Culturally Situated Design Tools: Animated Support Tools for Mathematics

by

Albanie Tremaine Bolton

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Computer Science and Software Engineering

Auburn, Alabama
May 14, 2010

Keywords: Culturally Situated Design Tools (CSDTs), educational gaming, ethnomathematics, mathematics, culture, computing

Approved by

Cheryl D. Seals, Chair, Associate Professor of Computer Science and Software Engineering
Richard Chapman, Associate Professor of Computer Science and Software Engineering
Sanjeev Baskiyar, Associate Professor of Computer Science and Software Engineering

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 paper 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 and Technology is essential for a discipline that relies on creativity and diverse perspectives. Creating equity in Computer Science learning opportunities is also a social justice issue. With the educational research community 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 paper and project is to address these issues. The project we focused on out of the CSDT Series is the Break Dancer tool, located 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. In this project, the Break Dancer tool integrates the youth subculture, promotes physical activity, and makes it into an educational math game.

Acknowledgments

I would like to express my thanks to Dr. Cheryl Seals for being a great mentor and advisor for the past two years. Not only this, her guidance has been truly appreciated and helpful along my journey here at Auburn University. I would also like to express my sincere thanks to my committee members Dr. Richard Chapman and Dr. Sanjeev Baskiyar. Your time and participation as a committee member is greatly appreciated in this study. A special thanks to Dr. Eglash, Rahul Potghan and Ravali Gondi. Thanks to Human Computer Interaction Lab and STARS alliance members in all their support during the course of this project.

Finally, I would like to thank my God and mom, Jeannie Bolton, and my immediate family for all their prayers and inspiration throughout these years. Also, very special thanks to my grandmother, Adeline Jackson, for being there and encouraging me while she was here. May your soul be at peace. Without them, I know that none of this would have ever been possible.

Table of Contents

Abstract ii

Acknowledgments iii

List of Tables vii

List of Figures viii

Chapter 1 Introduction 1

Chapter 2 Literature Review 4

2.1 Using CSDTs as Teaching Aides 4

2.2 What Can Educational Games Teach 8

2.3 Computer Games as a Part of Children's Culture 10

2.4 What Makes a Game Good 11

2.5 Issues Designing Cultural Games 12

2.6 Why do CSDTs Work 14

2.7 HCI Studies Culturally Situated Design Tools 15

Chapter 3 CSDT Framework 17

3.1 Framework 17

3.2 Interface Services 17

3.3 GUI Construction and Maintenance 18

3.4 CSDT Interrogation 19

3.5 Serialization 19

Chapter 4 Methodology 20
4.1 The Problem 20
4.2 Solution 21
Chapter 5 Development Tools 23
5.1 TortoiseSVN Subversion Server 23
5.2 Unity 3D 24
5.3 Wii 28
5.4 iPhone 29
5.5 Break Dancer Tutorial Development 30
Chapter 6 Data Analysis 34
6.1 Data from the Usability Testing of Tutorial 34
6.2 Suggestions from Data Collected 37
Chapter 7 Future Work and Conclusion 38
7.1 Future Work 38
7.2 Conclusion 39
References 40
Appendix A Download Instructions for Tortoise Server 43
Appendix B Download Instructions for Unity 3D 45
Appendix C Download Instructions for Netbeans 46
Appendix D Usability Questionnaire 47
Appendix E 3D Geometry Lesson Plan 48
Appendix F Sine Lesson Plan 49
Appendix G 3D Geometry Mini Test 50

Appendix H Sine Mini Test 51

List of Tables

Table 1: Usability Testing Current Tutorial 34

Table 2: Usability Testing Revised Tutorial 35

Table 3: Anova Comparison Results – Terrible & Wonderful 35

Table 4: Anova Comparison Results – Frustrating & Satisfying 36

Table 5: Anova Comparison Results – Difficult & Easy 36

Table 6: Anova Comparison Results – Boring & Fun 37

List of Figures

Figure 1: Break Dancer Tool 3

Figure 2: Cornrow Curves (Transformational Geometry 6

Figure 3: Graffiti Grapher (Cartesian and Polar Coordinates) 6

Figure 4: Virtual Bead Loom (Cartesian Coordinates) 7

Figure 5: Rhythm Wheels (Fractions and Least Common Multiple (LCM)) 8

Figure 6: Unity 3D Graphical User Interface 24

Figure 7: Wii Gaming Console 29

Figure 8: Online Break Dancer Tutorial 30

Figure 9: Revised Break Dancer Tutorial 32

Chapter 1: Introduction

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 actually differ quite a lot across cultures. This is where the term Culturally Situated Design Tools come into play.

