Social Networking and Culturally Situated Design Teaching Tools: Providing a Socially Interactive Computer Supported Collaborative Network Environment for K-12

by

Albanie Tremaine Bolton

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 4, 2014

Culturally Situated Design Tools (CSDTs), educational gaming,
ethnomathematics, mathematics, culture, Computer Supported Collaborative Work (CSCW)

Approved by

Cheryl D. Seals, Chair, Associate Professor of Computer Science and Software Engineering
Wei-Shinn Ku, Associate Professor of Computer Science and Software Engineering
Sanjeev Baskiyar, Associate Professor of Computer Science and Software Engineering
L. Octavia Tripp, Assistant Professor of Elementary Science Education

Abstract

Culturally Situated Design Tools (CSDTs) are web-based software applications that allow students to create simulations of cultural arts: Native American beadwork, African American cornrow hairstyles, urban graffiti, and so forth; using these underlying mathematical principles. CSDTs are the rationale of creating a set of culturally designed games utilizes gaming as a teaching tool to attract and instruct students with familiar methods and environments. This dissertation will review the development and evaluation of CSDTs, and discuss how various activities attempt to navigate through the potential dangers and rewards of this potent hybrid of information technology (CSDTs), traditional culture and individual creativity.

Increasing accessibility to Computer Science Technology is essential for a discipline that relies on creativity and diverse perspectives. Creating equity in representation of numbers of students and their access to Computer Science learning opportunities is also a social justice issue. Educational research communities having begun to explore the causes behind the underrepresentation of females and students of color in computing courses, outreach efforts have commenced to overcome these enrollment discrepancies. The purpose of this dissertation is to address issues of under representation. We have created an environment and developed a framework to support the creation of computing applications promoting more robust educational software to train the next generation of computer scientists.

Acknowledgments

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CSDT   Culturally Situated Design Tools

CSCW   Computer Supported Collaborative Work

HCI   Human Computer Interaction

CS   Computer Science

Qt   pronounced Q tee

JVM   Java Virtual Machine

1 INTRODUCTION

1.1 Motivation

Diversity does not refer only to ethnicity or race. It refers to differences in social class, family culture, geographic, religious backgrounds, and learning styles, which are all reflected in our classrooms as important components of diversity [6]. Culturally responsive instruction addresses the specific interests, concerns, and experiences of students in the classroom. Therefore, teaching math and computer science in culturally responsive ways means using students' own habits, experiences, and cultural references to connect to real-world experiences with numbers, shapes, patterns, chance, and measurement. Successful methods for learning, calculating, memorizing, and communicating about math differ across cultures. This is where the phrase *Culturally Situated Design Tools* (CSDTs) comes into play. This area of work also focuses on how to incorporate Computer Supported Collaborative Work (CSCW) in a way that combines relevant factors into a more effective resource for students and instructors.

The conventional use of CSCW inside various academic fields is wide. This research will require participants to converge specific tasks to develop a shared understanding of CSCW among members of communities of practice for the purposes of data collection, analysis, and evaluation to ascertain the impact of the study on subjects. Justus Nyagwencha is currently researching CSCW in parallel with CSDT research.

1.2 General Area of Research

For over 25 years, human computer interaction (HCI) researchers and developers have been challenged with improving usability of products. More recently, the CSCW community has focused on developing collaborative systems, albeit with an emphasis on social interaction at work [34]. The widespread use of the Internet by millions of diverse users for socializing is a

new phenomenon that raises new issues for researchers and developers. Just designing for usability is not enough; we need to understand how technology can support social interaction and then design technology for sociability. Sociability is concerned with developing software, policies, and practices to support social interaction online. Three key components contribute to good sociability [34]:

- *Purpose.* A community's shared focus on an interest, need, information, service, or support that provides a reason for individual members to belong to the community.

- *People.* The people who interact with each other in the community and who have individual, social and organization needs. Some of these people may take different roles in the community, such as leaders, protagonists, comedians, or moderators.

- *Policies.* The language and protocols that guide people's interactions and contribute to the development of folklore and rituals that bring a sense of history and accepted social norms. More formal policies may also be needed, such as registration policies and codes of behavior for moderators. Informal and formal policies provide community governance [34].

Decisions about purpose, people, and policies by community developers help determine the initial sociability of an online community. Later, the community evolves and gradually establishes an understanding of which social norms and policies are acceptable and which are not [34]. This is what the CSDT tool base has taken into account for its framework and design.

Culturally Situated Design Tools allow students and teachers to explore mathematics and computer science with depth and care, using cultural artifacts from specific times, places, and cultures. *Ethnomathematics* is the study of mathematical ideas and practices situated in their cultural context. The CSDT website provides free standards-based lessons and interactive

2

"applets" that help students and teachers explore mathematics and knowledge systems using ethnomathematics in African, African American, Youth Subculture, Native American, and Latino circles. The supporting materials for the CSDTs include lesson plans and evaluation instruments to ensure they are integrated into the curriculum through state and national standards. Based in K-12 schools with significant numbers of African-American, Latino, and Native American students, preliminary evaluations indicate statistically significant increases in both math achievement and attitudes toward technology-based careers. Current locations include Alaska, California, Idaho, Illinois, Michigan, New York, and Utah.

Dr. Ron Eglash and the Rensselaer Polytechnic Institute have copyrighted the CSDT simulation software and teaching materials. Eglash, a professor at Rensselaer Polytechnic Institute and the author of *African Fractals: Modern Computing and Indigenous Design,* provides the software on the web at *http://www.rpi.edu/~eglash/csdt.html* [8]. When instructors have a sense of what issues motivate and their students they may want to search various culturally effective ways to teach them. This site provides resources for this purpose. Although the use of mathematics and computing is universal, math is not culturally neutral. That is why this research seeks how to incorporate math and computer science into gaming.

Eglash recognizes a lack of diversity in these fields that limits what individuals can do to hold the attention of students. Therefore, he constructed these tools by researching and integrating aspects of ethnomathematics, mathematics, computing, and educational gaming. Eglash's work provided the sole inspiration for this present research project. We initially focused on the Break Dancer tool (Figure 1) on the CSDT website. The Break Dancer tool is a software-simulated game that teaches 3-dimensional space or solid geometry. Real-world objects exist in 3-dimensions. For example, a cuboid, or a box, is described by three parameters, *length, breadth,*

3

and *height.* Corresponding to that, each point in the Cartesian space has three coordinates - *x, y,* and *z* [2]. The Break Dancer tool integrates the youth subculture, promotes physical activity, and makes it into an educational game.
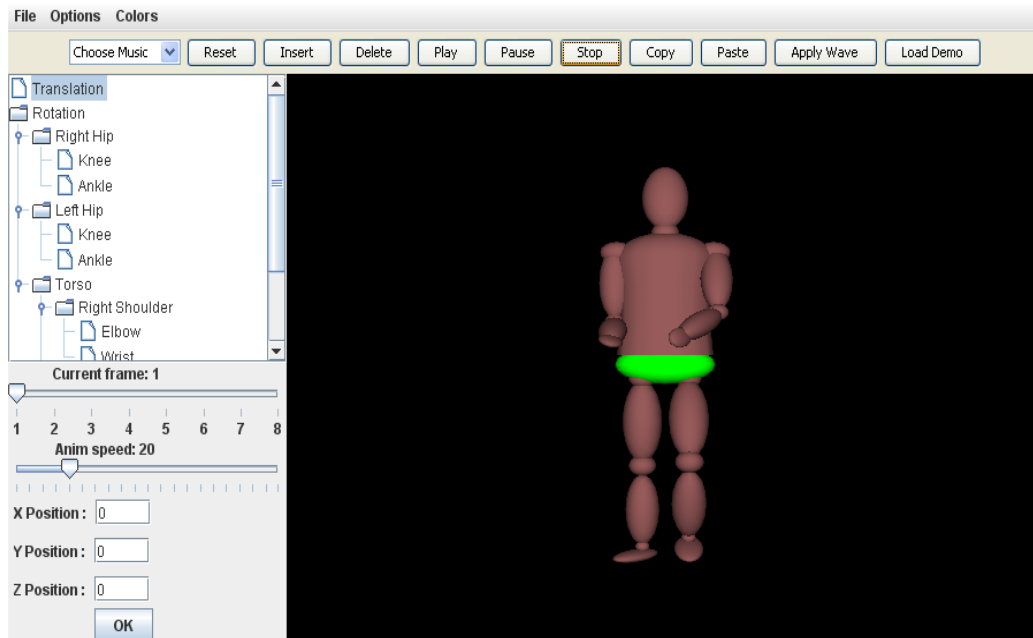
Figure 1: Break Dancer Tool

Studies focused on CSCW after researchers from various academic disciplines realized that computers should be designed according to the user's needs and that various technological designs and efforts can greatly benefit from the input of others in the areas of cognitive science and humanities. The research has led to a new theory of user-centered design. Emerging from previous research, this project focus on: (a) usability of collaboration tools and their effects on novice computer users, or HCI of system interfaces; (b) integration of CSDT tools into a classroom setting; (c) usage of CSDTs compared to traditional teaching, and (d) effects of gaming incorporation into a classroom setting. In this research, a community of instructors and students will use and evaluate a CSCW/CSDT tool to determine whether the designed system tool will meet minimum online usability standards and integrate easily into a classroom setting.

Moreover, the study will incorporate new tools into an online community and test their usability in a classroom setting.

1.3 Goals, Approach, and Contribution of the Research

The project aims to provide a source for increasing teaching aids in schools. Specifically, it seeks to develop a gaming convention that will hold the student's attention in the field of mathematics and introduce computing. In addition, it aims at evaluating and validating a tool or framework that can encourage the sharing of best practices within a community of practice. The sharing of best practices can enhance members' career aspirations significantly. By tapping into CSCW, K-12 teachers can benefit personally and enhance re-use and collaboration using technology. Students will benefit as well; mathematics scare most students and keep them away from computing and IT jobs because the careers require coursework in mathematics and in depth problem solving.

By collaborating in their use of CSCW tools, K-12 teachers can be encouraged to share and re-use best practices as a community to emulate the business industry, which has highly benefited from sharing best practices through collaboration. For example, the software development industry successfully utilizes code-re-use during software development through collaboration.

United States National Computing Data

Comparison of CIS Degrees Earned Nationally and Projected Average Annual Computing Jobs

Projected job openings that could be filled by computing degrees: 61%

- Associate's
- Bachelor's
- Master's
- Doctoral
- Projected Annual Openings

0   80,000   160,000

Education Indicators

AP Computer Science Test-takers
2009: **21,727**

Degrees Earned in Computer and Information Sciences
2007-2008 Academic Year
Associate's: **29,808**
Bachelor's: **39,701**
Master's: **17,312**
Doctoral: **1,340**

Projected Workforce Indicators (2008-2018)

Average Annual Computing Job Openings: **144,500**
10-year Percent Change in Computing Jobs: **22.19%**
10-year Change in Number of Computing Jobs: **745,700**

Figure 2: U.S. Data Comparison on CIS Degrees Correlation with Job Outcomes

1.4 Organization of the Dissertation

The rest of the dissertation is discussed in the following into eight chapters. Chapter 2 is a literature review that contains a brief history of CSDTs and computer supported collaborative environments. Chapter 3 discusses computer science (CS) principles and their relation to this project. Chapter 4 addresses the research issues; defines the research problems, and outlines approaches that attempt to solve the problems. Chapter 5 gives an outline of the system framework. Chapter 6 focuses on the framework for CSDTs. Chapter 7 gives an outline of the development tools that are used and will be used for future experiments. Chapter 8 describes the

methods and presents the human studies empirical results. Finally, Chapter 9 offers conclusions

and recommendations for future work.

2 LITERATURE REVIEW

2.1 Using CSDTs as Teaching Aides

Games reflect a cultural value that provide social contexts for learning. In other words, games are one place where the values of a society are embodied and passed on. Although games clearly do reflect cultural values and ideologies, they do not merely play a passive role. Games also help to instill or fortify a culture's values system. Seeing games as social contexts for cultural learning acknowledges how games replicate, reproduce, and sometimes transform cultural beliefs and principles. This way of looking at games forms the basis of a schema known as "Games as Cultural Rhetoric" [10].

Games contribute to the development of knowledge by having a positive effect on the atmosphere in a class that produces a better mental attitude towards math among students. Educational games provide a unique opportunity for integrating the cognitive, affective, and social aspects of learning [7]. Each CSDT topic comprises a number of resources that enable teachers to integrate topics into standards-based math instruction. Resources for each topic include:

- A section on cultural background and history

- A tutorial on the math topic and its connection to cultural artifacts and systems of knowledge

- Software (applets) that enable teachers and students to simulate the development of these artifacts

- Links to extensive teaching materials -- including lesson plans, pre- and post-tests, and samples of student work from a wide variety instructional settings.

The following sections describe some of the best examples from the design tools, highlighting the ways in which the tools emerged using computing, cultural rhetoric, and math education as their teaching foundation.

African and African American Design Tools

The tools offered with these were African Fractals, Mangbetu Design, Hexastrip Weaving, and Cornrow Curves (Figure 3). They combined African and African American tools that went hand in hand. Cornrow Curves, in particular, provide a prime example. Very few students knew that cornrows originated in Africa, so on the design tool website there is a historical background page. It includes images and information covering the indigenous styles and their meaning in Africa. There is also a series of "goal images:" photos of styles for students to simulate, including both professional styles and photos the students took of themselves and their friends. This helps students to make the connection from contemporary vernacular identity to heritage identity, which then opened up a wide selection of indigenous African fractal patterns for additional simulation.

Transformational geometry is the focus of cornrow curves. Each plait (braid) is given some particular rotation, translation (spacing), scaling, and reflection because these transformations are part of the standard curriculum.
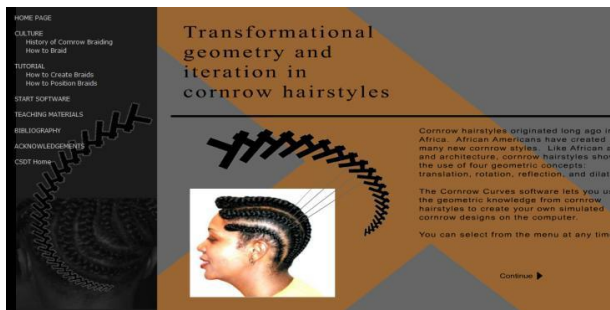
Figure 3: Cornrow Curves (Transformational Geometry)

9

Youth Subculture

There are two Youth Subculture design tools: Graffiti Grapher and Break Dancer. The Break Dancer tool is not completely finished yet. It is still going through the actual design phase. Graffiti Grapher (Figure 4) is a tool that reaches out to middle school and junior high school students. It gives them a chance to be artistic and express themselves. In turn, they are learning about Cartesian and Polar coordinates. This tool also raises the concern about students involvement in "inappropriate" activity: putting graffiti on actual building while enhancing their knowledge of subculture practices. We will look at these issues as we examine our game design in detail.

Figure 4: Graffiti Grapher (Cartesian and Polar Coordinates)

Native American Design Tools

The current Native American design tools include "SimShoBan" (Eglash 2001), Pacific Northwest Basket Weaver, Yupik Star Navigator, Yupik Parka Patterns, Navajo Rug Weaver, and the Virtual Bead Loom (VBL, Figure 5). The web-based software allows the user to enter *x, y* coordinates for bead positions; together with color choice, this allows the creation of patterns similar to those on the traditional loom. There is also a "cultural background" section showing how the concept of a Cartesian layout can be seen in a wide variety of native designs: Navajo

sand paintings, Yuptik parka decoration, Pawnee drum design, and other manifestations of the Four Winds" concept [7].
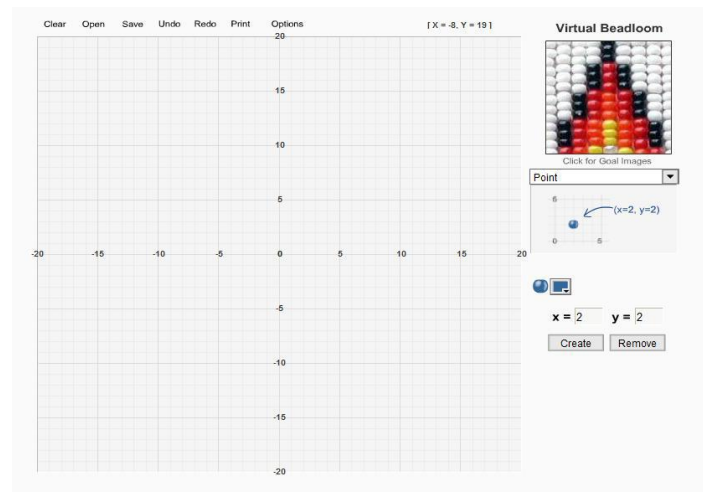
Figure 5: Virtual Bead Loom (Cartesian Coordinates)

Latino Design Tools

There are currently two Latino design tools: Pre-Columbian Pyramids, in which students create three-dimensional simulations of architecture from the ancient cultures of Central America and Rhythm Wheels (RW).

The RW software (Figure 6) makes use of the ratios between beats in percussion. There are six drumbeats for every eight-clave beats. The two instruments go out of phase, but come back into phase after 48 beats. It is this impression of separating and reuniting the rhythms that gives the music its "hook." The existence of a Least Common Multiple (LCM) is an important part of any drummers' understanding. Because ratios and LCM is part of the standard math curriculum, this gave students an opportunity to link ethnomathematics to the classroom.

The RW software allows a student to choose from a variety of percussion sounds and drag each sound into a position on a rotating wheel. The students select the number of beats per wheel (up to 16), the number of simultaneous wheels total (up to three), the number of times

each wheel loops, and the speed of the wheels (all wheels must rotate at the same speed). They can also separately vary the volume of each sound for accents. There are wide varieties of rhythms that can be reproduced.
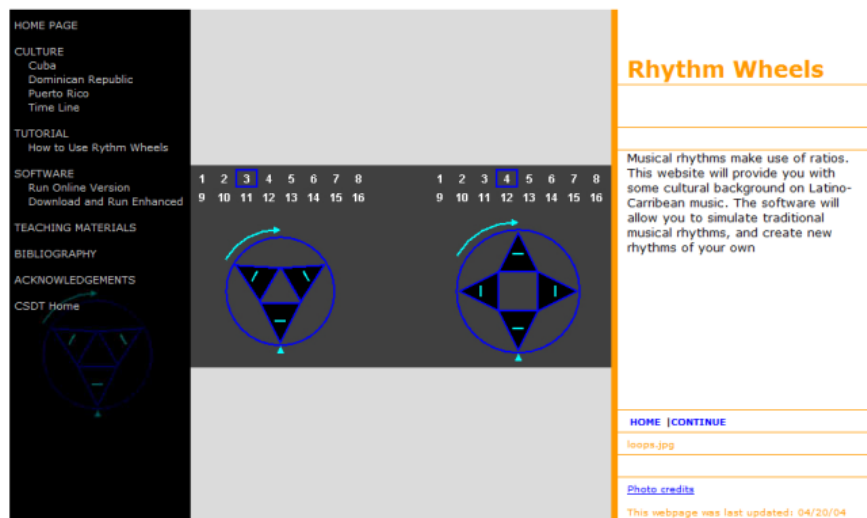
Figure 6: Rhythm Wheels (Fractions and Least Common Multiple [LCM])

2.2 What is the approach of CSCW?

By virtue of the first part of its name, the 'CS' part, the professed objective of CSCW is to support via computers a specific category of work -- cooperative work. Therefore, the term computer support conveys a commitment to focus on the actual needs and requirements of people engaged in cooperative work. Of course, new technologies of communication and interaction necessarily transform the way people cooperate and CSCW systems are likely to have tremendous impact on existing cooperative work practices. Cooperative work can be conceived as a specific category or aspect of human work with certain fundamental characteristics common to all cooperative work arrangements, irrespective of the technical facilities available now or in the future.

By virtue of its commitment to support cooperative work, CSCW should not be defined in terms of the techniques being applied. Computer supported collaborative work is a research

area aimed at the design of application systems for cooperative work, in all its forms. Like any other application area, CSCW, in its search for applicable techniques, potentially draws upon the whole field of computer science and information technology. Accordingly, a technology-driven approach to CSCW will inevitably dilute the field. To some extent, the current lack of unity of the CSCW field bears witness to that.

Computer supported collaborative work, in a sense, is an endeavor to understand the nature and requirements of cooperative work with the objective of designing computer-based technologies for cooperative work arrangement*s*. Because multiple individuals, situated in different work settings and situations, with different responsibilities, perspectives and propensities, interact and are mutually dependent in the conduct of their work has important implications for the design of computer systems intended to support them.

