

Applications of Multitouch & Gaming Technology for the Classroom

W. Muto, J. Dobies & P. Diefenbach¹

¹Antoinette Westphal College of Media Arts and Design, Drexel University, United States

Abstract

Multi-point touch screens have enjoyed recent popularity due to their natural tendency to create highly intuitive and user-friendly systems for even the novice user. We believe that multitouch is well-suited for educational purposes, since it engages users through an invisible interface and natural, gestural interaction, as well as promoting collaborative learning through equal access, as opposed to individual or "driver and co-pilot" learning at a traditional computer workstation. In this paper, we adopt this powerful interface and combine it with 3D simulation and gaming technology to create a novel teaching tool, incorporating digital learning content co-developed with educators and providing access for administration and student assessment.

Categories and Subject Descriptors (according to ACM CCS): K.3.1 [Computers and Education]: Computer Uses in Education—Computer-assisted instruction

1. Introduction

The educational merits of computer technology have long been understood, yet the narrower application of computer gaming and simulation techniques has largely remained outside the classroom, confined mainly to the military and private sectors. Computer games and simulations have been championed by educational theorists; carefully designed games, combined with modern processing power, can create learning environments that foster critical thinking and provide instant and constant feedback to the user, a necessity for young minds. Early experiments and prototypes have supported these claims. We believe that these successes, combined with novel input devices, can create powerful and engaging learning tools. The purpose of this project is to prototype a modular, expandable, and low-cost educational multitouch system that could be adopted for use in a classroom.

The remainder of the paper is structured as follows: Section 2 discusses previous research in gesture-based computing, multitouch interfaces, and simulation gaming as an educational tool. Section 3 describes our educational approach and system design, and how they drove our first prototype educational content. A case study describing how existing 3D simulation content was adapted for our system is illustrated in Section 4. Section 5 discusses our conclusions.

2. Related Work

2.1. Gesture-based Computing

Gesture-based computing has been explored as early as 1964; the Sketchpad Graphical Communication System utilized a light pen to send position information, using physical switches and knobs to change modes and control rotation and magnification [Sut64]. Since Sketchpad, many contributions have been made to gesture-based interfaces. Minsky's prototype, developed in 1984, used single-finger input (not a pen) to manipulate objects on screen, and was designed mainly for systems used by young children [Min84]. Stretch, X-Menu, and Slider gestural techniques have particular allure for novice users [BWB06].

Gesture recognition gaming is an increasingly popular way to develop for today's gaming industry, as can be seen with the recent success of Nintendo's two latest systems, the Nintendo DS and Wii. Both of these systems allow for the users to interact with the game through novel forms of gesture recognition.

2.2. Multitouch Interfaces

Multi-point touch systems have been evolving for several decades. [KGH85] demonstrated multiple-finger input in

VIDEOPLACE. The DiamondTouch [DL01] and SmartSkin [Rek02] tables both support input from multiple users, but that input could easily become ambiguous. The iPhone's gesture recognition system allows users to perform complex tasks such as typing and navigating the web on a small device without the use of a pen or keypad. Similar gestural recognition has been added to the trackpad in Apple's MacBook laptop line. In addition, a great deal of research has focused on developing large-scale multitouch displays. Many current implementations are based on frustrated total internal reflection (FTIR) [Han05]. Recently, systems such as Microsoft's Surface and Nortd's Touchkit are aiming at bringing affordable systems to the mass market.

2.3. Educational Gaming and Simulations

The effectiveness of simulations as a component of training was first recognized by military organizations. Origins of military simulations can be traced back nearly five millennia, however, it can be argued that not until the turn of the eighteenth century, when "war was becoming less of an art and more of a science" was gaming largely accepted as a viable option in military training [TW72]. Businesses recognized the success of this approach, and by the middle of the 20th century, began to adopt serious gaming into their training.

Despite success in the private and military sectors, gaming has not been similarly embraced by educators. Simulations did not appear in the classroom for nearly a decade after adoption by businesses, and early attempts, met initially with great initial enthusiasm but little critical evaluation, eventually led to failure [TW72]. In addition, technological limitations of the time made complex simulations somewhat unavailable to casual users. However, current computing power no longer presents the restrictions that previously minimized usefulness, and a great deal of educational theory is revisiting gaming as a viable learning tool. [Pre01] maintains that simulation and gaming engage students by "putting learning into context" and catering to the visual-spatial skills and inductive discovery methods of young minds.