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 Culturally Situated Design Tools website provides free standards-based lessons and interactive "applets" that help students and teachers explore the mathematics and knowledge systems using ethnomathematics in areas such as African, African American, Youth Subculture, Native American, and Latino. 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 (current locations include Alaska, California, Idaho, Illinois, Michigan, New York, and Utah), preliminary evaluations indicate statistically significant increase in both math achievement and attitudes toward technology-based careers

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

Ron Eglash sees that there is not enough diversity in these fields and what can individuals do to get the attention of students into these areas. Therefore, he constructed these tools on the basis of researching in the areas of ethnomathematics, mathematics, computing, and educational gaming. The sole inspiration of this project is based on Dr. Ron Eglash's research.

The project we focused on out of the CSDT Series is the Break Dancer tool (Figure 1), located 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, and height. Corresponding to that, each point in the Cartesian space has three coordinates - x, y, and z [2]. In this project, the Break Dancer tool integrates the youth subculture, promotes physical activity, and makes it into an educational game.

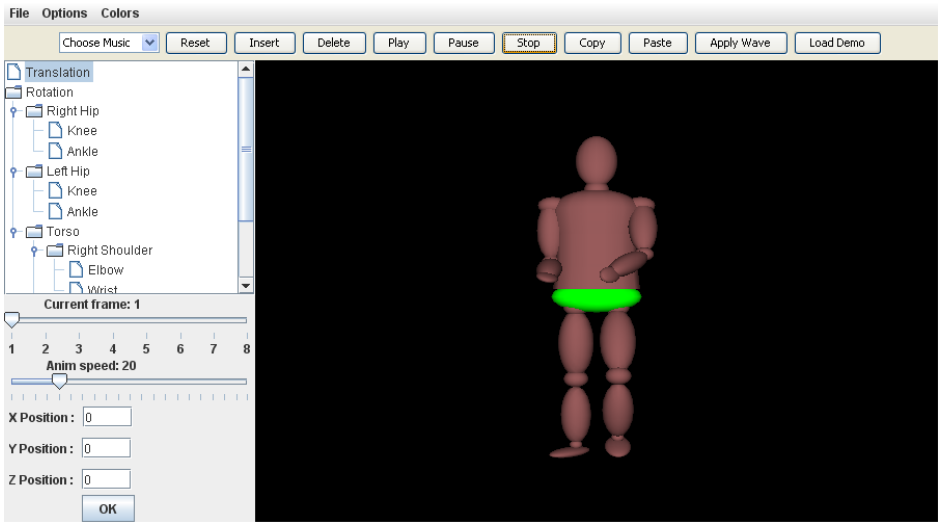


Figure 1: Break Dancer Tool

The project aims to provide a source for increasing teaching aide in schools. In addition, develop a gaming convention that will hold the student's attention in the field of mathematics. This field scares most students and keeps them away from Computing and IT jobs because this is required in the coursework. If we can help instill in students that math can be fun and innovative, this will bring more high school graduates entering college with a higher confidence level in math.

Chapter 2: Literature Review

2.1 Using CSDTs as Teaching Aides

Ways of saying games reflect a cultural value is that games are social contexts for cultural learning. This means that 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 this schema "Games as Cultural Rhetoric" [10]**.**

Games contribute to the development of knowledge by having a positive effect on the atmosphere in the class that produces a better mental attitude towards math in the pupils. 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 the 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 2). In this, they combined both African and African American tools since they went hand in hand.

The one that has deemed favorable is the Cornrow Curves. There are very few students knew that cornrows originated in Africa, so on the design tool website there is a historical background page. This includes images and information covering the indigenous styles and their meaning in Africa. There were 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, i.e., each plait (braid) is given some particular rotation, translation (spacing), scaling, and reflection. Because these transforms are part of the standard curriculum.

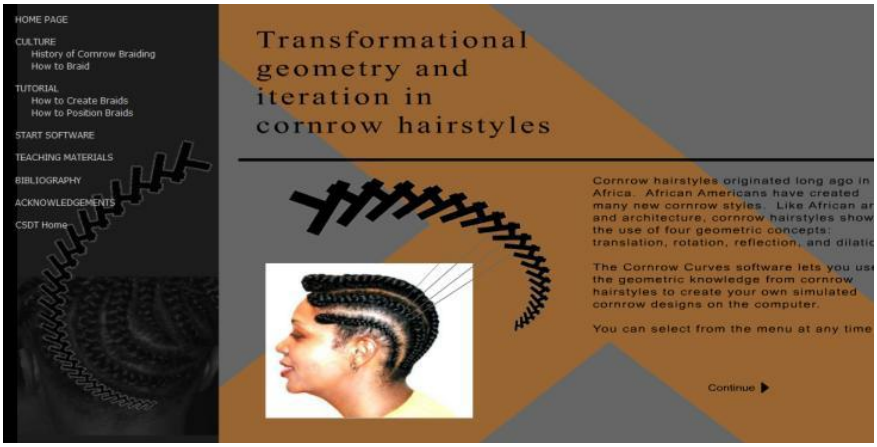


Figure 2: Cornrow Curves (Transformational Geometry)

