

## **3-D Virtual Reality for Education, Training and Improved Human Performance in Nuclear Applications**

**Chen Xi and Hsingtzu Wu**

Nuclear, Plasma and Radiological Engineering  
University of Illinois at Urbana-Champaign  
chenxi2@illinois.edu; u911730@gmail.com

**Arwa Joher**

Department of Business, University of Miami  
Coral Gables, Florida 33124-6520

**Leo Kirsch**

Lincoln-Way East High School,  
Frankfort, IL 60423

**Cheng Luo**

University Laboratory High School,  
Urbana, IL 61801

**Mohammed Khasawneh and Rizwan-uddin**

Nuclear, Plasma and Radiological Engineering  
University of Illinois at Urbana-Champaign  
mkha@ieee.org; rizwan@illinois.edu

### **ABSTRACT**

Three dimensional (3-D) virtual reality can be an effective tool for education, training and improved human performance in nuclear engineering applications. This technology can be used to model and simulate nuclear facilities with the help of custom-made packages and/or using well-known programming languages. We earlier reported the development of nuclear-specific virtual reality models using a well known game engine. This paper introduces some new features and scenarios which were developed for the models of interest. Specifically, some of the built-in features of the game engine are exploited to add nuclear-specific applications and to enhance interactivity. With this technology, Nuclear Power Plant (NPP) personnel and first responders can “play” and, hence, experience and emulate the operational and emergency scenarios of a plant through these specifically developed interactive and 3-D graphical models of nuclear facilities.

*Key Words:* Virtual Reality Training, Gaming for Nuclear Applications, Safety Training for NPP

### **1 INTRODUCTION**

Training of personnel at nuclear facilities is probably even more important and necessary than at most other facilities. Time and resource spent on education and training at nuclear facilities is significant. Traditional training methodologies involving written text and video tapes are useful but lack interactivity. Reactor simulators are used to train reactor operators but similar interactive mode of training is generally not available for other personnel. Assessment of the

training imparted is also difficult due to lack of a direct feedback in the absence of interactivity. It is for such reasons that an interactive and virtual training paradigm shows some promise towards achieving the training goals for utility personnel.

3-D virtual reality has been around for some time, and is being used in diverse fields, especially those involving high risk, such as in the military. This technology is also being fruitfully employed by the medical and chemical industry [1-3].

Virtual reality is also being studied and used in different ways in nuclear engineering. Use of virtual reality is being explored at Penn State University [4]. Several groups are working with virtual reality technology for dose rates assessment [5, 6]. A study at Oak Ridge National Laboratory analyzed the potential to train first responders using virtual reality and game engines [7]. Recently, a group in Brazil reported a virtual control desk used in nuclear facilities to enhance human factor [8]. The Virtual Education and Research Laboratory at the University of Illinois has also been exploring and developing uses of virtual reality in nuclear engineering [9, 10]. A fairly realistic model of the University of Illinois TRIGA reactor was developed using Unreal Engine [10]. In this paper, we report some recent progress in modeling and simulating scenarios with the help of built-in functions and specifically written scripts.

## 2 DEVELOPMENTAL PLATFORM: *THE UNREAL TOURNAMENT ENGINE*

Several 3-D software platforms that can be used to develop virtual models exist and are in use. Some of these, such as Maya [11] and 3dsMax [12], focus on 3-D, detailed model development of components such as a police car. Others are more suitable for the development of expansive floor and building plans, and in providing interactivity between players and models, as is the case with gaming engines. In this paper, the primary platform used for VR model development platform is the *Unreal Tournament* engine.

Unreal Tournament is a commercial PC game by Epic Games [13]. The first edition of Unreal Tournament was released in 1999, which then evolved into Unreal Championship in 2002, then to Unreal II in 2003 and finally into Unreal Tournament 2004. The newest edition, Unreal Tournament III, was released in 2007. All the models developed in this paper are developed in Unreal Tournament 2004.

One important reason to choose *Unreal Tournament* as the development platform is its multiplayer mode. It allows 16 players to act at the same time. If these players are connected to the internet, they are able to play on the same model simultaneously via remote access.

