# Game-based InfoSec Education Using OpenSim

Jungwoo Ryoo, Angsana Techatassanasoontorn, Dongwon Lee, and Jeremy Lothian, *Pennsylvania State University*

***Current information security education approaches tend to focus on theories and concepts. Although these conventional education strategies have their own advantages, students can also benefit from pedagogical strategies that are more interactive and scenario-driven. In particular, the current net-generation of students are often more likely to prefer learning in a feedback-rich and contextualized environment. Therefore, an environment in which learning occurs in a game-like context can be highly effective in teaching students information security topics, especially in introductory courses. Based on these observations, this paper proposes a novel security educational environment that aims to improve student engagement and learning effectiveness by using the recent developments in 3D, Web-based virtual world technologies in a game-like setting.***

**Index terms – Information Security Education, Game-based Learning, OpenSim**

## I. Introduction

In the past few years, the number and sophistication of threats in cyberspace have dramatically increased. For instance, according to the Federal Trade Commission, 3.7 percent of the survey population in the U.S. was victims of identity theft during 2005 [1]. Likewise, the lack of appropriate cyber security policy poses tremendous risks to the nation's security as suggested in the report to the President, entitled "Cyber Security: A Crisis of Prioritization" [2]. Among five key strategies that the U.S. has been pursuing is cyber security education that creates awareness about cyber vulnerabilities, trains appropriate security personnel, and promotes safe computing [3].

With a growing number of threats, securing cyberspace becomes an increasingly more important economic, social, and technical problem. The consequence of the complex and highly interconnected network is that computer security has become highly dependent on individuals' security practices. Therefore, enhancing cyberspace security requires both **behavioral** measures (i.e., engaging in safe and secure behaviors) and **technical** protection of computer systems (i.e., appropriate hardware/software installations and their proper use). For many users, this requirement can be a daunting task because computer security remains an abstract notion. As a consequence, there is a pressing need to effectively educate users on how to protect themselves against attacks.

We believe that the current efforts in security education have several limitations: (1) a narrow focus on technical aspects of security, (2) lack of user-centered learning modules, and (3) limited guidelines from learning theories. This paper describes the development of the **Immersive Security Education Environment** (**I-SEE**), intended to address these shortcomings. We utilized the integrated information assurance model by Maconachy et al. [4] as a framework to guide the development of contextualized security learning modules.

Our solution aims to develop a creative learning module in the immersive 3D environment of OpenSim [5], an open source 3D virtual world similar to that of Second Life with more than 3 million registered users around the world. This paper describes one of the learning modules to help students obtain a basic understanding of the sequence of events that occur during a security attack from the perspectives of both the attackers and defenders.

## II. GAME-BASED LEARNING (GBL)

A subset of educational Serious Games, game-based learning (GBL) uses competitive exercises to motivate students learning according to specific learning objectives [6]. The students can compete either with other students or against themselves. The games used in GBL usually employ an interesting narrative deliberately designed to engage the students in their learning. In GBL, scoring is essential to properly develop interest among the students. Although GBL does not necessarily have to rely on computers, the type of GBL discussed in this paper only refers to those implemented in digital media.

In addition to more effectively motivating students to learn, immersive GBL environments also encourage them to learn from and adapt to each other's tactics and play styles [7].

*Jungwoo Ryoo is an assistant professor of Information Sciences and Technology (IST) at the Pennsylvania State University (PSU)-Altoona. Angsana Techatassanasoontorn and Dongwon Lee are assistant and associate professors in the college of IST at PSU. Jeremy Lothian is a Ph.D. student in the college of IST at PSU.*

ISBN 1-933510-96-X/$15.00 © 2011 CISSE

These attributes are of particular importance when teaching the “net-generation” of students. As Van Eck suggests, these students “require multiple streams of information, prefer inductive reasoning, want frequent and quick interactions with content, and have exceptional visual literacy skills”, all of which, he notes, match features provided by GBL [8]. Immersive environments used in GBL activities not only provide these components, but also offer them in a potentially very detailed, engaging, and interactive manner.

Even in instances where the GBL task do not facilitate better performance, they do not generally result in worse performance, and may have the additional benefit of a more positive attitude toward the subject of the GBL [9]. This is important, particularly for introductory classes, as it could help encourage students to further pursue topics they otherwise would not have.

Additionally, a study found that when using an immersive environment, students were not only much more engaged with the material, but also showed significantly more improvement over those participating in learning with other educational software [10].