Youth Subculture

There are two Youth Subculture design tools: Graffiti Grapher and Break Dancer. The Break Dancer tool is not completely finish yet. It is still going through the actual design phase. Graffiti Grapher (Figure 3) 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. With this tool, it brings up the concern will this cause students to avoid "inappropriate" activity, i.e. putting graffiti on actual building, or will it teach and enhance their knowledge of subculture practices. This is the area upon which we will be look at our game design in detail.

Figure 3: 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 4).

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 was 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 FourWinds" concept [7].

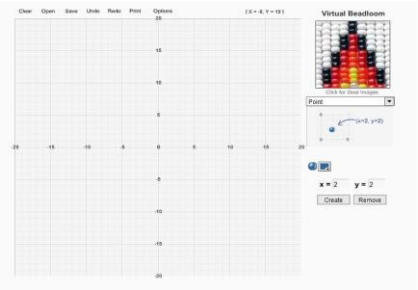


Figure 4: Virtual Bead Loom (Cartesian Coordinates)

Latino Design Tools

There are currently two Latino design tools: Precolumbian 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 5) 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 them an opportunity to link this ethnomathematics to the classroom.

The RW software allows a student to choose from a variety of percussion sounds and to then 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.

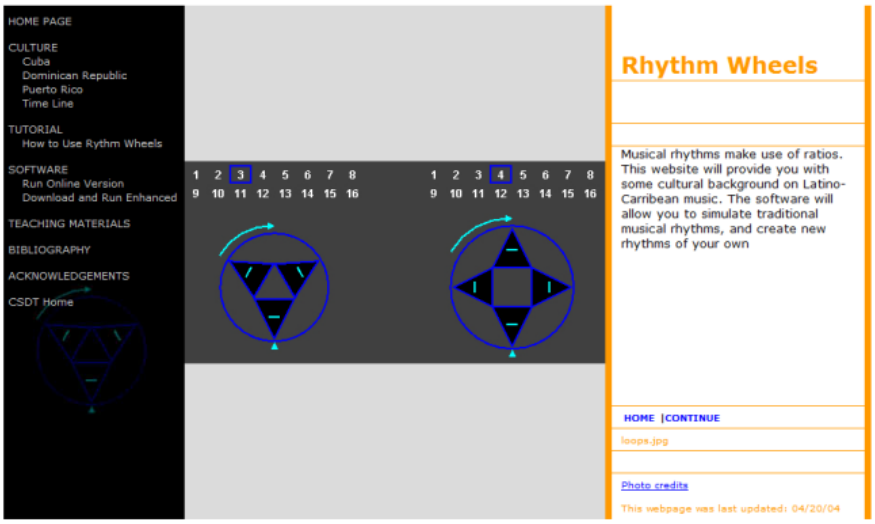


Figure 5: Rhythm Wheels (Fractions and Least Common Multiple (LCM))

2.2 What Can Educational Games Teach

In a Summit on educational gaming, they suggest that many game features, combined and designed effectively into educational gaming, could teach many things in an engaging and motivating manner. Games could be used for the expansion of cognitive abilities, as well as a platform for developing new or practicing existing skills in the context of real world goals, rules, and situations. Games could also be used to 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 structure of the game [5].

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

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 Training

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 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.

2.3 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 which 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 seems to be necessary in order 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. On one hand, this may be regarded as something that should be accepted or even supported, because children want and need to have their own spheres. In the view of this paper, the pedagogical task remains to actively and critically accompany the children's process of growing up and developing their relationship to the cultural world. Moreover, the task remains to secure a pluralitity of resources and challenges they can use to develop their cognitive, social, and physical abilities [24].

2.4 What Makes a Game Good

The perceived value of a game depends greatly on the individual preferences of those who play it according Wolfgang Kramer. When looking at CSDTs, its educational design should consider some of these characteristics. For instance, some players prefer games of luck; others prefer games of tactics; still others enjoy communicating with fellow players. Then there are those who like games based on reaction, manual skills, or memory, etc. The following are some of the 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

2.5 Issues Designing Cultural Games

As with anything, there are some issues when coming to designing cultural games. Most teachers think that it is a waste of time and that it would take 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 issues involved in developing videogames for educational use. The reasons for the 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 new innovations and as do making the necessary organization and instructions 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 that are 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, there are other barriers that also influence second thoughts on designing educational games. First, there is the fact that students are limited to computer availability at their schools. If a student is going to use an educational game, it does require computer access. Secondly, 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.