### 2.1 Unreal Engine

The core of a PC game is its gaming engine. The original Unreal Engine was used in the first edition of Unreal Tournament, while Unreal Engine II was used in Unreal Tournament 2003 and 2004. The latest edition of Unreal Tournament III runs on Unreal Engine III, which is the third generation of the Unreal engine [14]. Models presented in this paper are built on Unreal Engine II.

Basically, Unreal Engine consists of sound, graphics, and physics engine constructs. These sub-engines are capable of dealing with data inputs, such as maps, textures and music. Every scene and action in a game is controlled by this Unreal Engine.

The source code for Unreal engine is open source, which is called UnrealScript. This source code is similar in basic design principles to C++ and Java [15]. Open source nature of this platform offers the possibility for modification of the engine, as needed.

## 2.2 Unreal Editor

All basic elements of a model are developed in the “*level editor*” of the *Unreal Editor*. It provides the user with four windows, which show the top, front, side and 3-D views of the *level* developed. As details are added to the model, they are simultaneously updated in these four windows. Fig. 1 is the screenshot of the Unreal Editor.

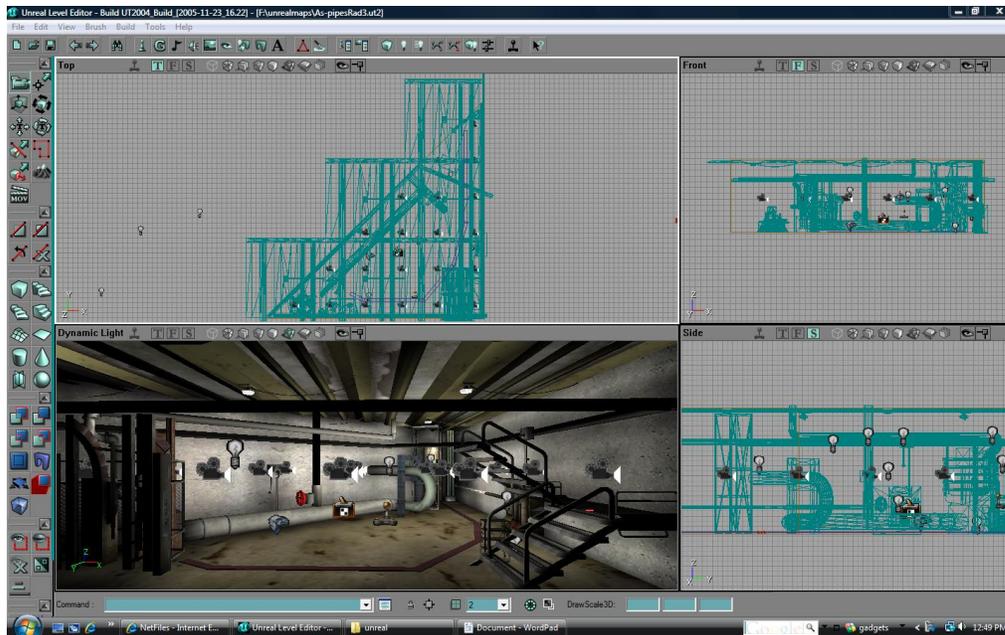


Figure 1. Unreal Editor with four windows

The major difference between the Unreal Editor and some other 3-D modeling platforms is the way the model is built. Whereas one needs to add items in an empty space in most other 3-D modeling platforms, Unreal Editor starts with a solid (filled) world that level designers need to subtract from to create the desired space.

## 3 FEATURES

Compared to other commonly used 3-D model development software, Unreal Engine has several advantages in showing active scenarios. The engine is capable of mixing sound, graphics and physics (interactions, motion) all under one platform. In addition to the ability to develop the structure of any facility, Unreal Engine has active and dynamic features embedded in its *level* that can be exploited to introduce additional *levels* of interactivity. Some of these built-in features meet the immediate needs for modeling and simulation of nuclear facilities. These features are described below.

