AC 2010-1489: REVAMP COMPUTER EDUCATION WITH MULTIMEDIA AND GAME TECHNOLOGIES

Suxia Cui, Prairie View A&M University

Dr. Suxia Cui is an assistant professor in the department of Engineering Technology at Prairie View A&M University. She received her BS and MS in Electrical Engineering from Beijing Polytechnic University in 1997 and 1999 respectively. She received her Ph.D. in Computer Engineering from Mississippi State University in 2003. Her research interests include digital signal processing, data compression, image processing, video coding, and wavelets.

Yonghui Wang, Prairie View A&M University

Dr. Yonghui Wang received the B.S. degree in technical physics from Xidian University, Xi'an, China, in 1993, the M.S. degree in electrical engineering from Beijing Polytechnic University, Beijing, China, in 1999, and the Ph.D. degree in computer engineering from Mississippi State University, Starkville, MS, in 2003. From 1993 to 1996, he was an Engineer with the 41st Electrical Research Institute, Bengbu, China. From 2000 to 2003, he was a research assistant with the Visualization, Analysis, and Imaging Laboratory (VAIL), the GeoResources Institute (GRI), Mississippi State. He is currently an Assistant Professor with the Department of Engineering Technology, Prairie View A&M University, Prairie View, TX. His research interests include image and signal processing, and image and video coding.

Siew Koay, Prairie View A&M University

Dr. S. T. Koay is currently professor at Department of Electrical and Computer Engineering, Prairie View A&M University. His current research interests are analytical modeling for spatial tracking of mobile objects for real time applications, statistical signal processing, modeling and simulation.

Yonggao Yang, Prairie View A&M University

Yonggao Yang is an associate professor in the Department of Computer Science at Prairie View A&M University in Texas. He joined the CS Department at PVAMU as an Assistant Professor in August 2002 immediately upon receiving a Ph.D. degree in Information Technology & Computer Science from George Mason University. His research interests include computer graphics, scientific visualization and virtual reality, computer animation, distributed virtual environments, and computer networks.

Revamp Computer Education with Multimedia and Game Technologies

Abstract

The booming computer technology has changed our daily lives dramatically in the past decade. We witnessed the microprocessor changing from 8-bit to 64-bit and the display from monochrome monitor to millions of colors LCD screen. However, the teaching of college level computer classes does not catch up with the pace. This widens the gap between advanced computing technology and the college computer education. Especially facing the fact that today's youngsters are grown up in a computerized multimedia world – they are enthusiasts of internet, video games, ipod, MySpace, face book, and etc. even before they enter college. Having known lots of popular fancy applications using computer techniques, their expectation of their first computer class is not the same as students of twenty years ago. But the struggling for their first computing class in college resulted in low retention rate reported by many institutions.

To address this issue, a group of faculty members from computing discipline investigated to teach the traditional topics of computer courses unconventionally. Instead of working in a console environment (with text-based outputs), we explored new teaching approaches that allow students to learn computing by writing programs that produce graphics, manipulate images, working with audio and video, and developing computer games. Focuses are placed on students' first computing class in two categories: Engineering major and non-Engineering major. C++ is the computing fundamental class for engineering majors. In order to avoid overwhelming students with game theory or the intricacies of a graphics library such as DirectX or OpenGL, we used the Microsoft product, Dark GDK (a free library that makes graphics programming simple enough for beginners), as the platform, so that students can focus on the fundamentals while creating interesting graphics and game programs. While covering the fundamental topics (such as data types, variables, input, output, control structures, functions, arrays, files classes, and objects), students learn to draw with primitive graphics, load and manipulate images, create sprites and animations, play music and sound effects. For non-engineering majors, the course is non C++ based computer application course. We choose Vizard—a popular virtual reality programming environment—to design course teaching modules to make the course learning full of fun since there is no need to teach this group of students as software developers, instead, they should be taught as tool modifiers.

In this project, faculty members developed new course modules and introduced them into the corresponding computing classes. Preliminary results were obtained and obstacles were discussed, and in the future, we are going to develop more new course modules and also involve more computing courses to be enhanced.