2.6 Why Do CSDTs Work

For starters, they only have some suggestive results, not wide-scale testing, but assuming that using CSDTs in the classroom actually may raise minority student math achievement and improve their technological aspirations. An explanation is simply that they are using a flexible computational medium, which allows students to pursue inquiry and discovery learning and that the cultural component is irrelevant. The belief is that the flexibility and discovery learning aspects are critical to CSDT success, but flexibility and creativity are just as integral to cultural identity. [11]

There are different conflicts between cultural identity and mathematics in the work of Eglash (2006) which are explained in detail. 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 the identity conflict. Ethnomath examples can decrease the perceived cultural distance between math and cultural identity (whether that 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. Then the myths of genetic determinism are also a conflict. Ethnomathematics offers strong counterevidence to primitivist and ethnocentric portraits of "simple" cultures. In addition to these conflicts between cultural identity and mathematics education, a second component for the poor mathematics performance is in these 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. They suspect that CSDTs ameliorate these barriers. Further studies will be needed to determine which ones or to what degree [4].

2.7 HCI Study Culturally Situated Design Tools

Understanding how users experience technologies necessitates a concern with 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]? This can be shown through Culturally Situated Design Tools.

This inquisitive thinking has led the Human Computer Interaction (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].

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 in order to better understand experience and aesthetics in technology design. Their study will explore the extent to which these disciplines can be used to 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.

Chapter 3: CSDT Framework

3.1 Framework

In general, the CSDT framework is reflective and interface-light. 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 CSDT. The primary services are as follows:

- Interface services
- GUI construction and maintenance
- CSDT interrogation
- Codelet environment mediation (Environment not complete, reason for section missing)
- XML serialization and deserialization [33]

3.2 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. All of these objects are objects that may be created by the user, and possibly, automated. The classes MUST extend the PObject abstract class. The PEngine may specify its own primitive properties, for example, a gravity parameter for CSDTs involving physics simulators, with the @PropertyPrimitive annotation.

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, all datatypes are wrapped by the PVariant class. 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].

3.3 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 the same as anything else that extends JApplet [33].

3.4 CSDT 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].

3.5 Serialization

Serialization is the smallest service provided by the framework. For the sake of compatibility, the serialization format is XML. Unless absolutely 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].

Chapter 4: Methodology

4.1 The 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 wanted to explore out the ones explained was the Break Dancer tool. The tool is designed to teach the sine and sine function. In the version of the tool now, it is hard to understand that concept. 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 the teachers a better way to get the information across to students. We were concerned if the current state was a good way to get the mathematical concepts across, or should we change the interface and try designing it in Unity 3D.

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 Dr. Eglash’s team is using with CSDTs have this feature and he would like to integrate this into this tool. This would aide in more user friendliness for not having to remember textual syntax. If the user is not an expert programmer, having a drag enable script allows them to

interact more with tool and implement what they want to do easily. 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 on what it is trying to express. There is no one really working on how to actually use Break Dancer to teach math. For example, it is not clear to me how to use it to teach 3D coordinates. It is also not clear on how to use the rotations to teach about angles. In addition, not clear on how to use the sine wave option to teach phase, frequency, etc. Dr. Eglash suggested that if our team wanted to examine that from an HCI perspective our investigations would be really valuable, because we would be creating lesson plans, evaluations, etc that are very important in teaching the students the concepts.

Finally, with this tool, there was a need for a better tutorial. Tutorials are sometimes where critical learning takes place. Dr. Eglash also suggested if we wanted to improve on the tutorial, and perhaps introduce some of the math concepts there with little games or even just better diagrams, that would be terrific.

4.2 Solution

To address the previously stated problem, we broke the teams up into two groups. The first group looked at improving the lesson plans and tutorials associated with the tool. We looked at improving the lesson plans and tutorials associated with the tool. In addition, there was extensive research done on the Break Dancer tool and we 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. In the new tutorial, we talked about every topic in extensive detail, also including examples with each explanation. We designed lesson plans that will be beneficial to the teachers and aide them in getting the information across to the

student more efficiently. We collected data analysis on what teachers thought the tool and lesson plans could add to the lessons being taught.

