

On Developing User Interfaces for Children in Educational Virtual Reality Environments

Lori Scarlatos, Rebecca Friedman

Department of Computer and Information Science
Brooklyn College, CUNY
2900 Bedford Ave.
Brooklyn, NY 11210 USA
+1 718 951 5000 x2063
lori@sci.brooklyn.cuny.edu

ABSTRACT

Virtual reality provides users with a truly immersive experience that can more fully engage them with the data. The goal of this project is to create learning environments where children can be fully immersed in and engaged with their learning materials. This technical report describes initial experiments in this area.

Keywords

Virtual reality, motion capture, math education, collaborative learning environment.

INTRODUCTION

In an age where our well-being – physical, economic, and military – relies so heavily on technology, educating our nation's future scientists and mathematicians has become increasingly critical. Educators and governments have come to recognize the importance of collaborative activities, learning through play, and teacher guidance [1, 22]. Increasingly, schools and museums are relying on computers to engage students in learning activities and provide some level of guidance.

Human-computer interfaces have unquestionably come a long way [20]. Yet even with our graphical user interfaces and design guidelines, the popular desktop mode of interaction leaves a lot to be desired. The key problem is that "direct manipulation" is not so direct. Good hand-eye coordination, and practice, is needed to accurately manipulate two-dimensional on-screen objects using subtle movements of one's hand or fingers on a separate device (such as a mouse or trackpad). For people with physical disabilities, this can prove to be particularly challenging. Furthermore, there are numerous tasks that do not translate well to a two-dimensional user interface, even when users have mastered use of the tools. For example, building with blocks or doing a jigsaw puzzle can never be as easy

to do on a computer as they are in real life. Finally, most popular input devices (and computer systems) allow only one person to interact with the computer at a time. This severely limits the possibilities for group activities and collaborative work.

Researchers have consequently been exploring alternative approaches to computer-human interfaces. Progress has been made in immersive interfaces, with virtual reality and ubiquitous computing, and alternative input modes, such as voice, gesture, and eye tracking. Scientists have even experimented with using objects in the environment as input devices (tangible interfaces) and giving those objects intelligence about where they are in relation to others (smart objects). Some of these efforts have materialized into practical applications that are creeping into our lives: a stuffed Barney toy that responds to a child's actions; a SpeedPass card that allows commuters to drive through tollbooths; a mapping system in your car that uses satellite data to keep track of where you are; even an AIBO robot that responds to verbal commands. Yet most educational applications still rely on a point-and-click interface with a two-dimensional playing field.

The goal of this project is creating learning environments where children can be fully immersed in and engaged with their learning materials. To make this a reality, it is necessary to pursue several research tracks simultaneously. Reliable tracking and representation of the objects (i.e. their positions and orientations relative to one another) and the children (i.e. their gestures relative to the virtual objects that they wish to manipulate) must be provided in real time so that the system can respond appropriately. Gestural inputs must be defined in such a way that the children can easily learn to control multiple parameters associated with large data sets of

educational information. Qualitative probabilistic reasoning strategies applied to the tracking functions can help us to rapidly adjust our algorithms to work with the specific applications that we have defined. Adaptive learning with dialog systems will help us to fine-tune the gestural interface, making it more natural and intuitive for the children.

BACKGROUND

Educators have long recognized that puzzles provide rich opportunities for learning. Young children develop spatial and problem-solving skills by playing with blocks and puzzles. Occupational therapists use puzzles to test – and subsequently develop – their patients' cognitive skills. The National Council of Teachers of Mathematics [13] states, "Students must learn mathematics with understanding [by] actively building new knowledge from experience ...". Holt argues that children, when encouraged to explore on their own, will come to understand mathematical principles that most educators would not dare to try to teach them [8]. Gardner theorizes that different people can be "intelligent" in a variety of different ways, and that teachers must therefore address these multiple intelligences in order to teach effectively [6].

As a result, math teachers are being encouraged to involve students in collaborative activities, plan differentiated tasks, and use "workstations" to present the same material in different ways [1]. In response, teachers increasingly rely on math manipulatives – such as Cuisenaire Rods, Pattern Blocks, and Tangrams – to illustrate math concepts and provide students with a way of further exploring those concepts. Apparently children who normally have trouble with math find the concepts more understandable when they are working with tangible objects. Talking about the activity also helps children to develop meta-cognitive connections that are useful in transference of the knowledge. Gokhale demonstrated that collaborative learning significantly improves students' critical thinking skills [7].

