

Unreal III BASED 3-D VIRTUAL MODELS FOR TRAINING AT NUCLEAR POWER PLANTS

Zachary Kriz¹, Russell Prochaska², Cody Aaron Morrow¹, Cesar Vasquez¹
Hsingtzu Wu¹ and Rizwan-uddin¹

¹Department of Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana Champaign, Urbana, IL 61801 USA

²Cornell College, Mount Vernon, IA 52314 USA
hwu24@illinois.edu

ABSTRACT

Three dimensional (3-D) virtual reality models can be an effective tool for education and training in nuclear engineering applications. This technology can be used to model and simulate nuclear facilities with the help of custom-made packages and programming. This paper introduces useful features of a well known game engine, Unreal Engine III, for nuclear-specific virtual reality models. Specifically, some of the built-in features of this edition are exploited to add nuclear-specific applications and to enhance interactivity. Through these specifically developed interactive and 3-D graphical simulators, Nuclear Power Plant (NPP) personnel and first responders can “play” and experience the operational and emergency scenarios of a plant.

1. INTRODUCTION

Effective and efficient personnel training are essential for safe operation of NPP. Traditional training methodologies involve written text and video tapes which are useful but lack interactivity. Reactor simulators can be used to train reactor operators but similar interactive mode of training is 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.

An interactive and virtual training paradigm shows some promise towards achieving training goals. 3-D virtual reality has been around for some time, and it is being manipulated in diverse fields, such as in the military which involves high risks. 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. For example, Virtual reality for training is being explored at Penn State University [4]; several groups are working on 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]; and a group in Brazil reported a virtual control desk used in nuclear facilities to enhance human performance [8]; Halden reactor project in Norway is also actively working on the application of virtual reality for training personnel [9]. The Virtual Education and Research Laboratory at the University of Illinois has also been exploring and developing uses of virtual reality in nuclear engineering [10-12]. A fairly realistic model of the University of Illinois TRIGA reactor was developed

using Unreal Engine 2004 [11, 12]. During summer of 2009, all models and scenarios were switched to the updated Unreal Engine, Unreal Engine III. Unreal Engine III has more desirable features than Unreal Engine 2004 such as a function of embedded video. Note that different game engines have different desirable features. Another game engine, the Garden of Eden Creation Kit (GECK), also is being used to develop training models, and the results are reported in an accompanying paper [13]. In this paper, we introduce Unreal Engine III and valuable features used in our upgraded scenarios.

2. DEVELOPMENTAL PLATFORM

The developmental platform for this paper is Unreal Tournament engine. It has two components, Unreal Engine III and Unreal Editor III. Unreal Engine III comes with a commercial PC game, Unreal Tournament III, which was released in 2007 by Epic Games [14]. The gaming platform which utilizes the Unreal Engine is Unreal Editor III. Many immersive 3-D games are currently being developed and run upon Unreal Editor III.

2.1. Unreal Engine III

Unreal Tournament III runs on Unreal Engine III which is the third generation of Unreal Engine. Unreal Engine III consists of sound, graphics, and physics engine constructs, such as maps input, textures input and music input. Unreal Engine III also controls every scene and action in the game. The source code for Unreal engine is an open source called UnrealScript that offers an opportunity to modify the engine as needed.

2.2. Unreal Editor III

Unreal Editor III is a powerful tool which a designer can use to create and model almost any shape they can think of inside of its level confines. Figure 1 is a screenshot of Unreal Editor III. To construct a model, the designer must first decide upon an additive or subtractive style of construction for their level. In the additive mode, the editor generates a massive *open nothingness* for the designer to work within, while the subtractive mode generates a massive block of mass from which the designer can cutaway interior areas to create an expansive “cave” for the level to reside within. Once the designer has chosen their style for the level, they can move onto the creation of the static elements of the map. This involves the adding of ‘brushes’ to the level which are geometric

shapes for the designer to mold to their desired shape and then place them in the level. These brushes can have materials then added to them to give them the desired appearance. When all of the static objects are in place, the base of the level is completed and active and dynamic aspects can then be added to create the desired interactivity.