Recent applications have shown great promise. The "My-Pet-Our-Pet" system [CCDC05] harnesses emotional attachment to strengthen motivation. This system simulates an animal companion, similar to a Tamagochi. To sustain the pet, students must obtain resources by participating in learning activities. Chen et. al.'s system, in addition to creating a dynamic learning experience which engages students, also encourages both self-reflection and group responsibility when caring for individual and team pets. [Bru02] recreated the kelp forest exhibit at the Monterey Bay Aquarium using VRML (Virtual Reality Modeling Language). This simulation allowed users to explore the exhibit environment through various cameras. This project was showcased to the public and was met with enthusiasm by participants of varying ages. In a joint initiative between educators and tech-

nologists, [SBGH04] developed SuperCharged!, a 3D game designed to teach the abstract concept of electromagnetism. Squire et. al. found that improvements could be made to educational gaming by avoiding level-to-level game design, effectively redirecting competitive aspects of play from "completion" of the game to mastery through exploration. [CZ06] continued study of simulation in the classroom by combining collaborative learning with a computer simulation. Chen and Zhang found that peer collaboration had notable positive effects during educational exploratory game play.

The successes of these prior applications, as well as possible improvements, informed our design goals. From this early work, we concluded that a game with optimal educational value must be an exploratory environment lacking standard level progression, should be designed with the teacher in mind as a supplement lesson plan and not a replacement, should emphasize emotional and collaborative involvement, and should not be developed in the tradition of a pencil and paper test but should instead encourage critical thinking and discovery.

2.4. Planet Diggum

Our early work with multitouch gaming provided an important framework to build upon [DMC*07]. Planet Diggum is a kiosk god-game, where users interact with creatures, called "diggums", using gross gestures.

The application was also developed in X3D, and at its completion, we had made several important extensions to the engine, including a set of X3D nodes, which would allow communication with the multitouch hardware. This extension was improved upon, and will be discussed further in the following section.



Figure 1: Planet Diggum

3. Design

3.1. Overview

From our educational research, we determined that the following goals must be met in order for our system to be successful:

- Encourage learning through discovery and play
- Increase learner motivation
- Encourage collaborative, shoulder-to-shoulder work
- Encourage learning outside the classroom
- Discourage "play-to-complete" and promote exploration
- Allow for easy expansion and promote system longevity
- Support and integrate with a standards-based lesson
- Provide open access to data
- Provide an administration interface for the facilitator

These goals drove both the educational design (content) and the system design (architecture).

3.2. Educational Design

The first prototype educational content developed for our system was a virtual terrarium. The simulation was designed as a sandbox environment, which lacks formal separation of levels or an endgame scenario and therefore discourages competition from overshadowing educational content. Students, working in groups, interact with a macroscopic world where they seed different plants and observe changes in the surrounding insect population. In this way, through the simulation, students will be able to observe and interact with an environment that is not readily available in the classroom, promoting understanding the complex relationships in an ecosystem through experimentation, and not through memorization and recitation. They can also track the progress of insects in the simulation by "capturing and observing" them, fostering a connection with the virtual organisms and therefore increasing motivation.

For the educational simulation to be effective it must support the lesson and its usage must be moderated by the educator. Therefore, integration into a practical lesson plan was a necessity. While developing the digital content, we worked with educators to develop a lesson plan. Our virtual terrarium would support a lesson on ecosystems, based on the following state standards:

- Demonstrate the dependency between living components and nonliving components in the ecosystem (PA Environment & Ecology Standard 4.6.7.A)
- Explain how change in an ecosystem relates to humans (PA Environment & Ecology Standard 4.6.7.C)
- Compare and contrast different biomes (PA Environment & Ecology Standard 4.6.7.A)
- Identify the environmental impact that waste has on the environment (PA Science & Technology Standard 4.6.7.A)
- Apply models to predict specific results and observations (PA Science & Technology Standard 4.6.7.B)
- Explain the complex, interactive relationships among members of an ecosystem PA Environment & Ecology Standard 4.3.7.B)

Ideally, before using the table, the educator will introduce

ecosystems and the concept of "balance", guiding a discussion with the class regarding what they already know and what they want to learn regarding the topic. The educator will present example ecosystems, and guide student research of remote ecosystems. Once the students are introduced to the simulation, the educator will ask students to make predictions about their activities in the simulation. In addition, the educator will strengthen the activity by reiterating learning goals during exploration, providing questions for critical thinking and initiating reflection after the activity is complete.