Educational Games

A number of math games have been developed both commercially and by researchers in an attempt to increase students' confidence and competence in the use of math skills. For example, Wang et al developed a framework for supporting mathematics education on the Internet, where students are expected to work primarily on their own [23]. Ke has experimented with tournament math games, and demonstrated that students who work collaboratively tend to develop a more positive attitude toward math [11]. Building on

the popularity of games among teenagers, Elliot and Bruckman developed single-player 3D games for high school students in which the players must use math skills to navigate obstacle courses [5]. To encourage collaboration, they also suggest that players share "challenges" with one another. Cypher and Smith developed a system that teaches physics by enabling end-users to program their own simulations [4]. Others have been using pedagogical agents to reinforce learning [3, 9]. Yet, in general, these games do not combine the benefits of physical activity and collaboration the way that games with tangible user interfaces do.

Collaborative Learning Environments

Scarlatos' Tangible Interfaces for Collaborative Learning Environments (TICLE) is a project that focuses on using puzzle pieces in a tangible interface that informs an educational application [16]. The computer system acts as "guide on the side" by responding appropriately to what children are doing with the puzzles, offering encouragement, suggestions, and hints. Yet the children can also choose to ignore the computer, leaving them free to focus on the learning activity instead of some awkward computer interface.

Scarlatos' primary approach has been to use computer vision to track the puzzle pieces. Working with puzzles on a 2D tabletop, a camera looking up tracks the pieces via a set of colored tags. Different tags have been developed for different types of puzzles, and a library for detecting these tags has been developed as an Xtra¹ [15]. Experiments with groups of school children have demonstrated that this approach to learning keeps the children on track, encourages discussion about the problem, and increases the likelihood that children will be better able to solve similar problems in the near future.

Other uses of alternative interfaces in educational applications have also been explored. I-Sign uses speech recognition to allow hearing parents to read aloud to their deaf children [19]. This interaction between parent and child reinforces the learning, while giving the child multiple ways to recognize words in a vocabulary list. Using "word books" greatly simplifies the speech recognition task, and

¹ An Xtra is a library of functions that supplements the capabilities of the Lingo language in an application developed with Director, a popular multimedia authoring program.

eliminates the need for training. Another application, SmartStep, uses touch sensors on the floor to help reinforce arithmetic skills in children [18]. Although this application was intended to be used by one child at a time, we have observed that children often like to work together with SmartStep, giving one another advice or hints, or even helping to step on the right answer.

Despite the advances that these applications represent, they are limited by their reliance on a traditional computer with a touch screen monitor for feedback. Most of these applications are also inherently two-dimensional. Although Scarlatos did some initial experiments with three-dimensional puzzles [17], they still did not provide the immersive experience that was sought.

Virtual Reality Environments

Although some "virtual reality" environments for learning, such as VRMath [24] are actually just 3D simulations or games – viewed on a traditional computer screen with mouse and keyboard for input – others involve moving through a very real space. For most of these applications, gestures are used to control the application.

Even though using gestures can be physically exhausting, it is convenient for sporadic use in virtual environments [2]. Recent experiments have used VR to educate children in museums, but limit the interaction to an individual [14]. In another experiment, where children learn about how the earth is round, only one child is allowed in the CAVE at once while the other children observe from a distance [12].

In many virtual environments, motion tracking technology is used for gesture recognition. Motion tracking can use prosthetic, acoustic, magnetic, or optical devices to detect positions [21]. Optical approaches are generally favored because a large number of points can be tracked simultaneously, without constraining the users' motions with wires. Of course, the biggest problem with optical tracking is occlusion.

APPROACH

Our approach is to develop a learning environment using the CAVE visualization environment, produced by FakeSpace Systems, Inc. Unlike most virtual reality systems, the CAVE allows multiple users to immerse themselves fully in the same virtual environment at the same time. Although we did not have the budget for stereoscopic viewing, we felt that

the environment would be sufficiently immersive with mono projections on an 8'x10' back wall and a 8'x10' floor space.



Figure 1. Reflective tags on a child.