As mentioned above, the source code for these features is an open architecture. Hence, *scripting* is used to modify some of the features involved. All *actors* under an Unreal Engine environment are characterized by a set of default properties. These include: size, brightness,

location, weather visible or invisible, etc. Script changes can control these properties and modify existing features to fit the particular needs of a model.

### 3.1 Triggers

Trigger is a class of *actors* that control other events in Unreal Engine. A trigger could be placed anywhere in the level and be hidden or visible. A trigger can be activated by pressing a specified key on the keyboard (or the joystick) when player is near the trigger. Once the trigger is activated, it initiates a pre-named Unreal Engine event, which leads to a sequence of pre-programmed activities.

There are several kinds of triggers. For example, the material/fabric trigger changes the texture of a surface. Using this trigger, script can be used to specify a sequence of pictures to be pasted to a wall. *Message* triggers can be used to show messages on the screen. There are also the *UseTrigger* and *ScriptedTrigger*, the use of which will be illustrated in the following implementations.

### 3.2 Virtual Security Cameras

Security camera is an advanced feature which combines material triggers and script. In a manner similar to placing a video camera at some prescribed location in the physical world to transmit live videos to some control room for monitoring purposes, virtual security cameras are meant to monitor virtual spots in the virtual model. Fig. 2 shows security cameras in a control room.



Figure 2. Security cameras showing water leakage

As shown in Fig. 2, several virtual video cameras can be placed at different locations in the model. The camera view is transmitted to monitors in the control room. This feature is very helpful for education and training purposes, especially when multiple trainees are involved. One of the trainees, for instance, can be in the control room and issue instructions to trainees based at

other locations receiving the sequence of commands and responding. Also, a supervisor who is in front of the monitors can observe the actions of others and make appropriate decisions.

### 3.3 Health meters

Health meter is a built-in feature in Unreal engine. Like in most games, health meter in Unreal indicates the player's health status. If a player gets hit (or shot), her health index decreases. After some modification, and with the help of a script, the health meter can be made to mimic a dosimeter. That is, based upon the local radiation level and the amount of time spent, a dose can be calculated by a script and an accumulated dose can be displayed in the health meter.

### 3.4 Avatars

Avatars are non-player controlled characters in the game engine. Their actions are programmed in advance by a script and not affected by the action of other players. To make the simulation more effective in training and improving human performance, avatars are introduced in our models. As in real life, these avatars work around the player in the simulation, like workmates of the players in real life, executing their pre-programmed chores. Fig. 3 shows an avatar standing in a room. [This figure also shows a floor that is color coded based on the local radiation level [10].]



Figure 3. An avatar standing in a room

## 4 MODELS AND SCENARIOS

### 4.1 Models

With the above features integrated, the models have active elements rather than just the customary static structures. So far, several models have been developed, including one for the UIUC TRIGA reactor [10]. Other models, including a model of a nuclear power plant, are being developed. Details are continually being added in to these models. Fig. 4 and Fig. 5 are pictures of the UIUC TRIGA reactor model and a nuclear power plant model respectively.

