior to the 3rd benchmark test, there were math lessons every single day. It's possible that between the previous week of practice and the daily math lessons, students did improve. This is one theory, however, the rate of improvement during the week without treatment was pretty significant. Regardless of why students improved greatly on the 3rd benchmark, the results make it difficult to conclude that the treatment alone helps to improve math fluency.

In section 5.2, all of the data is presented. See Chapter 6: for conclusions.

5.2 Data

Here are snapshots of the spreadsheets and graphs that were used to analyze the data:

Name	Correct	Out of	percent correct/attempted	percent of total	average
Student01	20	21	95%	37%	66%
Student02	26	28	93%	48%	71%
Student03	28	28	100%	52%	76%
Student04	40	42	95%	74%	85%
Student05	11	13	85%	20%	52%
Student06	0	9	0%	0%	0%
Student07	9	12	75%	17%	46%
Student08	23	24	96%	43%	69%
Student09	5	24	21%	9%	15%
Student10	34	39	87%	63%	75%
Student11	38	38	100%	70%	85%
Student12	20	20	100%	37%	69%
Student13	13	16	81%	24%	53%
Student14	11	29	38%	20%	29%
Student15	16	24	67%	30%	48%
Student16	10	18	56%	19%	37%
Student17	29	30	97%	54%	75%
Student18	23	24	96%	43%	69%
Student19	9	10	90%	17%	53%
Student20	18	19	95%	33%	64%
Averages	19	23	78%	35%	57%

Figure 6. Benchmark 1, February 28th

Name	Correct	Out of	percent correct/percent attempted	percent of total	average	diff in % compl	diff in % total
Student01	9	9	100%	17%	58%	4.76%	-20.37%
Student02	22	37	59%	41%	50%	-33.40%	-7.41%
Student03	22	27	81%	41%	61%	-18.52%	-11.11%
Student04	48	54	89%	89%	89%	-6.35%	14.81%
Student05	absent						
Student06	1	20	5%	2%	3%	5.00%	1.85%
Student07	17	18	94%	31%	63%	19.44%	14.81%
Student08	21	21	100%	39%	69%	4.17%	-3.70%
Student09	14	30	47%	26%	36%	25.83%	16.67%
Student10	45	47	96%	83%	90%	8.57%	20.37%
Student11	40	41	98%	74%	86%	-2.44%	3.70%
Student12	22	23	96%	41%	68%	-4.35%	3.70%
Student13	17	18	94%	31%	63%	13.19%	7.41%
Student14	2	3	67%	4%	35%	28.74%	-16.67%
Student15	18	30	60%	33%	47%	-6.67%	3.70%
Student16	17	28	61%	31%	46%	5.16%	12.96%
Student17	34	36	94%	63%	79%	-2.22%	9.26%
Student18	37	38	97%	69%	83%	1.54%	25.93%
Student19	20	20	100%	37%	69%	10.00%	20.37%
Student20	20	20	100%	37%	69%	5.26%	3.70%
Averages	22	27	81%	42%	61%		5.26%
Std Dev							13%

Figure 7. Benchmark 2, March 4th

Name	Correct	Out of	correct/attempted	percent correct	average	diff in % compl	diff in % total
Student01	32	33	97%	59%	78%	-3.03%	42.59%
Student02	25	30	83%	46%	65%	23.87%	5.56%
Student03	35	40	88%	65%	76%	6.02%	24.07%
Student04	51	54	94%	94%	94%	5.56%	5.56%
Student05	14	18	78%	26%	52%	77.78%	25.93%
Student06	7	10	70%	13%	41%	65.00%	11.11%
Student07	38	45	84%	70%	77%	-10.00%	38.89%
Student08	29	30	97%	54%	75%	-3.33%	14.81%
Student09	22	30	73%	41%	57%	26.67%	14.81%
Student10	49	54	91%	91%	91%	-5.00%	7.41%
Student11	45	45	100%	83%	92%	2.44%	9.26%
Student12	42	42	100%	78%	89%	4.35%	37.04%
Student13	27	29	93%	50%	72%	-1.34%	18.52%
Student14	25	30	83%	46%	65%	16.67%	42.59%
Student15	29	33	88%	54%	71%	27.88%	20.37%
Student16	28	38	74%	52%	63%	12.97%	20.37%
Student17	33	34	97%	61%	79%	2.61%	-1.85%
Student18	40	40	100%	74%	87%	2.63%	5.56%
Student19	30	30	100%	56%	78%	0.00%	18.52%
Student20	30	30	100%	56%	78%	0.00%	18.52%
Averages	32	35	90%	58%	74%		18.98%
Std Dev							13%