For more information on the theory, design, and evaluation of GBL activities and Serious Games refer to de Freitas [11] and Wilson et al. [12].

We believe that the positive characteristics of GBL, especially those of immersive environments, are highly compatible with the educational goals we set out to pursue in the I-SEE project.

## III. OPENSIM

Open Simulator (OpenSim) is an open source, multi-user 3D virtual world [5]. It is modeled after, and compatible with, the popular commercial virtual world, Second Life [13]. OpenSim can be used to create one or more virtual regions that can be accessed by various clients including the Second Life client. Unlike Second Life, which is fee-based, users can create custom 3D objects on their OpenSim islands free of charge as long as they have a server that can host the OpenSim server program. In addition, since the source code of the OpenSim server application is freely available, it is relatively easy to make necessary changes to the server framework to accommodate specific needs. OpenSim can be run on both Windows and Unix/Linux Operating Systems (OS).

Our I-SEE project was originally developed on the Second Life platform, but due to the many constraints imposed by Linden Labs (the company that owns Second Life), we decided to migrate to OpenSim. One of the obstacles we faced while implementing learning modules in Second Life was the use of a proprietary programming language, Linden Scripting Language (LSL), required by Second Life. OpenSim offers a more open environment by accommodating programming languages supported by the .Net platform to run scripts on 3D objects. This includes C#, which is much more widely used than LSL. Another shortcoming of Second Life is the lack of control over the usage environment. Second Life clients are constantly upgraded, making it difficult to keep up with changes, and the regions are often unavailable due to server upgrades and other maintenance activities.

However, OpenSim is not immune to problems either. Since its development is completely community-driven, users are dependent on the willingness of volunteers to provide feature upgrades and bug fixes. In addition, stability of the server implementation between versions is not always guaranteed. A positive note is that the primary developers are readily accessible on mailing lists and seem eager to help when they can. Based on our own cost-benefit analysis, we concluded that OpenSim has many advantages when compared to Second Life for our goals.

## IV. THE LEARNING MODULE

The learning module simulates a scenario in which two teams are competing against each other: one acting as a group of attackers (a.k.a. red team) and the other acting as a group of defenders (a.k.a. blue team). Each team has a base consisting of a packet assembly board (Figure 1), a building (Figure 2) containing a money counter and a console, and a pipeline (Figure 3) representing a network connection from the base to the gateway router (Figure 4).

The game begins when each team successfully creates a data packet according to the TCP/IP protocol on the packet assembly board. Using their team avatar, the team must select the packet headers in the appropriate order.

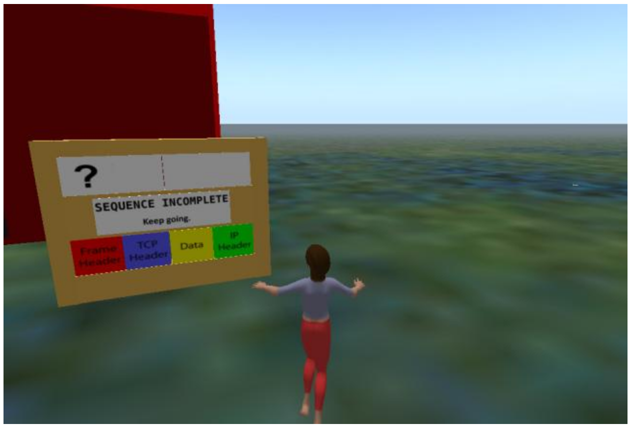

**Figure 1**: The Packet Assembly Board

The successful packet assembly will initiate the flow of packets in the pipeline as shown in Figure 2 and Figure 3.