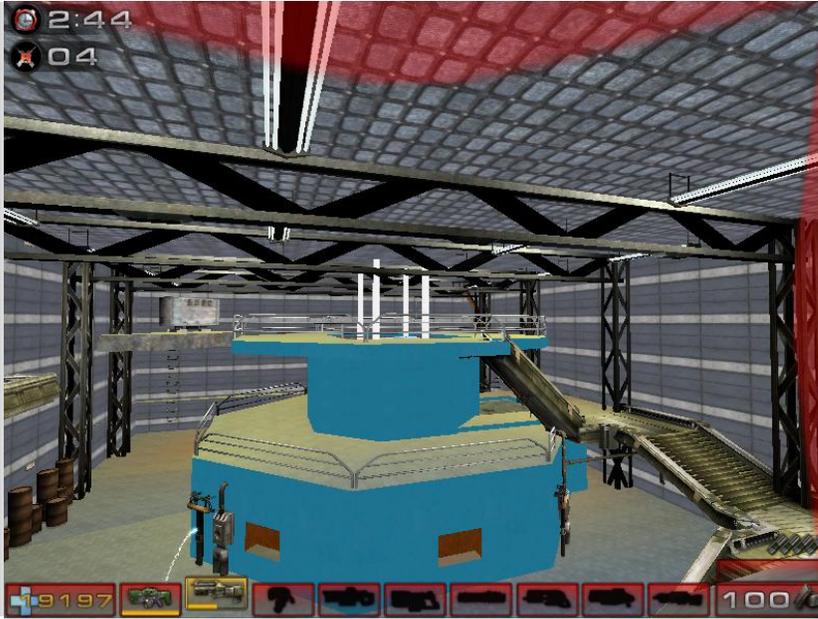


Figure 4. UIUC TRIGA model with irradiation level based color-coded floor [10]



Figure 5. Spent fuel pool in a nuclear power plant model

## 4.2 Scenarios

Besides realistic structures, scenarios are also needed to provide even better training and education. The built-in features in Unreal Engine are used to develop several hypothetical scenarios to simulate the events that take place at nuclear facilities. These scenarios are described in this section.

### 4.2.1 Fire and water leakage

Built-in features in UT allow easy implementation of water surfaces, flowing water (from a leak or down a tap), and fire. (These are commonly used items in video games.) These have also

been exploited in the development of models of nuclear facilities. For example, water surface feature is used in the model of the pool of the TRIGA reactor [10]. Fire is introduced in a model of a reactor facility. A picture of the room where fire breaks out is shown in Fig. 6. This fire is actually a part of a training scenario for fire fighters. Currently, the fire fighter needs to touch (get close to) the fire extinguisher to extinguish the fire. Dripping or leaking water feature is also a part of the same scenario in which a worker needs to respond to a water leak in the power plant. A picture of the leaking water is shown in Fig. 7. These scenarios are initiated by a trigger. The player responds by following a pre-programmed course of action.



Figure 6. Color coded floor displaying minimal dose path from the door to the fire, picking up the fire extinguisher along the way

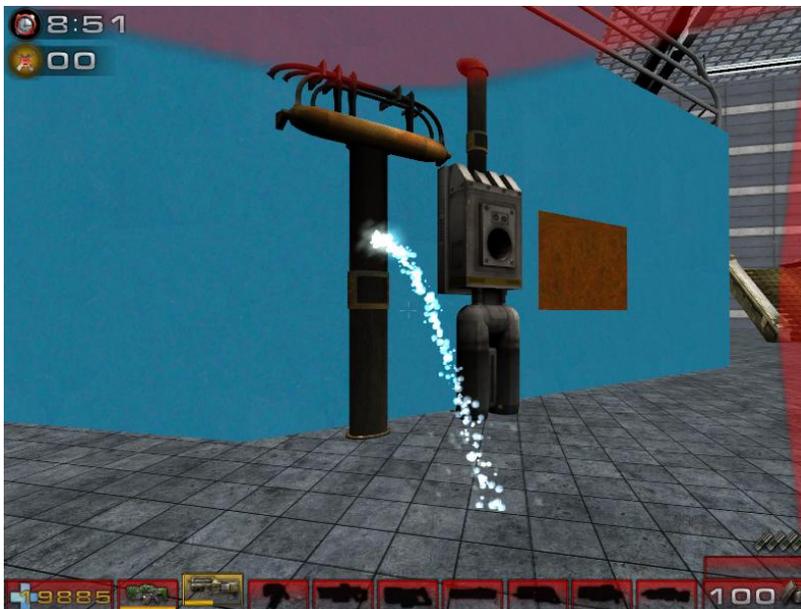


Figure 7. Leaking water in the TRIGA model

#### 4.2.2 Color-coded floors

The color coded radiation map is one of the developments which combine Unrealscript and other software [10]. Goal of color coded radiation map is to display the radiation dose level, for example, in a pump room as shown in Fig. 6. Radiation levels displayed may be based on measured data or via simulations using software. Scripts are written to display radiation on the map using different colors based on the radiation data. Red floor means higher radiation doses. Blue and green areas are used for increasingly lower level of radiation, respectively. Health meter is also used in this scenario as a dosimeter. Reading on the dosimeter, it increases faster when the player is in the red region than when standing in, for example, the yellow region.