Since the application is designed to support four students comfortably, students will rotate through other learning stations after their session on the multitouch display. Supplemental activities, with traditional computer terminals and without, will emphasize the learning goals and standards. Through the web portal, students at individual computers will be able to continue their research of their collections under the guidance of the teacher. Using print and web resources, students will create charts of observed food chains and relate their observations to broader topics such as renewable resources and waste management. Groups will then reconvene to share their findings with the rest of the class, and finally, reflect on their observations.

3.3. System Design

In order to meet the aforementioned educational goals, our system design had to be taken into careful consideration. For our system to be feasible for a classroom environment, it had to be built on affordable technologies. During development, no large-scale multitouch platform was commercially available. Therefore, a low-cost prototype was developed in a parallel project by Dr. Youngmoo Kim's MET lab. The display utilizes Han's FTIR method [Han05], in which infrared light transferred through a polished surface is scattered upon contact with the display. The scattered light is detected by a camera placed behind the display, and translated into point data using blob-detection software. Images are displayed using rear-projection on a diffuse screen under the touch surface. This method is extremely scalable and relatively inexpensive; the first-generation prototype with a display area measuring 27 by 36 inches was constructed for under \$4000.00.

X3D was chosen as our rendering engine allowed full access to source code through a commercial engine, so core functionality could be added as needed. X3D was also chosen for its flexibility; it is capable of rendering a wide range of multimedia formats, and is easily expanded. MySQL was chosen as a database solution because of its speed, flexibility, ease of use, and ability to access to data from multiple clients. Finally, a web interface was built using PHP, a powerful server-side language, running on an Apache HTTP server. Not only did utilization of open source and open stan-

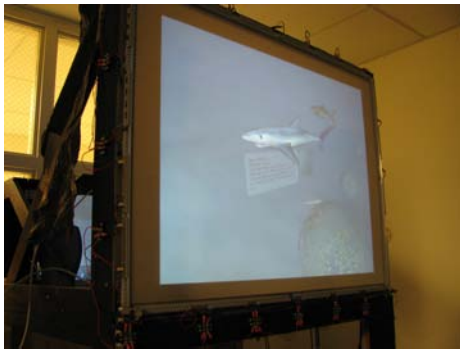


Figure 2: Prototype Multitouch Display

standard technologies keep our production costs to a minimum, but also helps ensure system longevity.

When a user touches the FTIR display, finger positions captured by a FireWire camera are tracked by a server. These coordinates, as well as any recognized gestures, are sent via TCP/IP to a client. In our case, the client is an X3D environment. In order for the multitouch server to communicate with X3D, we created a set of custom X3D nodes, which establishes a socket connection with the server, reads the input, and overrides the standard X3D input handler to loop each point received by the client node, treating each as if it was standard mouse input. The definitions for these nodes are:

```
MTClient {
  SFInt32  []  port  0
  SFString []  server "localhost"
}

MTTouchSensor {
  SFVec3f  [out] hitNormal_changed
  SFVec3f  [out] hitPoint_changed
  SFVec2f  [out] hitTexCoord_changed
  SFBool   [out] isActive
  SFBool   [out] isOver
  SFTime   [out] touchTime
  SFInt32  [out] gestureId
}
```

User actions and simulation data are stored in a MySQL database. Multiple users can simultaneously change the database which allows for a large number of editors to work on the project without causing issue. For this to be possible, another custom X3D node was created, which executes SQL (Structured Query Language) commands on a database. The definition of the node follows:

```
Mysql4vrml {
  SFString  []  server  "localhost"
  SFString  []  username ""
  SFString  []  password ""
  SFInt32   []  port    0
  SFString  []  database ""
  SFString  [in] query
```

```
MFString   [out] result
}
```

Once the node has been instantiated, query strings can be assembled and sent to the node. If a result is returned, it is stored in a string array. This way, simulation data can easily be accessed via a web browser through a server-side language such as PHP.

A web portal was created for several reasons. First, a web interface allows an educator to access multitouch hardware functionality and administer sessions on the hardware easily. Second, it helps to engage learners by giving them greater access to the learning activity when they are outside of the classroom. Through a web site, all relevant information from the database can be utilized, allowing students can track their progress, analyze data from a particular session on the multitouch hardware, or complete secondary learning activities.