ISBN 1-933510-96-X/$15.00 © 2011 CISSE

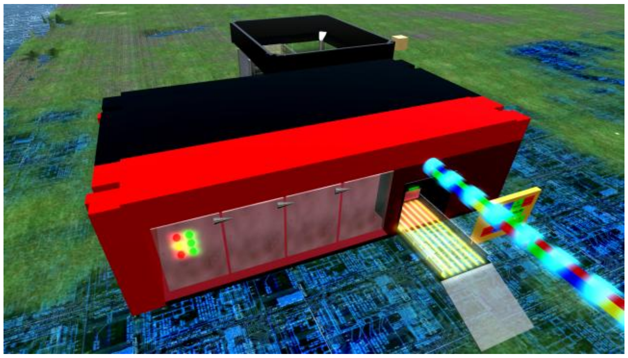

**Figure 2**: Red Team Base

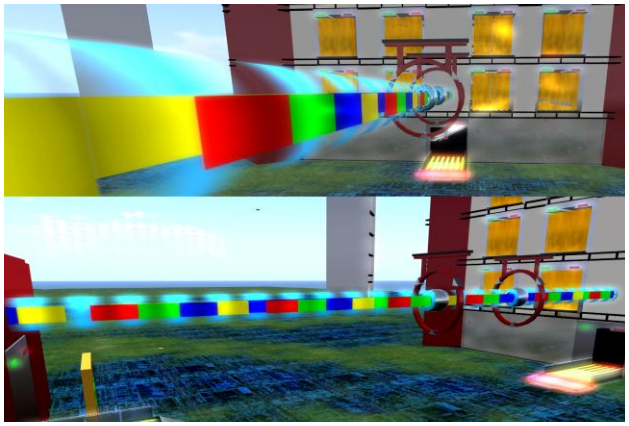

**Figure 3**: Pipeline Showing Packet Flow

The frames flow into the gateway router building as shown in Figure 4. The gateway is connected to the cloud (i.e., the Internet).

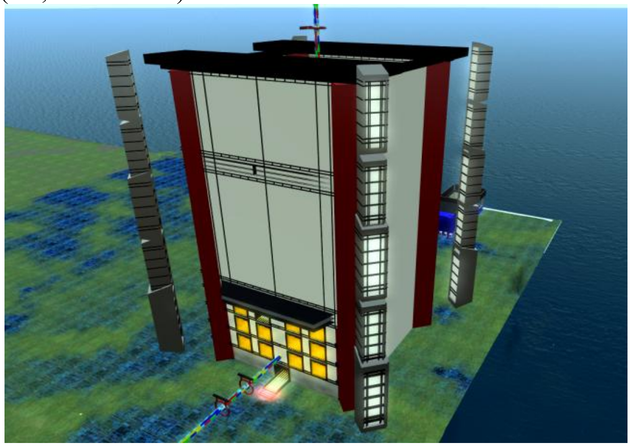

**Figure 4**: Gateway Connected to the Cloud (Internet)

As soon as frames start moving, the red team can begin launching an attack. To initiate an attack, the avatar needs to go inside the red team base and use the attack console to initiate a series of attack events (Figure 5).

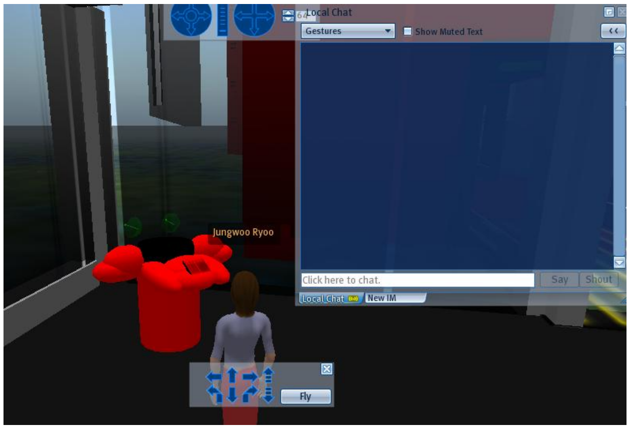

**Figure 5**: The Attack Console and Chat Window

Similar to a typical attack, the red team starts with *reconnaissance*. To scan the network for potential targets, the scan command with an IP address as a parameter needs to be entered into the chat window. Therefore, it is necessary for the red team to obtain potential IP address ranges of the blue team's computer. These IP addresses are embedded in small objects at a location next to the blue team base. The red team scans different IP addresses until it finds the one with known vulnerabilities (i.e., open ports flagged as being vulnerable). Next, the red team is able to initiate an attack, which is triggered when the team issues a command on its console such as: attack [IP address] [port number].

As the red team makes consecutive successful attacks, its money balance will increase as shown in Figure 6. The team (whether red or blue) with the highest balance will win the game.

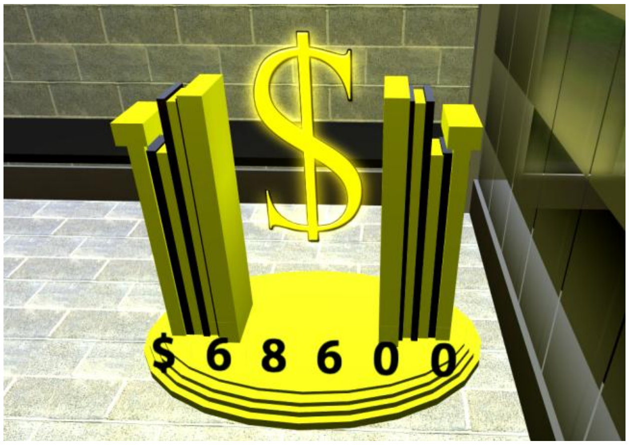

**Figure 6**: Money Counter Showing the Current Balance

When scanning is under way, an oversized magnifying glass will appear by hovering above the base of the blue team as shown in Figure 7.