The objective of social science contributions to CSCW should not be to cash in on the new wave and do what they have always done, but rather to explore exactly how insights springing from studies of cooperative work relations might be applied and exploited in the design of useful CSCW systems. This demand raises the issue of how to use insights already achieved in related fields to influence the design process. Moreover, it raises more fundamental issues such as: Which are the pertinent questions being pursued in field studies and evaluations for the findings to be of use to designers? And how are the findings to be conceptualized? If CSCW is to be taken seriously, the basic approach of CSCW research should not be descriptive but constructive.

As a research area devoted to exploring and meeting the support requirements of real world cooperative work arrangements, CSCW requires that technologists extend from a strict technical focus and investigate how their artifacts are and potentially used in actual settings. In

short, the drive of CSCW should be directed towards designing systems embodying an ever-deepening understanding of the nature of cooperative work forms and practices.

While this conceptualization of the general approach of CSCW facilitates an understanding the nature of cooperative work that betters supports people in their collaborations, it does not prescribe a particular research strategy. Of course, field studies of cooperative work in diverse domains with the objective of identifying the research requirements of various kinds and aspects of cooperative work is much needed, but the design and application of experimental CSCW systems may also yield deep and valid insights into the nature and requirements of cooperative work. We thus concur with the drive of Groupware enthusiasts to construct working artifacts to support cooperative work processes [36].

2.3 Computer Supportive Collaboration Work to Mold Communities

The phrase *computer supported cooperative work* was first coined by Greif and Cashman in 1984 at a workshop attended by individuals interested in using technology to support people in their work[43]. In 1987, Findley presented the concept of collaborative learning-work. Many authors consider CSCW and groupware to be the same, but they are different. Groupware refers to computer-based systems; CSCW is the study of tools and techniques of groupware as well as their psychological, social, and organizational effects. Wilson (1991) expresses the difference between these two concepts: "CSCW is a generic term, which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques " [54].

Computer supported collaborative work is a design-oriented academic field bringing together social psychologists, sociologists, and computer scientists, among others. Despite the variety of disciplines, CSCW is an identifiable research field focused on understanding

characteristics of interdependent group work with the objective of designing adequate computer-based technology to support such cooperative work.

Over the years, researchers have identified three CSCW core dimensions of cooperative work:

- **Awareness:** Refers to the need for individuals working together to gain some level of shared knowledge about each other's activities [36].

- **Articulation work**: Refers to ability of cooperating individuals to partition work into units, divide it amongst themselves, and after the work is performed, reintegrate it [37, 38].

- **Appropriation (or tailorability**): refers how an individual or group adapts a technology to their own particular situation; the technology may be appropriated in a manner completely unintended by the designers [39].

These concepts are derived through the analysis of systems designed by researchers in the CSCW community or through studies of existing systems (e.g. Wikipedia). Computer supported collaborative work designs and builds systems to address core concepts in novel ways. However, the complexity of the domain makes it difficult to produce conclusive results; the success of CSCW systems are often so contingent on the peculiarities of the social context that it is hard to generalize. Consequently, CSCW systems based on the design of successful models may fail to be appropriated in other seemingly similar contexts for a variety of reasons that are nearly impossible to identify a priori [40]. Computer supported collaborative work researcher Mark Ackerman calls this "divide between what we know, what we must support socially, and what we can support technically" the *socio-technical gap* and describes CSCW's main research agenda to be "exploring, understanding, and hopefully ameliorating" this gap [41].

15

In order to implement CSCW effectively, Ackman's social technical divide must be addressed. The gap of what technology can support from a social context introduces the challenge of "how we can replicate the social events virtually." Bridging this gap will ensure that CSCW tools are effective in satisfying the need they are designed to mitigate or solve. To address the social-technology divide Kaufmann uses a Time/Space matrix and divides CSCWs into groups; same time-same place, different time-same place, and same time-different space, and different time-different space. The matrix is a replica of real life social situations that CSCWs designers will have to address when creating or refining CSCWs. The Time/Space Groupware Matrix shown below, courtesy of Morgan Kaufmann publishers, outlines the different ways people collaborate [41].
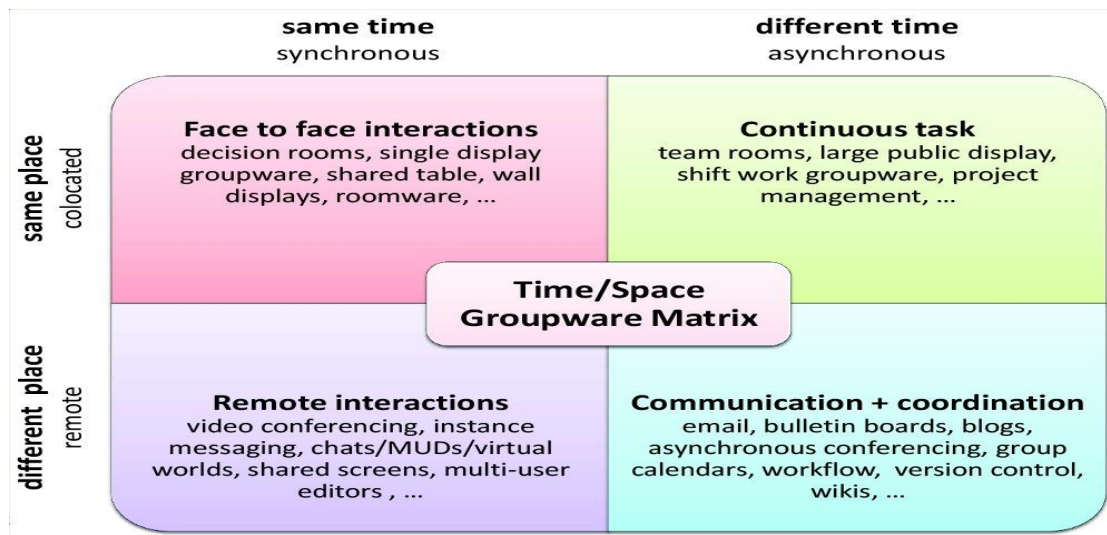
Figure 7: Computer Supported Collaborative Work Matrix. (Source: Johansen, R. 1988 "Groupware: Computer Support for Business Teams" The Free Press.)

Both time and space facets are bipolar (i.e. same time or different time and same place different or place). Thus the time space groupware matrix has online communities divided into four categories:

Same Time, Same Place -- Synchronous Co-located: Characterized by such things as face to face interactions in decision rooms, single displays, groupware, shared table, wall displays, and room ware [41].

Same Place, Different Time -- Asynchronous: A major collaboration between a group working on continuous tasks through tea rooms, large public displays, shift work groupware, project management and the like [41].

Different Place, Same Time -- Asynchronous Remote: Remote interactions accomplished through such means as video conferencing, instant messaging, charts/MUDs/virtual worlds, shared screens, and multi-user editors [41].

Different Place, Same Time -- Asynchronous-Remote: Involves communication and coordination with tools such as e-mail, bulletin boards, blogs, asynchronous conferencing, group calendars, workflow, version control, and wikis [41].

The CSCW paradigm provides a framework of what we know we can support socially but social-technical mapping still remains the main problem. Many researchers are looking for ways to bridge the disparity between social needs and the capability to support the need technically from a computer science perspective [45].Communities can be formed to support almost every daily activity. This creates a need for CSCW in a multitude of areas that need support through computer collaborative work. In real life today, there community supported computer-based collaborative services being used by major commercial, social, and academic activities around the world. IBM uses the use the term *social computing* to describe the field of computer collaborative work. This is an attempt to infuse the concept with social conventions, in opposition to the technological characteristics, that are associated with computer systems and software.

The outlined tools have been successfully implemented and accepted by many users as a way of social life. Many educational and commercial institutions are in the forefront of advancing their services using CSCW tools. Some of the major services offered to clients include online degrees and online banking services by most major banks in the commercial service sector. Various social forums have been implemented to serve communities. The forums are an intended meeting "spot" for individuals to gather and socialize. In the academic world, systems are used as pedagogical agents to enhance teaching and sharing knowledge (e.g. blackboard, WebCT, and Moodle) [42]. The public has many forums to support social interaction, including Facebook, Yahoo Chat, and Twitter. Our goal to leverage Facebook to provide a system where members of a community of practice will spend time together and exchange knowledge with peers in a user friendly and secure manner through social networking.

2.4 Communities of Practice

Schools and districts are organizations in their own right, and they too face increasing knowledge challenges. The first applications of communities for practice have been in teacher training and in providing isolated administrators access to colleagues. There is a wave of interest in these peer-to-peer professional-development activities. But in the education sector, learning is not only a means to an end: it is the end product. In schools, changing learning theory is a much deeper transformation. This will inevitably take longer. The perspective of communities of practice affects educational practices along three dimensions. Our system plans to reorganize educational experiences' three dimensions and evaluate the reorganization's effects through surveys:

- Internally: We will organize educational experiences that ground school learning in practice through participation in communities around subject matters within the system and receive feedback from members.

- Externally: We will focus on how to connect the experience of students to actual practice through peripheral forms of participation in broader communities beyond the walls of the school, through blogs, team pages, and forums within the system.

- Over the lifetime of students: Our system will also serve the lifelong learning needs of students by organizing communities of practice focused on topics of continuing interest beyond the initial schooling period in the realms of agriculture and basic science, responsibility, and good citizenship.

2.5 What Can Educational Games Teach

A summit on educational gaming suggested that many game features, combined and designed effectively into educational gaming, could teach many things in an engaging and motivating manner. Games expand cognitive abilities and serve as a platform for developing new or practicing existing skills in the context of real world goals, rules, and situations. Games could teach old subjects in new ways. For example, in the civilization-building games, players may explore subjects such as math, how computers work, and geography within the rules and structure of the game [5].

There are skills that help with the use of games and simulations: higher order skills, practical skills training, responses high performance situations, rarely used skills, expertise, and team building skills.

Higher Order Skills

When individuals play many commercial video and computer games, they must employ a wide range of higher-order skills. This suggests that games may be effective in teaching these skills. For example, in various games players must: [5]

- Think strategically

- Master resource management

- Interact with systems and understand the interaction of variables

- Multi-task, manage complexity, respond to rapidly changing scenarios, and make decisions;

- Learn compromise and trade-off in satisfying the needs of diverse constituencies

- Manage complex relationships

- Exercise leadership, team building, negotiation, and collaboration.

Practical Skills

Through games and simulations, learners can exercise practical skills such as operating sophisticated aircraft, building a bridge, performing surgery, or controlling scientific equipment. This allows learners to move up the learning curve, without risking life, limb, or damage to expensive equipment in the early part of training and practice. If learners fail in the tasks they are learning, little harm is done, and they can try repeatedly to gain mastery of the required knowledge and skills.

High Performance Situations

Games and simulations show promise in training individuals for high-performance situations that require complex and multi-component decision-making.

Rarely Used Skills

Simulations are particularly important for reinforcing skills that are seldom used. For example, it can allow managers to practice their responses to a terrorist attack, school shooting, and natural disaster scenarios.

Developing Expertise

Games offer a way to "walk in the shoes" of experts and learn how experts approach problems. Knowledge is compiled over time and organized in the minds of experts, creating mental models, or templates that they apply to different situations in their work.

Team Building

Some elements of multiplayer games foster information sharing, goal-directed cooperation, and the spontaneous formation of networks -- all of critical importance in business today. Games and simulations hold promise for training team members to work effectively as a team, especially in decision-making, exercising judgment, and solving problems under pressure.

2.6 Computer Games as a Part of Children's Culture

The cultural and social significance of electronic games is pedagogically relevant because any educational or teaching effort that aims at mediating so-called "media competency," computer literacy, or ICT skills is preceded by informal and non-formal learning processes of children within their "computer gaming culture." A better knowledge about informal learning processes and their background is necessary to avoid a "clash of media cultures." This metaphoric notion implies the following: teachers, parents, and others engaged in education and tuition are members of a generation, which during its primary socialization, has grown up in a different media culture and has different media experiences than the young generation of today. These experiences do not only influence their private values and attitudes towards new media, but they also have an impact on their educational concepts and actions [24].

In most cases, parents or other adults do not participate in children's gaming cultures in an active or interactive way. Some may regard this as something that should be accepted or even supported, because children want and need to have their own spheres. However, in the view of this paper, the pedagogical task remains to accompany actively and critically the children's process of growing up and developing their relationship to the cultural world. Moreover, the task remains to secure a plurality of resources and challenges they can use to develop their cognitive, social, and physical abilities [24].

2.7 What Makes a Game Good

According to Kramer, the perceived value of a game depends greatly on the individual preferences of those who play it. When constructing CSDTs, designers should consider some of these characteristics. For instance, some players prefer games of luck, others prefer games of tactics, and still others enjoy communicating with fellow players. Then there are those who like games based on reaction, manual skills, or memory. The following are some items Kramer felt important in the designing of a "Good Game" [8]:

- Originality

- Freshness and replayability

- Surprise

- Equal Opportunity

- Winning Chances

- No "Kingmaker" Effect

- No early Elimination

- Reasonable Awaiting Times

- Creative Control

- Uniformity

- Quality of Components

- Target groups and Consistency of Rules

- Tension

- Learning and Mastering a Game

- Complexity and Influence

2.8 Issues Designing Cultural Games

Most teachers think that designing cultural games is a waste of time that takes away from the actual verbal teaching time. In 2006, the Federation of American Scientists hosted a *Summit on Educational Games: Harnessing the Power of Video Games for Learning* in Washington D.C. The final *Summit* report addressed the reluctance involved in developing videogames for educational use. The reasons for such reluctance include [23]:

- High development costs in an uncertain market makes investment in educational gaming innovations too risky for the commercial video game producers and even the educational materials industries.

- Change in schools comes slowly in terms of adopting any innovations and incorporating the necessary organizational and instructional changes that allow use of new learning technologies.

- There is reluctance on the part of schools to give up textbooks in order to purchase educational gaming products.

- The specific educational values tied to state standards have not been proven through in-depth research, which is a requirement of the No Child Left Behind Act.

- Some parents and teachers have very negative attitudes about the use of videogames in the classroom.

- Games are especially good at teaching higher order skills, which are not typically assessed in standards examinations.

- Access to computers in many schools is so low that they cannot play a mainstream role in student learning.

Besides these reluctances, other barriers also raise second thoughts on designing educational games. First, students have limited access to computers at their schools. If a student is going to use an educational game, it does require computer access. Second, the majority of games are not designed to be played in a short or specific time. Most games cannot be played in the typical 50-minute class period of most middle and high schools. Therefore, the effective use of educational games requires changes in pedagogy, content, and a re-thinking of the role of teachers. Game design must eliminate the many hours it currently takes to learn to play a game [23].

Finally, some teachers and parents have negative attitudes about videogames and are often unfamiliar with the games. This brings up the issue that they will not willingly motivate their children to do the work and participate accordingly. If they can get the teachers and parents motivated then this will open a door to get the students' attention.

2.9 Why Do CSDTs Work?

Studies of CSDTs are limited and only have some suggestive results with no wide-scale testing. However, assuming that using CSDTs in the classroom may raise minority student math achievement and improve their technological aspirations, it may be simply because they use a flexible computational medium, which allows students to pursue inquiry and discovery learning

in a context in which the cultural component is irrelevant. However, this study believes that the flexibility and discovery learning aspects are critical to not only to CSDT success, but are just as integral to cultural identity [11].

Eglash (2006) identifies and explains different conflicts between cultural identity and mathematics. He begins with the "acting white" phenomenon. It is difficult to accuse someone of "acting white" if they are using materials based on black culture -- difficult but not impossible. Such pedagogies always risk the accusation of being patronizing or illegitimately appropriating minority culture, and in some cases, those accusations are correct [4].

Next, there is identity conflict. Ethnomath examples can decrease the perceived cultural distance between math and cultural identity whether such an identify is an experienced home culture, an imagined heritage culture, or some hybrid. The distance can be diminished from either end -- that is, students might change their perception of the minority culture as more mathematical, or change their perception of mathematics as being more cultural [4].

Social irrelevance is another conflict. Ethnomathematics is particularly effective in the context of class discussions of colonialism, primitivism, racism, and other histories of stereotypes. Its relevance is thus in its ability to provide alternative portraits. There are also practical applications of ethnomathematics to design. Here the challenges are commensurate with any such discussion. The myths of genetic determinism also raise conflict. Ethnomathematics offers strong counterevidence to primitivist and ethnocentric portraits of "simple" cultures. In addition to these conflicts between cultural identity and mathematics education, we must be aware of the role of perceived poor mathematics performance within certain minority groups [4].

In summary, these four cultural features -- the "acting white" accusation, identity conflict, social irrelevance, and myths of genetic determinism -- create cultural barriers to high

academic performance by minority students in subjects associated with science and technology careers. We suspect that CSDTs can ameliorate these barriers. Further studies will be needed to determine which barriers are most affect and to what degree [4].

2.10 Human Computer Interaction and Culturally Situated Design Tools

Understanding how users experience technologies raises concerns about social and cultural meaning. What does the product mean to the user? What does it mean in the context of particular cultures? And, what does it mean in terms of its broad impacts on the social and global environment [8]? Culturally situated design tools can address these concerns.

This inquisitive thinking has led the HCI community to turn its attention from the workplace and productivity tools towards domestic design environments. While usability is still central to the field, HCI is beginning to address other considerations such as pleasure, fun, emotional effect, aesthetics, the experience of use, and the social and cultural impact of new technologies [8].

The HCI community is concerned with the user experience and the ways in which technology can take on social meaning. It draws on non-engineering disciplines such as ethnography and design to understand better experience and aesthetics in technology design. Their study will explore the extent to which these disciplines can help develop innovative approaches to design through deeper understandings of the social and cultural meanings of domestic technologies [8]. In turn, the studies will contribute to the foundation of CSDTs.

2.11 Users and User Experiences with CSCW Systems

In the spring of 2010, researchers associated with Social Networking Teaching Tools: A Computer Supported Collaborative Interactive Learning Social Networking Environment for K-12 conducted a preliminary study that surveyed 33 teachers in North Carolina city schools with

different backgrounds and levels of education. The researchers used a forum based prototype system. The surveyed group completed the usability survey to express their experiences of the system. The results were encouraging since 70% of those surveyed felt that a forum type virtual tool would be good for K-12 education and expressed confidence in using the proposed tool to teach if it were available [25] .

To confirm and validate the preliminary results, this study extends the previous study and focuses on creating a secure and user friendly environment for a community of practice to share best practices. The proposed system will ensure the safety and privacy of the community members while online. For the success of the system, the stakeholder's opinion will weigh heavily on the adoption and usability of the system. As stakeholders evaluate the system, they will give their opinions and suggestions to improve chances for the future adoption and improved usability of the system. To verify the usability, the study will conduct a survey based on Norman's seven usability principals as it relates to the enhanced system, i.e. visibility, simplicity, designing for error, etc. The results will be published as part of the contribution of this dissertation.

2.12 Sociability and Usability in Online Communities

Various techniques exist for evaluating sociability and usability, but little focus has been directed towards identifying and measuring the determinants of success for online communities. This article begins to fill this gap by identifying key determinants of sociability and usability for online communities [34]. Measures of sociability include numbers of participants, amount of reciprocity, trustworthiness, and other factors. Measures of usability include numbers of errors, productivity, user satisfaction, and other factors.

The dissertation describes how usability is the nature of human-computer interaction, whereas sociability describes the nature of social interaction in an online community. A community's focus, the people who belong to it, and the policies that guide social interaction are key components of sociability. Dialogue and social interaction support, information design, navigation support, and accessibility are key components for good software usability. Usability is concerned with how intuitive and easy it is for individuals to learn to use and interact with a product. Various definitions of *usability* have been proposed. For example, "usability means that people who use the product can do so quickly and easily to accomplish their own tasks" [34]. While sociability closely relates to usability and could be considered a new genre of usability, it also has significant differences. In theory, usability is primarily concerned with how users interact with technology and sociability is concerned with how members of a community interact with each other via the supporting technology. The focus of usability is therefore interaction across the human computer interface. The focus of sociability is human interaction supported by technology [34].

The main usability issues for online communities are similar to those for most other web-based software, but the following four components are particularly important because they are concerned with the software's role as a medium and a place for social interaction.

- *Dialogue and social interaction support.* Includes such aspects as the prompts and feedback that support interaction, the ease with which commands can be executed, the ease with which avatars can be moved, spatial relationships in the environment, etc.

- *Information design.* Relates to how easy to read, how understandable, and how aesthetically pleasing the information associated with the community is.

- *Navigation.* Refers to the ease with which users can move around and what they want in the community and associated websites. Many online community users have suffered from the inconsistencies of data transfer and differences in interaction style between imported software modules and the website housing the community.

