A Scenario for Collaborative Learning in Virtual Engineering Laboratories

Felipe Arango, Chenghung Chang, Sven K. Esche and Constantin Chassapis
Stevens Institute of Technology, Department of Mechanical Engineering, Hoboken, New Jersey 07030, USA
FArango@stevens.edu, CChang@stevens.edu, SEsche@stevens.edu, CChassap@stevens.edu

Abstract - **The feasibility of developing interactive collaborative virtual environments for undergraduate student laboratory experiments is currently being explored at various institutions. Such environments can be implemented using commercial multiplayer game engines together with the associated software development kits. Such immersive environments are expected to provide students with an opportunity for exercising their problem solving skills by collaboratively interacting with each other and the virtual laboratory exercises. This paper discusses the requirements for designing game-based virtual laboratory environments from pedagogical as well as technical point of view. Also, the practical implementation of these design requirements in a prototype system is discussed and a sample scenario for a virtual laboratory based on an existing laboratory exercise from a junior-level mechanical engineering course on mechanisms and machine dynamics is presented. The described virtual laboratory environment represents an attempt to mimic important aspects of the learning experience gained through conventional hands-on experiments and even augment this experience by certain features that are not achievable in the traditional hands-on laboratory setting.**

Index Terms – Collaborative learning environment, Computer games, Game engine, Online laboratory, Remote experiment, Software development kit, Virtual experiment.

INTRODUCTION

At present, the vast majority of computer simulation programs [1],[2],[3] used for educational or training purposes basically represent mathematical models of physical phenomena. These models are accessed through graphical user interfaces, which allow for data input and the subsequent presentation and analysis of the simulation results [4]. Typically, they are designed for usage by a single user and therefore do not facilitate collaborative learning modes. In the past several years, rapid and significant advances in semiconductor technology have enabled the computer game developers and console producers to develop very realistic massive multiplayer game environments (e.g. World of Warcraft [5], Everquest II [6], Second Life [7]).

This paper describes the exploration of the potential for using multiplayer game engines to develop virtual interactive laboratory environments for undergraduate engineering education. Such environments are expected to provide students with an opportunity for exercising their problem solving skills by collaboratively interacting with each other and the virtual laboratory exercises. In the first section, this paper will discuss the requirements for designing game-based virtual laboratory environments from pedagogical as well as technical point of view. Next, the practical implementation of these design requirements is discussed. Finally, a sample scenario of a virtual laboratory is presented.

SYSTEM DESIGN REQUIREMENTS

General Guidelines for Providing an Effective Learning Experience

Any educational activity aiming to provide students with an effective learning experience should exhibit certain characteristics. Some of the desirable features are that it must be contextualized, goal-oriented, challenging, anchored, exploratory, attention-drawing and feedback-providing [8].

<u>Contextualized</u>: The learning should be in a setting where the students' actions make sense to them. In the context of developing virtual laboratory environments, this means that all elements within the environment should be within the engineering context. Thus, the virtual laboratory scenarios should be scripted so as to convey the essence of the engineering phenomenon investigated. In addition, examples of industrial or commercial implementations of the learned concepts can be presented to the students. Furthermore, complementary content might be offered for interactive exploration by the student.

<u>Goal-oriented</u>: Any educational activity that the students are asked to complete should have clearly defined goals. For laboratory experiments, the goal is to understand how the simplified theoretical model of the experiment to be conducted represents the behavior of a real engineering phenomenon. Therefore, the developer must aim at providing the students with an efficient method for making the connection between the simplified virtual model and the underlying complex engineering phenomenon (Figure 1).

<u>Challenging</u>: The level of difficulty of any learning task has to be somewhat beyond the students' current capabilities, but not so far beyond that they cannot accomplish the task. In a game-based virtual laboratory environment, the students face challenges such as:

- understanding the limitations of the virtual environment

- controlling the avatars (i.e. game characters controlled by students)
- understanding the game objects' programmed functionality and purpose
- interacting with and manipulating the game objects
- interacting and collaborating with other students and instructors
- following the instructions implemented in the scripted laboratory scenarios
- analyzing and interpreting the experimental results
- understanding when things go wrong and recognizing undesired outcomes