#### 4.2.3 Minimal dose paths

Once the color coded radiation map is complete, it could be used to display minimal dose paths. A parallel effort is underway to develop algorithm to determine minimal dose path given the radiation levels at several different points [16]. This path can be displayed in the model as well. Fig. 6 shows the minimum dose path for a fire fighter that takes the fire fighter first to the fire extinguisher and then to the fire.

### 5 3-D, VIRTUAL AND IMMERSIVE DISPLAYS

While these models can be fruitful utilized on commonly available display monitors, they can also be “exported” to virtual reality formats and then displayed in the 3-D, immersive virtual reality environment for an even more realistic training experience. Visbox is such an environment that displays 3-D models [9, 10]. New graphics cards [17] also allow the models to be displayed in 3-D immersive modes on ordinary desktop and laptop monitors (to be viewed using special red-blue glasses).

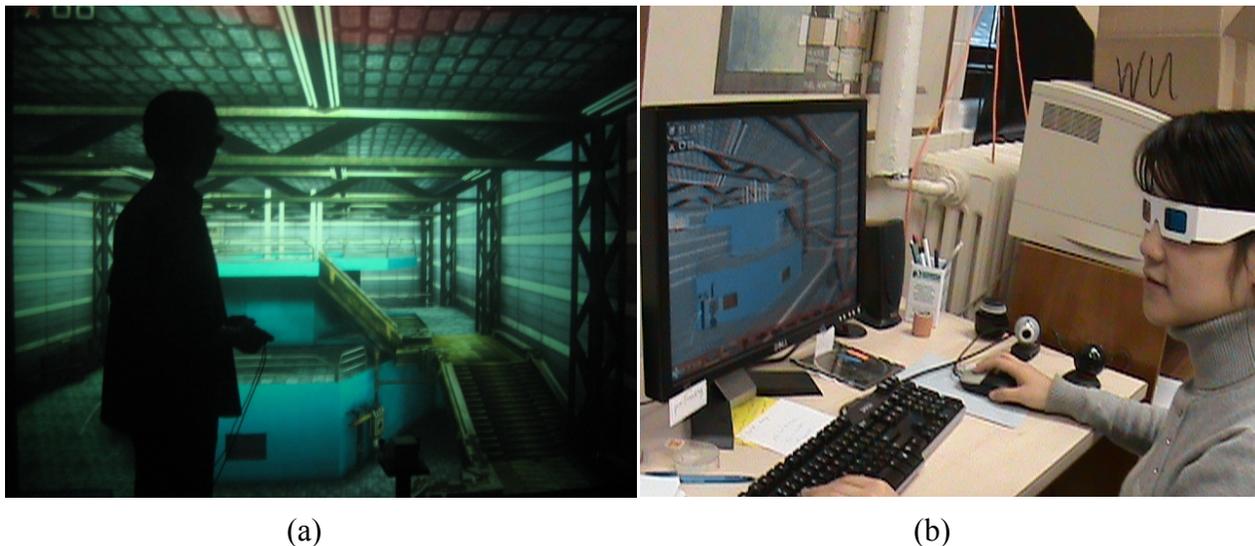


Figure 8. Model of a TRIGA reactor: (a) on Visbox; (b) on a PC

Fig. 8a and Fig. 8b show the TRIGA model developed in UT2004 on two different platforms. Fig. 8a shows a trainee maneuvering around the reactor using a joystick in the 3D immersive environment of Visbox. Trainee is wearing special tracked glasses. Fig. 8b shows a trainee using a mouse and experiencing the same model on a PC (that has the latest Nvidia graphic card). The

card displays the image on the monitor in two colors. When seen with the special red-blue glasses, the images give a feel of the third dimension as well.

## 6 CONCLUSIONS

This paper reports some of the recent progress made toward virtual 3-D models for nuclear applications. Additional features and scenarios have been developed and added to models and scenarios reported earlier. With some script changes and modifications, existing features in Unreal have been modified to meet the demand of the models and nuclear specific cases. These developments can be fruitfully utilized for training and education at power plants as well as at universities.