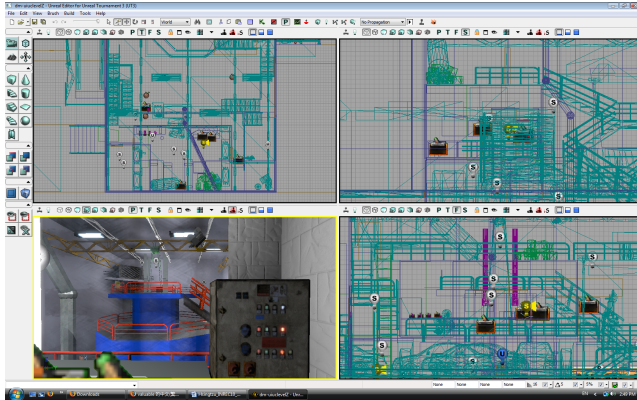


Figure 1. A screenshot shows the updated TRIGA reactor in Unreal Editor III.

3. FEATURES

Compared to other commonly used 3-D model development software, Unreal Engine III has several advantages in showing active scenarios. This engine is capable of mixing sound, graphics and dynamics, all under one platform to create a realistic feeling and an interactive environment. In addition, features of this engine can be manipulated to develop several hypothetical scenarios to simulate the events that take place at nuclear facilities. Some of these features come with the game, while other features involve the manipulation of the games, elements by the designer, and still other features which the game does not ship with can be created by the designers through the use of Unreal Script. Features used in our upgraded models and scenarios are described below.

3.1. Multiplayer Options

One of the most useful features of Unreal III is the ability to have several players in any given map at one time. Currently the game engine can support up to sixteen players on a single map at the same time. This could help better learn what an individual's role in a given project is and learn how to fix probable mistakes made by their colleagues. For example, in a multiplayer scenario a *class choosing* system can be used, where all of the players assemble in a small room outside the normal level to be assigned their tasks. Upon their job are decided they can then proceed to the mission they are assigned. The individual classes can also be setup so that tasks cannot be carried out until the proper time in a scenario. This helps prevent mis-steps such as fire-fighters showing up an hour after the fire has been put out already. These scenarios provide a live training of an individual at a facility, alongside the learning of facility layout, and safety practices.

3.2. Textures And Materials

Textures and Materials are crucial attributes to making models look realistic. Within Unreal Editor III, basic shapes of things modeled are wire frames which the designer can manipulate into the desired shapes. A material is a texture which can be applied to the surfaces of these meshes or static meshes in this Editor. The designer has control over applying different effects to the textures, ranging from adding differing color levels, and subtracting out different portions of a picture, to adding normal maps to the texture in order to gain a simulated 3-D effect on a 2-D texture through "light map" information. Once applied these materials supply an extremely realistic look to objects and coupled with the Unreal engines ability to "light map" surfaces, a seemingly 3-D surface can be seen. Figure 2 is a screenshot of a material editor window.

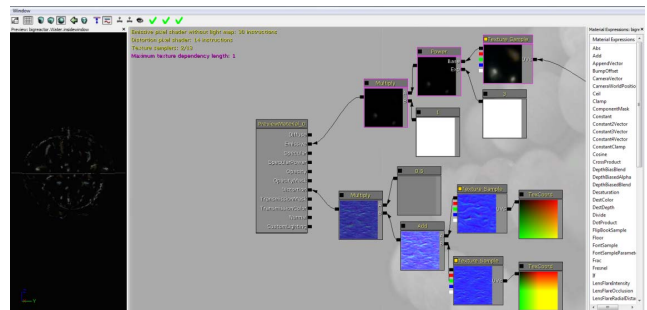


Figure 2. A screenshot of a material editor window.

3.3. Virtual Security Cameras

Virtual Security Cameras is an application of textures and materials. This is a feature to remotely view other locations in a map. This is based on a special material applied to a screen in this Editor. The texture used in this material comes from a camera the designer creates at a location in the map. This feature realistically provides real-time feedback of what is happening in the facility. 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 give instructions to trainees who receive the sequence of commands and respond in different locations.

3.4. Static Meshes.

Other than basic geometry meshes being used to create the levels, foundation layer and several other types of meshes can be used as well. One of these being is the static mesh. This is a prebuilt form either coming out of the editors, preloaded archive of meshes, or else being created by a third party outside of the game in a 3-D modeling software such as Maya, or 3dsMax. These meshes commonly have a material already applied to them when they are placed in the level, or the designers can change this texture manually to fit their need. Static meshes are generally used to create more intricately shaped things such as desks, or walkways which differ in shape. Other types of static meshes include trees, crates, and many other simple items.