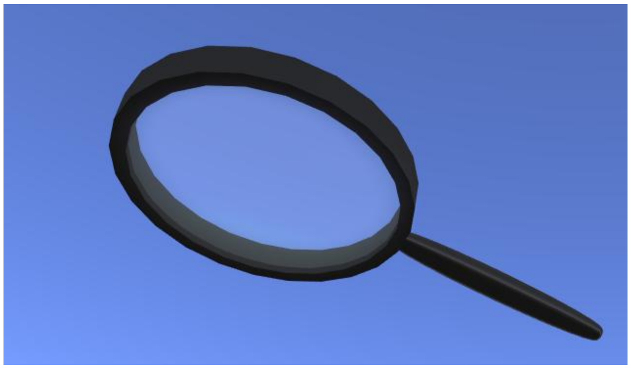

**Figure 7**: Scanning Process Visualized as an Oversized Magnifying Glass

The magnifying glass offers a visual cue as a warning sign that encourages the defending team (blue team) to take proactive actions to prevent the attack from happening. There are two ways to prevent attacks. One is to close all the communication ports residing in the blue team base as shown in Figure 8. Each port is labeled with a port number. For example, the first port is labeled as 80, the port used by Hyper Text Transfer Protocol (HTTP). The light representing an active port is lit blue while the light representing the closed ports are turned off and glow in red.

The drawback of closing all the ports is that the blue team cannot perform any normal business operations. While the blue team should fend off any potential attacks from the red team, it also has a parallel requirement to maintain an e-commerce site and offer Web hosting services through a secure shell server. When the ports for HTTP and SSH are open, the blue team is making money which can be observed through a money counter similar to the one shown in Figure 6. However, the money counter stops when the ports necessary for the business transactions are closed. Therefore, the blue team needs to use the strategy of closing all ports sparingly.

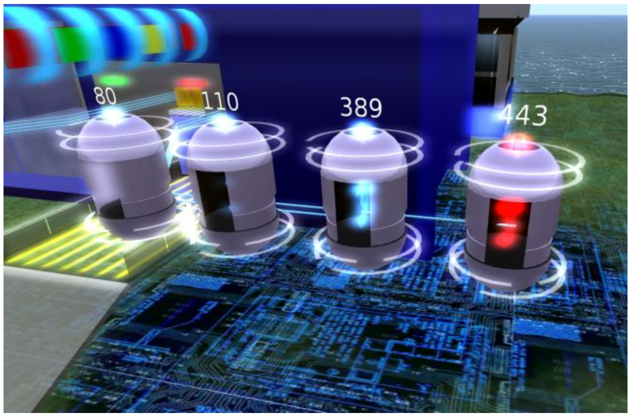

**Figure 8**: Ports at the Blue Team Base

The second strategy for preventing the potential attacks is to retrieve patches available in the form of objects next to the red team base. If the blue team applies its patch for the known vulnerabilities existing in its network (represented by the blue team base) before the red team starts launching its attack, the blue team can successfully thwart the attack attempts made by the red team. The patches are not available until the blue team subscribes to the patch delivery service as shown in Figure 9.

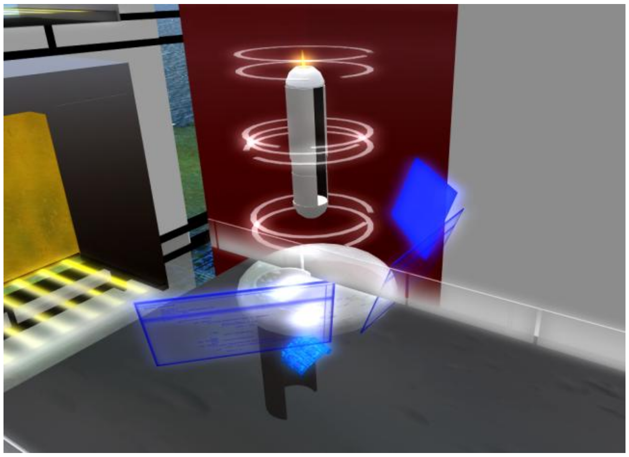

**Figure 9**: Patch Delivery Service

To make the defense process more challenging, the blue team is required to answer a quiz question before it is allowed to apply the patches. The quiz question tests the information security knowledge of the blue team members by asking questions such as: "Is a plain FTP service available via the TCP port 20 and 21 safe? (Yes/No)." If the blue team members anwers the question correctly, the patch is automatically installed. Otherwise, the blue team needs to answer another question. The quiz questions are triggered by touching the Consultant Bot shown in Figure 10.