Figure 3: Demonstration student portal

Since learning applications can vary greatly in requirements and scope, the web framework had to be extremely flexible. This was accomplished through creation of a module system. Modules are written in PHP and placed in a directory with a simple configuration file. To install a module, it is simply copied to a specified location in the filesystem, where it is loaded at runtime. Once loaded, the module can be activated for student access through an administration panel.

4. Case Study: Virtual Aquarium

4.1. Overview

To demonstrate the flexibility of this new framework, we also adapted existing learning content as a multitouch learning module. This content is an aquarium simulation written in X3D and developed as an information and exploration program that recreates a set of natural behaviors exhibited in wild animals while allowing users to explore and learn more about the animals. The simulation was built in a modular fashion that allows for the creation of various scenarios. This serves two purposes: allowing users to build their own ecosystem to learn about the necessary balance for maintain

a self-sustaining ecosystem and allowing a curator to tailor the simulation to a particular real world environment. As a learning tool, this simulation allows the user to explore and learn at his or her own rate, rather than be forced into a pre-defined format. Adapting this program to multitouch would allow multiple users to explore the environment, shoulder-to-shoulder, ideal for classroom use. Users are able to explore the environment and observe its inhabitants as they interact with one another. Behaviors include feeding, dying, spawning, aging and schooling. Learning takes place by simply clicking on a fish to find out more about the particular species.

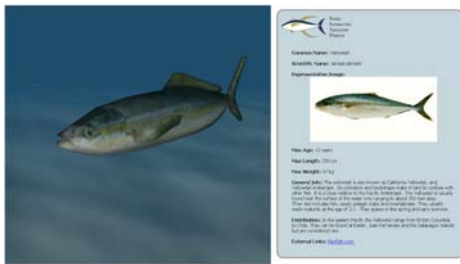


Figure 4: X3D Aquarium Simulation

4.2. Multitouch Functionality

To integrate this simulation, we first had to add multitouch capability. To do so, we added our custom MTClient node so the X3D could retrieve input data from our blob tracking software. Next, we replaced the standard X3D touch sensors with custom touch sensors, which can recognize multiple input points. Finally, in order for the simulation to communicate with a web module, we added our database connection node.

Once the required nodes were added to the X3D file, we needed to define gestures that users could utilize to interact with the simulation. We defined the following interaction goals which would have an accompanying gesture vocabulary: calling focus to a particular specimen, controlling the camera, opening and closing an information dialog for a particular specimen, and adding new fish to the simulation at runtime.

Once our gesture recognition system was able to discern these inputs, we inserted ECMAScript logic to translate this data to actions within the simulation and connected our custom touchsensors to the ECMAScript. The multitouch client assigns each gesture it recognizes a unique ID, which is then passed to the ECMAScript.

```
DEF TOUCH MTPlaneSensor { enabled TRUE }
DEF movelogic Script {
  eventIn SFInt32 gestureId
  url"ECMAScript:
    function gestureId(val, t) {
```

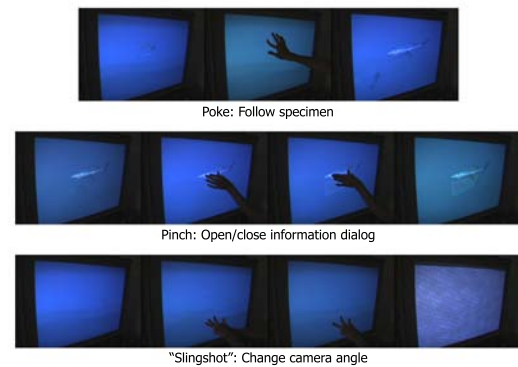


Figure 5: Gesture vocabulary

```
    switch (val) {
      // handle based on gesture id
    }
  }
}
ROUTE TOUCH.gestureId TO movelogic.gestureId
```

4.3. Database Connectivity

Any data we wanted exposed to the web module can be stored in the database. We did this for significant simulation events such as specimen birth and death. After adding the X3D database communication node, the interface was exposed to all objects that required database access. Then, for significant events in the simulation, a query was built and sent to the MySQL node. The following code demonstrates how such a query is constructed.

```
DEF MysqlNode Mysql4vrm1 {}
DEF fishlogic Script {
  url"ECMAScript:
    function death(val, t) {
      var string = '
      INSERT INTO sim_history
      VALUES (NULL, \" + globalID +
      '\', \'3\', NULL, \"+ t + '\')';
      query_changed = string;
    }
  }
}
ROUTE BlueShark1.query_changed TO
MysqlNode.set_query
ROUTE MysqlNode.result_changed TO
BlueShark1.result_changed
```

4.4. Web Interface

Finally, a module was built for the web framework. The module displayed a timeline of the simulation's history, so students could track a particular fish in the simulation when

they were apart from the multitouch hardware. A small PHP page was created to perform this task, and then placed in a folder with a configuration file. Once copied into the correct location, the web framework loaded the module, which then could be viewed by students.