- *Access.* Pertains to the requirements to download and run online community software. Thes requirements must be clear. In addition, if high bandwidth and state of the art technology is needed to run the community, there should be low bandwidth text only versions and clear instructions about how to obtain it [34].

3 COMPUTER SCIENCE PRINCIPLES PROJECT

3.1 Computer Science Principles Project Overview

The Computer Science Principles project is part of a national effort to reach a wide and diverse audience of students and share with them the deep, rich intellectual, and practical contributions of computing. Members of the CS principles community are building a new course designed to be an introductory college level course for everyone a course rich in computer science content and exciting in its pedagogy. The course provides an academic foundation for understanding these contributions. It emphasizes the intellectual, practical, and creative aspects of the field of computer science.

A set of Computational Thinking Practices and Big Ideas specifies the CS Principles curriculum by identifying the content, practices, thinking, and skills central to the discipline of computing and computer science. This current research project has developed a course focused on the concepts of the Big Ideas and how they relate to cultural aspects within computer science.

3.2 Big Ideas of Computer Science

I. Computing is a creative activity. (CREATIVITY)

    Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact. At the same time, computing and computer science facilitate exploration and the creation of knowledge. This course will emphasize these creative aspects of computing. Students in this course will create interesting and relevant artifacts with the tools and techniques of computing and computer science.

II. Abstraction reduces information and detail to facilitate focus on relevant concepts. (ABSTRACTION)

Everyone uses abstraction on a daily basis to cope effectively with our world. In computer science, abstraction is a central problem solving technique. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems. This course will include examples of abstractions used in modeling the world, managing complexity, and communicating with people as well as with machines. Students in this course will learn to work with multiple levels of abstraction while engaging with computational problems and systems.

III.   Data and information facilitate the creation of knowledge. (DATA)

Computing enables and empowers new methods of information processing that have led to monumental change across disciplines, from art to business to science. Managing and interpreting an overwhelming amount of raw data is part of the foundation of our information society and economy. People use computers and computation to translate, process, and visualize raw data, thereby creating information. Computation and computer science facilitate and enable a new understanding of data and information that contributes knowledge to the world. Students in this course will work with data using a variety of tools and techniques to understand better the many ways in which data is transformed into information and knowledge.

IV.   Algorithms are used to develop and express solutions to computational problems. (ALGORITHMS)

Algorithms are fundamental to even the most basic everyday tasks. Algorithms realized in software have affected the world in profound and lasting ways. The development, use, and analysis of algorithms are some of the most fundamental aspects of computing. Students in this course will work with algorithms in many ways: they will develop and

express original algorithms, they will implement algorithms in some language, and they will analyze algorithms both analytically and empirically.

V.   Programming enables problem solving, human expression, and creation of knowledge. (PROGRAMMING)

Programming and the creation of software have changed our lives. Programming results in the creation of software, but it facilitates the creation of more general computational artifacts including music, images, visualizations, and more. In this course, programming will enable exploration as well as being the object of study. This course will introduce students to the concepts and techniques used in writing programs and to the ways in which people develop and use programs. The focus of the course is not programming per se, but on all aspects of computation. Students in this course will create programs, translating human intention into computational artifacts.

VI.   The Internet pervades modern computing. (INTERNET)

The Internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration. The principles of systems and networks that helped enable the Internet are also critical in the implementation of computational solutions. Students in this course will gain insight into how the Internet operates, will study characteristics of the Internet and systems built upon it, and will analyze important concerns such as cybersecurity.

VII.   Computing has global impacts. (IMPACT)

Computation has changed the way people think, work, live, and play. Our methods for communicating, collaborating, problem solving, and doing business have changed and are changing due to innovations enabled by computing. Advances in computing have

fostered many innovations in other fields. Computational approaches lead to new understandings, new discoveries, and new disciplines. Students in this course will become familiar with many ways in which computing enables innovation, and will analyze the potential benefits and harmful effects of computing in a number of contexts.

These Seven Big Ideas discussed above are designed to specify the set of student learning objectives for the course. Refinements to the curriculum framework and learning objectives are ongoing, aided by data from a series of pilot offerings.

4 STATEMENT OF THE PROBLEM

4.1 Research Problem

Culturally Situated Design Tools allow students and teachers to explore mathematics and computer science with depth and care, using cultural artifacts from specific times, places, and cultures. The tool that we explored in our preliminary investigation was the Break Dancer tool. The design of this tool is to teach the sine and sine function. In the current version of the tool it presents a somewhat complicated introduction to the concept and tutorial. So, our initial objective is to make a few changes to the current version. This will allow it to be more user friendly and give teachers a better way to get the information across to students. We were concerned whether the current state was sufficient to instruct complex mathematical concepts and computing and wondered whether we should change the interface and try designing it in Unity 3D or other environments.

We would like to have a Scratch-like "drag and drop" interface. Drag and drop is the action of clicking on a virtual object and dragging it to a different location or onto another virtual object. In general, it can be used to invoke many kinds of actions, or create various types of associations between two abstract objects. As a feature, support for drag and drop is not found in all software, though it is sometimes a fast and easy-to-learn technique for users to perform tasks. However, the lack of affordances in drag-and-drop implementations means that it is not always obvious that an item can be dragged. In this tool, drag and drop refers to codelets, which are dragged into a scripting area. We based the GUI on MIT's Scratch "http://scratch.mit.edu." This is why the new version of the tools are called pCSDTs -- for *programmable* CSDTs. Most of the tools that Eglash's team are using with CSDTs have this feature and he would like to integrate this into this tool. This would offer more user friendliness because users would not have to

remember textual syntax. If the users are not an expert programmer, having a drag enabled script allows them to interact more with the tool and easily implement what they want to do. In addition, it allows the user to have a more iconic syntax that is useful in visual programming environments.

The next thing considered is that the tool, as a whole, needs more clarity and stepwise introduction on what it is trying to express. The concept has a wonderful application, but we have found that there needs to be a clearer and direct mapping to the desired educational content. For example, it is not clear how to use it to teach 3D coordinates. It is also not clear how to use the rotations to teach angles, nor is it clear on how to use the sine wave option to teach phase and frequency. Eglash suggested that it would be valuable for our team to examine that from an HCI perspective would be creating improved lessons, lesson plans and evaluations that are very important in teaching the students the concepts. In addition, suggestions of applications that more clearly conveys the intended concepts.

Finally, with this tool, there was a need for a better tutorial. Tutorials are sometimes where critical learning takes place. Eglash also suggested that we improve the tutorial, and perhaps introduce some of the math concepts using games or even better diagrams.

This chapter introduces the study's research problem, hypotheses, and research questions. It also introduces the characteristics of the empirical/experimental research that are general to all studies. These characteristics will be dealt with in detail in subsequent chapters. In this project, study questions and hypotheses will be investigated and further benefits of using an online tool for education using CSDT. The analytic and empirical methods used for requirements analysis appear in later chapters.

In the current educational systems, K-12 teachers as a community of practice have no time, desire, or motivation to learn new tools, let alone develop and refine tools that serve their community. Thus, there is need to develop a tool to serve their needs as a community and help them share and re-use best practices. An easy way to provide a secure and trustworthy community of practice tool in virtual space is by adopting an existing collaboration tool/system, tailor it, and evaluate it with users. Many of these applications have very little evaluation of the user experience with novices. In this study, we will focus on communities of practice comprised of K-12 teachers and students, who in some instances will be novice computer users. This research project focuses on K-12 education teachers as a community of practice to understand the need for a virtual space tool to share best practices, build a new a tool that addresses those needs, and evaluate the effectiveness of the tool in comparison with the existing tools or traditional methods.

There are some empirical studies of supporting K-12 teachers collaborating and sharing best practices in virtual space [49] [34]. However, many studies have focused in higher education support content managers like Moodle, Blackboard and WebCT that support both teachers and students in the learning process [47][48]. Although there are a number of tools that support collaboration, most do not encourage the sharing of best practices among K-12 teachers. This study will primarily focus on providing a secure tool for K-12 teachers to share best practices with a capability to keep private information within the community, thereby improving cyber trust among the community of practice. This project will leverage an environment that supports educational learning in partnership with traditional learning. We want the application to be easy to use, user friendly, and robust enough for students, teachers, and groups that may report as computer illiterate but are able to use the environment intiutively with little to no training.

We will conduct surveys and query the stakeholders on the usability and learnability of this type of application. The study will survey the potential stakeholders -- K-12 teachers, students, and colleagues -- to see if they will accept this concept as a noble idea of collaboration and sharing the community's best practices in virtual space. After the system is fully developed with features and user interfaces, what will be the users' response to its usability? How will the usability of this computer based collaborative environment affect the user's motivation in using the tool for future educational purposes? The answer to these questions, we will provide the framework for implementing a successful environment for supporting a community of practice.

4.2 Research Approach

The three main goals of the study are to (a) enhance technical skills of novice users, (b) encourage users to adopt the use the technology for collaboration instead of traditional methods, and (c) introduce new technical skills to novice computer users.

The study's main target is to encourage the learning of a new environment for collaboration within groups of communities of practice. The study will also produce a tool that can be used with Android, Internet, or Wii. The study will focus on the usability of the tool and by perform a comparative study of its usefulness. The data will be gathered from the K-12 teacher population in the initial stages of this study to find out if the project is suitable for the intended participants. We will also gather information from students ranging from middle school to high school. A broad array of questions to be answered by the experimental participants has been created to gather data for the research through a survey. The survey requires users to identify themselves as a novice or has having advanced computer skills for the purpose of assessing the impact of the tool on subjects.

37

To gather more data on the usability and effectiveness of the environment, experimental participants will perform a series of tasks to test the usability of the environment. At the end of series, users will be required to complete a detailed survey questionnaire to provide feedback on their experiences with the system.

4.3 Research Questions

In the first phase of the study, we plan to investigate the security and usability hurdles facing communities of practice in virtual space as well as explore the suitability of user interfaces of existing tools to the specific community of practice -- K-12 teachers meeting the requirements for enlistment. In the next chapter, the findings will lead to a comprehensive focus on the usability and problems for users to learn, interact easily, and trust the current tools appropriately. What key usability issues face communities of practice (e.g. K-12 teachers) when trying to share best practices in virtual space? What are the key security concerns for communities of practice (e.g. K-12 teachers) in virtual space?

4.4 Hypotheses

For K-12 teachers and students to collaborate successfully in virtual space, they must share information. The CSCW environment will provide a framework for community members to express their ideas realistically in real time and through various types of media. The CSCW environment has incorporated features that are paramount for sharing quality information with ease through the use of templates. To validate the CSCW feature's ease of use, we will conduct a series of survey studies with participatory design, scenario-based design, qualitative evaluation, and usability analysis.

With the use of example templates to create successful best practices, users receive with a simple task of sharing best practices by simply clicking icons on the screen. The environment

will offer standard templates that will be used for sharing information. The empirical study will focus on the effectiveness of user interface and the usability of the forms -- CSCW templates-in comparison with the traditional methods of sharing best practice among the selected research groups.

This research is based on the two research hypotheses listed above to test the hypotheses at the end of the study. The empirical study compares the traditional method and the virtual community of practice methods. It will examine the usability, trust, and analysis of the user interface by experts and other test group participants. The data collected during the experiment and through qualitative observations and surveys will be presented in consecutive tables and other statistical methods.

4.4.1 Hypotheses I

I. $H0_1$: There is no learning difference when using CSDTs compared to the CSCW environment.

II. $H0_A$: There is a learning difference when using CSDTs compared to the CSCW environment.

4.4.2 Hypotheses II

I. $H0_2$: There is no difference in misconceptions that learners have when using CSDTs compared to traditional environments.

II. $H0_A$: There is a difference in misconceptions that learners have when using CSDTs compared to traditional environments.

4.5 Experiment

Traditionally, members of communities of practice collaborated through traditional means: word of mouth, conferences, lectures, and hand written notes. With the emergence of

online environments, many groups have resorted to e-mails and other web based tools like dedicated websites and blogs to collaborate.

However, most communities of practice have members with a variety of different technological skills and varied access to technology based upon the location and focus of the community. Due to the variety of collaboration tools available, there is debate on which environment will be more suitable for these communities of practice to share best practices easily. Most existing educational collaboration environments were created to satisfy the need for content management and cater to structured learning situations. None of these tools have been tailored to cater to novice users or designed with a loose structure that allows novice users to express themselves comfortably despite their limited technological skills.

Although there are many ways of collaborating between members of communities of practice in virtual space, many of them have not fully ascribed to the available technological tools to share and re-use best practices because of the time required to acquire the skills. Therefore, encouraging the use of technology to share and re-use best practices among members of various communities of practice requires an environment that is easy to use, user friendly, easy to learn, and does not require any programming or advanced technological skills.

In this experiment, we designed new user interfaces for virtual space aimed at fostering educational communities of practice. We are considering three styles of login wireframes (i.e. design dashboards) to be rated and evaluated by user interface design experts and members of the targeted community of practice. The basic design will be the current interface: the version that we will be comparing for improvement will be the next design version of this system with added features.

Our methodology will be to gather demographics from user groups before they begin this experiment to find their level of computer efficacy, general educational background, and technophobia levels. The experimental portion of the work will begin with a list of tasks that provides design experts an opportunity to evaluate usability, usefulness, and aesthetics of design. At the conclusion of the experiment, the users will complete a post -- questionnaire.

4.5.1 Setup of Experiment

To perform the experimental tasks, participants must have access to the Internet through a web browser. The specifications of the machines will not be considered, but the latest browsers are recommended. The system can also be accessed through hand held devices, iPad, iPod, Blackberries, web accessible cell phones, and the Android. We will use hand held devices; smart phones, iPods, Androids, Blackberries are prevalent in most rural areas for Internet connectivity. We will take into account that there is limited fast Internet connection of most rural areas. Nevertheless, with available satellite communication, a majority of the areas have fast Internet accessible through hand-held devices.

The experiment has pre-selected as the first group of participants usability experts because they will give reliable feedback from a designer point of view. The group is considered highly technical and with a minimum of four years of professional usability testing experience -- HCI graduate students. The second group will comprised of K-12 teachers who are certified to teach in their respective states in the United States. A pre-survey questionnaire will categorize the teacher group further into computer novices and advanced computer skills groups. The third group of participants will be students. For the purposes of this research, the individuals in this group are considered novice users.

4.5.2 Experimental Procedure

The research experiment is web based. The subjects who have agreed to participate will be sent a link through e-mail or given one on site with the details of the study. They will be able to complete the study at the own convenience. Before taking part in the study, participants will be provided with an Institutional Review Board (IRB) form to inform them of their rights. The IRB also notes that their participation is purely voluntary and that they can withdraw from the study at any time without any giving any reason. The experiment process will have three phases: (a) Pre-questionnaire, (b) Visiting website (or using tools) (c) Post-questionnaires (see Figure 8).
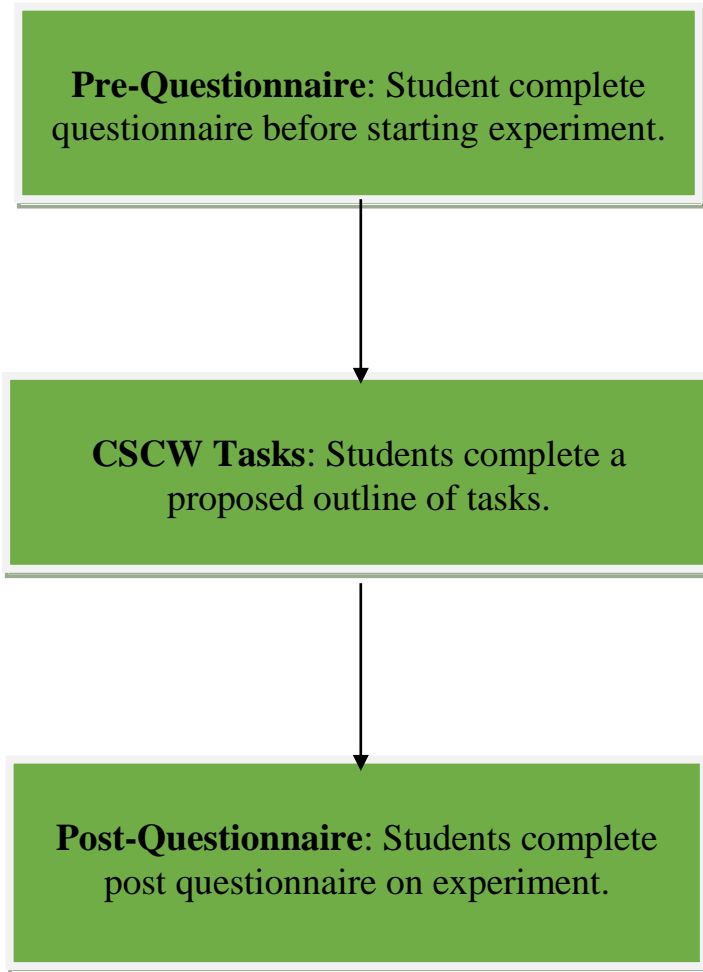
Figure 8: Experiment Procedure Diagram

5 SYSTEM FRAMEWORK IMPLEMENTATION &

PROCEDURE

This chapter outlines in detail the refined requirements for the development of the system based on the initial requirements analysis and system implementation. It also includes the details of the scenario system in phase I to the prototype system in phase III. The implementation has been outlined with Unified Modeling Language (UML) as an object-oriented (OO) design standard to capture requirements.

5.1 Initial Phase I

We approached the project by first identifying a number of tools that were suitable for a CSDT environment and CSCW community of practice with which to collaborate. We needed scenarios for each Big Idea for games that would be built into our CSCW environment. For more feedback and insight, we conducted a few case studies with physicists, whose perspective would benefit our project. For the CSCW portion of the project, we decided to use an evolutionary prototyping. It is quite different from regular prototyping and its main goal is to build a robust prototype in a structured yet dynamic manner. Thus, an evolutionary prototype is the foundation of the main system whereas the new system is a product of improvements on the initial system based on new requirements and changes from users [30]. This process allows a continuous refinement of the system and is based on an acknowledgement that designers do not necessarily understand all the requirements initially. They will build on the system, adding features as they understand the requirements more.
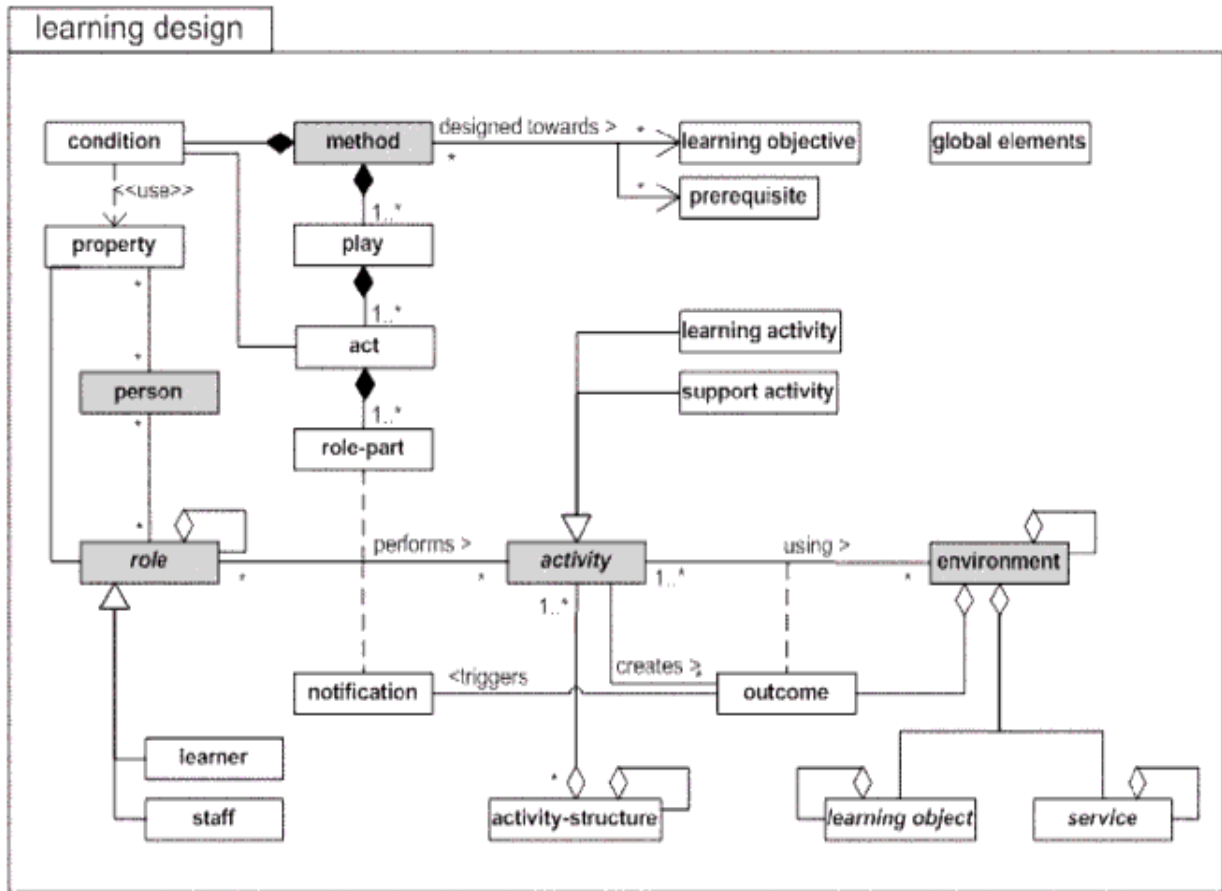
Figure 9: Educational Game Scenario Diagram

5.2 Case Studies: Break Dancer Tool (Phase II)

The case studies provided useful information. One interviewee offered this conceptual

idea for teaching break dancing:

"If I [R. Jones] were to teach breakdancing to middle/high school students from a physics

background, I would define breakdancing (give background history), describe the various

moves in breakdancing, and explain how the moves are maneuvered using physics and

laws of physics.