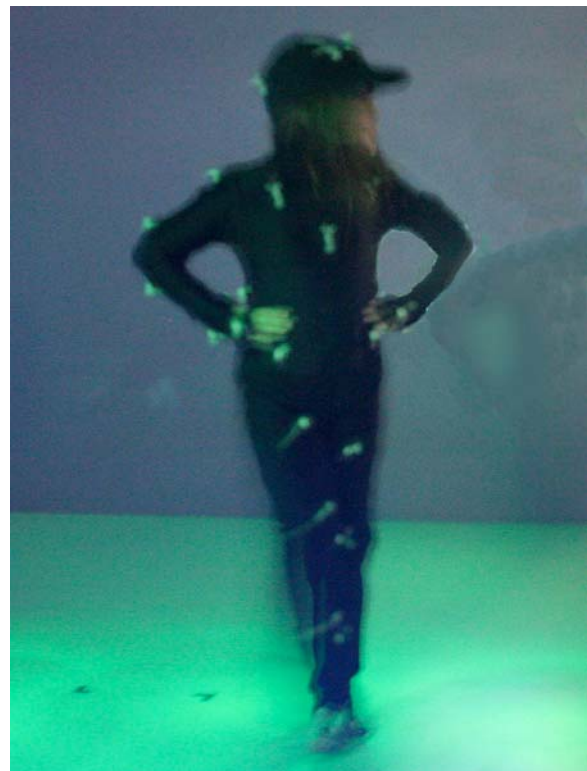


Figure 2. Tracking motion in the CAVE environment.

To track the gestures and positions of the children in the space, we are using a Vicon motion capture system with twelve cameras. Each camera emits a strobing light frequency that is reflected back by special reflectors in the environment. The Vicon software automatically correlates this data to produce 3D points that map to a model. Figure 1 shows a child wearing these reflectors in our CAVE space. Figure 2 shows another child being tracked in the CAVE environment.

Our approach has been to systematically familiarize ourselves with the system, with an eye toward developing strategies for implementing fully immersive educational applications. Steps in this process are to 1) track human motion in non-real-time, and use it to control a 3D object offline; 2) track human motion in real time, and use it to control a 3D object online; and 3) track multiple humans simultaneously, using their motion to control a collaborative interface.

Tracking Human Subjects

Vicon 2.0 is Vicon's motion capturing software used for recording optical markers in motion and labeling and rigging those markers. Figure 3 shows a Vicon screen displaying what is seen by nine cameras. Although our space contains twelve cameras, we had a difficult time getting an accurate reading from the dots; automatic calibration of the cameras is a time-consuming process that is subject to error. Two of the cameras had problems making a connection with the software.

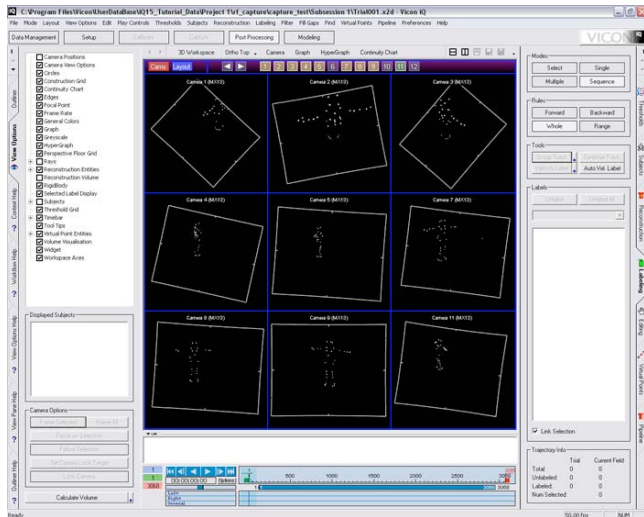


Figure 3. Motion capture data in 2D

To address the correlation problems, we adjusted the calculations for converting the two-dimensional camera images into usable three-dimensional optical

data. The “CircleFit, Reconstruct, Trajectory Fit frame Range” function in the pipeline calculates and reconstructs the 2D optical data from each camera and tries to construct a 3D version of it. Because of occlusion, the spots on the figure tend to disappear and reappear, which results in problems in reconstructing the whole animation. Parameters in the “CircleFit, Reconstruct, Trajectory Fit frame Range” function allow one to modify the number of cameras being used to calculate the positions, increase the maximum range of seeing the dots, and even read only from selected cameras. To better understand the rest of the Vicon 2 software, we used the pre-recorded optical data that came with the software.

