Integrating Radiation Transport Models in a 3D Video Game to Train Nuclear Detection Techniques

James Winso[1], *Member, IEEE*, John Rolando[1], *Member, IEEE*, David Olivares[1], Henry Yu[2], Ronen Shaham[2], Cliff Halcom[2], Vythilingam Wijekumar[3]
[1]Spectral Labs Incorporated, San Diego, CA 92128 USA, [2]Kalloc Studios, Carlsbad, CA 92010 USA,
[3]Indiana University of Pennsylvania, Indiana, PA 15705 USA

***Abstract*– A novel technique has been developed under a Department of Homeland Security Small Business Innovative Research (SBIR) Program to exploit Video Games to dramatically improve the quality and availability of law enforcement and first responder training to locate and adjudicate nuclear sources. This training method is implemented in a software system which allows trainees to practice finding, identifying, and determining the threat level of radioactive sources in a 3D video game environment and returns results to a learning management system to enable tracking progress.**

I. Introduction

Spectral Labs Incorporated assembled a team that complemented Spectral Labs radiation measuring equipment design experience with the responder training experience of the Indiana University of Pennsylvania and the Video Game Engine development expertise of Kalloc Studios to develop the Realistic and Adaptive Interactive Learning System (RAILS) software based training system. RAILS enables a student or "trainee" to log into a web application that gives them access to written and video training content, allows them to communicate with instructors or other trainees, and, most significantly, enables them to launch the appropriate RAILS 3D video games as shown in Figure 1.

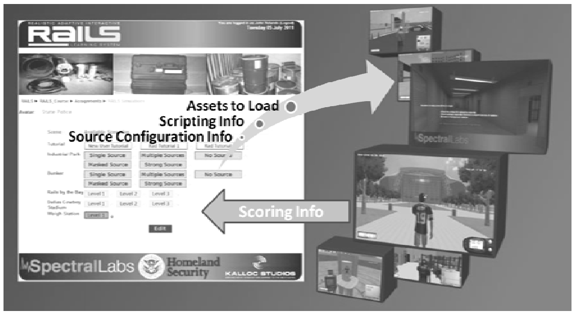

Fig. 1: RAILS Web Page shown on the left, interfaces with the Video Game environments.

The RAILS Games provide physically realistic PRND training by integrating radiation physics algorithms with the video game engine which is then used to generate training scenarios based on real world locations (e.g. the Dallas Cowboy's Stadium). The games include tutorial levels that enable the trainee to practice instrument operation and experience basic radiation dose rate concepts, as well as scored challenge levels, which when completed successfully unlock progressively more difficult levels.

Each challenge level is followed by a debrief in which the user is asked follow-up questions and given their scores for metrics such as time to find sources, ability to identify the sources and practicing of safety protocols.

RAILS also has a special interface for instructors so that they may add or modify training content, monitor user scores, communicate with trainees, and adjust which radiation source or sources are present, the source activities, and their locations for 3D simulations. Additionally, instructors can control other in game parameters like time limit and available instruments.

The Kalloc video game engine has been modified to accurately represent radiation behavior based on distance from the source, energy dependent shielding attenuation, and scatter effects by using buildup factors. SLI also tested the accuracy of the simulation through photon transport modeling and by comparing simulated environments with data collected from real world radiation environments. The initial user for this "serious game" is the Law Enforcement community, but its use will also benefit first responders, fire departments, medical and industrial radiation workers and nuclear power plant staff.

The Kalloc game engine used for RAILS was originally designed for the Play Station 3 platform. It has been modified to operate on laptops available in the law enforcement user community, such as the Panasonic Toughbook. Many of these laptops have relatively primitive graphic accelerators providing a unique software development challenge for Kalloc Studios. The current RAILS release has been tested on a wide range of laptops and shown to be compatible with virtually any system purchased over the last six to seven years.

II. Software Structure

RAILS consists of four discrete components: the video game engine (VGE), the game assets, the web based user interface program (UIP), and the RAILS daemon program. Figure 2 is a top level illustration of how the different RAILS components interact. Note that the RAILS User Interface Program can be located on a server hosted anywhere, so that anyone with internet access can log into RAILS through a web browser. Alternatively, the server can be hosted on a local department network, or even on the client PC.

This work was funded in part by the Department of Homeland Security (DHS), Domestic Nuclear Detection Office (DNDO) under contracts HSHQDC-09-C-00106 and HSHQDC-10-C-00119.

The RAILS Video Game Engine (VGE) is an executable program that is responsible for rendering the virtual environments that make up the 3D simulations. During each frame (i.e. every time a new screen image is rendered, about 30 times per second depending on platform), the RAILS VGE executes a function that calculates the Gamma Radiation flux at the PRND equipment's sensor location, taking into account all of the radiation sources in the virtual environment along with their gamma emission energies, all intervening shielding materials, and the simulated background. The VGE then translates that flux to a dose rate and counts per second for the specific instrument being simulated.