For example, we use physics in our everyday lives and breakdancing is no

exception. Physics is in effect in every movement of the body in breakdancing.

Breakdancing may seem to defy the laws of physics but they are really displaying and using the laws with great control. Some of the common moves of breakdancing are flares, float, spins, and windmills. Windmills display the laws of physics. One of the most common used moves among breakdancers, the windmill requires the dancer to propel his or her legs in the air, preventing their contact with the ground. The legs are spread and move around the body in a circular fashion, resembling the blades of a windmill pivoting around one point. The windmill involves many aspects of the laws of physics such as friction, Newton's Law of Inertia, centripetal motion and uniform circular motion:

Friction: In order for the dancer to move, he or she must first overcome the force of friction between their clothes and the floor. Friction is a force that involves two surfaces moving against each other. The rubbing of the two substances causes them to pull at each other and thus absorbing some of the force applied to the substance. That's why dancers dance only on smooth surfaces such as linoleum or a cardboard box to reduce the friction between their clothes and the floor, thereby maximizing their speed. The windmill looks better when it is done at high speed.

Newton's Law of Inertia: In addition to friction, the windmill also demonstrates the laws of motion. When the windmill is executed, the legs are moving in a circular motion. The legs want to remain in their state of motion (a straight path); however, since the legs are attached to the body, the legs swing in a circular motion. This motion demonstrates Newton's First Law of Motion, a body at rest or motion will remain at rest or motion unless an outside force is acted upon it. Therefore, the legs will remain in motion until an outside force is acted upon them.

Centripetal Force and Uniform Circular Acceleration: The windmill also demonstrates the centripetal force: any motion in a curved path represents accelerated motion and requires a force directed toward the center of curvature of the path. Centripetal force means "center seeking" force. The force has the magnitude F=mv^2/r, where m is mass, v is velocity, and r is radius. During a windmill, dancers use their legs to drive their bodies in a circular motion. During this movement dancers are constantly accelerating as they spin. As the dancers spin in a uniform circular motion, their legs act as their radius and their feet are the points where velocity is tangent to the circle they are creating."

This overall idea was a great concept and will be implemented in a future version of Break Dancer. In addition, the user found a few weakness in Break Dancer that will have to be considered. The user felt that it would be better received if the application was in modules. The modules would allow the user review the function of each module in the application. The user could then assess understanding by requiring each student to create a similar module to create a different executable.

5.3 CSCW Environment Prototype (Phase III)

We designed multiple wireframes and outlines to capture the essence of what we thought would be attractive and appealing to the user. In addition, we considered usability when focusing on our environment design. The initial versions are shown in Figures 10, 11, 12, & 13. Figure 14 shows the final product.
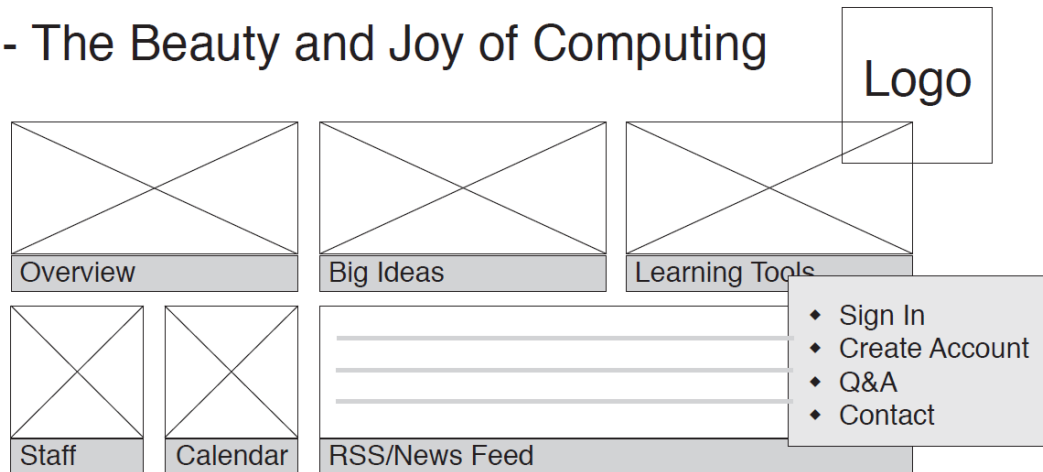
Figure 10: Conceptual CSCW Environment Home Page Model
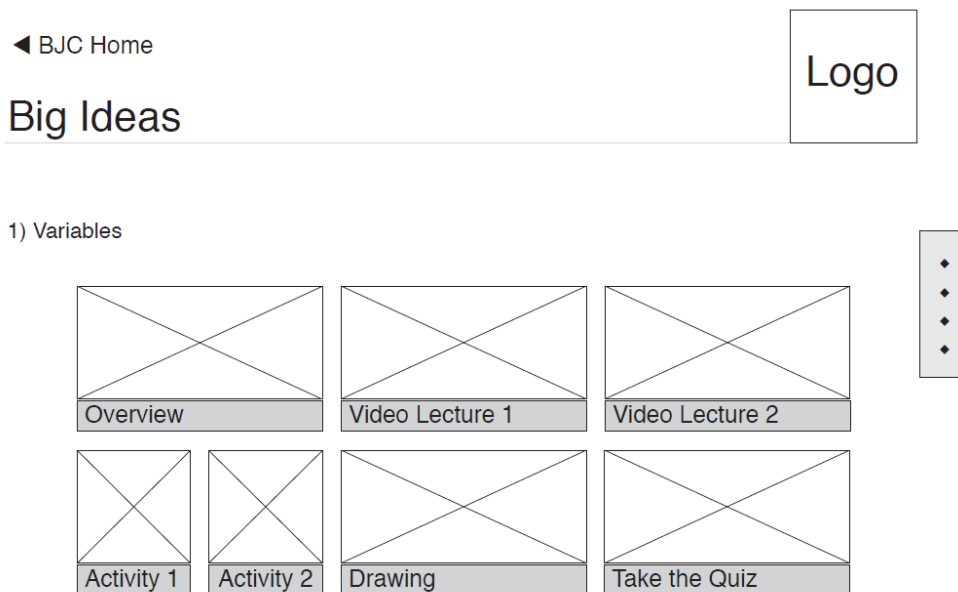
Figure 11: Conceptual CSCW Environment Sub Page Model

Figure 12: Conceptual CSCW Environment Home Page Model Implemented

Figure 13: Conceptual CSCW Environment Sub Page Model Implemented
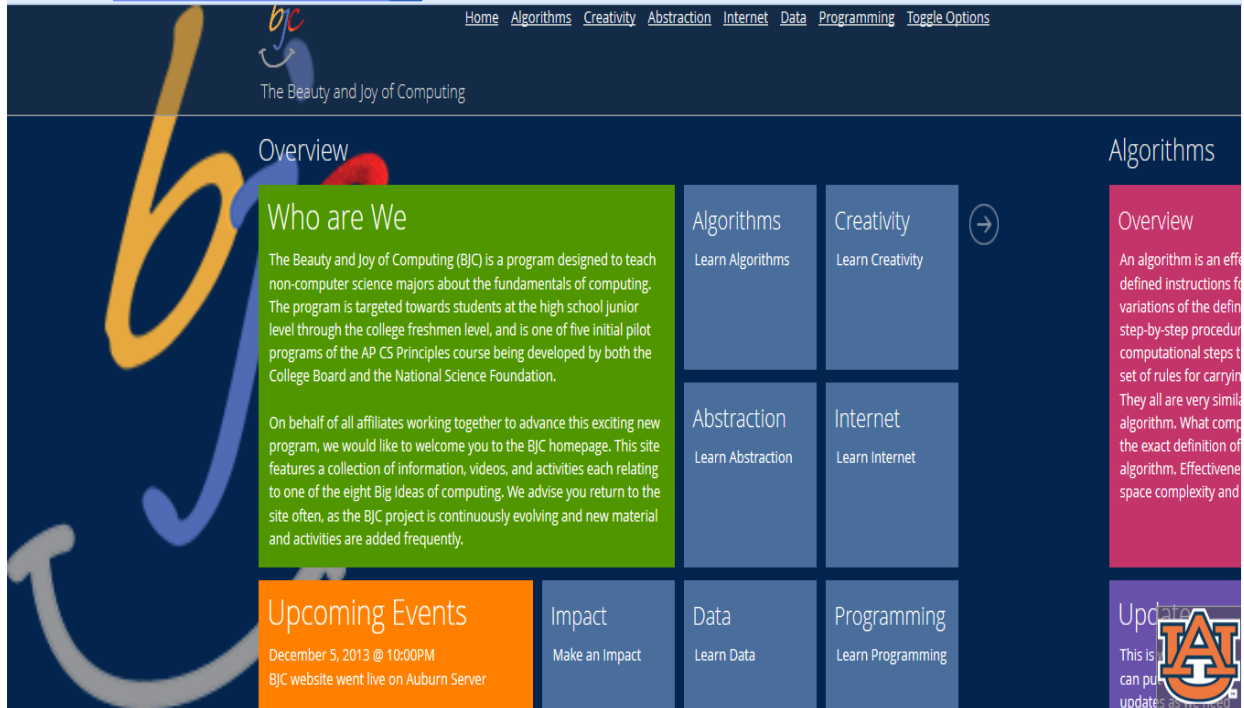
Figure 14: Final CSCW Environment Version
http://spider.eng.auburn.edu/bjc/homepage/index.php

5.3.1 Requirements

In considering functional requirements, we thought that the website itself should provide a gateway for users to find information about comp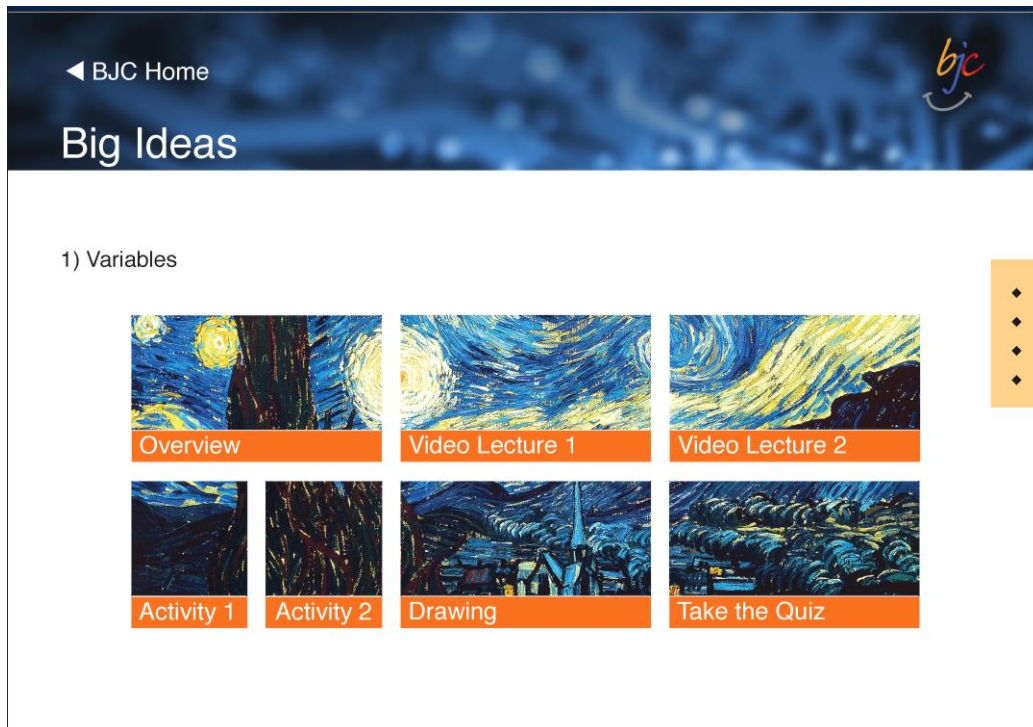uting. The website will contain lectures and games that have been designed increase the user's knowledge of a specific computing aspect. It will provide a safe and reliable medium for the stimulation learning computing ideals. We created this website with the purpose of making the overall topics in computing easily accessible to students who in most cases would not have access to this information.

I. Homepage Interface Requirements

    a. The home page should contain the tabs to the other parts of the site in a tiled-app format.

    b. Each tile should contain links to different information throughout the site.

c. After logging into the website, professional users should have certain options: upload a new game or lecture, post, or comment inside of the community blog about a specific teaching topic.

II. Security Requirements

a. Users should create an account to access the full site.

b. A user is defined as a teacher, parent, or student.

c. Any unregistered user of the website can access the homepage.

III. Computing Content Pages Requirements

a. After selecting an area they would like to explore, unregistered users may then choose to play a game or lecture. After that game or lecture has finished, the user will be prompted to either create an account or log into an existing account if they wish to continue to use the site.

b. Student users have full access to all games and lectures with no time limitations.

IV. User Registration in Pages

a. Users should indicate if they are a professional or not.

b. Each professional user will be verified by system administrators.

c. A user will be prompted to enter certain information to determine the demographic using the website.

V. Data Requirements

a. The content pages should contain games and videos related to the page topic.

VI. Interface Requirements

a. The information should be easily visible for users so they can select what they are interested in.

b. The information should be shown in a rotational manner that offers brief

descriptions of what the lecture or game is about
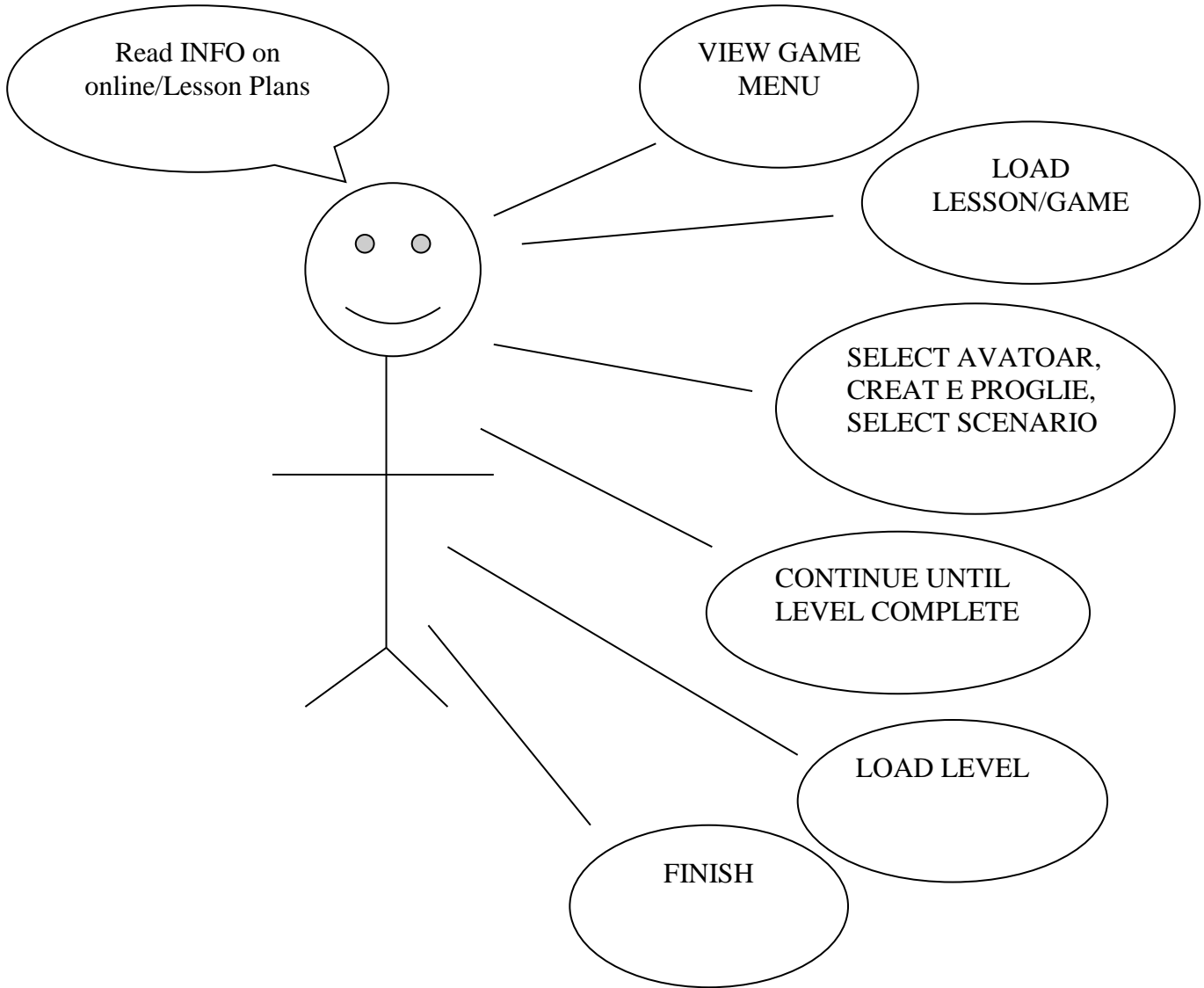
Figure 15: Use Case Diagram for Game/Scenario System
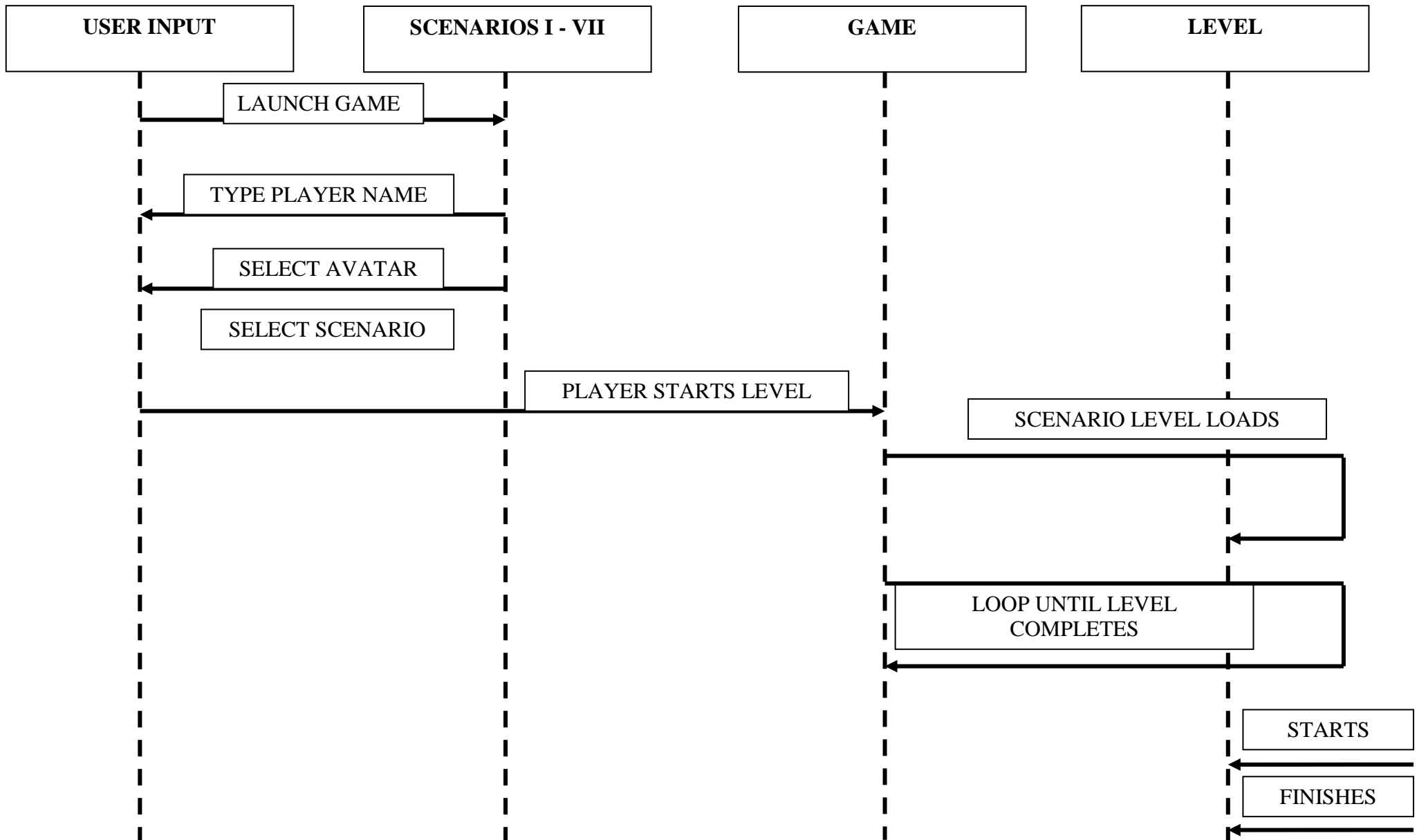
Figure 16: Scenario Sequence Diagram

53

5.4 Use Case and Sequence Diagrams

As Figures 15 and 16 display, we designed a use case diagram to have a set of usability requirements to support our game design among novice users and expert groups. This will allow for the sharing of best practices in implementing the software and getting the most out of the system. Our research plans are to provide a user-friendly means for novice computer users to create, adapt, and re-use existing best practices for such items as artifacts, videos, and curriculum examples. System requirements serve as our guide in building design cases that support the functionality and usability of the system.