## 7 ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of Joel Dixon and Stefano Markidis. This work was supported in part by a grant from the US Nuclear Regulatory Commission.

## 8 REFERENCES

1. "Medical Virtual Reality Center" (2008), <<http://www.mvrc.pitt.edu/>> (13December, 2008).
2. "The Virtual Reality Medical Center" (2008), <<http://www.vrphobia.com/>> (13 December, 2008).
3. John T. Bell, H. Scott Fogler, "Virtual Reality in Chemical Engineering Education", *Proceedings of the 1998 ASEE North Central Section Meeting*, University of Detroit Mercy, Detroit, MI, April 3-4, 1998, <<http://www.vrupl.evl.uic.edu/vrichel/Papers/aseepap7.pdf>>.
4. Whisker, V.E., Warren, M.E., Baratta, A.J., Shaw, T.S., "Development of a Radiation Dose Model for Immersive Virtual Environments", *Trans. ANS/ENS International Winter Meeting*, New Orleans, LA (2003). Also see: Whisker, V.E., "Assessment of Radiation Awareness Training in Immersive Virtual Environments," Ph.D. dissertation, Penn State University, 2008.
5. J. Ródenas, I. Zarza, M. C. Burgos, A. Felipe and M. L. Sánchez-Mayoral, "Developing a Virtual Reality Application for Training Nuclear Power Plant Operators: Setting Up a Database Containing Dose Rates in the Refueling Plant", *Radiation Protection Dosimetry*, Vol. 111, No. 2, pp. 173---180 (2004).
6. Antonio Carlos A. Mol, Carlos Alexandre F. Jorge, Pedro M. Couto, Silas C. Augusto, Gerson G. Cunha, Luiz Landau , "Virtual Environments Simulation for Dose Assessment in Nuclear Plants" *Progress in Nuclear Energy* 51, 382–387 (2009).
7. Robert L. Sanders, Joseph E. Lake, "Training First Responder to Nuclear Facilities Using 3-D Visualization Technology", *Proceedings of the 2005 Winter Simulation Conference*, December 4-7 (2005).
8. Isaac Jose´ Antonio Luquetti dos Santos, Cla´udio Henrique dos Santos Grecco, Antonio Carlos, Abreu Mol, Paulo Victor Rodrigues Carvalho , "The Use of Questionnaire and Virtual Reality in the Verification of the Human Factors Issues in the Design of Nuclear Control Desk", *International Journal of Industrial Ergonomics* 39 ,159–166 (2009).

9. Nick Karancevic, Rizwan-uddin, “The Virtual Nuclear Laboratory”, *Proc. Fourth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls and Human-Machine Interface Technologies*, Columbus, Ohio (2004).
10. Joel Dixon, Stefano Markidis, Cheng Luo, Jared Reynolds, Rizwan-uddin, “Three Dimensional, Virtual, Game-Like Environments for Education and Training”, *Trans. ANS*, 97, 915-917 (2007).
11. “Maya” (2009), <<http://usa.autodesk.com/adsk/servlet/index?id=7635018&siteID=123112>> (15 January, 2009).
12. “3ds Max” (2009), <<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302>> (15 January, 2009).
13. “Unreal Tournament 3Wikipedia” (13 December 2008), <[http://en.wikipedia.org/wiki/Unreal\\_Tournament\\_3](http://en.wikipedia.org/wiki/Unreal_Tournament_3)> (13 December, 2008).
14. “Unreal Engine Wikipedia” (11 December 2008), <[http://en.wikipedia.org/wiki/Unreal\\_Engine](http://en.wikipedia.org/wiki/Unreal_Engine)> (13 December, 2008).
15. “UnrealScript Wikipedia” (18 August 2008), <<http://en.wikipedia.org/wiki/UnrealScript>> (13 December, 2008).
16. Abdul-Qadim Alzalloum, MS thesis, University of Illinois, in progress.
17. “Nvidia” (2009), <<http://www.nvidia.com/page/home.html>> (15 January, 2009).