The next item of interest looked at was incorporating the “drag and drop interface”. This was an extensive and long experience. This idea was not implemented and will continue to be worked on for a future project. The problem we encountered is that the current interface only has two panels, and if you want to implement a scripting drag and drop interface, you need three panels. However, to manipulate and change this, we would need permission from the initial designer. This task was a hardship because we could never get in contact with the designer to obtain permission and/or instruction on how to implement the interface. So as a whole, we concluded that this task will be for a future project and that by then the initial designer will have directed us in the path that the group in the path that needs to be taken.

Chapter 5: Development Tools

5.1 Tortoise Subversion Server

TortoiseSVN is an easy to use Revision control / version control / source control software for Windows, is the management of changes to documents, programs, and other information stored as computer files. It is 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 you do not know the GPL too well: you can use TortoiseSVN to develop commercial applications or just use it in your company without any restrictions. Since it is not an integration for a *specific* IDE like Visual Studio, Eclipse or others, you can use it with whatever development tools you like.

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 the windows explorer. The only commands that make sense for the selected file/folder are shown. You will not see any commands that you cannot use in your situation. You can see the status of your files directly in the Windows explorer. The descriptive dialogs, constantly improved due to user feedback. It allows moving files by right dragging them in the windows explorer.

5.2 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, however, 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 server, audio and video.

Figure 6: Unity 3D Graphical User Interface

Fully Integrated Editor

- *Play, pause, and step*: Click the Play button to instantly run your game. 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 instantiated 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 as pie. 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 effect. 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. See the bottom of this page for details.
- *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. It's that simple. See the bottom of this page for a full list of supported formats.
- *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, be relieved as 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 rock-solid support for almost all hardware/software combinations, in both DirectX and OpenGL. Unity has undergone extensive compatibility testing, which means you don't need to put an ounce of effort into making sure your customers can run your games. Instead, you can expect everything to work well the first time. This helps you reduce your QA and customer support needs, and allows you to focus on making the game great instead of making the game work.
- *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 a 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 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 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.

Today 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].

5.3Wii

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 7: Wii Gaming Console

In order to evaluate or license Unity 3D engine for use on the Wii console developers must meet the following requirements:

- You must own Unity Pro 2.x
- You 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 your intended distribution. The retail price is $30,000 per title and the for WiiWare it is $15,000 per title [28].

5.4 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 (also 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)—using 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 you can choose to follow. The first is to pick up the iPhone SDK and start learning to code in Objective C. The other is to use a commercial engine and build your 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. You can view all the license prices and a comparison of features at the Unity website [28, 31].

5.5 Break Dancer Tutorial Development

The tutorial development was done to improve the current tutorial at hand. The current tutorial (Figure 8) needed more information added to get a better understanding for what was being taught.

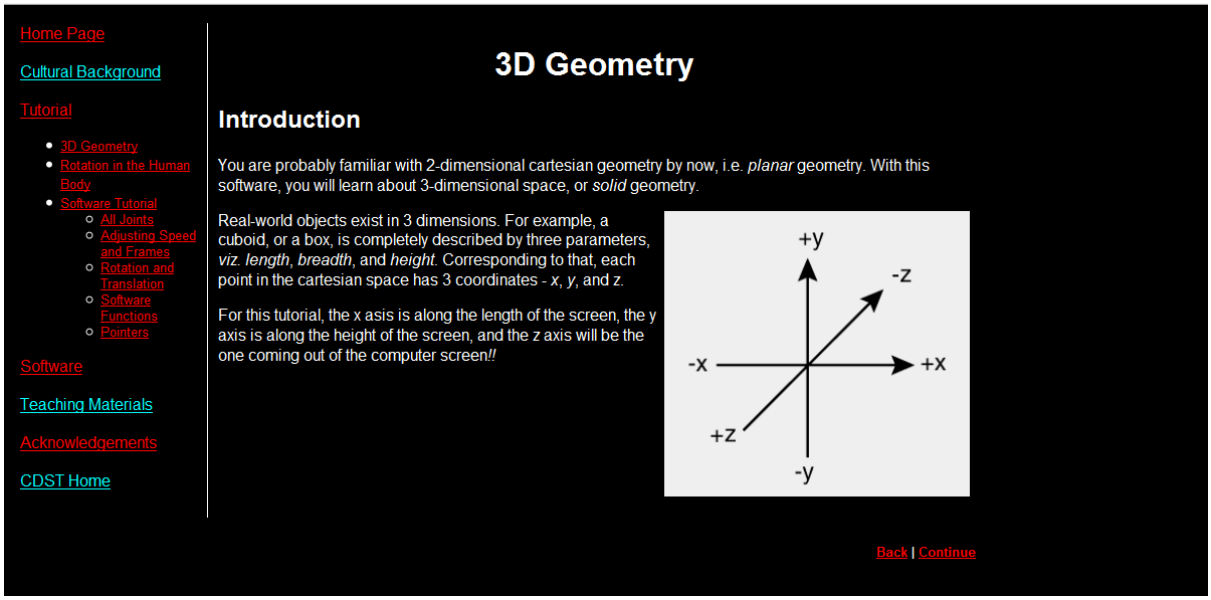


Figure 8: Online Break Dancer Tutorial

The current tutorial is in 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 doing the revised tutorial, we did extensive research on the Break Dancer tool and we 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 that were discussed in the tutorial.