To accomplish a detailed analysis and design of the system, we will use Unified Modeling Language (UML) for object modeling and as a specification language to state the system requirements and functionality. In addition, we employed use cases and user scenarios to analyze, capture, and document specifications in relation to novice users, who are the potential users of the system.

Members of the CSCW community (i.e. K-12 teachers and students) will interact with the system directly. Teachers and System administrators will be responsible for content generation. They will create and modify lessons plans, tests, and tutorials on an "as-needed" basis. The system will also allow guests to view content, but not provide guests the level of permission to access restricted member functions such as editing, creating, and downloading unless they are enjoined as members of the community by the administrator.

6 CSDT Framework

6.1 CSDT Framework

In general, the CSCW framework used here will resemble much of what is used in CSDTs. It is reflective and interface-light. The interfaces tend to have only a few methods, and abstract classes tend to provide default implementations for methods that should be overridden. Methods that may be automated are marked with the @PropertyMethod annotation. Primitive member variables that may be automated are marked with the @PropertyPrimitive annotation. Events that may be fired must be explicitly constructed and made available to the framework. The framework itself provides a variety of services to the CSCW tool. The primary services are as follows:

- Interface services

- GUI construction and maintenance

- CSCW interrogation

- XML serialization and deserialization [33]

6.2 Interface Services

The framework is able to manage most of the housekeeping needed to mediate automation via a set of abstract classes that the consumer must extend. These classes are loosely hierarchical in nature. These classes are documented in more depth in the JavaDocs, but the most important classes are described here. The top-level class is the PEngine. The consumer provides a list of classes to the PEngine through PEngine's constructor. The user may create and possibly automate all of these. The classes *must* extend the PObject abstract class. The PEngine may specify its own primitive properties with the @PropertyPrimitive annotation such as, for example, a gravity parameter for CSDTs involving physics simulators. PEngines can also specify

events, to which the user may attach scriptlets to be invoked at relevant times during execution. The PObject has a HitTest method, into which an X and Y coordinate is passed. Hit testing allows the framework to determine which object a user is clicking on in order to present the user with the properties and codelets for that object. In addition to properties, a PObject may also specify events that are fired by the CSDT at relevant times during the execution of a codelet. A PObject *may only* fire an event during runtime, never during design time. If, for instance, an event exists that is fired every second, this event must stop firing when the user stops codelet execution.

A PObject can specify a collection of member methods that may be invoked on the PObject. For instance, a PObject for a physics-aware skateboarder may support a SetVelocity method that sets the skateboarder's velocity. The CSDT can also pass primitive parameters into an event, which is then passed to the codelet. For instance, an OnGoingVeryFast event could pass in a dx and dy parameter that represents the current speed of an object when an object is going very fast. There is currently one built-in event for the PEngine and PObject classes that serves as a hardpoints for user codelets, and it is OnStart. With it, users can programmatically set things in motion.

An important design note is that, internally, the PVariant class wraps all datatypes. If the consumer intends to use a datatype for a property, method, or event that is *not* one of these datatypes, a runtime exception will be thrown. The supported datatypes are int, float, Boolean, and String [33].

6.3 CSDT GUI Construction and Maintenance

The framework namespace contains a master GUI class, helpfully named GUI, which a consumer is expected to extend and implement. The GUI class is abstract due to one method,

GetEngine. GetEngine recovers a PEngine instance that will be automated. Note that GUI extends JApplet, so any extension of GUI may be embedded in a webpage in the same way as anything else that extends JApplet [33].

6.4 CSCW Interrogation

After acquiring the single PEngine, GUI queries the PEngine for its list of supported objects, properties, and events, and then queries the known objects for their list of properties, methods, and events [33].

6.5 Serialization

Serialization is the smallest service provided by the framework. For the sake of compatibility, the serialization format is XML. Unless otherwise necessary, consumers are encouraged to use XML formats. The consumer *must not* serialize any of the following:

- Any property marked with @PropertyPrimitive. The framework will serialize these, and will internally handle string conversion and format validation.

- Any state whose value is not available before codelet execution. These states are generated by the codelets and therefore should not be assigned except by codelet execution.

- Any property present in a CSDT base class. Properties present in a base class are serialized if they are relevant. If they are not serialized there is a good reason for it.

- Any codelets or event bindings. The framework is responsible for managing relationships between codelets and events. In most cases, consumers will find that they do not have to write any serialization code at all [33].

7 DEVELOPMENT TOOLS

7.1 Tortoise Subversion Server

TortoiseSVN is easy to use revision control / version control / source control software for Windows. It manages changes to documents, programs, and other information stored as computer files. Based on Subversion. TortoiseSVN provides a nice and easy user interface for Subversion. It is developed under the GPL, which means it is completely free, including the source code. However, just in case users do not know the GPL too well: they can use TortoiseSVN to develop commercial applications or use it in their company without any restrictions. Since it is not an integration for a *specific* IDE like Visual Studio, Eclipse or others, they can use it with whatever development tools they choose.

As a Subversion client, TortoiseSVN has all the features of Subversion itself, including:

- Most current CVS features.

- Directories, renames, and file meta-data are versioned.

- Commits are truly atomic.

- Branching and tagging are cheap constant time operations.

- Efficient handling of binary files [32].

The main reason for using any tool is its ease of use. All commands are available directly from Windows Explorer. Only commands that make sense for the selected file/folder are shown. Users will not see any commands that they cannot use in their situation. They can see the status of their files directly in Windows Explorer. The descriptive dialogs are constantly improved due to user feedback. It allows moving files by right dragging them in Windows Explorer.

7.2 Java Programming Language

Java is the language of choice for this project. Java will also work great with the selected development environment. It is a general-purpose, concurrent, class-based, object-oriented computer programming language specifically designed to have as few implementation dependencies as possible. Java is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture [37].

One characteristic of Java that will be beneficial to this project is portability. Computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but they are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets [37].

7.3 QT Creator Integrated Development Environment (IDE)

Qt (pronounced Q Tee or Cute) Creator is a cross-platform C++ integrated development environment that is part of the Qt SDK. It includes a visual debugger and an integrated GUI layout and forms designer. The editor's features include syntax highlighting and auto completion, but not tabs. Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux and FreeBSD. On Windows, it can use MinGW or MSVC with the default install and can use Microsoft Console Debugger (cdb) when compiled from source [35].

Qt Creator includes a code editor and integrates Qt Designer for designing and building GUIs from Qt widgets. The code editor in Qt Creator supports syntax highlighting for various languages [35]. In addition, the code editor can parse code in C++ and QML (QT Model/Meta Language) languages and as a result provide code completion, context-sensitive help, and semantic navigation.

Qt Designer is a tool for designing and building GUIs from Qt widgets. It is possible to compose and customize the widgets or dialogs and test those using different styles and resolutions directly in the editor. Widgets and forms created with Qt Designer are integrated with programmed code, using the Qt signals and slots mechanism.

Qt Creator provides support for building and running Qt applications for desktop environments (Windows, Linux, FreeBSD, and Mac OS) and mobile devices (Android, Blackberry, Maemo, and MeeGo). Build settings allow the user to switch between build targets. For mobile device targets, Qt Creator can generate an installation package, install it to a mobile device attached to the development computer, and run it there.
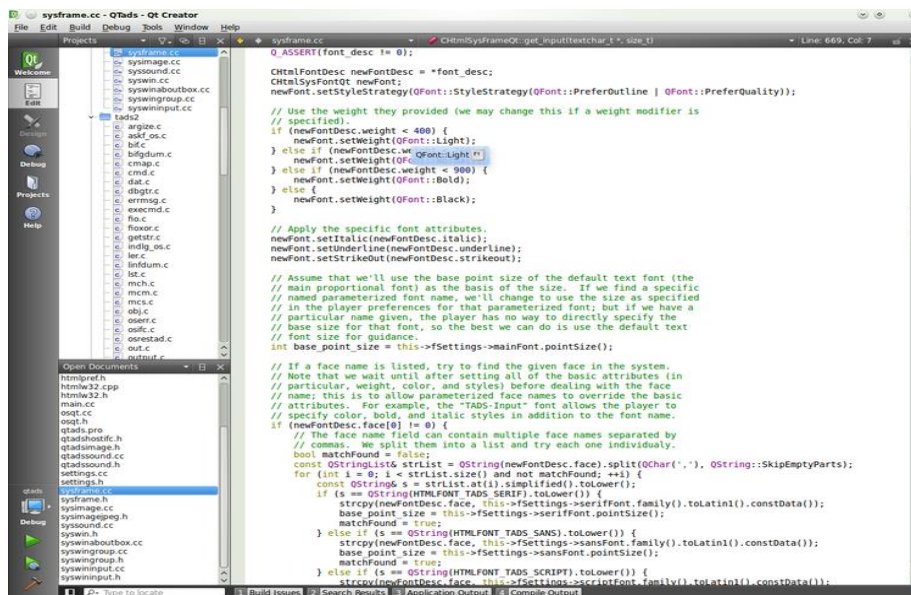

Figure 17: QT IDE Window

60

There are multiple range bindings for Qt. The one we want to focus on is the Java binding language. To implement this in Qt we would use Qt Jambi. Qt Jambi is a Java binding of the cross-platform application framework Qt. It enables Java developers to use Qt within Java programming language. In addition, Qt Jambi generator create Java bindings for other Qt libraries and future versions of Qt. Unlike GTK, there are no Swing LAF implementations that use Qt for rendering [36].

7.4 Unity 3D Development

Unity 3D is an integrated authoring tool for creating 3D video games or other interactive content such as architectural visualizations or real-time 3D animations. Unity is similar to Director, Blender game engine, Virtools, or Torque Game Builder in the sense that an integrated graphical environment is the primary method of development. The editor runs on Windows and Mac OS X and can produce games for Windows, Mac, Wii, or iPhone platforms. Linux support may be introduced in the future, but it is not currently a priority. There are no plans to port the authoring tool to Linux, as "the cost/benefit ratio is simply not there." It can also produce browser games that use the Unity web player plugin supported on Mac and Windows. The tool consists of a fully integrated editor, graphics, asset importing, and deployment [28]. It also supports shaders, advanced physics, scripting, terrains, unity asset servers, audio, and video.

Figure 18: Unity 3D Graphical User Interface

61

Fully Integrated Editor

- *Play, pause, and step*: Click the Play button to run your game instantly. Pause and Step forward to analyze complex behavior in detail. While the game is running (or paused), you can alter values, assets, or even scripts. This means you can do more experimentation and more testing for a better final game.

- *Easy Editor Customization*: Streamline your workflow using simple, Editor-specific scripts. We wrote the Editor using Unity's own scripting, so anything we can do in the Editor, you can do as well.

- *Customizable Editor Layout*: Choose from one of many built-in application arrangements, or create and save your own. Different tasks might benefit from different arrangements, so use easy hotkeys to switch arrangements as needed.

- *Drag and Drop*: Visually drag assets and objects in the editor to assign Textures, Audio, Behaviors, and script variables. Create logical GameObject hierarchies to manage and maximize your game's functionality.

- *Prefabs*: To streamline the repeated use of complex GameObjects, you can turn one or more of them into a *prefab*. This prefab can then be easily placed throughout the game or initiated at runtime. Any changes to the original prefab are propagated to all dependents, so both major and minor adjustments can be made very quickly to a large number of GameObjects. [28]

Graphics

- *Incredible Speed:* Unity defines fast. Rendering is sorted to minimize state changes, taking lights and shadows into account. On beefy hardware, Unity renders millions of polygons per second.

- *Particle Systems:* Visual creation and manipulation of particle systems is simple. Create rain, sparks, dust trails -- anything that you can imagine.

- *Direct 3D 9 and Open GL:* Unity makes sure that your games run everywhere. It includes a full DirectX and an OpenGL renderer. In our testing lab, we test Unity across hundreds of graphics hardware and driver combinations.

- *Flexible Pipeline:* Hook into the rendering pipeline to create special effects. Use low level rendering commands to achieve exactly what you want [28].

Asset Importing

- *Instantaneous, Automatic Importing*: When any asset file is saved, it is seamlessly imported without exception. Each asset's import settings are remembered, so you can set them once and never worry again.

- *Incredible 3D Packages support*: Unity can import 3D models, bones, and animations from almost all 3D applications.

- *Just Hit Save*: Hit Save in Maya, 3ds Max, Cinema 4D, Cheetah3D or Blender, and Unity will pick up all changes across your entire project.

- *True Type Font Support*: Unity handles pixel-perfect rendering of TrueType fonts. Drop in any TTF font and start making great-looking text. When localization time rolls around, Unicode fonts are supported, as are Unicode strings.

- *Texture Handling*: Save your multi-layer Photoshop files normally and let Unity automatically compress your images with high quality DXT texture compression. It's all automatic without a single required click.

- *Height-map to Normal-map Conversion:* Any texture can be converted into a Normal-map. This process is automatic and instantaneous, even when you later change your image files.

- *High-quality Mipmap Generation*: Unity supports several different mipmap generation methods: Detail Fade, Kaiser Filters, Gamma Correction, and more.

- *Audio Support*: Unity can import any audio format that is supported by QuickTime. Audio can be internally converted and distributed as Ogg Vorbis for keeping down your game's published file size [28].

Deployment

- *Standalone Mac & Windows*: Publish standalone builds for Mac OS X (Universal Binary, or specific, smaller Intel/PPC-only builds) and Windows 2000/XP/Vista/7.

- *Support for Old Hardware and Drivers:* Many potential players are using outdated graphics hardware and drivers. Even many common computer configurations are much less than ideal for games. Unity has built-in fallbacks and workarounds for compatibility problems. Unity has support for almost all hardware/software combinations, in both DirectX and OpenGL. Unity has undergone extensive compatibility testing, which means you do not need to effort into making users can run their games.

- *Web Deployment*: Unity-made games can be played inside a web browser thanks to the Unity Web Player Plug-in. The plug-in download is small (about 3 MB); auto-installs without a browser restart, and already has an 8-digit distribution. It works on all modern browsers including Internet Explorer, Firefox, Safari, and most Mozilla-based browsers. You can publish a web game that is identical to a standalone in visual fidelity from the same project. Auto-streaming web players and additional WWW streaming all work

64

together to reduce load times. It is also possible to customize the loading screen colors, progress bar, and graphics to make the short wait a pleasant one.

- *Mac OS X Dashboard Widgets*: Still using the same project as your standalone, you can create a 3D Dashboard Widget. Using a Dashboard Widget is a great way to advertise your content, give a pick-up-and-play preview, or extend your game's functionality [28].

There are two main licenses: *Unity* and *Unity Pro*. The Pro version has additional features, like render-to-texture and post-processing effects. The free version also displays a splash screen in standalone games and a watermark in web games. Both Unity and Unity Pro include the development environment, tutorials, sample projects and content, support via forum, wiki, and future updates in the same major version.

At the time of writing, Unity Technologies, the company behind the multiplatform engine, has updated Unity for iPhone to v1.5. With this release, the development team has implemented a number of enhancements and new features. The biggest of these is a major performance gain, which can be as much as three-times what was possible with the previous engine release [31].

7.5 Wii

Wii is a home video game console released by Nintendo. As a seventh-generation console, the Wii primarily competes with Microsoft's Xbox 360 and Sony's PlayStation 3. Nintendo states that its console targets a broader demographic than that of the two others. As of February 2010, the Wii leads the generation over the PlayStation 3 and Xbox 360 in worldwide sales and in December 2009 broke the record for best-selling console in a single month in the United States.

Unity's game engine allows developers to create, modify and iterate on Wii game functionality with several features optimized for the console, including Live Preview for instant

previews regardless of the development phase; Scriptable Controller scripting class for reading data from the Wii Remote, Nunchuk controller and Classic Controller; Optimized Character Animation for creating characters without required exports, imports, or modifications; Scriptable Shaders for built-in shaders optimized for Wii or custom shaders; and Click to Publish for running a game on a Wii development kit, building it with one click [29].

Figure 19: Wii Gaming Console

To evaluate or license Unity 3D engine for use on the Wii console, developers must meet the following requirements:

- They must own Unity Pro 2.x.

- They must be a an Authorized Developer for the Wii console and obtain a Wii development kit.

The Unity engine is licensed for use on the Wii console on a per-title basis and the licensing fee will depend on intended distribution. The retail price is $30,000 per title and for WiiWare it is $15,000 per title [28].

7.6 iPhone

The iPhone is a line of Internet and multimedia-enabled smart-phones designed and marketed by Apple Inc. The iPhone functions as a camera phone including text messaging and visual voicemail, a portable media player equivalent to a video iPod, and an Internet client with

e-mail, web browsing, and Wi-Fi connectivity. It uses the phone's multi-touch screen to provide a virtual keyboard in lieu of a physical keyboard.

When it comes to developing applications, and in particular games, for the iPhone, there are two different roots developers can choose to follow. The first is to acquire the iPhone SDK and start learning to code in Objective C. The other is to use a commercial engine and build an app on top of the functionality it offers. Unity falls into the latter category and is widely regarded as one of the best engines for developing games both on iPhone and across other platforms [31].

The Unity engine is available for a 30-day unlimited free trial, after which an iPhone Basic license costs $399 and the advanced license costs $1,499. Cheaper licenses are available for those not needing iPhone developments. The Unity website displays all the license prices and a comparison of features [28, 31].

7.7 Break Dancer Tutorial Development

We further developed the tutorial to incorporate improvements to the existing tutorial. The existing tutorial (Figure 8) needed more information to offer a better understanding of what was being taught.

Figure 20: Online Break Dancer Tutorial

The current tutorial has the following outline:

- 3D Geometry

- Rotation in the human body

- Software Tutorial

  - All Joints

  - Adjusting Speeds and Frames

  - Rotation and Translation

  - Software Functions

- Pointers

In revising the tutorial, we extensively researched the Break Dancer tool and contacted math teachers to get an understanding of what would be needed to compile a lesson plan. The previous version of Break Dancer's tutorial only highlighted a few of the topics discussed in the tutorial.

In the new tutorial, we talked about every topic in extensive detail, including examples with each explanation (Figure 18).
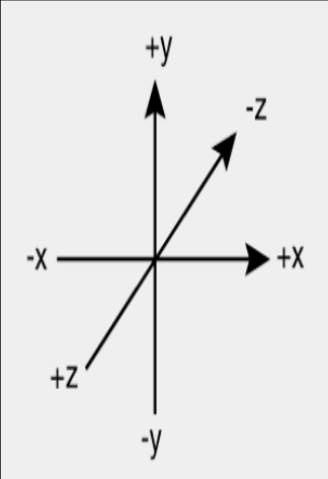
Introduction to 3D Geometry & Space

You are probably familiar with 2-dimensional Cartesian geometry by now, i.e. *planar* geometry. With this software, you will learn about 3-dimensional space, or *solid* geometry.