Once we had the optical data reconstructed to 3D (see Figure 4), we had to label the dots. Labeling gives each dot, or marker, a unique identity so that the software knows exactly where each marker is. Once the markers are labeled, we can calibrate the model, or subject.

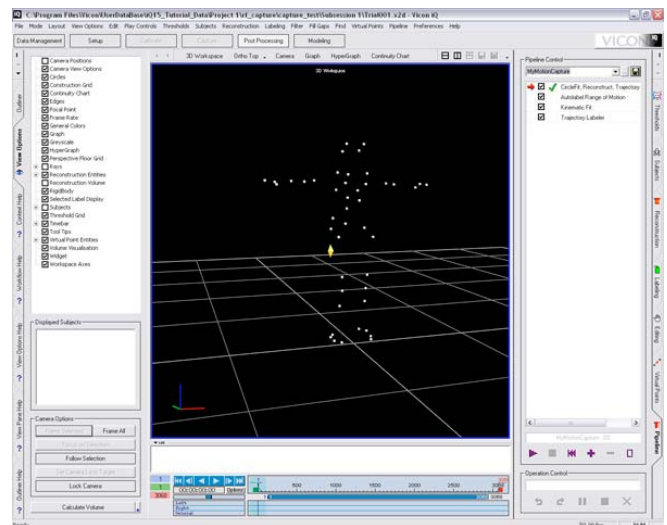


Figure 4. Reconstruction of 3D points from 2D data

Before we began labeling, we created a generic subject (VST), which is a Vicon Skeleton Template. The subject model holds a list of markers that the user must assign a dot to, which can later be mapped to an external model. Each marker represents a part of the body, or where the dot is placed on the human subject. Labels must be assigned to the dots in a particular order. For example, the first marker shown on the list is the “root”. This is the main point of the model, which is always located at the waist. The next five markers are four points around the waist and one point at the pelvis.

As we assign markers to the dots, Vicon creates connections from dot to dot to form a skeleton. When we play the whole animation, the connected skeleton moves along with the optical data. At certain frames, some of the connections disappear because some of the markers become unlabeled and the information is lost. To fix this problem, we used “Autolabel Range of Movement”. This automatically labeled the rest of the animation, based on the initial labels that were made on the first frame.

After all the optical data were labeled with its corresponding markers, we had to calibrate the template model (VST) with the markers. The initial frame shows the optical data in a T-Pose, a generic pose for all models. When modelers create a 3D character, it is always modeled in a T-Pose, so it will be easier for the animators to apply the skeleton and skin the bones to the mesh. We then calibrated the subject from the subject menu. This calculates the average error between the real markers on the actor’s body, and where the “Subject Calibration” process believes where the markers should be.

The next operation to add to the pipeline is called the “Kinetic Fit”. The operation connects the subject model with the label we have assigned to each marker. This was used to export the information in *.C3D format, which is a format read by most 3D software packages. We needed to calibrate the model subjects before exporting because it prevents unlabeled unknown markers from appearing when importing the file into another program.

The exported data could then be used by an animation package. We used Alias Motion Builder 7, which supports real time animation with ease. With this tool, 3D characters and objects can be animated manually or with imported motion capture data applied to the models. We used this software to animate a 3D character, which Friedman created with 3D Studio Max 8. The 3D Studio Max file was exported as a *.FBX file, a cross-platform format that is supported by several applications..

With the optical motion data imported and the actor in the scene, we created a marker set for the actor (figure 5). Cells are generated for each body part of the template model, onto which markers from the optical data are dragged and dropped. In order for this process to work correctly, the naming and positions of the skeleton must be recognized by Motion Builder 7. This allows Motion Builder to modify the skeleton so

it will save the user time in labeling and setting up the skeletal rig in the software.