Figure 8. Benchmark 3, March 11th

Name	02/28	03/01	03/02	03/03	03/04
Student01	12	12		11	8
Student02	10	13	13	19	20
Student03	10	6	11	10	11
Student06	6	9	7	15	12
Student07	10		12	15	12
Student08	8	9	9	10	9
Student09	6	8	8	16	12
Student10	20	18	18	20	19
Student11	16	16		15	15
Student12	13	9	11	16	9
Student13	9	11	20	12	8
Student14	7	1		10	11
Student15	13	14	13	20	14
Student17	16	14	17	14	13
Student18	13	0	13	14	12
Student19	14	13	12	16	18
Student20	15	15	11	13	14

Figure 9. Treatment Week, Daily Top Scores

Name	02/28	03/01	03/02	03/03	03/04
Student01	6.5	10		4	4
Student02	8	9	11	12	11
Student03	7.25	5	9	5	9
Student06	2.5	4	6	9	12
Student07	5.25		8	10	8
Student08	5.75	6	7	7	9
Student09	3	4	4	5	4
Student10	16.5	18	11	10	16
Student11	13.25	12		12	7
Student12	6.25	6	9	13	2
Student13	7.25	9	13	8	4
Student14	2.25	0		3	6
Student15	6.25	10	11	8	7
Student17	12.25	13	15	6	6
Student18	10.25	0	12	9	4
Student19	8.5	10	9	10	11
Student20	9.75	12	10	7	8

Figure 10. Treatment Week, Daily Average Scores

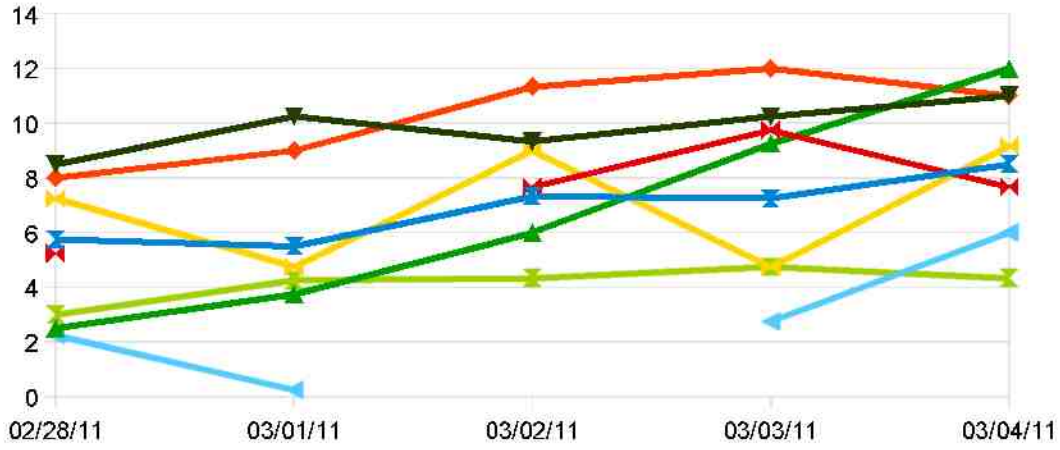

Figure 11. Treatment Week, Daily Average Scores of Selected Students

Name	Slope	>0
Student01	-1.0834	
Student02	0.9	0.9
Student03	0.3833332	0.3833332
Student06	2.45	2.45
Student07	0.8000057143	0.8000057143
Student08	0.725	0.725
Student09	0.31666	0.31666
Student10	-0.85834	
Student11	-1.31668	
Student12	-0.183334	
Student13	-0.766668	
Student14	1	1
Student15	-0.083334	
Student17	-1.95	
Student18	-0.325	
Student19	0.5	0.5
Student20	-0.816668	
Mean:	-0.0181426521	
StdDev:	1.0699682555	