Three-dimensional space is a geometric model of the Physical universe in which we live. The three dimensions are commonly called length, width, and depth (or height), although any three mutually Perpendicular directions can serve as the three dimensions.

Real-world objects exist in 3 dimensions. For example, a cuboid, or a box, is completely described by three parameters, *length*, *breadth*, and *height*. Corresponding to that, each point in the Cartesian space has 3 coordinates x, y, and z.

For this tutorial, the x axis is along the length of the screen, the y axis is along the height of the screen, and the z axis will be the one coming out of the computer screen!
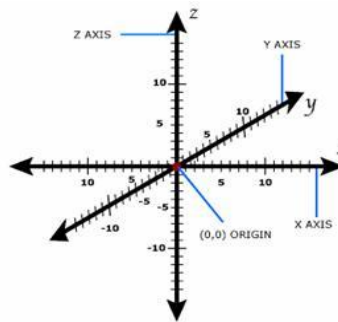
Tait-Bryan Angles

Yaw, pitch, and roll, also known as **Tait–Bryan angles**, named after Peter Guthrie Tait and George Bryan, are a specific kind of Euler angles very often used in aerospace applications to define the relative orientation of a vehicle respect a reference frame. The three angles specified in this formulation are defined as the roll angle, pitch angle, and yaw angle.

These angles are particularly seen when looking at the rotation of an object in 3D space. The rotations can be split into three parts. This will be further discussed under the Transformations heading.
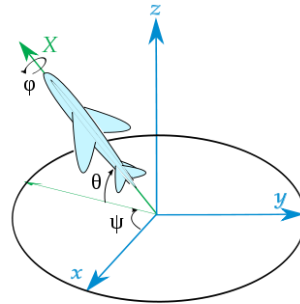
Figure 21: Revised Break Dancer Tutorial

The revised tutorial has the following outline:

- Introduction to 3D Geometry & Space

- Tait-Bryan Angles

- Cartesian Coordinates in 3D Space

- Transformations (Translations & Rotations)

- Sine Function

The revised lesson plans should be equally if not more beneficial to the students and the teachers. The tool we used to design the new tutorial is PowerPoint and Captivate. We also created lessons plans (Appendix B &C) and mini tests (Appendix D & E). We conducted data analysis on the tutorial, lesson plans, and mini tests. The next chapter presents the results.

8 EXPERIMENTAL EVALUATION AND DATA ANALYSIS

As mentioned previously, we modified validated versions of the *Computer Understanding and Experience - Potosky & Bobko, Computer System Usability Questionnaire* (CSUQ) -- Lewis, 1995 *IBM and Perceived Usefulness and Ease of Use* to collect data from usability experts, teachers, and students. This helps ensure a better usability experience with each user.

The experimental procedure included a pre-questionnaire, a task list, and post-questionnaire to collect data from the experts for the second phase of the study. The phase surveyed 100 participants. The outcome results of the study led to the recommendation of the use of CSCW to support communities of practice to share best practices.

Gender	% Response	Count (n)
Male	55%	57
Female	45%	43

Table 8.1: The overall groups gender distribution table

The survey collected gender distribution data to ensure that the expert feedback data was not biased, was representative of both genders and could be considered valid and credible. The survey group was a collection of students and teachers.

72

Age Group	% Response	Count(N)
17 or younger	0.00%	0
18-20	0.00%	0
21-29	41.67%	42
30-39	8.33%	10
40-49	8.33%	10
50-59	33.33%	28
60 or older	8.33%	10

Table 8.2: Group age distribution table
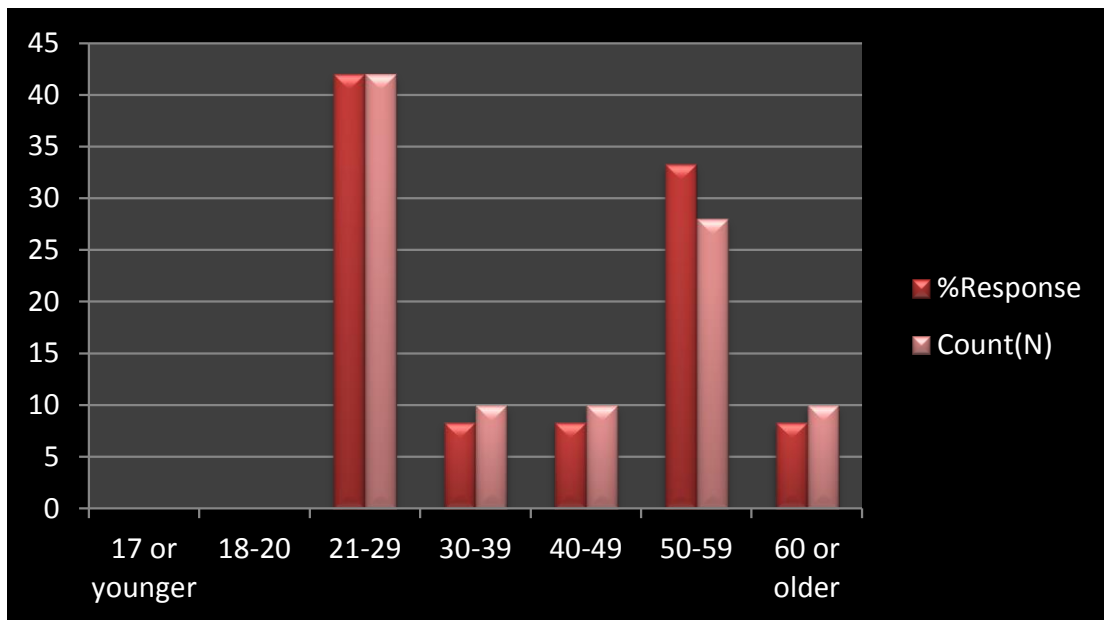
Figure 22: Group Age Distribution Chart

We used a benchmark to assess the CSCW environment. The benchmark is necessary to evaluate the time involved performing collaborative tasks. If the benchmark revealed that tasks were tedious or unappealing for a participant, we ascertained that this function was too difficult to use or needed improvement. The maximum benchmark time of a task for a novice user was set by doubling the average time an expert took to perform the task.

We queried users to assess the need to modify the feature before a final testing and acceptance test. Table 8.3 shows the comments elicited from the group on the positives and negatives of the system.

Positives
"Website was wonderful"
"Very well setup, updates are visible. Games are needed to take a break. Found it very
      enjoyable."
"Great idea. Website is user-friendly."
"Very interaction, looks, and works great.
"Concept is great"
"Website was helpful"
"Nice, clean, and modern layout. I like that there are multiple routes to access the same data."
"Interface is good and great choice of colors."

Negatives
"Colored Background makes it hard to read text at times"
"Contrast between the background colors"
"No pictures."
"BJC menu should be visible when navigating from page, easier to revert back to home page."

Table 8.3: Experts elicited comments table CSCW Environment tool

We addressed the negatives by paying close attention to the color issues that seem to trouble a few people. We chose to do multiple runs to assess whether it was an issue that could eventually lead to more issues in future situations. In general, the general responses from the self-reported feedback approved the suggested that the overall CSCW environment was very usable.

To validate our model, we based our analysis on the current solution, which is intractable. We used the older version of Break Dancer to test what users felt was more visually effective, user-friendly, and gave better explanation of the lessons.

8.1 Data from the Usability Testing of Break Dancer

The questionnaire used for the testing of the tutorial contained four questions that focused on teachability and learnability.There were ten participants who were teachers and not previously exposed to the CSDTs. The result showed that the participants particularly liked the tool and how the tutorial conveyed the information to the students and teachers.

	Average General User Satisfaction
Terrible ------------------Wonderful	3.5
Frustrating----------------Satisfying	3.8
Difficult-------------------------Easy	3.5
Boring-----------------------------Fun	2.0
Strongly Agree = 5, Agree =4, Neutral = 3, Disagree = 2, Strongly Disagree = 1	

Table 8.4: Usability Testing Current Tutorial

Table 8.5 represents data taken on the current tutorial on the site. Overall, people did not like the initial tutorial d and commented that it needed drastic improvement. The participants noted that the current tutorial did not convey enough information to the user and it would be too complex to teach lessons.

	Average General User Satisfaction
Terrible ------------------Wonderful	4.5
Frustrating----------------Satisfying	4.8
Difficult-------------------------Easy	4.5
Boring----------------------------Fun	5.0
Strongly Agree = 5, Agree =4, Neutral = 3, Disagree = 2, Strongly Disagree = 1	

Table 8.5: Usability Testing of Revised Tutorial

A moderate percentage of the participants answered that the tutorial was fun and overall satisfying in comparison to the older tutorial. The Anova results for the level of boring and fun are listed in table 8.7. Our initial null hypothesis was done at a level of significance of 5.0. This means that by looking at the P-Value we see that there is a significant different between the two results.

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Column 1 (table 1)	10	20	2	0.222222
Column 2 (table 2)	10	50	5	0

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	45	1	45	405	8.65E-14	4.413873
Within Groups	2	18	0.111111			
Total	47	19				

Table 8.6: Anova Results

76

The results reveal that the participants found this tutorial conveyed the appropriate information to the teacher and student. Therefore, the overall revised tutorial makes it easier for the student to learn the suggested lesson with the Break Dancer tool.

The experimental evaluations will explain evaluations reported in chapter8.4, which includes the Experimental Design Sec. 8.2, Data Collection Sec. 8.3, and Experimental Results Sec. 8.4. The data collection section will present methods for the work, materials used, experimental data (i.e. demographics, user satisfaction questionnaires), procedures, and experimental observations. The chapter concludes with a discussion of the experimental hypothesis and the implications of the study. The usability results and implications support the adoption of CSCW system as suitable tool for a community of practice to share and re-use best practices.

8.2 Experimental Design

We conducted a heuristic review and cognitive walkthroughs study to gain insight and understanding of how end users (i.e. potential novice users) would interact and perceive the CSCW environment. We sought to ascertain support for our hypothesis that a collaborative environment is a useful for forum for user groups to engage through interactive learning. We did a comparative analysis with the old CSDT website versus the new CSCW environment. This work supports our hypothesis through usability ratings among potential user groups. Our goal was to answer the following questions:

1. Can the students learn better using Culturally Situated Design Tools (CSDTs) when compared to using traditional teaching techniques?

2. Can CSDTs clear up misconceptions students typically have when listening to the instructor?

3. Do students retain knowledge better from using the CSDT?

4. When students use CSDTs, what is the reaction from interacting with the environment?

5. How satisfied are the students and instructors with using CSDTs?

In answering these questions, we provide data on the feasibility of the CSCW tool as a collaborative tool supporting for enhancing the learning experience. These results can serve as a general guideline for choosing and developing online collaborative tools for future groups.

8.3 Heuristic Review

The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces. We used our five expert users in this area. Usability consultant Jakob Nielsen developed this method based on several years of experience in teaching and consulting about usability engineering.

Nielsen's heuristics are probably the most-used usability heuristics for user interface design. He developed the heuristics based on work together with Rolf Molich in 1990. The final set of heuristics still used today were released by Nielsen in 1994. The heuristics as published in Nielsen's book *Usability Engineering* are as follows:

A. Visibility of system status:

    The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

B. Match between system and the real world:

    The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

C. User control and freedom:

    Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

D. Consistency and standards:

    Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

E. Error prevention:

    Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

F. Recognition rather than recall:

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

G. Flexibility and efficiency of use:

Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

H. Aesthetic and minimalist design:

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

I. Help users recognize, diagnose, and recover from errors:

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

J. Help and documentation:

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Below are the results emerging from the heuristic review. There were 80 questions that ranged from how the home paged look to the writing and content quality of

the sites. The experts were given time to review the CSDT site and the CSCW site before

taking this survey. The ratings for the heuristic review were based on a 10 point scale as
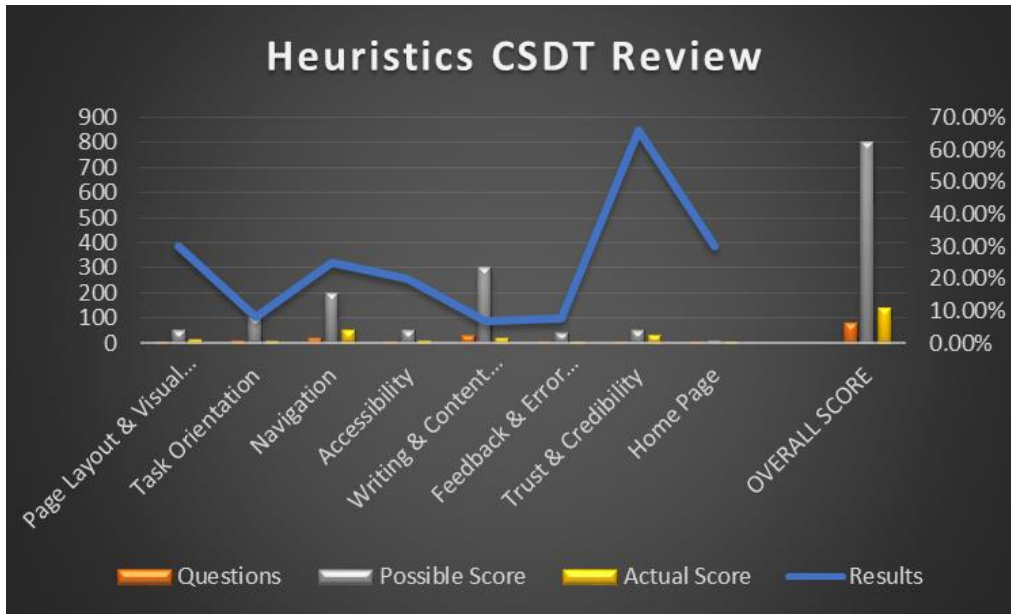
follows:

- A. 0 points if it falls short of a metric

- B. 3 points if it is halfway there

- C. 5 points if it does the job

- D. 10 points if it is "out of this world awesome"
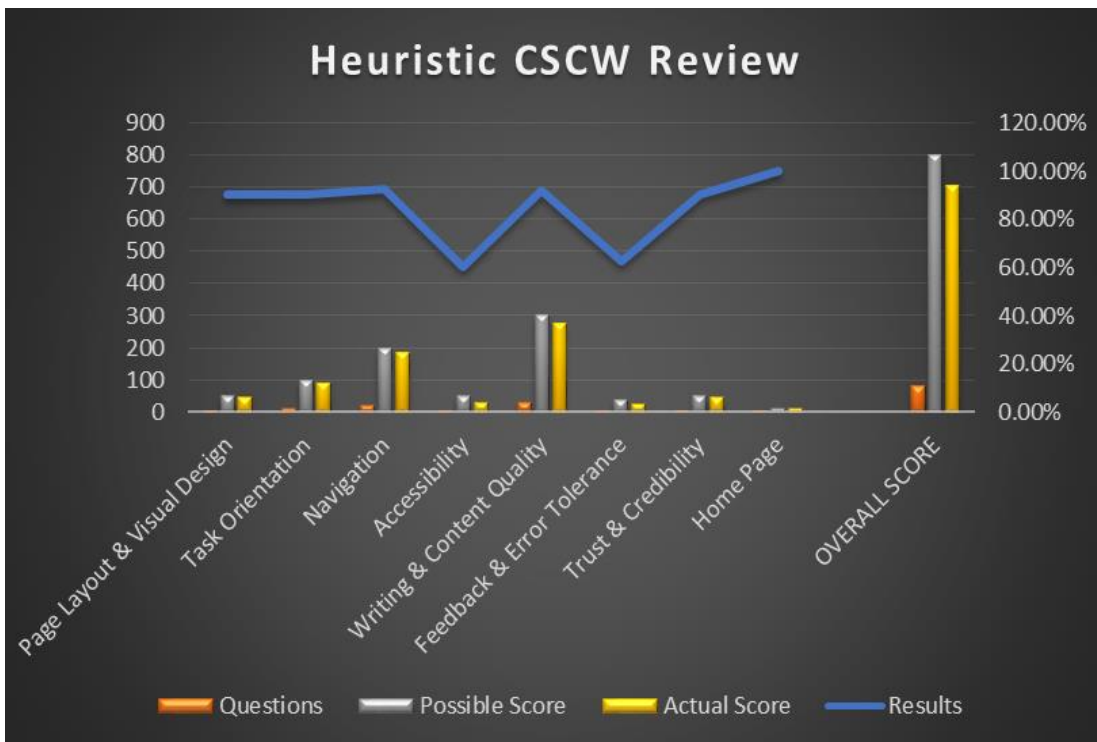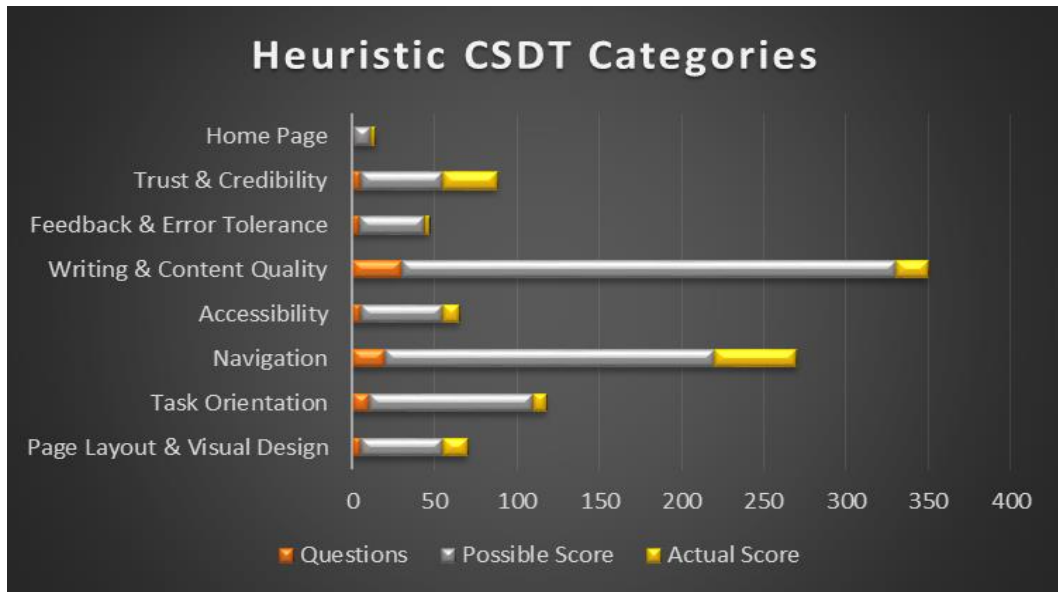

Figure 23: Heuristic Review Chart from CSCW
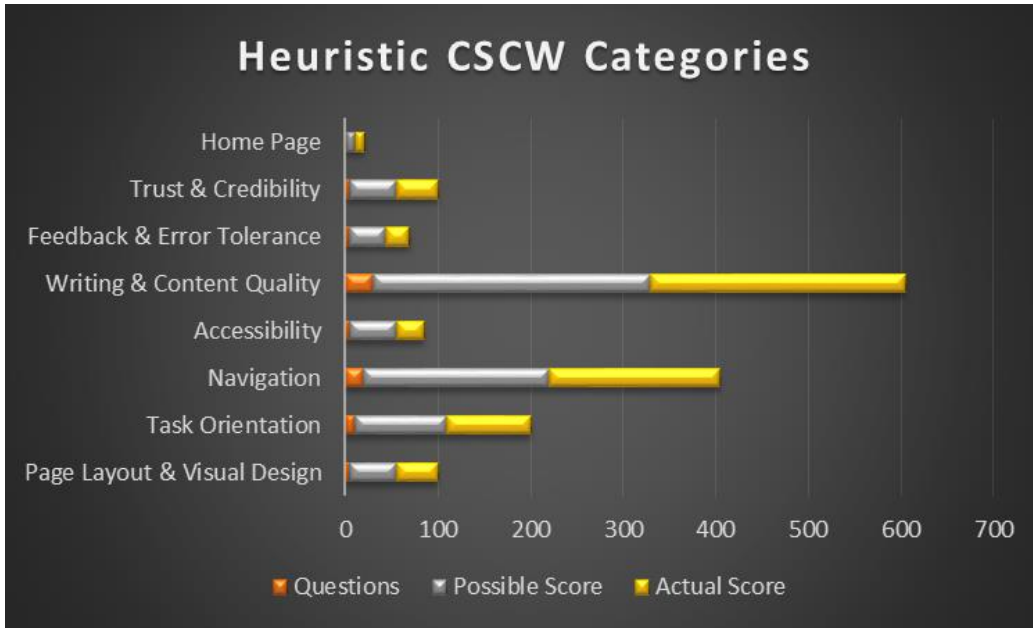
Figure 24: Heuristic Categories at a Closer Look (CSCW)

Figure 25: Heuristic Review Chart from CSDT

Figure 26: Heuristic Categories at a Closer Look (CSDT)

The review reveals that the majority of the experts in the group found the CSCW site more aesthetically pleasing and that the content quality better than the CSDT ssite. We received feedback that the CSDT site was overall confusing and typically hard to navigate. This leads into the next phase of the experiment of doing a cognitive walkthrough.