Background

Teaching computer courses is challenging these days since this generation of students witnessed the fast development and wide spread of computer techniques. Most of them are already enthusiasts in popular computer applications, such as video games. However, the teaching of college level computer classes does not change much. This enlarges the gap between advanced computing technology and the college computer education, which creates following problems:

Problem 1

There exists a feeling among youngsters that computing jobs are of less fun; computer experts have no social skills [1,2]. This stereotype also drives college students away from computing discipline. Although the truth is computer is a modern tool for human communication and socialization—a growing number of people use social networking site (such as Facebook and Myspace) to keep in contact with friends and family [3]—there is a missing educational track to let non-computing major students notice that computer has been accepted in the circle of communication and socialization. Thus it is of great importance to advance the skills of computer mediate communication for students outside of computing discipline.

Problem 2

Those who choose computer related programs as their majors are attracted by the exciting application of computers nowadays. But their first computer class at freshmen year, usually an introduction class for C++, might disappoint them since it looks just like another math class. The concepts such as pointers, objects and classes are too abstract for them to understand and there lacks of a path to link what they learn in computer to the real world examples. As a matter of fact, students' interests in computing discipline faded during the struggle among lines and lines of instructions. Lack of student motivation may contribute to some known problems in computer education, such as high DWF (D-grade, withdrawal, and failure) rates reported by many institutions [4] and low enrollment among females and minorities [5]. As an example, high fail rate of C++ class lead to high drop rate from computer related majors as freshmen.

On the other hand, due to the aging workforce and emerging techniques [6, 7], the industry demands for qualified graduates with expertise in computer science, computer engineering and technology are tremendous. The shortage of qualified computer-related workforce in US coupled with the challenges from foreign fast growing countries, like China and India, may end up with more computing jobs shifted abroad every year. This tendency will cause U.S. gradually losing its leading position in computer industry.

Methodology

In order to fight with the cliché and introduce the real picture of computing territory, faculty members from computing programs at the Prairie View A&M University propose to revitalize core undergraduate computing courses with up-to-date computer techniques, for instance, computer graphics, game programming and multimedia. The methodology has been approved to be effective and similar approaches include: "Warriors of the Net" short movie clip developed by MediaLab [8]; "Alice" [9, 10], developed by Carnegie Mellon University; EM-Viz [11]; Video Games for Learning [12]; and Culture Framing of Computer Games [13]. Our scheme focuses on interactive gaming environment and hands-on experience of manipulation image or sound as multimedia tools.

At the mean time, based on the nature of the audience, we developed different modules. At the Prairie View A&M University, undergraduate computing curricula are of the same structure that most universities follow. We can category them into two levels:

- Level 1: University requires all the undergraduate students to take at least 3 credit hours computing class. Most non-engineering /computer science / technology students select non C++ oriented classes, such as CPET1013 Computer Applications for Technology I (offered by Engineering Technology department), COMP1003 Introduction to Computer Education (offered by Computer Science department), and COMP1013 Introduction to Computer Science (offered by Computer Science department).
- Level 2: The CoE requires all majors to have a C++ class at freshmen year. With different focus of each curriculum, three C++ classes offered by three departments in CoE are shown in table 3 below.

Table 3 Key C++ courses serve for different programs in College of Engineering

Key C++ Course	Programs Served	Department Offered
ELEG 1043 Computer Applications in Engineering	Chemical Engineering	Electrical and Computer Engineering
	Civil Engineering	
	Electrical Engineering	
	Mechanical Engineering	
COMP 1213 Computer Science I	Computer Science	Computer Science
	Computer Engineering	
CPET 1023 Computer Application to Technology II	Computer Engineering Technology	Engineering Technology
	Electrical Engineering Technology	
	Computer Aided Drafting	

It is clear that all undergraduate students will take a computing class, but only CoE students require C++ based computing course. We can naturally map the two existing problems to the two groups of students with different levels of computing requirements to guide our strategies. Problem 1 associates with level 1 requirement which includes all university students other than CoE. Problem 2 is linked to computing level 2 courses mostly serve CoE students. According to the enrollment sample taken at fall 2008, level 1 enrollment is 5077 and level 2 is 971 at the Prairie View A&M University.

Strategy and Preliminary Results

The project involves designing new educational modules for multi-level computing needs by applying streamlined computer techniques in undergraduate courses to motivate students' interests in computing and strengthen the aptitude of computing problem solving skills in the future U.S. workforce. Our goal is to: (1) make computer classes teaching/learning "full of fun" and "exciting" by imbedding multimedia and computer game components while still retaining the underlying contents and (2) foster students' critical thinking skills through unveiling the mystery of image, audio, video processing, and game programming; enhance student's problem solving capability with novel computing methods.