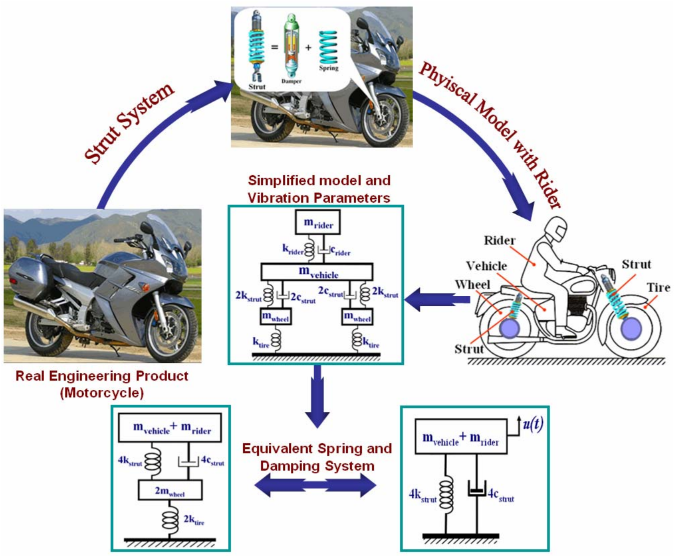

FIGURE 1:
CORRELATION BETWEEN ACTUAL MOTORCYCLE AND VARIOUS MODELS [4]

Anchored: All individual actions within a learning activity must have a meaningful effect on the final learning outcome, i.e. there should be no actions that do not directly or indirectly contribute towards achieving the goals. In fact, students learn best when they understand the reason for taking the steps they are asked to take. In a virtual laboratory setting, the major actions scripted in the laboratory scenario as well as their role in the learning process must be explicitly provided to the students and be comprehensible for them.

Exploratory: Learning is said to be most efficient when the students have to make choices while engaging in an educational activity and then face the consequences of those choices. This represents an exploratory learning mode. This means that a virtual laboratory environment should have a certain variety of possible choices and give the students the ability to try different things. The search for additional complementary knowledge can be encouraged by providing means for gathering additional information (e.g. from the Internet, databases, etc.) and interacting with other students (e.g. through voice, instant messaging, etc.).

Attention-drawing: Properly motivating the students before a learning activity can significantly improve the learning outcomes by speeding up the learning process and increasing the retention rate [9]. In the context of developing virtual laboratory environments, emphasis should be placed on making them visually appealing (e.g. high-quality renderings), customizable (e.g. changing the avatars' appearance, adjusting the controller settings), realistic (e.g. sound effects, proper physics, true-motion animation) and engaging (e.g. interactive and collaborative).

Feedback-providing: In general, feedback keeps students motivated and engaged. Therefore, a virtual laboratory environment should provide the following types of feedback:

- Error warnings
- Time constraints
- Student performance

Effective Pedagogies in Educational Laboratories

The pedagogical effectiveness of any educational activity is judged by whether or not the intended learning outcomes are achieved. In general, engineering laboratories aim at instilling the students with various competencies, such as a thorough understanding of the physical phenomena and principles that underlie certain engineering systems, the ability to measure, analyze and subsequently optimize the behavior of those systems, as well as professional skills (e.g. communication, collaboration, planning, project management, etc.). Therefore, in designing a virtual laboratory environment, the above-mentioned goals should be addressed.

In the context of game-based virtual laboratory environments, successful educational experiences represent the combination of three delivery elements [10]. Simulation and game elements should be surrounded by pedagogical or didactic elements, thus ensuring that the students' time is spent productively. However, if there are too many pedagogical elements, the student may feel they are mindlessly following directions given in a laboratory manual.

Pedagogical elements: Some of the desirable pedagogical features of laboratory exercises are that they should [10]:

- offer background material and the underlying theory
- give an introduction to the experiment to be performed
- guide the students through the exercise via text, voice, graphics, etc.
- facilitate collaboration between the students
- force the students to reflect on their actions and the resulting consequences
- give diagnostic capabilities (i.e. student self-evaluation)
- provide means of communication (e.g. access to forums or chat rooms)
- offer acronyms or other mnemonic devices to trigger the memorization of processes
- enable error prompting and help

Game Elements: Game elements provide familiar and engaging interactions and increase the appeal of an educational experience. Thus, they create goodwill in the students, but more importantly, they tend to increase time on task, which undoubtedly improves learning. Some of the most important game elements that should be incorporated into a virtual learning environment are:

- Use of first person game perspective so that each student feels as close to the real experience of a laboratory course
- Customizing the avatars' appearance so that they look like casually attired (Figure 2) enhances the students' feel of immersion
- Tasks that challenge the students and drive collaboration
- Switching between simplified and detailed interfaces while providing an appealing access to in-depth information on relevant concepts
- Certain exaggerations in responses of interaction between game objects, such as damage after collisions
- A pause button that would allow the students to take a time-out (e.g. to find complementary information)
- A speed-up/slow-down switch to provide the students with a means to perform the tasks at their desired pace
- A replay option to have a chance to repeat a task
- Accessible communities for sense of belonging through virtual laboratory multi-user servers with interaction among the students
- Choosing between multiple skill levels to better align difficulty with capability (i.e. raising the challenge as the students' capabilities increase)

FIGURE 2
MODIFIED SAMPLE AVATAR

Simulation Elements: Simulations are based on model representations of reality. Simulation elements can rigorously or selectively represent objects, situations as well as user interactions, depending on the limitations of the available computing infrastructure. Different simulation elements enable discovery, experimentation, concrete examples, practice and active construction of systems [10]. In the learning process, the use of simulation elements provides the students with a deeper understanding of the traditional educational materials.

IMPLEMENTATION

The implementation of a collaborative virtual laboratory environment for undergraduate engineering is an attempt to mimic important aspects of the learning experience gained through conventional hands-on experiments. Such a virtual environment can even augment the students' interactions by certain features that are not achievable in the traditional laboratory setting [11]. A suitable means for developing such computer applications is a commercial multiplayer game engine. With a game engine as the foundation of the learning application, scenarios must be scripted and structured in order to implement the desired pedagogical intent.

Software Tools

"Source" game engine: The computer game "Half-Life 2" [12] is classified as a "first person shooter" game and utilizes the "Source" game engine [13]. In this game engine, the first person perspective is employed (i.e. the screen display matches what one would see if being inside the game), which gives the user a more immersive experience (i.e. a feeling of being physically present in the simulated 3D game environment). Also, the game incorporates the "Havok" physics engine [14], which simulates Newtonian physics and includes tools that mimic real world phenomena (e.g. the oscillating behavior of springs-mass systems, the application of accelerations and torques, collisions, etc.). Furthermore, by using a "PhysX" physics processing card [15] in conjunction with the "Havok" physics engine, the interactions between the objects of the virtual environment can be made even more realistic. Outside the realm of entertainment, computer game technology was first used successfully for military applications [16]. By using "Half-Life 2" and "Havok" for the development of a virtual laboratory environment as described here, students are enabled to maneuver within a simulated laboratory environment, interact with experimental setups, communicate with each other and carry out experimental procedures cooperatively with other students who are simultaneously connected to the same game server.

"Source" SDK: The corresponding "Source" Software Development Kit (SDK) [17] is available as an optional download for users of the "Half-Life 2" game. Included in the SDK are a number of software tools, which let users modify existing content as well as add new customized content into the game. For creating a modification of the game, users are also enabled to create a copy of the original game C++ source code, modify it and finally recompile it.

"Hammer" map editor: One of the software tools included in the "Source" SDK is the "Hammer" [18] map editor (Figure 3). It provides the functionality for creating customized virtual environments that the users interact with when running the game. The creation of simple primitive geometric shapes is enabled, which can be further refined using Boolean addition and subtraction with other primitives. Third-party 3D modeling software can be utilized to create the more complex models. Subsequently, these models can be converted to a game-compatible format and imported into the virtual environment. The spatial boundaries of the virtual environment are defined using flat rectangular prisms, thus forming a closed volume with walls, floors and ceilings. Next, materials can be assigned to the primitives, providing visual details, physical behaviors as well as proper sound effect responses. Furthermore, a vast library of functions can be embedded into the primitives (e.g. a small cube functioning as a button capable of outputting commands; a flat rectangular geometry function as a movable door triggered by input commands or game characters). Additional logic entities are available to relay output commands in order to aid in scripting causality within the game.