8.4 Cognitive Walkthrough

The cognitive walkthrough method is a usability inspection method used to identify usability issues in interactive systems, focusing on how easy it is for new users to accomplish tasks with the system. Cognitive walkthrough is task-specific, whereas heuristic evaluation takes a holistic view to catch problems not caught by other usability inspection methods. The method is rooted in the notion that users typically prefer to learn a system by using it to accomplish tasks, rather than, for example, studying a manual.

The cognitive walkthrough used a group of evaluators inspecting a user interface by going through a set of tasks and evaluating its understandability and ease of learning. The user interface is often presented in the form of a paper mock-up or a working prototype, but it can

also be a fully developed interface. We conducted our walkthrough using a working prototype. The evaluators included engineers, software developers, and teachers.

8.4.1 Defining the Input to the Walkthrough

- *Who will be the users of the system?* This included specific background experience or technical knowledge that could influence users as they attempted to deal with a new interface. We considered the users' knowledge of the task and of the interface.

- *What tasks will be analyzed?* The tasks were limited to a reasonable but representative collection of benchmark tasks. Task selection was based on the results of concept testing and requirements analyses.

- *What is the correct action sequence for each task?* For each task, there must be a description of how the user is expected to view the task before learning the interface. A description of the sequence of actions should aid in the accomplishment of the task with the current definition of the interface. Example actions are "Click on one of the Big Ideas," and "Move to Big Ideas block and click to enter."

- How is the interface defined? The definition of the interface described the prompts preceding every action required to accomplish the tasks being analyzed, as well as the reaction of the interface to each of these actions. In the interface that was implemented, all information was available from the implementation.

8.4.2 Walking through the Actions

As the walkthrough proceeded, the evaluators asked the following four questions:

- *Will the users try to achieve the right effect?* For example, their task is to play a game but the first thing they have to do is get to a big idea and go into the game. Will they know how to do that easily?

- *Will the user notice that the correct action is available?* This relates to the visibility and understandability of actions in the interface.

- *Will the user associate the correct action with the effect to be achieved?* Users often use the "label-following" strategy, which leads them to select an action if the label for that action matches the task description.

- *If the correct action is performed, will the user see that progress is being made toward solution of the task?* This question allows us to check the system feedback after the user executes the action.

The evaluators constructed a success story for each step in the task cases. General conditions where a success story can be told is given next in "features of success" section. In the case that a success story cannot be told, we would construct a failure story, providing the criteria (one or more of the four questions above) and the reason why the user may fail.

8.4.3 Features of Success

The following are criteria used to construct success stories.

A. Users may know "what effect to achieve":

- Because it is part of their original task, or

- Because they have experience using a system, or

- Because the system tells them to do it

B. Users may know "an action is available":

- By experience, or

- By seeing some device (like a button) or

- By seeing a representation of an action (line a menu entry)

C. Users may know "an action is appropriate" for the effect they are trying to achieve:

- By experience, or

- Because the interface provides a prompt or label that connects the action to what they are trying to do, or

- Because all other actions look wrong

D. Users may know "things are going OK" after an action:

- By experience, or

- By recognizing a connection between a system response and what they were trying to do

8.5 Experiment

These sections outline the methodological aspect of the comparative studies.

8.5.1 Experiment Setup

The study was conducted in Auburn University Classroom, Huntsville City Schools' District in Huntsville, AL, Perry County School District in New Augusta, MS, and Redstone Arsenal Army Base. There was one expert from each area with a minimum of 25 participants from each area.

8.5.2 Experiment Design

The experimental design for the study involved a comparative study of users looking at both sites, CSDT and CSCW, so they can gain insight. The users performed tasks and gave feedback on usability.

8.5.3 Materials

The materials for this experiment included the informed consent for users to sign before undertaking the experiment and the tutorials that were prepared to guide users through their learning and evaluation sessions.

8.5.4 Informed Consent

The Auburn University Institutional Revise Board requires researchers to have an informed consent approval of research designs when conducting any type of research involving surveys, interviews, or human factors. The informed consent stated to the participant the purpose of the study, justification, procedures, benefits, and risks of the project. It guarantees the participants that all responses will be held confidential and will only be used anonymously.

8.5.5 Pre-Test & Post Test Questionnaire

The pre-questionnaire gathered general information about the participants to assess whether they met the criteria established for classification as both novice and content area experts. It also grouped them into age sets.

The post questionnaire gathered detailed information about how participants assessed their performance and the system. It also judged on a qualitative level the extent to which users learned more about using the system for personal purposes.

8.5.6 Procedures

The experiment began by informing participants of Auburn University Institutional Review Board approval for the experiment and affirming the informed consent. This familiarized participants with the experiment and allowed them the opportunity to either sign the informed consent and become a participant or decline to participate.

Following the consent, participants completed an online background pre-questionnaire. This helped to identify that our expert users were capable of completing the tasks at the rate we assumed. In addition, it determined if other users they met the minimum qualifications and were suitable for the experiment. After completing the tasks, the users completed a post questionnaire survey.

8.6 Data Collection & Analysis

The following table is an overview of the experimental instruments and measures that were used to collect data in this study:

Instrument	Description
Pre-test Questionnaire	User background, demographics and expectations
Performance data	Time, types # of rules and errors per creation
User observations	Qualitative observation and critical incidents
Post-test Questionnaire	User satisfaction and system ratings
Retrospective Interviews	Debriefing and elaboration

Table 8.7: Experimental Instrument Overview

8.6.1 Performance Data

This was collected in terms of user created practices with aid of minimalist tutorials.

8.6.2 User Observations

This was collected in the form of user observations as well as formal observations in the form of critical incidents as self-reported data from participants.

8.6.3 Interviews

We used retrospective interviews to capture any information and statements the users provided either as critiques or affirmations of the system's success in addressing individual needs and, in some cases, community needs.

8.6.4 Statistics

We reported and analyzed the data collected to evaluate whether the tool was easy to use, supported novice users, and promoted of online informal education. The results confirmed and validated usability criteria to guide us in adopting the CSDT system and creating the CSCW environment for sharing information between students and teachers.

8.6.5 Experimental Results

The goals of the heuristic review, cognitive walkthrough, and usability testing were to discover whether the CSCW environment meets the planned usability specifications and develop suggestions for improving the design. To fulfill these goals as evaluators, we have to understand not just what the participants did during the test tasks, but why they behaved and reacted as they did. This was accomplished by characterizing the test participants, probing the details of the responses to the tasks (e.g. time, and errors), and noting subjective reactions (e.g. comments while using the system and ratings or opinions provided after tests) [50].

8.6.6 Participant Background

Using the pre-questionnaire survey form, we were able to document various types of user characteristics. Figure 27 represent the age range of participants with a group distribution. The numbers in percentages (e.g. 8%) represent the percentages of test subjects respectively.
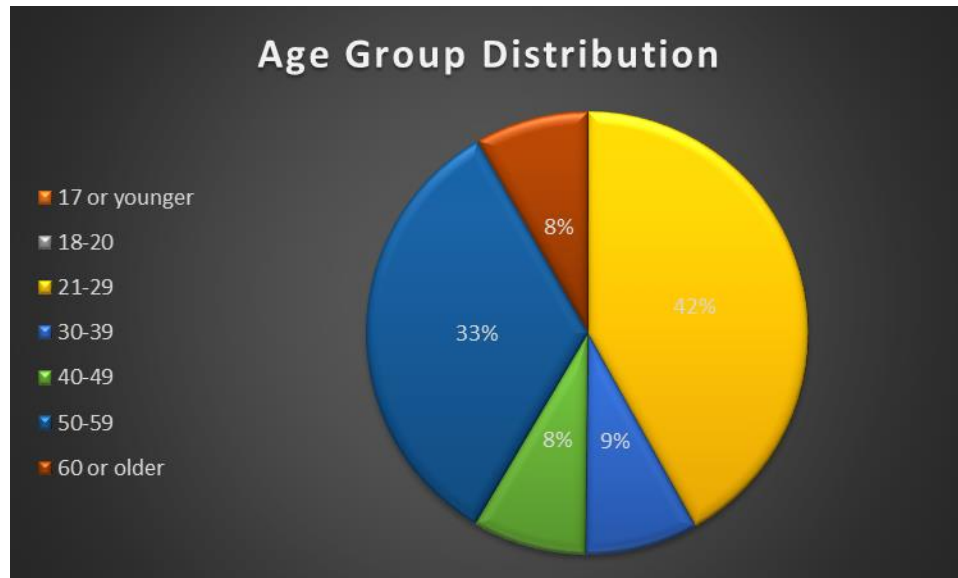
Figure 27: Age Distribution Results

Studies show that some define occupations as a categorical variable. In this study, participants' self-reported occupations were grouped into engineers, developers, students, teachers, and other (those involved in the survey with no direct relation to the group). The majority of our participants were teachers.
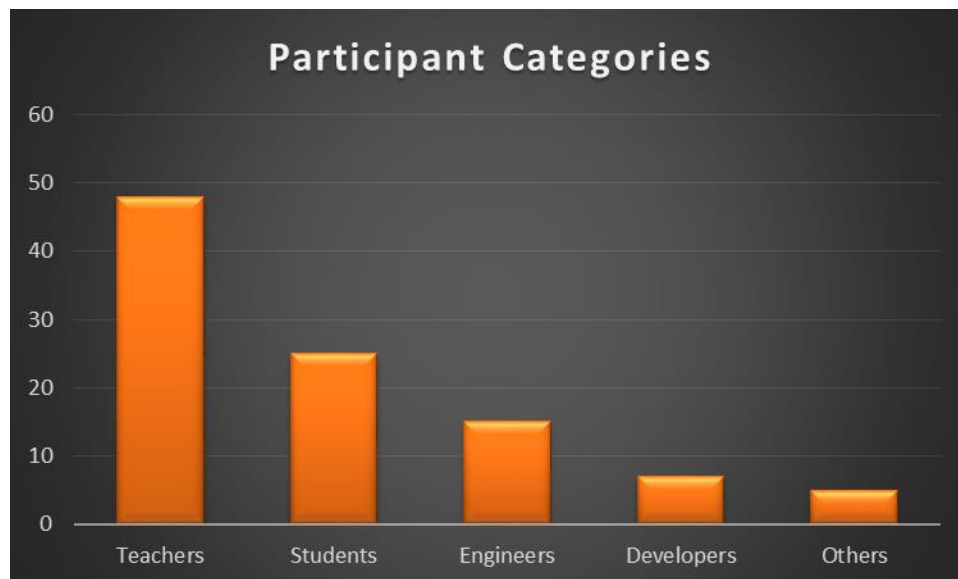
Figure 28: Categories of Participants

Figure 29 summarizes the level of education of various user groups. It also shows number of year's computer experience to aid in categorizing participants as either novice or expert user.
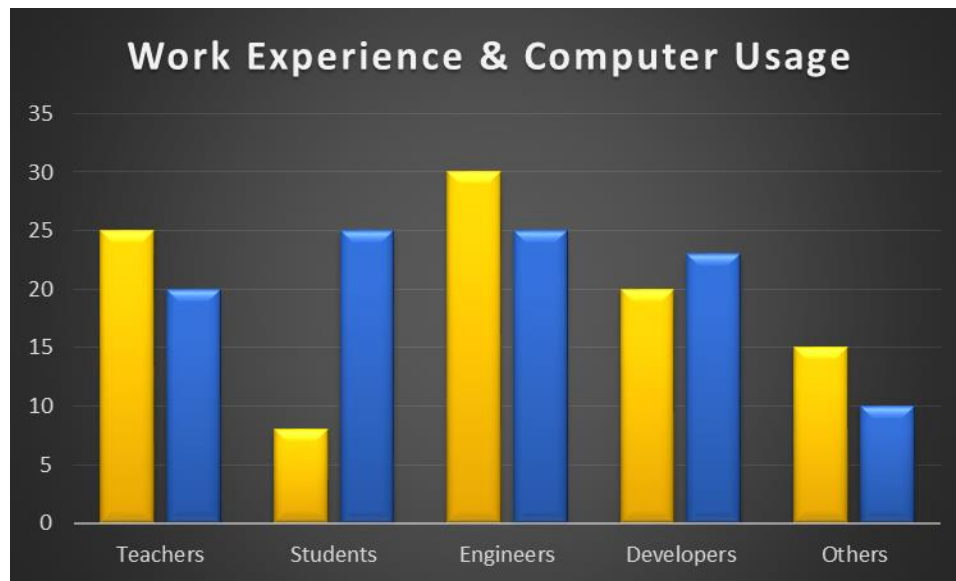
Figure 29: Work Experience and Computer Usage

9 FUTURE WORK & CONCLUSION

9.1 Future Work

The future work for this project will be focused on implementing the software in Unity 3D for future use with Wii. This is our final goal. Work still needs to be done from evaluation viewpoint, including qualitative and long-term evaluation. One hypothesis is that a math game that engages students will motivate them in other Computer Science and IT courses and encourage them to pursue these careers in college. Measuring student grades in the math pre-class and post-class could determine if the game engaged students sufficiently to improve their performance in later classes.

In addition to implementing Break Dancer in Unity 3D, we need to find a way to implement the drag and drop interface. This idea was not implemented and we will continue to work on it for a future project. As stated earlier, the problem encountered is that the current interface only has two panels and a scripting drag and drop interface needs three panels. Therefore, we concluded that by the time we embark on a future project the initial designer will have directed us in the path we need to take.

The CSCW environment needs to be tested more and additions should be made accordingly. The potential for this project is to extend this to many schools across the United States. The environment will gradually evolve and change with the needs of future users.

9.2 Conclusion

The primary expected outcome of this research was to have created a novel CSCW framework environment that would attract teachers and students. In turn, this would open doors for a new era of teaching science, technology, engineering and math methods (STEM) to enter the classroom. The framework will house all CSDT related information and other pertinent

information that the user deems feasible. While HCI has continuously made great strides, there is still always room for development and improvement in our world of technological advancements. This CSCW environment will serve as a catalyst toward the continued development of collaboration among the teaching community and students. The findings presented here will directly contribute to the goals of the HCI/CSCW discipline.

We implemented a better tutorial, lesson plan, and mini tests to help teach the lesson better and grab the user's attention. This helps make the Break Dancer tool a better instrument for teachers as well as students. The CSDTs offer an exciting convergence of both pedagogical and cultural advantages. In addition, we created a CSCW environment that can be used across a wide range of venues. We must keep in mind that computer games are very popular among children and adolescents. Consequently, they could be exploited by educational software designers to render educational software more attractive and motivating [10].

Unlike many other ethnomathematics examples, we can modify the interface to allow a close fit to the math curriculum that makes it easy for teachers to incorporate into their class. At the same time, the ability of computer games to move between virtual and physical implementations allows use in the arts; and their historical connections provide teaching opportunities in history and social science. Most importantly, they allow for a flexible, creative space in which students can reconfigure their relations between culture, mathematics, and technology.

We learned that the adoption and success of an informal educational online tool and its value as a secure and easy to use learning tool depends heavily on its usability. Computer supportive collaborative work theory and human computer interaction research provide usability acceptance test knowledge that can support effective user evaluation acceptance tests of online

based collaborative tools. However, the formulation of an effective and efficient acceptance testing process is complicated by a plethora of design theories and models. The premise of this research is that user acceptance tests can provide a mechanism for identifying a suitable CSCW tool that is understandable and easy to use for a novice user. We base the practicality of this approach on previous research efforts in human studies and HCI fields that tested the suitability of software products before deployment.

Research could help make games more attractive to different types of uses, and address the differences in the types of games that appeal to either gender. Females tend to be more attracted to games that involve relationship building than do males, who tend to prefer action games. Educational games also need to be culturally sensitive.

This research examined the issue of proving a collaborative tool to support communities of practice engaged in informal learning by sharing of best practice. It developed a model for managing groups that emerge within the community of practice. This research postulated a hypothesis that novice communities will benefit this CSWC tool, interface, and interaction design. This research also took a usability evaluation approach to assess effectively the environment created to support CSCW.

BIBLIOGRAPHY

1. Bell, G., Blythe, M., Gaver, B., Sengers, P., Wright, P. 2003. Designing Culturally Situated Technologies for the home. In *Chi '03 extended abstracts on Human factors in computing systems.* 1062-1063.

2. Culturally Situated Design Tools. Available at http://www.rpi.edu/~eglash/csdt.html

3. Eglash, R., Bennett, A.,O'Donnell, C., Jennings, S., Cintorino, M.. Culturally Situated Design Tools: Ethnocomputing from Field Site to Classroom. In *American Anthropologist*. Available at: http://www.ccd.rpi.edu/Eglash/csdt/teaching/papers/aa.2006.108.2.pdf

4. Eglash, R.. Ethnocomputing with Native American Design. In *Information Technology and Indigenous People*. Available at: http://www.ccd.rpi.edu/Eglash/csdt/teaching/papers/indig_it.htm

5. Game attributes for application in learning. In *Summit on Educational Games.* Available at: http://www.fas.org/gamesummit/Resources/Summit%20on%20Educational%20Games.pdf . Oct 5, 2005. (53 pages)

6. Math and culture. Available at: http://www.commcorp.org/dys/pdf/DYSMath-MathCulture

7. Mathematical Games as an Aid to teaching Mathematics. Available at: http://www.bbc.co.uk/dna/h2g2/A798221

8. RAMER, W. 2000. What Makes a Game Good? In *The Games Journal.* Available at: http://www.thegamesjournal.com/articles/WhatMakesaGame.shtml

9. Salen, K. and Zimmerman, E. 2004. Rules of Play Game Design and Fundamentals. Games As Cultural Rhetoric. Massachusetts Institute of Technology. 515-534.

10. Virvou, M., Katsionis, G., & Manos, K. 2005. Combining software games with education: Evaluation of its educational effectiveness. In *Educational Technology & Society*. Available at: http://www.ifets.info/journals/8_2/5.pdf 54-65

11. Bates, B. 2004. Game Design(2nd ed.) Boston: Thompson Course Technology.

12. Claypool, K. and Claypool, M. 2005. Teaching software engineering through game design. In Proceeding of the 10th annual SIGCSE Conference on innovation and Technology in Computer Science Education (Caparica, Protugal, June 27-29, 2005, ITiCSE '05) ACM Press, New York, NY, 123-127.

13. Fullerton, T., Swain, C. & Hoffman, S. 2004. Game Design Workshop: Designing, Prototyping and Playtesting Games, San Francisco, CA: CMP Books.

14. Rabin, S. 2005. Introduction to game development. Hingham, MA: Charles River Media, Inc.

15. KONZACK, L. 2002. Computer Game Criticism: A Method for Computer Game Analysi*s*. In *Proceedings of Computer Games and Digital Cultures*. Tampere, Finland: Digital Games Research Association. Available at: http://www.vrmedialab.dk/~konzack/tampere2002.pdf (12 pages)

16. Integrate Math & Computer Science Math Lessons That Introduce Computer Science to K-8 Students. Available at: http://trainingpd.suite101.com/article.cfm/math_lessons

17. Salen, K. and Zimmerman, E. 2004. Rules of Play Game Design and Fundamentals. Defining Culture. Massachusetts Institute of Technology. 505 – 513

18. Salen, K. and Zimmerman, E. 2004. Rules of Play Game Design and Fundamentals. Games As Cultural Rhetoric. Massachusetts Institute of Technology. 515-534

19. Goode, J. 2008. Increasing Diversity in K-12 computer science: Strategies from the field. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 362-366

20. JUUL, J. 2000. What computer games can and can't do. In *Digital Arts and Culture* conference, Bergen, Norway, August 2000. Available: http://www.jesperjuul.net/text/wcgcacd.html (7 pages)

21. CRAWFORD, C. 1984. The art of computer game design, Osborne/McGraw-Hill, Berkeley, Calif. Chapter 1. Available at:
 http://www.vancouver.wsu.edu/fac/peabody/game-book/Chapter1.html (9 pages)

22. Educational Gaming Has to Engage the Students . . . . but How? Available at: http://edutechie.com/2006/11/29/educational-gaming-how/

23. Issues Involved in the Development of Educational Videogames. In *Summit on Educational Games: Harnessing the Power of Video Games for Learning.* Available at: http://dpi.wi.gov/pld/pdf/gamedevissues.pdf

24. Fromme, J. Computer Games as a Part of Children's Culture. Game Studies the international journal of computer game research. Volume 3. 2003. Available at: http://www.gamestudies.org/0301/fromme/

25. Video games have 'role in school'. *BBC News.* Available at:
    http://news.bbc.co.uk/2/hi/technology/5398230.stm

26. ME1071 Computer Games- An Introduction. Available at:
    http://w3.msi.vxu.se/medieteknik/kurser/ME1071ht08/2.htm

27. Education and Simulation/Gaming and Computers. Available at:
    http://www.cofc.edu/~seay/cb/simgames.html

28. Unity 3D http://unity3D.com

29. Caoili, Eric. "Unity to Support Wii Console as Authorized Middleware Provider", June
    2008. http://www.gamasutra.com/php-bin/news_index.php?story=18840

30. Smith, Roger A. 1976. Educational games in today's learning. Paper presented at the
    American Industrial Arts Association Annual Convention, Des Moines, Iowa, April 23.
    ERIC Document ED 129 304.