In the new tutorial, we talked about every topic in extensive detail, also including examples with each explanation (Figure 9).

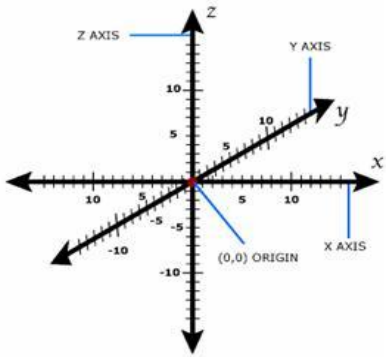
Introduction to 3D Geometry & Space

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.

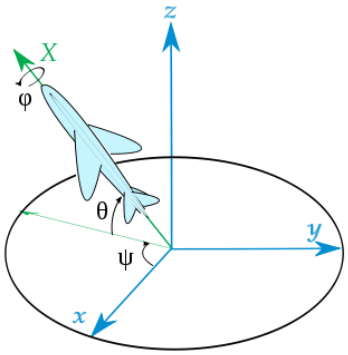


Figure 9: Revised Break Dancer Tutorial

The revised tutorial outline was changed to the following:

- 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 E & F) and mini tests (Appendix G & H). We did data analysis on the tutorial, lesson plans and mini tests. The results are expressed in the next chapter.

Chapter 6: Data Analysis

6.1 Data from the Usability Testing of Tutorial

The questionnaire used for the testing of the tutorial contained four questions that focused on the teachability and learnability.

There were ten participants who were teachers and not previously exposed to the CSDTs. For most cases, we would normally have about 30 participants for a full analysis. Due to the response level, we went with an approach of opportunistic sampling. It consists of taking the sample from people who are available at the time the study is carried out and fit the criteria your are looking for. We felt that it was adequate in the sense that it was the population we wanted and the participants supplied very detailed information. The result showed that the participants particularly like 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 1: Usability Testing Current Tutorial

Table 1 represents data taken on the current tutorial that is on the site. Overall, people did not like the tutorial that was initially used and commented that it need drastic improvement. The participants noted that the current tutorial did not convey enough information to the user and would be complex to teach lesson from what was in the tutorial.

	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 2: 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 each level listed in tables 3, 4, 5, and 6. Our initial null hypothesis was done at a level of significance of 3.0. By looking at the P-Value we see that there is are differences between the two results.

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Column 1 (table 1)	10	35	3.5	0.277778
Column 2 (table 2)	10	45	4.5	0.277778

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5	1	5	18	0.00049	4.413873
Within Groups	5	18	0.277778			
Total	10	19				

Table 3: Anova Comparison Results – Terrible & Wonderful

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Column 1 (table 1)	10	38	3.8	0.177778
Column 2 (table 2)	10	48	4.8	0.177778

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5	1	5	28.125	4.84E-05	4.413873
Within Groups	3.2	18	0.177778			
Total	8.2	19				

Table 4: Anova Comparison Results – Frustrating & Satisfying

Anova: Single Factor

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Column 1(table 1)	10	35	3.5	0.277778
Column 2 (table 2)	10	45	4.5	0.277778

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	5	1	5	18	0.00049	4.413873
Within Groups	5	18	0.277778			
Total	10	19				

Table 5: Anova Comparison Results – Frustrating & Satisfying

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 6: Anova Comparison Results – Boring & Fun

From the results with see that the participants found that 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.

6.2 Suggestions from Data Collected

The main suggestions that were collected from the data was to incorporate more lesson plans and to include more tests. In addition, one suggested that the lesson should be taught in a 3D lab to allow for more of an appropriate atmosphere for the Break Dancer tool. The participants did think that what we have now is very nice and would recommend that any teacher use this tool in their classroom. The participants found the Break Dancer tool fun and innovative. In addition, a great way to grab the students attention and teach a math lesson at the same time.