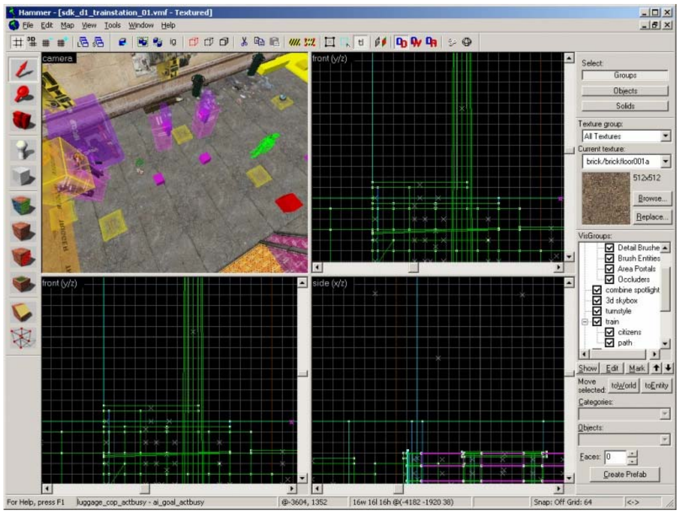

FIGURE 3
BUILDING INTERFACE OF "HAMMER" MAP EDITOR

Scenario Structuring and Scripting

In the context of computer game programming, scripting is a concept that is similar to that of a storyboard for a movie. The result of the scripting process is a scenario. This scenario has to be developed in order to establish a meaningful sequence of events. Within the programmed logic of a game script, there is a set of rules that governs how a user goes through the scenario in order to reach the desired goal of the game.

When adopting the concept of a scripted scenario to the development of virtual laboratory environments, the educational outcome becomes the goal (of the laboratory exercise) and the layout of the major laboratory tasks in order to achieve this outcome is structured in a pedagogically sound way in the form of a laboratory script. By this script, the developer defines the interactions amongst the students and between the students and the objects forming the virtual laboratory environment. These interactions are triggered as one or more students carry out individual or collaborative tasks set forth in the programmed laboratory script (e.g. the measurement of the vibration of a certain object using a virtual accelerometer). Furthermore, the laboratory script also incorporates commands that activate the interactions between the virtual objects (e.g. the multi-meter created using "3ds Max 9" [19] shown in Figure 4) and enables the integration of remote experiments (based on actual hardware) and virtual experiments (representing pure computer simulations) into the laboratory scenario (Figure 5).

Using this framework, the educational content can be tailored to address the students' different learning modalities. A number of predefined scenarios can be scripted, which exercise the students' problem solving skills by mimicking typical problems that might occur when carrying out actual hands-on experiments [20]. In addition, the experimental scripts imbedded within the system allow one to monitor - and possibly even enforce - active participation and collaboration by all students of a laboratory group, which are considered two crucial factors in improving learning.

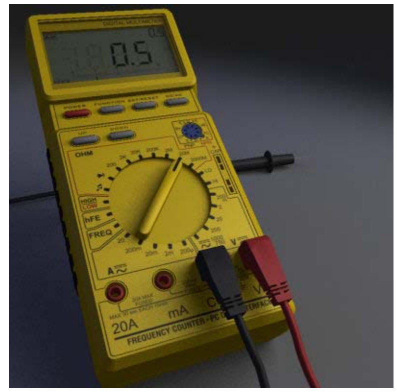

FIGURE 4
MULTI-METER MODELED USING 3DS MAX 9

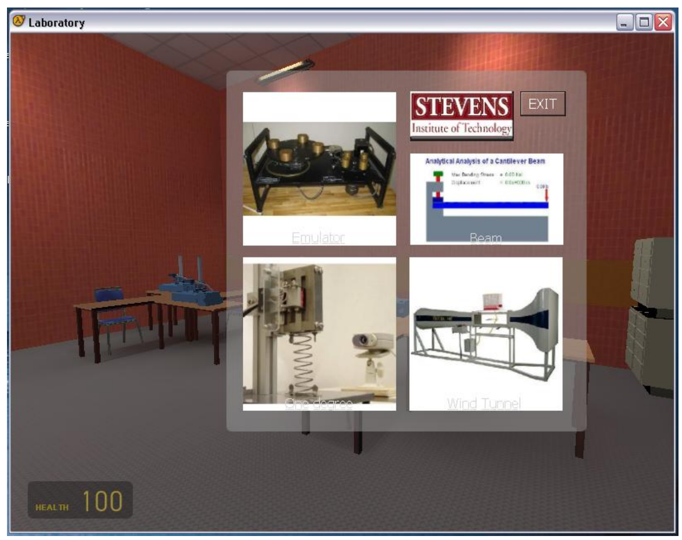

FIGURE 5
INTEGRATED ACCESS TO LABORATORY SIMULATIONS

Performance feedback is essential for reaching the desired educational outcomes. While performing the laboratory exercises, the students face a variety of choices. Different forms of feedback can be used to reinforce correct actions, ranging from subtle (e.g. positive sounds, correct text pop-ups) to elaborate feedback (e.g. performance charts) while wrong choices can also trigger corresponding feedback (e.g. a spark if a cable is plugged into the wrong outlet). All these interactive types of feedback are part of the laboratory scenario scripting. In addition, the actual remote experiments or experimental simulations launched from inside the game environment can generate data driven feedback of their own.