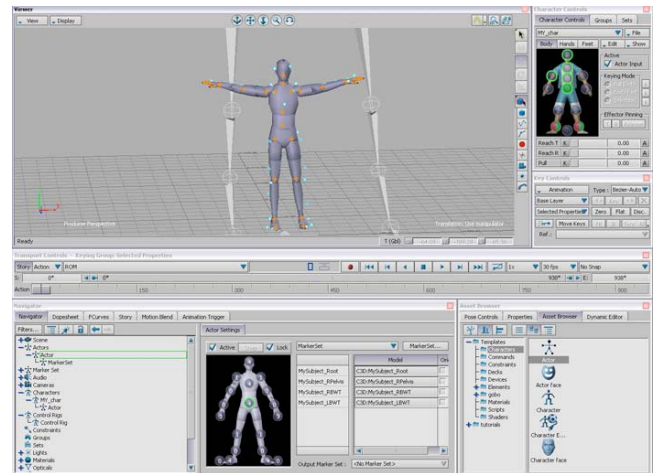


Figure 5. Actor template with motion capture data

Once all the motion data is applied correctly to the actor, the actor template is applied to the 3D character. The character’s movement will then match the movements of the actor template. Figure 6 shows the motion data applied to Friedman's model.

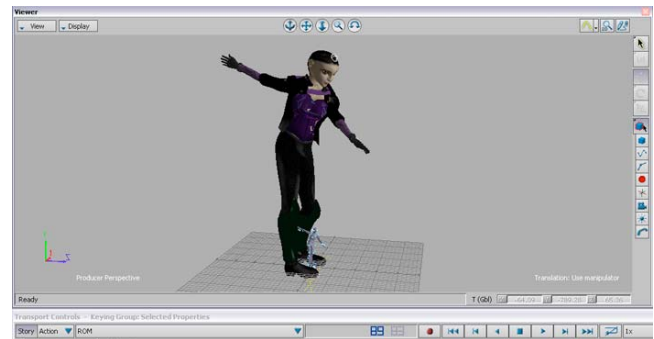


Figure 6. Motion capture applied to 3D model

Collaborative Learning Environments

Most motion capture works with single actors linked to a single jointed model, primarily because these types of models are easiest to work with. By considering the constraints of the jointed model (i.e. following the rotations of the bones), an expert-systems approach can be used to estimate where the bones might be, and where they cannot be [21]. Adding actors complicates the problem, because actors can move independently of one another.

To simplify the problem, we decided to initially work on a task where we need to track only one marker per participant. We decided to create a game from the gallery problem, which is a variation of the coloring problem: given a concave polygonal space, how can you arrange guards such that all parts of the polygon

are visible to at least one guard? The goal is to minimize the number of guards needed to cover the space.

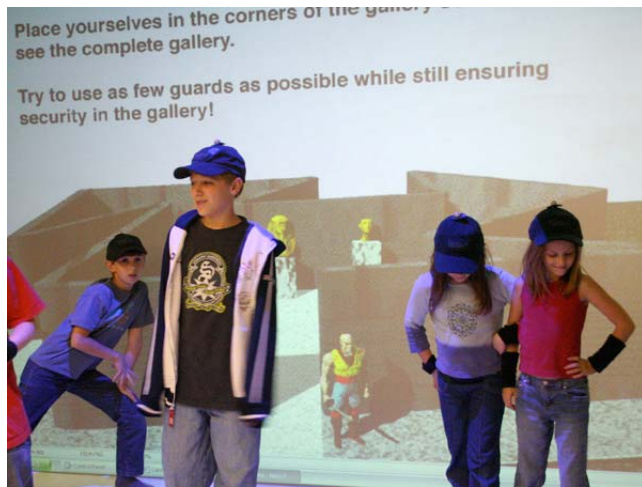


Figure 7. Gallery game.

In this game, instructions are projected on the back wall and the polygonal space is projected onto the floor (figure 7). As the children move about on the floor, the floor projection is updated to show where each child (or guard) can "see".

Each child's position in the space is indicated by a single reflector on top of a black cap. For this application, orientation – or where the child is looking – is irrelevant. We begin by calibrating the system – which involves removing extraneous spots and correlating the remaining spots – and then marking the 3D spots. Each spot is then tracked as the child moves around the space.

CONCLUSIONS

In using the motion capture data to animate a 3D character off-line, we have become intimately familiar with both the capabilities and pitfalls of the motion capture system. We discovered that calibrating the system, and labeling the markers, chews up most of the motion capture time.

In our simple game, we still need to calibrate and mark the spots associated with each child. Corners of the projection area are also marked, to accurately relate the children's positions to the positions of the virtual gallery walls. This setup takes a very long time, and therefore would be problematic for most groups of school children.