3.5. Triggers

Triggers are essentially nodes which are placed inside of the Editor at the desired locations. When a player uses an object they are actually interacting with this node telling the engine that they wish to use this object, the simulation then enacts the predetermined action that the trigger is tied to. These events are triggered off of various events that happen such as a player interacting with a portion of a level or a timer running out.

3.6. Kismet

The kismet system is in charge of the simulations responses to players' actions in a game. This system is initially a large empty space when a level is created. By placing triggers within the space and "attaching" them to the other triggers placed within the window, a chain of events is created. When the first action on the chain is initiated, the system follows the chain to the end activating all of the other triggers along the way. These activated triggers can perform many functions such as activating sounds or prompting messages to appear on the screens of players. In the case of a sequence of events a player must perform, the game can check to see if a previous event has occurred using different methods. The simplest is via a Boolean logic system in the Kismet system. For example, if what follows step B depends upon whether step A was performed earlier or not, a node is inserted at step A in the chain which is switched to True if A is performed. Later, when the chain of events reaches step B, the value of the logical variable corresponding to step A is checked, and depending upon its value one of the two paths can be followed. In this way the designer can simulate the result of an interaction with an object. Figure 3 is a simple kismet chain example.

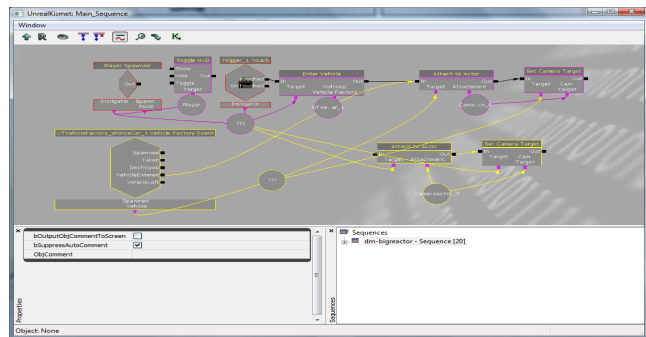


Figure 3. A simple kismet chain in which a player spawns, and is placed inside of a police car vehicle.

3.7. Embedded Video And Sound Tracks

Within the texture importing ability of the Editor, is the ability to import video into the Editor as well. This feature is extremely useful when it comes to training purposes as it allows a trainee to walk up to a workstation and trigger a video displaying the workings of the device. Along with the video importing, the Editor also allows the importation of sound only tracks which can then be played at different times.

3.8. Matinee

This Unreal Engine contains a feature called Matinee which is used to animate all moving objects in the level in this Engine. All doors, elevators, or moving parts must be entered into matinee to gain the ability to move. Along with the movement of meshes in the game, Matinee can be used to create in-level movies and cut-scenes which the player can witness. This can help players understand how an event happened in the first place by showing a clip of the initial event occurring, or a clip of the actions they trigger to play out.

3.9. Color-Coded Floors

This is a feature which takes colors the floor square-by-square based on the designers specified radiation level distribution in a plant. Figure 4 is a color-coded floor in a corner of the NPP model. In this model, the colors are purple (low to no radiation levels), green (moderate radiation levels), and red (high radioactive levels). The feature can be used to give a visual representation of where radiation levels can be expected. This colored floor feature is toggle-able by the use of a trigger on the map. This feature has several settings which can be changed, such as the number of times it can be activated in a map. Whether it stays permanently or shuts off on its own after a certain amount of time. Both of these aspects can be used in a testing capacity for a trainee. For instance, they can only turn the floor on three times during their run through a scenario. This lets a scoring system be utilized based on how well they maneuvered around the radiation levels, and how many times they used the visual aid.