The game engine itself is capable of tracking some of the activities carried out by the game users when interacting with the virtual environment. Therefore, by adding certain scripts to the laboratory scenario, it is possible to monitor the students' activities. A log of these activities can then be used by the instructor for grading purposes or for isolating areas of deficiencies of the students. For instance, when according to the log a particular student shows poor performance in a specific aspect of the laboratory procedure, the instructor can then address this problem with that particular student within the context of a one-on-one interaction. On the other hand, if considerable numbers of students are underperforming regarding a certain aspect of the laboratory exercise, then the problem likely actually lies within the laboratory implementation itself, and thus the scripted scenarios might need to be analyzed and improved accordingly.

Finally, collaborative participation of all students of a laboratory group in performing the required laboratory tasks can also be monitored using the tracking features inherent to the game engine. Group interactions should be encouraged by strategically designing the laboratory environment and the laboratory scenarios such that interacting cooperatively will yield more efficient results. Therefore, the ability to log and monitor the students' activities can be a means for encouraging or even enforcing group participation.

SAMPLE SCENARIO

Consider a laboratory exercise used at Stevens in the laboratory component of a junior-level mechanical engineering course on mechanisms and machine dynamics. The exercise involves an industrial emulator designed for experiments with different rotating bodies connected by a gear-belt mechanism (Figure 6). It allows one to establish the inertia of the device itself and of the weights placed at various locations within the mechanism as well as to experiment with different gear ratios and belt stiffnesses. The laboratory assignment for the laboratory exercise using this industrial emulator involves the following: (a) assembling of mechanism in right logical order according to a manual, (b) running experiments with/without weights and rigid/flexible belt, (c) analyzing experimental results, and (e) writing report.

FIGURE 6
EXPERIMENTAL SETUP OF INDUSTRIAL EMULATOR

A sample laboratory scenario can be scripted in the game-based laboratory environment as follows:

- When logging into the virtual laboratory environment, the students first enter the building housing the laboratory facility (Figure 7) and proceed to the appropriate room.
- First, a pre-test can be given to the students assessing their level of preparation for the laboratory to be conducted. Depending on the outcomes of the pre-test, the students can be directed to remedial instruction (in text, audio or video format).
- Then, the students are presented with the laboratory description in text form (Figure 8) or in audio/video form.
- Next, the students negotiate the division of the various laboratory tasks amongst the group members using communication features (e.g. instant messaging, voice/video, Figure 9).

FIGURE 7
BUILDING HOUSING THE VIRTUAL LABORATORY FACILITY

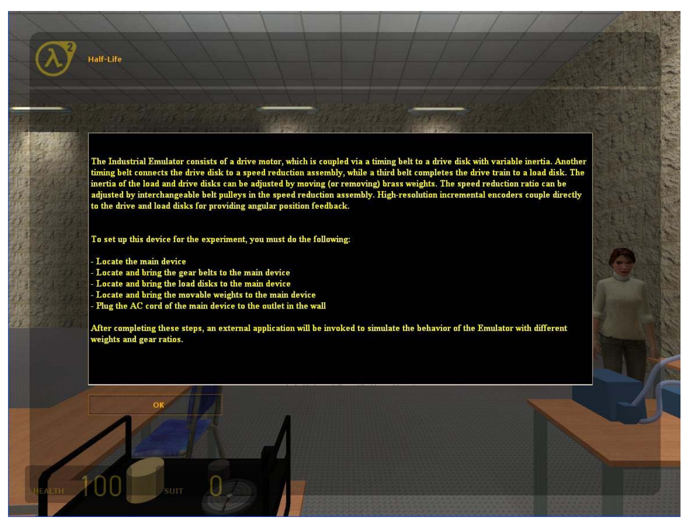

FIGURE 8
IN-GAME INSTRUCTIONS USING TEXT WINDOW

FIGURE 9
IN-GAME TEXT-BASED COMMUNICATION BETWEEN STUDENTS

- Then, the students collaboratively assemble the experimental setup according to the provided laboratory instructions using a set of in-game contextualized tools (Figure 10).
- Next, the students launch an input/output menu to carry out the actual experimental procedure (either a simulation or a remote experiment, Figure 5).
- Upon completion of the experimental procedure, the students analyze the results and divide the report writing tasks amongst each other.
- Finally, a post-test can be given to the individual students to assess the learning outcomes.