The strategy of teaching the traditional topics of computing classes unconventionally is explored through two categories aligned with the problems perceived above**.**

Category 1: Teach C++ concepts with projects on graphics, game, video, image, etc.

At all undergraduate students are required to take a computing course to fulfill the university requirement, while all CoE students all starts with C++ in their freshmen year. As all majors in

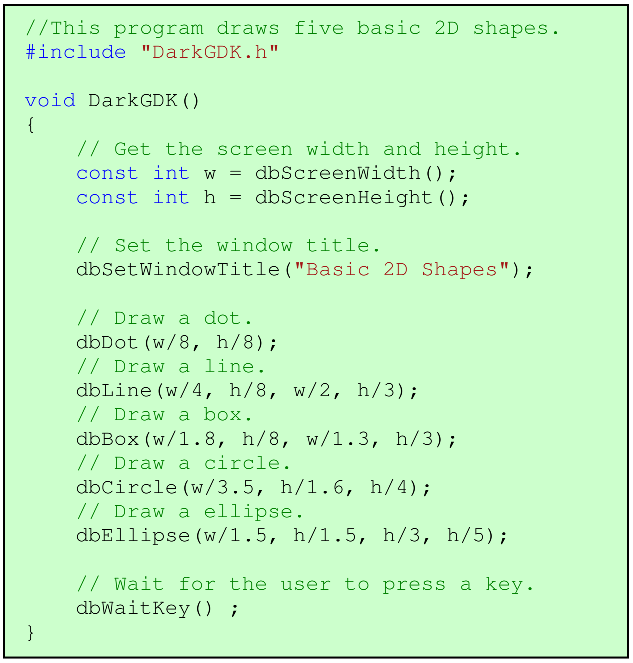
CoE have similar admission criteria and curricula, an introductory programming course with this group of students begins at level 2 computing level.

Instead of working in a console environment (with text-based outputs), we explore new teaching approaches that allow students to learn C++ by writing programs that produce graphics, manipulate images, work with audio and video, and play games. All these procedures are based on the assumption that students have no programming background. In order to avoid overwhelming students with game theory or the intricacies of a graphics library such as DirectX or OpenGL, we propose to use the Microsoft product, Dark GDK (a free library that makes graphics programming simple enough for beginners), as the platform so that students can focus on the fundamentals while creating interesting graphics and game programs.

While covering the fundamental topics (such as data types, variables, input, output, control structures, functions, arrays, files classes, and objects), students learn to draw with primitive graphics, load and manipulate images, create sprites and animations, play music and sound effects, and detect collisions between the graphical elements of a program. Students also learn to combine these skills to create their own interactive video games.

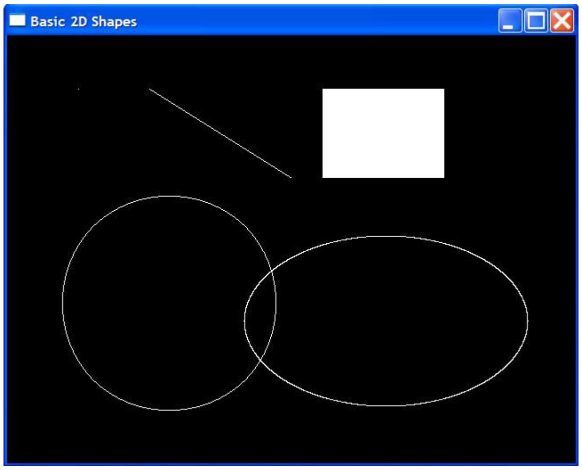
Example 1:

This example is to illustrate how students can learn basic concepts without knowing game theory or having graphics background. The program (Fig. 3a) draws five basic 2D shapes—dot, line, box, circle, and ellipse—at some places within the window (Fig. 3b). It introduces C++ concepts as comments, constant variable, data types, variables, functions, and external library. By writing this program, students do not need to know what are inside of the dbXX functions, instead, they just focus on the fundamental concepts.



```
//This program draws five basic 2D shapes.
#include "DarkGDK.h"

void DarkGDK()
{
    // Get the screen width and height.
    const int w = dbScreenWidth();
    const int h = dbScreenHeight();

    // Set the window title.
    dbSetWindowTitle("Basic 2D Shapes");

    // Draw a dot.
    dbDot(w/8, h/8);
    // Draw a line.
    dbLine(w/4, h/8, w/2, h/3);
    // Draw a box.
    dbBox(w/1.8, h/8, w/1.3, h/3);
    // Draw a circle.
    dbCircle(w/3.5, h/1.6, h/4);
    // Draw a ellipse.
    dbEllipse(w/1.5, h/1.5, h/3, h/5);

    // Wait for the user to press a key.
    dbWaitKey() ;
}
```

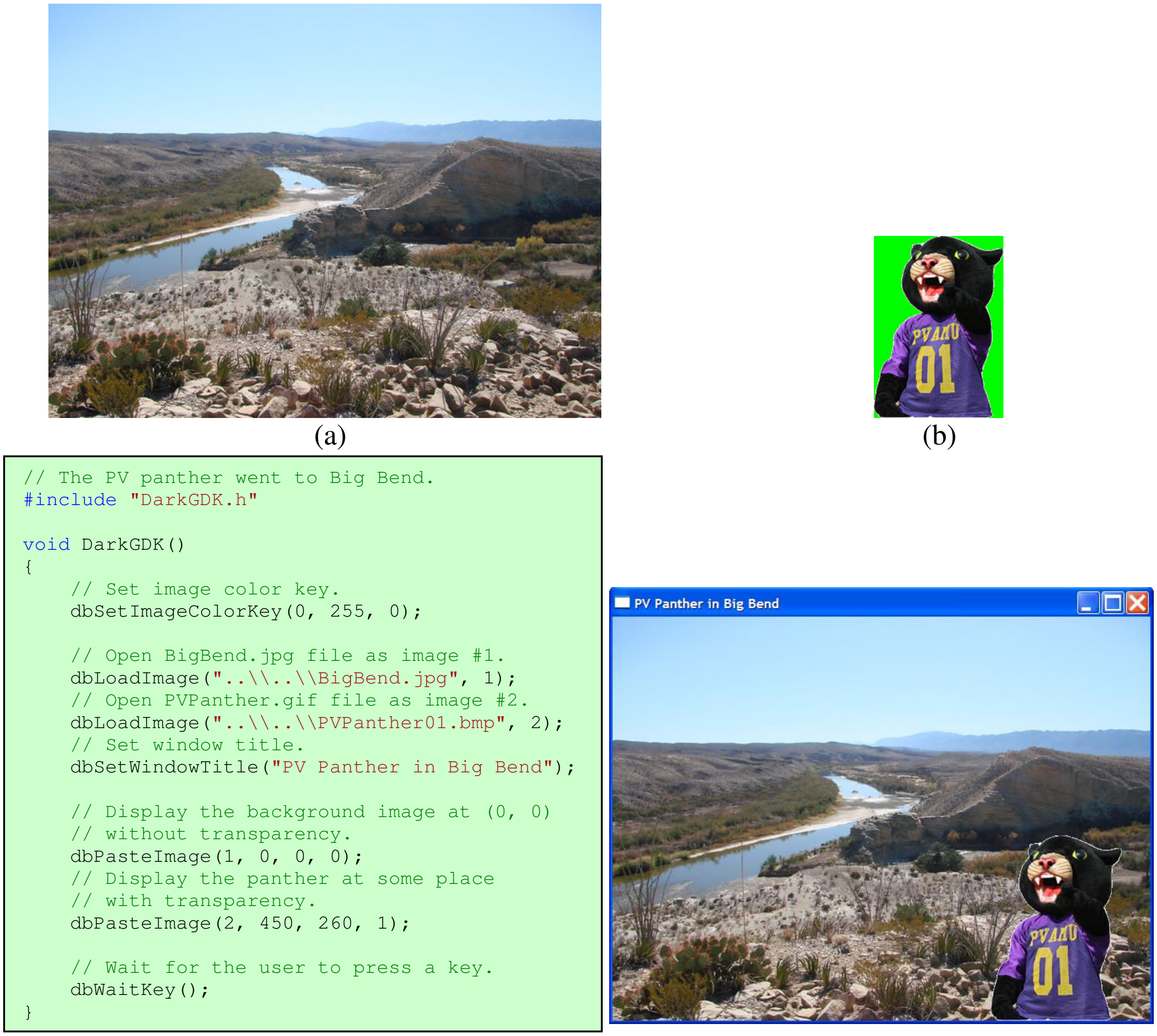
(a)