5. Conclusions

Many possibilities exist for this system. Network capability could feed real-world, real-time data to inform the simulations. Using available application programming interfaces (APIs), data generated from student use of learning modules can be standardized to conform to larger learning management systems such as SCORM. As demonstrated with the fish habitat simulation, new environments and other educational content can be easily added through X3D with little modification.

A limitation for the platform lies in availability of hardware. Large-scale multitouch displays are only starting to become commercially available. While inexpensive, our multitouch display had to be custom-built, and since it was a prototype, needed a great deal of structural improvements before deployment in an actual classroom. Another limitation of the system relies on streamlining all processes involved in the framework. Our system is comprised of many different components working together, including a web server, gesture recognition server, and 3D renderer, as well as required hardware, such as a digital camera. The system can be greatly improved by creating an installer and single executable that starts and manages all required processes.

We believe that the invisible interface and multi-user tendency of the multitouch displays will encourage student engagement and promote a collaborative environment which is more conducive to learning. While deployment in an actual classroom environment was outside the scope of this project, we believe we have developed an expandable, collaborative platform ripe for further development, built on low-cost technology, informed by education theory as well as the successes and failures of educational gaming. Developing the multitouch display as a mainstay in the classroom will require additional feedback and collaboration with educators and students to expand the educational content and refine the interface.

References

- [Bru02] BRUTZMAN D.: Teaching 3d modeling and simulation: virtual kelp forest case study. In *Proc. of the seventh international conference on 3D Web technology* (2002), pp. 93–101.
- [BWB06] BENKO H., WILSON A. D., BAUDISCH P.: Precise selection techniques for multi-touch screens. In *Proc. of the SIGCHI conference on Human Factors in computing systems* (2006), pp. 1263–1272.
- [CCDC05] CHEN Z.-H., CHOU C.-Y., DENG Y.-C., CHAN T.-W.: Animal companions as motivators for teammates helping each other learn. In *Proc. of th 2005 conference on Computer support for collaborative learning* (2005), pp. 43–47.
- [CZ06] CHEN Q., ZHANG J.: *Collaborative Discovery Learning Based on Computer Simulation*. L. Erlbaum Associates, 2006.
- [DL01] DIETZ P., LEIGH D.: Diamondtouch: a multi-user touch technology. In *Proc. of the 14th annual ACM symposium on User interface software and technology* (2001), pp. 219–226.
- [DMC*07] DIEFENBACH P., MUTO W., COLUCCI S., CUNANAN C., DOBIES J., GROW J., SMITH M.: Planet diggum: Multi-point touch and gesture control of gaming environments. In *Proc. of Distributed Multimedia Systems* (2007).
- [Han05] HAN J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. of the 18th annual ACM symposium on User interface software and technology* (2005), pp. 115–118.
- [KGH85] KRUEGER M. W., GIONFRIDDO T., HINRICHSEN K.: Videoplace - an artificial reality. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1985), pp. 35–40.
- [Min84] MINSKY M. R.: Manipulating simulated objects with real-world gestures using a force and position sensitive screen. In *Proc. of the 11th annual conference on Computer graphics and interactive techniques* (1984), pp. 195–203.
- [Pre01] PRENSKY M.: *Digital Game-Based Learning*. McGraw-Hill, 2001.
- [Rek02] REKIMOTO J.: Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *Proc. of the SIGCHI conference on Human factors in computing systems* (2002), pp. 113–120.
- [SBGH04] SQUIRE K., BARNETT M., GRANT J. M., HIGGINBOTHAM T.: Electromagnetism supercharged!: learning physics with digital simulation games. In *Proc. of the 6th international conference on Learning sciences* (2004), pp. 513–520.
- [Sut64] SUTHERLAND I. E.: Sketch pad a man-machine graphical communication system. In *Proc. of the SHARE design automation workshop* (1964), pp. 6.329–6.346.
- [TW72] TAYLOR J. G., WALFORD R.: *Simulation in the classroom*. Penguin Books, 1972.