FIGURE 10
IN-GAME MENU OF CONTEXTUALIZED TOOLS

CONCLUSIONS

A virtual laboratory environment based on a computer game engine was introduced as an alternative for existing online laboratories in the form of remote experiments or software simulations). This laboratory approach represents an attempt to mimic important aspects of the learning experience gained through conventional hands-on experiments and to even augment this experience by certain features that are not achievable in the traditional hands-on laboratory setting. First, general design requirements for game-based virtual laboratory environments were discussed, considering both pedagogical as well as technical aspects. Then, the practical implementation of these design requirements in a prototype system was described. Finally, a sample scenario for a virtual laboratory based on an existing laboratory exercise from a junior-level mechanical engineering course on mechanisms and machine dynamics was presented.

ACKNOWLEDGMENT

The multi-disciplinary research described here is being carried out under a grant by the National Science Foundation's Information Technology Research program [21]. This support is gratefully acknowledged.

REFERENCES

[1] Portfolio of free simulations from the University of Florida, http://vam.anest.ufl.edu/instructorsims/simulationportfolio.php
[2] Integrated Land Use, Transportation, Environment, Modeling System, http://www.civ.utoronto.ca/sect/traeng/ilute/ilute_the_model.htm
[3] Nanorobotics Simulation - Computational Nanomechatronics Laboratory at Center for Automation in Nanobiotech (CAN), http://www.nanorobotdesign.com/
[4] Aziz, E.-S., Esche, S. K. & Chassapis, C. (2007). Enhancing the learning experience using simulation and experimentation to teach mechanical vibrations. *2007 ASEE Annual Conference and Exposition*, Honolulu, Hawaii.
[5] "World of Warcraft"; http://www.worldofwarcraft.com/index.xml
[6] "Everquest II"; http://everquest2.station.sony.com/
[7] "Second Life", Inc.; http://secondlife.com/
[8] Quinn, C. N. (2005). Soapbox: making learning fun. http://www.gamasutra.com/features/20050818/quinn_pfv.htm
[9] Quinn, C. N. (2005). Engaging Learning: Designing e-Learning *Simulation Games*. Pfeiffer.
[10] Aldrich, C. Six criteria of an educational simulation; http://www.learningcircuits.org/NR/rdonlyres/F2ED000A-7A59-4108-A6CB-1BE4F4CC1CA5/4719/clark_e2.pdf
[11] Chang, C., Arango, F., Kodman, D., Esche, S. K. & Chassapis, C. (2006). Utilization of immersive collaborative student laboratory simulations developed using a game engine. *ASME International Mechanical Engineering Congress and Exposition*, Chicago, Illinois.
[12] "Half-Life 2" computer game; http://www.half-life2.com/
[13] "Source" game engine; http://www.valvesoftware.com/
[14] "Havok" physics engine; http://www.havok.com/
[15] "PhysX" physics processing card by AGEIA; http;//www.ageia.com/
[16] Manojlovich, J., Prasithsangaree, P., Hughes, S., Chen, J. & Lewis, M. (2003). UTSAF: A multi-agent-based framework for supporting military-based distributed interactive simulations in 3-D virtual environments. *Winter Simulation Conference*, Vol. 1, pp. 960-968.
[17] "Source" SDK, http://developer.valvesoftware.com/wiki/SDK_Docs
[18] "Hammer" map editor**,** http://developer.valvesoftware.com/wiki/Hammer
[19] "3ds Max 9", http://usa.autodesk.com/adsk/servlet/index?id=5659302&siteID=123112
[20] Chang, C., Kodman, D., Chassapis, C. & Esche, S. K. (2007). Immersive collaborative laboratory simulations using a game engine. Accepted for publication in *Computers in Education Journal*, Vol. 17, No. 3, 2007.
[21] Esche, S. K., Corter, J. E., Nickerson, J. V. & Chassapis, C. (2003). ITR: An Infrastructure for Designing and Conducting Remote Laboratories, NSF-ITR Grant No. 0326309.