We intend to explore solutions to the following issues as next steps in this project:

- Streamline the calibration process, so that children can start playing the game quickly.
- Develop a simple skeletal model for tracking children's gestures.
- Define gestures that will act as inputs to the system.
- Develop games that test the effectiveness of the gesture recognition. Incorporate a speech interface, with a finite number of commands, to be used in conjunction with the gestures.

ACKNOWLEDGMENTS

We are grateful to Dr. Tenenbaum for his continuing support of this project. This work has been supported by the National Science Foundation under grant number CNS-0420996.

REFERENCES

1. Bransford, J.D. et al, editors (2000) *How People Learn*, National Academy Press.
2. Cabral, M.C., Morimoto, C.H. and Zuffo, M.K. (2005) On the usability of gesture interfaces in virtual reality environments. *Proceedings of the 2005 Latin American conference on Human-Computer Interaction*, 100-108.
3. Conati, C. and Zhao, X. (2004) Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game, *Proceedings of the 9th international conference on Intelligent user interface*, pp. 6 - 13.
4. Cypher, A. and Smith D.C. (1995) KidSim: End User Programming of Simulations. *Proceedings of CHI '95*, pp. 27-34.
5. Elliot, J. and Bruckman, A. (2002) Design of a 3D interactive math learning environment, *Proceedings of the conference on Designing interactive systems*, pp. 64 - 74.
6. Gardner, H. (1983) Frames of Mind: The Theory of Multiple Intelligences, Basic Books.
7. Gokhale, A.A. (1995) Collaborative Learning Enhances Critical Thinking, *Journal of Technology Education*, Volume 7, Number 1, online at http://scholar.lib.vt.edu/ejournals/JTE/v7n1/gokhal_e.jte-v7n1.html.
8. Holt, J. (1983) How Children Learn, Da Capo Press.

9. A. Kapoor, S. Mota, and R. W. Picard (2001). Towards a learning companion that recognizes affect. In *AAAI Fall Symposium*, Nov 2001.
10. Katsunori, K. and Ormond, R. (2005) Discover the Game with Alias. Alias Learning Tools: Alias System Corp, 2005.
11. Ke, F. (2006) Classroom goal structures for educational math game application, *Proceedings of the 7th international conference on Learning sciences ICLS '06*, pp. 314-320.
12. Moher, T., Johnson, A., Ohlsson, S. and Gillingham, M. (1999) Bridging strategies for VR-based learning. *Proceedings of CHI '99*, 536-543.
13. National Council of Teachers of Mathematics (2002) Principles and Standards for School Mathematics.
14. Roussou, M. (2004) Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Computers in Entertainment*, 2 (1), 1-23.
15. Scarlatos, L.L. (2006) Tangible Math. To appear in the special issue on Computer Game-Based Learning in the *International Journal of Interactive Technology and Smart Education*.
16. Scarlatos, L.L. (2002) TICLE: Using Multimedia Multimodal Guidance to Enhance Learning. *Information Sciences 140*, 85-103.
17. Scarlatos, L. Landy, S. and Qureshi, (2002). Tracking 3D Puzzle Pieces for Collaborative Learning Environments, *SIGGRAPH 2002 Conference Abstracts and Applications*, 270.
18. Scarlatos, T. (2005) SmartStep: A Physical Computing Application to Teach Basic Math Skills. Available online at <http://www.cs.sunysb.edu/~tony/research/SmartStep/SmartStep.html>.
19. Scarlatos, T. Scarlatos, L. and Gallarotti, F. (2003). iSign: Making the Benefits of Reading Aloud Accessible to Families with Deaf Children. In *Proceedings of the Sixth IASTED International Conference on Computer Graphics and Imaging*.
20. Shneiderman, B. (1998) Designing the User Interface, 3rd Edition, Addison-Wesley, Reading, MA.
21. Trager, W. (1999) A Practical Approach to Motion Capture: Acclaim's optical motion capture system. Available online at http://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/motion_capture/motion_optical.htm.
22. Van de Walle, J.A. (2001) Elementary and Middle School Mathematics: Teaching Developmentally, 4th Edition, Addison Wesley Longman.
23. Wang, P.S., Kajler, N., Zhou, Y. and Xiao, Z. (2003) WME: towards a web for mathematics education, *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pp. 258 - 265.
24. Yeh, A., & Nason, R. (2004). VRMath: A 3D Microworld for Learning 3D Geometry. *World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004(1)*, 2183-2194.