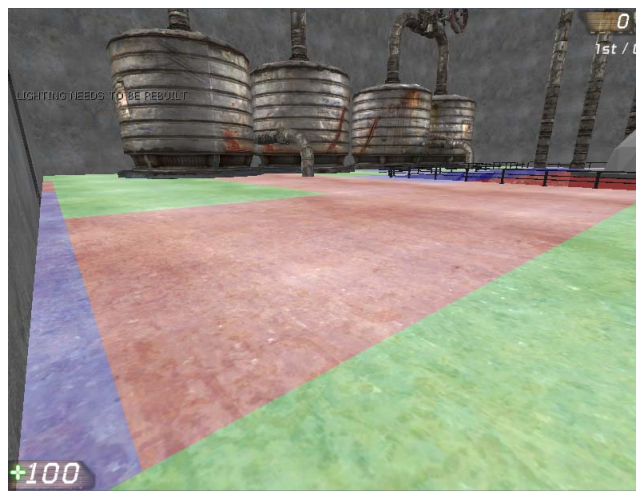


Figure 4. Colored floor shows radiation levels within a nuclear facility model.

3.10. Dosage Zones.

Along with the colored floor feature, the Editor allows for volumes to be placed within the game which *damage* the avatar. These can be tuned to correspond with the color coded radiation levels such that as a player walks around the map, they get more and more "damaged" by the radiation. This would also be utilizable in testing a trainee to check if they have learned the

most efficient ways of getting around the facility with the least exposure.

3.11. Emitters

An emitter is an Actor which spawns and emits 2-D *sprites* which always face the player. In this way different effects on a level can be created to enhance the realism of the environment, as well as create different events. Figure 5 is an example which shows fire emergency in the pipe room of the upgraded TRIGA reactor model and sprinklers putting it out.



Figure 5. Fire emitters in the pipe room of the reactor model. Sprinklers (water emitters) blur the top-left side of this figure.

3.12. Non Player Controlled Characters (NPCs)

NPC is a feature which allows for the creation of a scripted computer player. The NPCs can be made to do any task a player could, and will trigger its actions in the order a designer wishes when they are told to. Control of these NPCs can be scripted within Unreal script, or they can have command chains created within Kismet that tells them what to do when other events occur, such as a player initiating an event that tells a NPC to put out a fire.

4. Model of a TRIGA Facility

A detailed model of the University of Illinois TRIGA facility has been developed using Unreal Engine III. Features described above have been exploited to make the facility realistic and interactive. Figure 6 shows a screenshot of a trainee—the player—watching a training video on how to operate a valve standing next to the valve.



Figure 6. A screenshot of a training video a player watches in the game. This video demonstrates how to use valve at left hand side of the video.

5. CONCLUSION

This paper introduces valuable features of Unreal Engine III for 3-D virtual reality in nuclear engineering application. With some script changes and modifications, existing features in Unreal Engine III have been modified to meet the demand of nuclear specific cases. These developments can be fruitfully utilized for training and education at power plants as well as at universities.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of Chen Xi. This work was supported in part by a grant from the US Nuclear Regulatory Commission.

7. 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.
- [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*, Vol. 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* Vol. 39 ,159–166 (2009).
- [9] Iguchi, Y., Y. Kanehira, M. Tachibana, and T. Johnsen. “Development of decommissioning engineering support system (DEXUS) of the fügen nuclear power station”, *Journal of Nuclear Science and Technology*, Vol. 41, No. 3: 367-375 (2004).
- [10] 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).
- [11] Joel Dixon, Stefano Markidis, Cheng Luo, Jared Reynolds, Rizwan-uddin, “Three Dimensional, Virtual, Game-Like Environments for Education and Training”, *Trans. ANS*, Vol. 97, 915-917 (2007).
- [12] Chen Xi and Hsingtzu Wu, Arwa Joher, Leo Kirsch, Cheng Luo, Mohammed Khasawneh and Rizwan-uddin, “3D Virtual Reality for Education, Training and Improved Human Performance in Nuclear Engineering,” in Proc. *ANS NPIC HMIT 2009 Topical Meeting - Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology*, Knoxville, Tennessee, ANS, April 5-9, 2009.
- [13] Carl Rytych, Lewis Conley, Hsingtzu Wu, Rizwan-uddin, “Game-Like Environments for Nuclear Engineering Education using GECK,” submitted for presentation at *the 1st International Nuclear and Renewable Energy Conference (INREC10), Amman, Jordan, March 21-24, 2010*
- [14] “Epic Games”
<<http://www.epicgames.com/>> (23th December 2009)