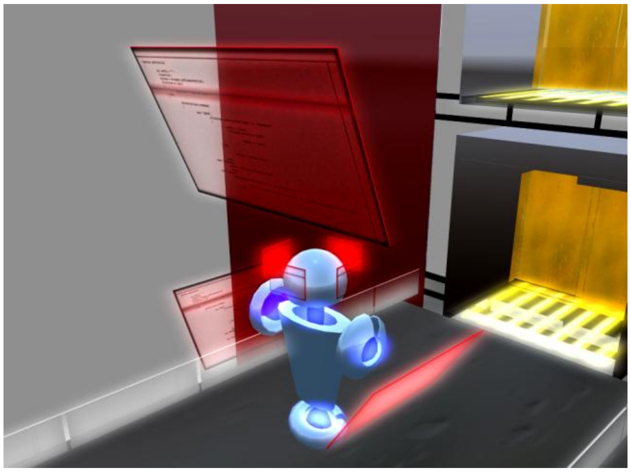

**Figure 10**: The Consultant Bot Issuing Quiz Questions

Periodically, the blue team will receive an alert from one of the customer bots in the main gateway building (Figure 11). The customer bots pose quiz questions to the blue team. A typical question given by the customer bot would be related to computer security practices for everyday users. For example, one of the questions could be: “Could I use my pet name as my password? (Yes/No).” If the blue team fails to provide a correct answer, the money counter will decrease its current balance. Otherwise, the balance grows if the answer is correct.

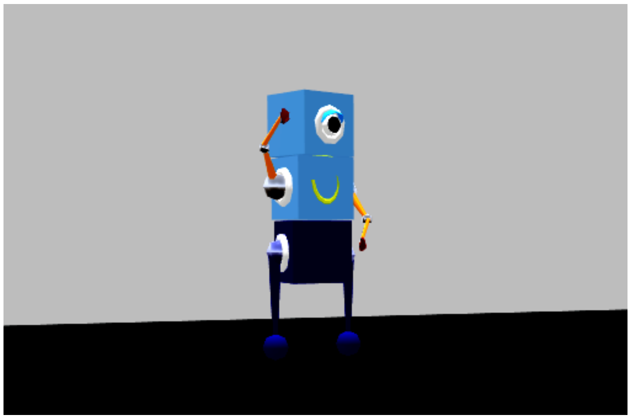

**Figure 11**: The Customer Bot

A divide-and-conquer approach is desirable for the blue team to effectively address the many different tasks it is supposed to perform. For instance, some of the blue team members can deal with patch application while the other team members work on the customer bots. This way, the game tests not only the security knowledge of the blue team members but also their organizational skills.

## V. Deployment

To expedite the adoption process for those interested in incorporating our learning modules in their curricula, we are making efforts to streamline the deployment process as much as possible.

After considering various deployment options including the creation of a custom Linux distribution fully configured to run our leaning module, we opted for developing a virtual machine (VM) image containing a working OpenSim environment with the 3D objects necessary to play the educational game as well as a Web-based management program that helps instructors configure the learning module and reset it if necessary.

This way all that needs to be done is to copy the master VM image into a new environment. No additional installation and configuration steps are necessary. Instructors can deploy the learning module within minutes with a quick start guide provided with the VM image.

Students can also use the management system to obtain more detailed feedback regarding how well they are playing the game. For example, they can view the fine-grained breakdown of which of their attack/defense activities contribute to the current money balance both negatively and positively.

## VI. ASSESSMENT

We plan to incorporate the proposed I-SEE learning module into the Fundamentals of Information Security courses at four Penn State campuses during the fall semester in 2011. Our project goal is to improve the security knowledge and skills of the students. Therefore, our evaluation plan is to assess changes (pre- and post-usage of the system) in three key constructs: (1) self-efficacy in the ability to take actions against security attacks, (2) perceived subject matter learning, and (3) perceived skill development.