(b)

Fig. 3: An example of Dark GDK

As everyone knows, people like to work on or be attracted by things relevant to them or they are familiar with [14, 15]. Nowadays, multimedia technologies penetrate every aspect in our daily lives, and most students are easily attracted by such information [15]. We design course projects for students manipulating their own media, such as sounds, images, etc. In fact, without the help of Dark GDK, such design work is not trivial. However, in Dark GDK, we can easily load an image or a sound file by using the functions of dbLoadImage() and dbLoadSound(), respectively. Then, to display the image or play the sound, functions dbPasteImage() and dbPlaySound() can be called. From the example shown in Fig. 4, students can mix 2 images (Figs. 4a and 4b) together fairly easy by calling the abovementioned functions (Fig. 4c), and the result image is shown in Fig. 4d.



(a)

(b)

```
// The PV panther went to Big Bend.
#include "DarkGDK.h"

void DarkGDK()
{
    // Set image color key.
    dbSetImageColorKey(0, 255, 0);

    // Open BigBend.jpg file as image #1.
    dbLoadImage("..\\..\\BigBend.jpg", 1);
    // Open PVPanther.gif file as image #2.
    dbLoadImage("..\\..\\PVPanther01.bmp", 2);
    // Set window title.
    dbSetWindowTitle("PV Panther in Big Bend");

    // Display the background image at (0, 0)
    // without transparency.
    dbPasteImage(1, 0, 0, 0);
    // Display the panther at some place
    // with transparency.
    dbPasteImage(2, 450, 260, 1);

    // Wait for the user to press a key.
    dbWaitKey();
}
```

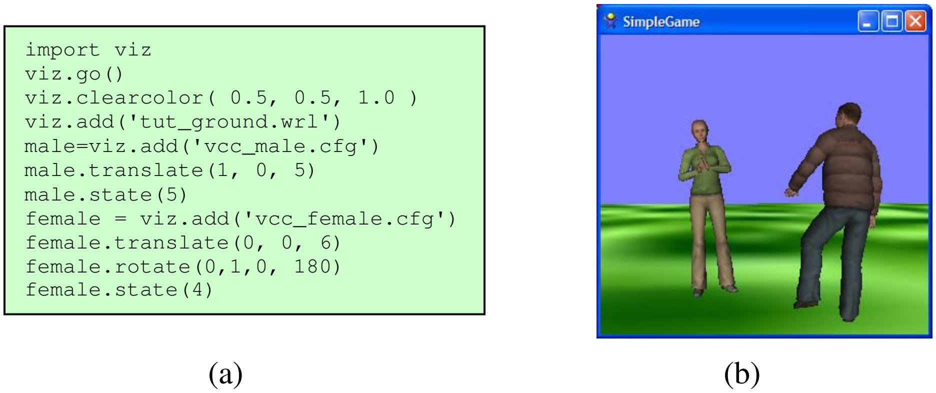
(c)

(d)

Fig. 4: An example of displaying image

Category 2: Non-CoE majors' intro to computer course

In CoE, three courses—COMP1003, COMP1013 and CPET1013—are designed for introduction to computer, which is also open for non-CoE major students. For this group of students, there is no need to teach them as software developers, instead, they should be taught as tool modifiers. Usually, for such students, they will rarely create a program exceeding 100 lines [15]. Attractive topics are even more desired by these students. Based on these observations, we introduce multimedia computing and game programming in these three courses in a similar way as described in step 1. We use Vizard—a popular virtual reality programming environment—to design course teaching modules to make the course learning full of fun, as shown in the following example as well as described in [16, 17].



```
import viz
viz.go()
viz.clearcolor( 0.5, 0.5, 1.0 )
viz.add('tut_ground.wrl')
male=viz.add('vcc_male.cfg')
male.translate(1, 0, 5)
male.state(5)
female = viz.add('vcc_female.cfg')
female.translate(0, 0, 6)
female.rotate(0,1,0, 180)
female.state(4)
```

Figure 5: An example of Vizard

Example 2:

We start a programming topic with a simple game program. As shown in Fig. 5, the program with eleven sentences (Fig. 5a) creates a playable game where the two avatars are interacting with each other and performing certain actions (Fig. 5b). This simple program can be used to introduce syntax, class, data types, functions, parameters, file loading, etc.