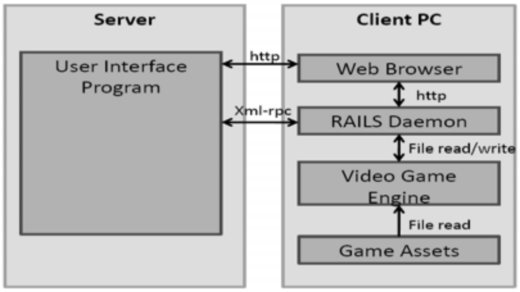

Fig.2. Simplified RAILS Software Component Interaction Diagram

The RAILS VGE calculates distance attenuation for both gammas and neutrons from multiple sources using $1/4\pi r_i^2$ where r_i is the distance between the ith source and sensor. Material attenuation is calculated as shown in (1), with μ (□$_i$) being the energy dependant linear attenuation coefficient for the jth material and t_j being the material thickness.

$$e^{-\mu(\varepsilon_i)t_j} \tag{1}$$

After validating the attenuation equation (1), Buildup factors, β, were added to the RAILS VGE as a correction for the narrow beam assumption made by (1), in which gammas that are scattered before reaching the detector are not accounted [1]. The attenuation equation shown in (2) is corrected with buildup factors, which are dependent on both the gamma energy (□) and the material thickness (t). The RAILS software determines β by using bilinear interpolation of table data preloaded into the game's configuration files.

$$\beta(t_i, \varepsilon_i)\ (e^{-\mu(\varepsilon_i)t_j}) \tag{2}$$

Once the distance and material attenuation factors have been calculated for each source, the results are multiplied by the source activity in Becquerel units to obtain the flux at the sensor caused by each source. The results are combined for each source to determine the flux observed by the virtual detector. Adding realistic background to the RAILS VGE greatly improved the perceived realism of the simulation. The readout on PRND equipment that gives dose rate and counts per second is never constant due to the statistical nature of background radiation. RAILS background, approximated by (3), is added to the flux calculated from the sources to give a more realistic result

$$\gamma_B = \gamma_{Bmean} + G(\gamma_{Bmean})^{1/2} \tag{3}$$

After the flux has been calculated, additional, PRND instrument specific algorithms are used to calculate the dose rate and counts per second displayed on the avatar's instrument as would be observed in the real world. This calculation is repeated for each frame as the game executes.

III. Interaction of Software Components

The RAILS VGE knows where to place radiation sources based on a configuration file that is dynamically created by the User Interface Program. This file specifies source types, locations, and activities. An additional configuration file houses the specific data for each source that defines the photo peaks and branching ratios for that source. Material data for any material that will be used in the simulations is also stored in that configuration file.

The VGE loads asset files when a simulation is started, and these asset files determine which objects will be present, as well as where they are located and with what orientation. Likewise, the loaded assets determine the character played by the Avatar, and any animation and sounds in the simulation. Unlike assets in conventional games, the RAILS assets have additional dimensions to account for the penetrating radiation. Thus an additional step is required for RAILS assets in which a material property is applied to the asset mesh so that the RAILS VGE knows which linear attenuation coefficients to use when that object obstructs the detector from a source.

Additionally, the VGE passes scoring information via an output file back to the RAILS Daemon which forwards the information to the User Interface Program when the simulation exits. This data enables integration of the game results into a Learning Management System. Fig. 3 illustrates how the RAILS VGE interacts with other RAILS components.

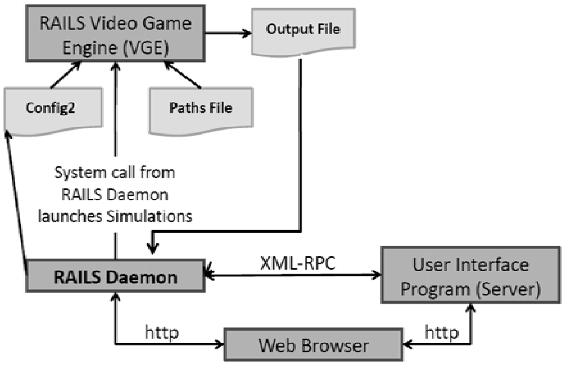
RAILS Video Game Engine (VGE)
Output File
Config2
Paths File
System call from RAILS Daemon launches Simulations
RAILS Daemon
XML-RPC
User Interface Program (Server)
http
Web Browser
http

Fig. 3. Diagram Depicting RAILS Software Component Interactions.

IV. In Closing

This paper was intended to provide a very cursory introduction of the Realistic and Adaptive Interactive Learning System. For additional information please download the video on the Spectral Labs website and scroll down to the "Download Demo Video" tab.

V. Acknowledgment

We thank our customer, the Domestic Nuclear Detection Office, for providing us with access to the members of their Training Advisory Forum that gave our team the opportunity to gain extremely valuable perspectives from scores of users throughout the country on the developmental RAILS versions.

References

[1] American National Standard N42.37-2006: *Training Requirements for Homeland Security Purposes Using Radiation Detection Instrumentation for Interdiction and Prevention*

[2] *Analysis of Transmitted Gamma-Rays by Multiple Scattering Method,* Yashiko Harima, Yosushi Nishiwaki, Journal of Nuclear Science and Technology, August 1970

[3] Moodle, version 1.9 (2007) http://moodle.org

[4] Python Programming Language – Official Website. http://www.python.org

[5] Glenn F. Knoll, *Radiation Detection and Measurement*, 2nd ed. New York: John Wiley & Sons 1989, pp. 56-57

[6] *Effects of Visual and Verbal Presentation on Cognitive Load in Vigilance, Memory and Arithmetic Tasks,* Barbara Tversky, Jeff Klingner, Pat Hanrahan, Stanford University

[7] American Nuclear Society. *ANSI/ANS-6.4.3-1991: gamma-ray attenuation coefficients and buildup factors for engineering materials*. La Grange Park: American Nuclear Society.