31. Humphries, Matthew. "Unity 1.5 engine released for iPhone bringing major performance
    improvements." Aug. 25, 2009. http://www.geek.com/articles/games/unity-15-engine-
    released-for-iphone-bringing-major-performance-improvements-20090825/

32. TortoiseSVN http://tortoisesvn.net/

33. CSDT Framework Documentation

34. Preece, Jenny. "Sociabilty and usability in online communities: determining and
    measuring success." Behavior and Information Technology. 2001.

35. Qt Creator. http://qt.digia.com/Product/Developer-Tools/

36. Qt Jambi. http://qt-jambi.org/

37. Java. http://www.oracle.com/technetwork/java/index-138747.html

38. Cain, C. (2010). Social Networking Teaching Tools: A Computer Supported
    Collaborative Interactive Learning Social Networking Environment for K-12. Auburn,
    AL: Maters Thesis Auburn University.

39. Dillenbourg, Pierre. "What do you mean by collaborative learning: Cognitive and
    Computation Approaches." Dillenbourg, Pierre. What do you mean by collaborative
    learning? Geneva: Oxford:Elsevier, 1999. pp.1-19.

40. Bannon, Lian J. "Perspectives on CSCW: From HCI and CMC to CSCW." International Conference on Human-Computer Interaction (EW-HCI'92). St. Petersburg: BCS HICOM electronic conference system, 1992. pp 148-158.

41. Grudin, J. (1988). "Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces". Proceedings of the 1988 ACM conference on Computer-supported cooperative work. ACM Press New York, NY, USA. pp. 85–93.

42. Ackerman, M." (2000). "The Intellectual Challenge of CSCW: The gap between social requirements and technical feasibility". Human-Computer Interaction: pp 179–203.

43. Goodsell A., Maher M, Tinto V, Smith B. B and MacGregor J(1992). "What Is Collaborative Learning?" National Center on Postsecondary Teaching, Learning, and Assessment at Pennsylvania State University

44. Fjuk, A. (1998). Computer support for distributed collaborative learning. Exploring a com-plex problem area. Dr. Scient. Thesis 5, University of Oslo, Dept. of Informatics.

45. Fåhræus, E. R. (1999a). Tutoring group learning at a distance. In J. D. Price, J. Willis, D. A, Willis, M. Jost, & S. Boger-Mehall (Eds.), Proceedings of the Society for Information Technology and Teacher Education International Conference (SITE'99, March, 1999), San Antonio, Texas.

46. Fåhræus, E. R. (1999b). Creative tutoring of electronic collaborative learning groups: Sustaining the human factor. Notes from a workshop at the TET'99 Conference on Telecommunications for Education and Training, Gjøvik, Norway.

47. Fåhræus, E. R. (2000a). Growing knowledge – How to support collaborative learning e-discussions in forum systems. Dept. of Computer and Systems Sciences, Stockholm University/KTH, Report series No. 00-005

48. Mary Beth Rosson, John M. Carroll. Usability Engineering. San Ndiego: Morgan Kuafmann, 2002.

49. Baker M. J., Lund K. Promoting reflective interactions in a computer-supported collaborative learning environment, University of Geneva(1990)

50. Maryam A. Computer-Mediated Collaborative Learning: An Empirical Evaluation MIS Quarterly, Vol. 18, No. 2 (Jun., 1994), pp. 159-174

51. Baecker, R.M.; Others, (1995). Readings in human-computer interaction: toward the year 2000. Morgan Kaufmann Publishers.

52. Grudin, J. (1994). "Computer-Supported Cooperative Work: History and Focus". Computer 27 (5): 19–26.

53. Carstensen, P. H. Schmidt, K. Computer Supported Cooperative Work: new challenges to systems design. To appears in Handbook of Human Factors, Kenji Itoh, Tokio, 1999 (23p)

54. Wilson, P. (1991). Computer Supported Cooperative Work: An Introduction. Oxford, Intellect Books. Wooley, DR (1994, July).

55. Dourish, P.; Bellotti, V. (1992). "Awareness and coordination in shared workspaces". Proceedings of the 1992 ACM conference on Computer-supported cooperative workcc. ACM Press New York, NY, USA. pp. 107–114.

56. Nworie, John and Noela Haughton. "Good Intetions unanticipated effects: The Unintended Consequences of the Application of Technology in Teaching and Learning Environments." TechTrends (2008): 52-58.

57. Bosworth, K. and S. J. and Hamilton. Collaborative Learning: Underlying Processes and Effective Techniques, New Directions for Teaching and Learning . Education Research Consumer Guide. Baltimore: US Department of Education, Office of Research, 1994.

58. Schmidt, K.; Bannon, L. (1992). "Taking CSCW seriously". Computer Supported Cooperative Work 1 (1): 7–40.

59. Dillenbourg, P., M., Blaye A. Baker And O'malley C. "Learning in Humans and Machine: Towards an interdisciplinary learning science." The evolution of research on collaborative learning. Oxford: E. Spada & P. Reiman (Eds), 1996. 189-211

60. Grudin, Jonathan. "Computer-Supported Cooperative Work: History and Focus." IEEE Computer (1994): 19 – 26

APPENDIX A

Answer the following questions:
Go to the culturally Situated Design Tool Website http://www.rpi.edu/~eglash/csdt.html and click on the Break Dancer link to answer the following questions.

1. Click on the tutorial link. Explore the link and please respond by circling the reaction that best reflects your overall viewpoint of the tutorial provided. Give additional feedback on your response.

   Terrible ---Wonderful
          1   2   3   4   5

   Frustrating--------------------------------------Satisfying
          1   2   3   4   5

   Difficult--Easy
          1   2   3   4   5

   Boring--Fun
          1   2   3   4   5

2. Now look at the PowerPoint tutorial, lesson plans, and mini tests provided and please respond by circling the reaction that best reflects your overall viewpoint of the tutorial provided. Give additional feedback to your responses.

   Terrible ---Wonderful
          1   2   3   4   5

   Frustrating--------------------------------------Satisfying
          1   2   3   4   5

   Difficult--Easy
          1   2   3   4   5

   Boring--Fun
          1   2   3   4   5

3. What do you think can be added to revised tutorial, lessons plans, and mini tests that can aide teachers in teaching this lesson in a classroom setting?

4. Additional comments:

APPENDIX B

Three Dimensional Break Dancer Tool
Lesson Plan

Lesson Start Date:
Lesson Title: 3D Geometry
Class:
Lesson End Date:
Objective: Three-dimensional solid geometry is part of mathematics teaching programs starting early on with shape recognition.
- Present the idea that real world objects exist in three dimensions
- Three Dimensional Space or planar geometry
- Cartesian Space Coordinates (X,Y, and Z)
- Rotation on X, Y, and Z axes
- Compare, classify, and construct transformations (translations and rotations)

Teaching Strategy
Using a manipulative such as the 3D Break Dancer Tool to model the process of showing three Dimensional Space and Rotation

The teacher will verbalize the usage of the tool and how it relates to the lesson. By doing a hands on activity, it will engage students as they are not feeling forced to learn but motivated. Once this has been done, the students will continue to practice and work with the tool. The students can then begin to apply it independently to other math problems.

Three-dimensional space is a geometric model of the physical universe in which we live. The three dimensions are commonly called length, width, and depth (or height), although any three mutually perpendicular directions can serve as the three dimensions.

Cartesian coordinates in three dimensions Choosing a Cartesian coordinate system for a three-dimensional space means choosing an ordered triplet of lines (axes), any two of them being perpendicular; a single unit of length for all three axes; and an orientation for each axis. As in the two-dimensional case, each axis becomes a number line. The coordinates of a point p are obtained by drawing a line through p perpendicular to each coordinate axis, and reading the points where these lines meet the axes as three numbers of these number lines.
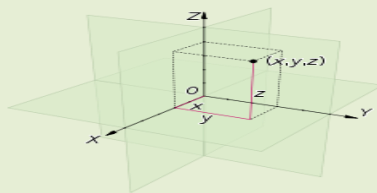
Evaluation/Assessment- Test Items:
Located under teaching materials on site. Should test students performance levels and depth of knowledge.

APPENDIX C

Three Dimensional Break Dancer Tool
Lesson Plan

Lesson Start Date:
Lesson Title: Sine Function
Class:
Lesson End Date:

Objective: Three-dimensional solid geometry is part of mathematics teaching programs starting early on with shape recognition.
- Amplitude & Frequency
- Periods
- Phase Shifts
- Vertical Shifts/Offset
- Apply sine wave function for translations

Teaching Strategy
- Using a manipulative such as the 3D Break Dancer Tool (apply wave function button) to model the process of showing Three Dimensional Space wave translation

The teacher will verbalize the usage of the tool and how it relates to the lesson. By doing a hands on activity, it will engage students as they are not feeling forced to learn but motivated. Once this has been done, the students will continue to practice and work with the tool. The students can then begin to apply it independently to other math problems.
The trigonometric sine function
$$f(x) = a*sin(bx + c)+d$$
its amplitude, period and phase shift are explored interactively using the Break Dancer applet. The investigation is carried out by changing the parameters a, b, c and d. To deeply understand the effects of each parameter on the graph of the function, we change one parameter at the time at the start. Then later we may change more than one parameter. Exploration and understanding of the phase shift is done by comparing the shift between the graphs of the two functions:
$$f(x) = a*sin(bx + c) + d$$
in blue and

$$g(x) = a*sin(bx) + d$$
in red as shown in the figure below.

Evaluation/Assessment- Test Items:
Located under teaching materials on site. Should test students performance levels and depth of knowledge.

APPENDIX D

Break Dancer Software Exam   Name: _____

Material covered for this lesson:
- Present the idea that real world objects exist in three dimensions
- Three Dimensional Space or planar geometry
- Cartesian Space Coordinates (X,Y, and Z)
- Rotation on X, Y, and Z axes
- Compare, classify, and construct transformations (translations and rotations)
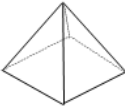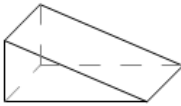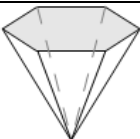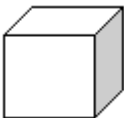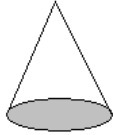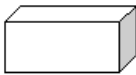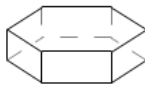
1. Every object you can see or touch has three dimensions that can be measured: _____, _____, and _____.
2. Cartesian Space, in relation to this lesson, has how many coordinates.
3. Describe rotation as it relates to this lesson.
4. What are three of the most important transformations that you many encounter?
5. In 3D Space, Euler angles can be used to describe the rotation of an object. Besides the typical name X, Y, and Z, how does each relate to the Tait-Bryan angles?
6. Name each of the objects below in the diagram. Word bank provided:

   Hexagonal Prism Cube   Rectangular Prism   Triangular Prism   Cylinder
                Cone   Rectangular Pyramid   Hexagonal Pyramid
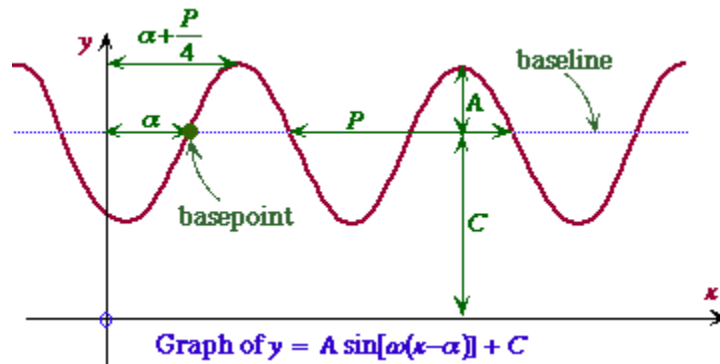
APPENDIX E

Break Dancer Software Exam   Name: _____

Material covered for this lesson:
- Amplitude & Frequency
- Periods
- Phase Shifts
- Vertical Shifts/Offset
- Apply sine wave function for translations

1. What is the equation for a sine wave function? _____

2. **General Sine Curve**

Graph of $y = A \sin[\omega(x-\alpha)] + C$

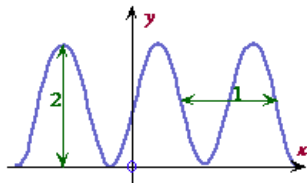A is called the _____ (the height of each peak above the baseline)

C is the _____(height of the baseline)

P is the _____ or _____ (the length of each cycle)
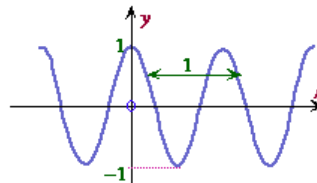
ω is the _____, given by $\omega = 2\pi/P$

α is the _____(the horizontal offset of the basepoint; where the curve crosses the baseline as it ascends)

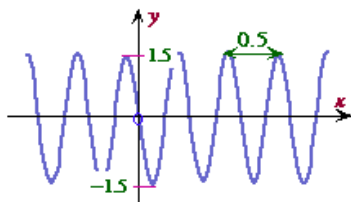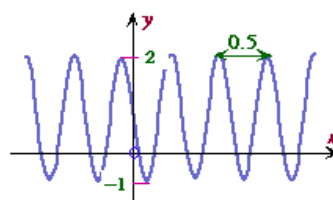Model each of the following curves with a sine function.

3.

4.

5.

6.

104

APPENDIX F

Scenario Title		
Keywords		
Whom do I want to teach?		
Age Range & Grade of Students		
Student Special Characteristics		
What do I want to teach?		
Subject/Field/Skills		
Goals		
How do I want to teach	**Rate 0-5**	
Learning Metaphor that can support learning objectives	Acquisition	0 1 2 3 4 5
	Imitation	0 1 2 3 4 5
	Discovery	0 1 2 3 4 5
	Participation	0 1 2 3 4 5
	Experimentation	0 1 2 3 4 5
Description of the game	Narrative Description of the game plot	
	Goals	
	Characters	
	Scenes	
Narrative description of learning activities – step by step organization and structuring		Learning Settings	Estimated Time
	Before the game:		
	During the game:		
	After the game:		
		Total	
How will I evaluate Students		
Evaluation Approach		
What will learners need in order to achieve learning objectives		
Prerequisite		
Settings and materials		
What do I need for implementing the scenario?		
Application Involved	Mandatory	
	Optional	
Infrastructure/Equipment	Mandatory	
	Optional	
Learning Resource Type		
Time/Space Resources		
Other things to remember		

APPENDIX G

Pre-Questionnaire
1. What is your gender?
    a. Male
    b. Female

2. Which category below includes your age?
    a. 17 or younger
    b. 18-20
    c. 21-29
    d. 30-39
    e. 40-49
    f. 50-59
    g. 60 or older

3. What is your occupation?
    a. Teacher
    b. Student
    c. Counselor
    d. Parent
    e. Other (Please Specify)

4. What do you normally use a computer for? (Select all that apply)
    a. Email
    b. Surfing the net
    c. Homework
    d. Blogging
    e. Maintaining social network (Facebook, Twitter, Etc.)
    f. Work
    g. Other (Please Specify

5. What internet browser do you use mostly?
    a. Internet Explorer
    b. Safari
    c. Firefox
    d. Opera
    e. Google Chrome
    f. Other (Please Specify)

6. Do you use a mobile device for web browsing?
    a. Yes
    b. No

7. Do you use a mobile device for browsing of class materials?
   a. Yes
   b. No

8. Have you taken any course over the internet?
   a. Yes
   b. No
   If yes, please specify the name of the course.

9. Do you have prior experience working in an online learning environment?
   a. Yes
   b. No

10. Do you feel that online materials can enhance traditional classroom materials?
    a. Yes
    b. No

APPENDIX H

Post-Questionnaire
1. It is easy to navigate through this website.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

2. It is easy to find what I want on the website.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

3. Is is eary to use this site upon my first visit.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

4. Clicking on links takes me to what I expect.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

5. The organization of information on the system screens is clear.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

6. How would you rate of the following tools/features in terms of importance for the BJC website:

	Not Important at All	Not Very Important	Neutral	Somewhat Important	Very Important	Not Sure What It Is
Upcoming Events						
Updates						
Course Work						
Games						
Videos						

7. What overall rating would you give the website?
Very Poor   Poor   Average   Good   Excellent
8. I will likely return to this in the future.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

9. I would recommend someone to this site in the future.
Strongly Disagree   Disagree   Neutral   Agree   Strongly Agree

10. Please use the space below to provide any additional comments about your visit our website today.

APPENDIX I

INFORMATION LETTER
for a Research Study entitled
"Social Networking and Culturally Situated Design Teaching Tools:
Providing a Socially Interactive Computer Supported Collaborative Network
Environment for K-12"

You are invited to participate in a research study to establish a more collaborative work environment between students and teachers. The purpose is to determine through evaluation and studies whether a more collaborative environment can enhance learning in the classroom. The study is being conducted by Albanie T. Bolton, under the direction of Dr. Cheryl D. Seals in the Auburn University Department of Computer Science and Software Engineering. You were selected as a possible participant because you are associated within the STEM field, education and/or the targeted potential users for the tool.

What will be involved if you participate? Your participation in this research study is voluntary. If you decide to participate, the study will be conducted in three parts. If you decide to participate in this research study, you will first be asked to fill out a questionnaire for general background information. The purpose of the questionnaire is to obtain demographic information and some basic information regarding your experience with social computing and online educational tools. The total time commitment for part one is approximately 5 minutes. At the end of the pre-questionnaire you will be presented with a task-list and link to the prototype environment. The time to accomplish part two the task list for guided exploration of how to muse the collaboration tool is approximately 20 minutes. The purpose of the task-list will be to introduce you to the collaborative environment. After completion of the task-list, successfully or unsuccessfully, you will be given a link to the final questionnaire. Your total time commitment to part three will be approximately 10 minutes.

Are there any risks or discomforts? There are no perceived risks associated with this study.
Are there any benefits to yourself or others? Information collected during this study will help us identify how effective and intuitive the environment is to students and teachers.

Will you receive compensation for participating? No compensation will be

given for participating in this survey.

Are there any costs? There are no costs associated with participating.

If you change your mind about participating, you can withdraw at any time during the study by simply closing your browser or returning your handout survey to the supervisor. One you've submitted anonymous data, it cannot be withdrawn since it will be unidentifiable. Your decision about whether or not to participate or to stop participating will not jeopardize your future relations with Auburn University, Department of Computer Science and Software Engineering.

Any data obtained in connection with this study will remain anonymous. All data is stored in a password protected electronic format. To help protect your confidentiality, the surveys will not contain information that will personally identify you. The results of this study will be used for scholarly purposes (publishing in professional journals, theses and dissertations and presentations in professional meetings) only and may be shared with Auburn University representatives

If you have questions about this study, please contact Albanie T. Bolton at the listed email address or number, atb0010@auburn.edu (256)544-2190, or Cheryl D. Seals at sealscd@auburn.edu (334) 844-6319.

If you have questions about your rights as a research participant, you may contact the Auburn University Office of Human Subjects Research or the Institutional Review Board by phone (334) 844-5966 or e-mail at hsubjec@auburn.edu or IRBChair@auburn.edu.

HAVING READ THE INFORMATION ABOVE, YOU MUST DECIDE IF YOU WANT TO PARTICIPATE IN THIS RESEARCH PROJECT. IF YOU DECIDE TO PARTICIPATE, PLEASE CLICK ON THE LINK BELOW.
YOU MAY PRINT A COPY OF THIS LETTER TO KEEP.

Albanie T. Bolton   November 1, 2013
Investigator   Date

_____
Co-Investigator   Date