Figure 12. Slopes of Linear Trendlines for Average Daily Scores

Chapter 6: Conclusions

The hypothesis was that using a clicker system would help remedial math students improve their fluency with basic math facts. There were two aspects to the experiment and subsequent analysis. The first aspect was a set of benchmark tests used to assess whether students improved. Students did improve in their benchmark test scores after one week of treatment. However, they improved even more after one week without treatment.

The second aspect of the experiment was the data that the treatment itself generated. After sifting through the treatment data, it was discovered that 8 out of 17 (47%) of the students did show improvement in fluency throughout the week. A more detailed statistical analysis did now show whether or not this indicated that the greater population would benefit from this treatment. Even so, the fact that 47% of students did show improvement over a 5 day period of treatment provides hope that this system could be used to help some students improve their math fluency.

Aside from the math fluency, the clicker drill software has a lot of potential for helping students with math. For one thing, it definitely encourages students to want to practice their basic math facts. Since the class was introduced to the clicker system, they have asked to use it almost every day. And, during clicker use, almost all students remain engaged for as much as 30 minutes at a time.

There is a lot of potential for expanding and improving the clicker software and those ideas are discussed in Chapter 7:.

Chapter 7: Future Work

7.1 Ideas for Future Improvement and Expansion

Making an API or library that allows this type of application to work with more than one vendor's Classroom Response System.

Adding more types of problems including equation solving simplifying more complicated expressions. It would be nice to use the program for higher level math classes with drills on common powers, negative exponents, fractional exponents, logarithms, factoring and others.

Adding team and tournament functionality.

Possibly adding a deceleration function that makes the questions more challenging as the participant gets to higher question numbers.

Progressing to problems with larger numbers.

Moving from simple operations to more complex operations. Initially, changing to mixed simple operations (plus, minus, multiply, divide) to more advanced operations (exponents, radicals, parentheses).

It would be nice to be able to display more elaborate equations possibly using either HTML or LaTeX.

Adding more dynamics to the display. For example, something innovative like Hans Rosling's GapMinder[13]. One question t consider here is: is it possible to create a visual display that allows 40 users to see their own status and a useful way?

References

[1] http://en.wikipedia.org/wiki/New_Math

[2] http://en.wikipedia.org/wiki/A_Nation_at_Risk

[3] http://en.wikipedia.org/wiki/Reform_mathematics

[4] Cleary. A. (2008). Using Wireless Response Systems to Replicate Behavioral Research Findings in the Classroom. Teaching of Psychology, 35(1). 42-44. Retrieved from ERIC database.

[5] Salemi M. Clickenomics: Using a Classroom Response System to Increase Student Engagement in a Large-Enrollment Principles of Economics Course. Journal of Economic Education [serial online]. January 1, 2009;40(4):385-404. Available from: ERIC, Ipswich, MA. Accessed June 30, 2010.

[6] Gebauer M. Extending the Personal Response System. March 9. 2006.

[7] Ofsted (1996) Successful Teaching of Literacy and Numeracy in Primary Schools: A Starting Point. London: Ofsted.

[8] Orton, A.. & Frobisher, L. J. (1996). Insights into teaching mathematics. London: Cassell.

[9] Kilpatrick, J., Swafford, J., Findell, B., & National Research Council (U.S.). (2001). Adding it up: Helping children learn mathematics. Washington, DC: National Academy Press.

[10] Orton, A. (1999). Pattern in the teaching and learning of mathematics. London: Cassell. P.54-59.

[11] http://www.einstruction.com/products/assessment/prs/index.html, July 1, 2010

[12] "Staggering iPhone App Development Statistics Unveiled at Macworld 2010" http://prmac.com/release-id-10695.htm

[13] "Hans Rosling shows the best stats you've ever seen" TED. 2006, http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html