Chapter 7: Future Work & Conclusion

7.1 Future Work

The future work for this project is basically focused on implementing the software in Unity 3D for future use with Wii. That is our final ultimate goal for this project. There is work that 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 course and encourage them to pursue these careers in college. By measuring student grades in the math pre-class and post-class, this could determine if the game engaged students sufficiently to improve their performance in later classes. We would also like to do a more rigorous evaluation over a six to nine week time span with some validated instrument to look more into our data analysis portion. We propose that we do a mini ethnography and sit in the classroom with the teacher to see exactly how effective the tool can actually be in a classroom setting.

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 will continue to be worked on for a future project. As stated earlier, the problem encountered is that the current interface only has two panels, and if you want to implement a scripting drag and drop interface, you need three panels. Therefore, we concluded that this task will be for a future project and that by then the initial designer will have directed us in the path that the group in the path that needs to be taken.

Other possible future work could be to explore other opportunities for game enhanced versions of other math courses, with the intention to motivate and improve the education of possible undergraduate students.

7.2 Conclusion

We implemented a better tutorial, lesson plan, and mini tests to help teach the lesson better and grab the user's attention. This factor will help to 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. Computer games are very popular among children and adolescents. In this respect, 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, which makes it easy for teachers to incorporate into their class. At the same time, their ability 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.

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 sex. 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.

References

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 Analysis. 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

Appendix A

Downloading and Installation of Tortoise SVN Subversion Server in Windows

The installation of the server allows for users to modify the tool and implement any changes that may be needed. After you install the SVN client (Tortoise) it enables a set of items in the right click menu of file explorer and file folders to check out the directory.

1. Go to http://tortoisesvn.net/downloads and get the 32 bit or 64 installer for your windows machine. Make sure to check your system properties or the server will not work with the wrong one installed.
2. Install tortoise
3. Create a folder on the desktop, i.e. CSDTSVN
4. Right click on the folder, you should see some new options than the normal options you usually see, i.e. Tortoise SVN or SVN Checkout. (If you do not see new options, as shown in the figure, you may to reinstall or you have downloaded the wrong version)
5. Clicking on TortoiseSVN you see:
    a. Repro browser – great way to look through the folders. Can also "check out" a copy for revision from here.
    b. You can also use SVN checkout.
    c. Selecting either: You'll get a dialog box, put https://subversion.hss.rpi.edu/svn/csdt_src in the field named "url of repository".
    d. Checkout directory will be filled in with the folder you right clicked on and Head Revision will be the default (most recent version) which is what you want
6. Press ok and the whole version will be downloaded into the folder.

Follow-up Items

- To add new file or folder: right click on file or folder, select TortoiseSVN, select add.
- To “commit” either the added file/folder or a revised file/folder to the archive: right click on it and select SVN Commit.
- Update will come from the server to your hard drive’s respository, so you can get the latest version.
- To transfer to the production server (the one user’s actually see) use export (or you can just copy and paste, but expert let SVN keep track).
- If Subversion says it is 'locked', it usually means that someone else has checked out files from subversion but has not checked them back in yet. In order to prevent one person from making changes, submitting them, and having them overwritten by someone else who commits to the same files, subversion locks them. Is someone else working on the same files? If so, ask them to commit the files. If that doesn't work, try getting them from subversion again and see if that will fix it.

Appendix B

Downloading and Installing Unity 3D for Windows

1. Go to http://unity3d.com/unity/download/ and make sure you are downloading for the Windows machine.
2. Click download
3. It will ask to save the file, click on save file.
4. When finished loading, then double click to open the download and it will begin running.
5. Follow each step that is given to you, leave everything set to default.
6. When finished, open the Unity 3D application, it will ask you to register. Make sure to register, you can either register for the full version or 30 Day trial.

Appendix C

Downloading and Installing Netbeans 6.8 for Windows

1. Go to http://netbeans.org/ and click on download Netbeans IDE 6.8
2. It will ask to save the file, click on save file.
3. When finished loading, then double click to open the download and it will begin running.
4. Follow each step that is given to you, leave everything set to default.

Appendix D

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 E

3D Geometry

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 F

3D Geometry & Sine Function

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:

$$f(x) = a*\sin(bx + c) + d$$
$$g(x) = a*\sin(bx) + d$$

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

Appendix G

Break Dancer Software Exam

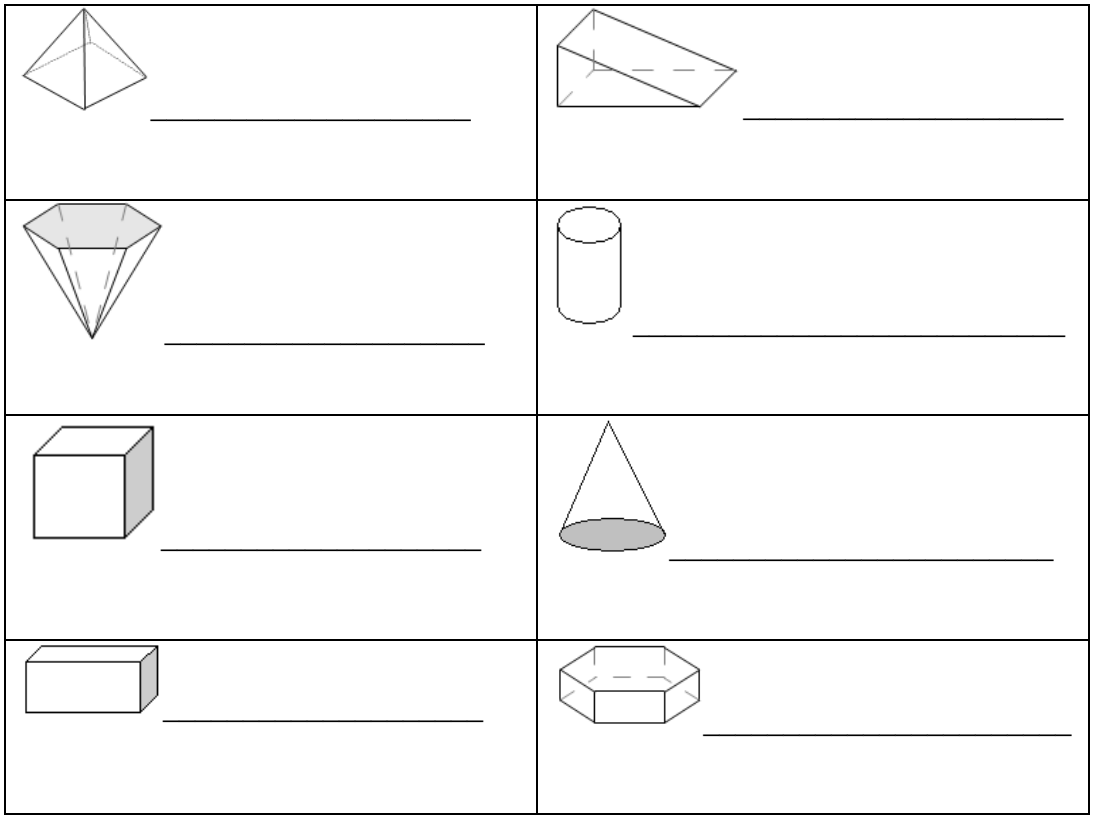
Name: _____

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 H

Break Dancer Software Exam

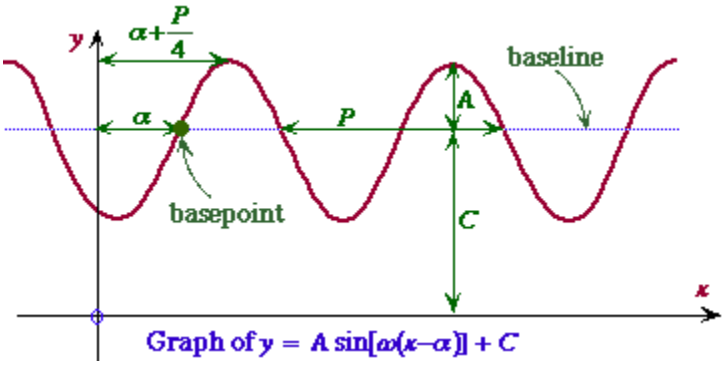
Name: _____

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? _____

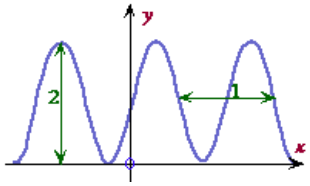
2. **General Sine Curve**



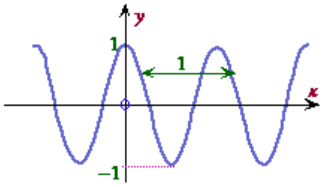
A is called the _____ (the height of each peak above the baseline)
C is the _____(height of the baseline)
P is the _____ or _____ (the length of each cycle)
ω is the _____, given by $\omega = 2\pi/P$
$\alpha$is the _____(the horizontal offset of the basepoint; where the curve crosses the baseline as it ascends)

Model each of the following curves with a sine function.

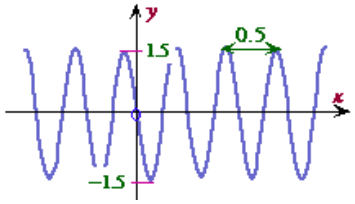
3.



4.



5.



6.