Based on the successful experience in our previous projects, we will design project assignments by using Vizard to help students to learn programming concepts within the three courses in a fun way. In fact, Vizard is based on Python, which is a popular programming language for multimedia programming. Besides game projects (as shown in Fig. 5), image manipulation as well as audio processing course projects are also developed.

Conclusions

As the completed teaching modules were introduced in class and favored by both instructors and students, we can conclude that 1) it is very necessary to update current computing class with modern tools to attract more youngsters to computing discipline; 2) game and multimedia are effective examples to motivate students' anxious of learning.

Future Work

With the success obtained from the first class of computing, we are going to extend our project to other computer related courses. It will not be limited to software related courses, but embrace hardware courses as well, for example computer architecture and computer systems design.

Bibliography

1. M. Sands, N. Moukhine, and G. Blank, "Widening the Pipeline of K-12 Students with Flash", Journal of Consortium for Computing Sciences in Colleges, vol. 23, issue 5, pp. 52-57.
2. G. Blank, S. Hiestand, and F. Wei, "Overcoming Misconceptions about Computer Science with Multimedia", Proceedings of 35th SIGCSE Technical Symposium on Computer Science Education, March, Norfolk, VA. At http://www.cse.lehigh.edu/~cimel/papers/sigcse04.pdf
3. T. Stewart, "You are never alone with a computer – social issues", Behavior & Information, Vol. 27, Issue 2, March 2008, pp. 95-96.
4. N. Herrmann, and etc. "Redesigning introductory computer programming using multi-level online modules for a mixed audience", In Proc. 34th SIGCSE Technical Symposium on Computer Science Education, 2003, pp. 196-200.
5. J. Margolis and A. Fisher, "Unlocking the Clubhouse: Women in Computing", Cambridge, MA: MIT Press, 2002.
6. ICAF Industries Studies 2006 Report, National Defense University.
7. ICAF Industries Studies 2001 Report, National Defense University.
8. Warriors of the Net, (http://www.warriorsofthe.net/)
9. Alice, (http://www.alice.org/)
10. Barbara Moskal, Deborah Lurie, Stephen Cooper, "Evaluatinig the Effectiveness of a New Instructional Approach," in *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pp. 75-79, Norfolk, Virginia, 2004.
11. Craig Scott, Jumoke Ladeji-Osias, Tanya Capers, and Kofi Nyarko, "The Development and Implementation of EM-Viz, a 3D Undergraduate Electromagnetic Engineering Visualization Application, with an Assessment of its Relative Efficacy for Minority Visual Literacy and Achievement," *2003 ASEE Annual Conference & Exposition: Staying in Tune with Engineering Education*, Nashville, TN, June 22-25, 2003.
12. James P. Gee, "What Video Games Have to Teach Us About Learning and Literacy," *ACM Computers in Entertainment*, Vol. 1, No. 1, October 2003, BOOK01.
13. Kurt Squire, "Cultural Framing of Computer/Video Games," *the international journal of computer game research*, volume 2, issue 1, July 2002
14. M. Guzdial, and A.E. Tew, "Imagineering Inauthentic Legitimate Peripheral Participation: An Instructional Design Approach for Motivating Computing Education," in *Proceedings of the Second International Computing Education Research Workshop*, Canterbury, UK, 2006, pp. 51-58.
15. M. Guzdial and A. Forte. "Design Process for a Non-majors Computing Course," in *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, St. Louis, Missouri, 2005, pp. 361-365.
16. Y. Wang, Y. Yang, J. Lian, "Make Math Learning Game Like and Bridge between Math and Engineering," in *Proceedings of the 2009 International Conference on Computer Graphics and Virtual Reality*, Las Vegas, Nevada, July 2009, to appear.
17. Y. Wang, S. Cui, Y. Yang, and J. Lian, "Virtual Reality Mathematic Learning Module for Engineering Students," the *Technology Interface Journal*, submitted.