## VII. DISCUSSION

The game described in this paper demonstrates all the major characteristics of effective GBL. The competitive nature of the game appeals to the students and is designed to be engaging. Scoring is done through money balance. Since the learning module is still going through beta testing, no assessment of the effectiveness of our approach has been conducted.

The beta testing process has revealed some shortcomings that we are currently addressing. One of the improvements we currently work on is the expansion of the red team activities. This team has a relatively easier set of tasks compared to the number and complexity of the blue team activities. One of the solutions we are developing is to introduce a mechanism to force the red team to go through a set of quiz questions before granting access to the vulnerability details of the blue team. The questions could be similar to those used for the patch retrieval by the blue team. These quiz questions can also be invoked when the scan or attack commands are issued.

Since the security learning module is only *simulating* a red team-blue team exercise, some may argue that the true value of such a game is somewhat limited because students do not actually get exposed to all the technical details of how to exploit a vulnerability on a victim’s system or how to best harden a host.

On the other hand, it can be argued that our learning module is highly suitable for students who are not quite ready to handle all the technical intricacies of the red team-blue team attack-defense exercise. Our learning module offers students a broad understanding of a series of events that lead up to attacks and ways to prevent them.

We believe that both of these arguments are reasonable. In fact, similar arguments can be made for teaching students how to program. User-friendly tools such as Alice [14] help students understand important programming concepts without the in-depth knowledge of a programing language such as Java.

In the case of the I-SEE learning module presented in this paper, it may be best to use the system to trigger students' interest in learning about cyber security and to promote a high-level understanding of security attacks and defenses in the introductory courses. Once the students grasp the overall idea and are eager to learn more, a real red team-blue team exercise can be conducted. Note that setting up an environment for such an exercise often becomes a monumental task due to the many technical hurdles to overcome. The I-SEE learning module provides a portable and time-saving solution to those mainly seeking a higher order understanding without the need of knowing the implementation details.

## VIII. Conclusion

This paper presented I-SEE, a novel 3D information security learning environment. We focused on a particular learning module of I-SEE, which is GBL-based. The proposed learning module provides a simulated cyber attack/defense experience. One of the main benefits of the system is its quick and easy deployment process. In addition, the user-friendly nature of our solution largely removes complex configuration steps and promotes a learning without overwhelming the students with unnecessary low-level technical details.

## IX. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0817376. We would also like to thank three undergraduate research assistants, Zafer Bilaloglu, Martin Bakiev, and David Kopczyk for their help in implementing I-SEE.

## X. References

[1] Federal Trade Commission (2007) Federal Trade Commission – 2006 Identity Theft Survey Report, http://www.ftc.gov/os/2007/11/SynovateFinalReportIDTheft2006.pdf.

[2] President's Information Technology Advisory Committee (PITAC) (2005) Cyber Security: A Crisis of Prioritization.

[3] The White House (2003) The National Strategy to Secure Cyberspace.

[4] Maconachy, W.V., Schou, C.D., Ragsdale, D., and Welch, D. (2001) "A Model for Information Assurance: An Integrated Approach," Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY.

[5] Open Simulator, http://opensimulator.org/wiki/Main_Page.

[6] Rebecca Teed, Game-Based Learning, http://serc.carleton.edu/introgeo/games/.

[7] Oblinger, D. "Simulations, games, and learning," *Educause Learning Initiative,* 2006.

[8] Van Eck, R. "Digital game-based learning: It's not just the digital natives who are restless," *EDUCAUSE review,* vol. 41, pp. 16-16, 2006.

[9] Ke, F. "A case study of computer gaming for math: Engaged learning from gameplay?," *Computers & Education,* vol. 51, pp. 1609-1620, 2008.

[10] Virvou, M., *et al.*, "Combining software games with education: Evaluation of its educational effectiveness," *Educational Technology & Society,* vol. 8, pp. 54-65, 2005.

[11] de Freitas, S. "Learning in immersive worlds," *A review of game-based learning. JISC e-Learning Programme,* 2007.

[12] Wilson, K. A., *et al.*, "Relationships between game attributes and learning outcomes," *Simulation and Gaming,* vol. 40, pp. 217-266, 2009.

[13] Second Life, http://secondlife.com/

[14] Alice, http://secondlife.com/

[15] Alavi, M. "Computer-mediated Collaborative Learning: An Empirical Evaluation,"MIS Quarterly, 18(2), 1994, pp. 159-174.

[16] Hiltz, S.R. The Virtual Classroom: Learning without Limits via Computer Networks, 1994, Norwood, NJ: Ablex.