Architectural/Engineering Visualization using Game Engine

Mohammed E. Haque, Pallab Dasgupta
Construction Science Department
Texas A&M University

Abstract

Innovations and advancements in animation and visualization technology have greatly enhanced the graphical interface between computers and humans. Today’s 3D game engines offer an ultra-realistic virtual world with high speed rendering quality, and user interactivity, which is difficult to obtain using traditional 3D pre-rendered and non-interactive visualization tools. This paper describes an approach for developing a realistic 3D architectural walkthrough visualization of a residential building using the “Unreal Engine 2”. 3D Game Engine based virtual walkthrough allows user to navigate a virtual environment as if in the real world. Architectural/Engineering students can experience their design from first person shooter’s point of view using Game Engine based architectural walkthrough, and can have enhanced understanding of their design in relation to time, space environment and scale. The walkthrough visualization using 3D Game Engine as demonstrated provides a broader and better understanding of architectural planning and design concept, exterior, interiors and feel of aesthetics as compared to paper based models currently used by most architectural students. In addition, by allowing the students to enter a virtual space at full scale, it is possible to add more pragmatism to their design experiences.

Introduction

Use of 3D technology to present virtual buildings has traditionally been hampered by long rendering times, and the non-interactivity of a pre-rendered walkthrough. Movie render times of several days to a week are not unheard of, depending on the complexity of the sequence.¹

Today, 3D computer games are highly complex systems that consist of a universal game engine and the specific game elements like the game rules and game data (e.g. geometry, textures and sound files). Main emphasis is put here in the game engine. This module is the heart of the computer game and represents the basic framework independent of the game. This general purpose feature allows the use of the engine for other applications, e.g. the indoor visualization of building elements. Game engines incorporates all sorts of elements that are vital to a game like physics, graphical user interface (GUI), artificial intelligence, network functionality, sound and event engine.²

Fritsch & Kada² stated that the Unreal Engine 2 (developed by Epic Games) is one of the most widely used game engines to date. Because it is a cross-platform solution, a broad range of products from PC and video games to architectural visualizations have been already developed with it. Being optimized for both indoor and outdoor environments, it is one of the most modern and versatile engines.³ The ability to create ultra-realistic virtual worlds, coupled with the technology to present
and collaborate on the same worlds from anywhere in the real one, gives the Unreality concept the greatest benefit over any standalone design tool, or even over using the Unreal engine to enhance design and conceptualization.¹

Like most other game engines, the technology is encapsulated in a binary runtime library, while the game related parts of the Unreal games are available as source code in a scripting language called UnrealScript. The novel approach of Epic Games is that they released the Unreal Engine 2 Runtime free for non-commercial and educational use. The runtime even includes the map editor UnrealEd and header files for C++ programmers. Beginners do find lots of technical documents and even video tutorial that teach level design, script programming and much more.²

The use of a game engine based collaborative virtual environment has enabled the development of software to support architectural design education.⁴ The focus of Moloney & Harvey’s⁴ paper has been on establishing the role of a CVE to allow an exploration of design context in a manner that is not possible when using typical architectural visualization software. The advantages of working in a real time environment where early design iterations can be tested from multiple points of view, and the value of using sound to evoke occupancy and materiality, allow students to understand architecture as place for the framing of events over time.

With the rapid development and implementing of new visualization by the Gaming industry many previous versions of game engine and game related libraries are available for free or almost no cost. Game engines can be used to give our clients a tour of the project rather than making them mere spectators of a predetermined view.⁵

Harrison⁶ states that a game engine is normally designed for a particular game such as first personal shooters, real-time strategy and vehicle simulations. One advantage of a game engine is that it can be reused to the development of many different kinds of games, in an automated way, saving a lot of programming time. Since the time computer graphics has been introduced, the demands for visualization techniques have grown continuously. Today, the visualization of three-dimensional worlds seems to be a demanding task requested by many geo related disciplines. Due to the increasing interest in the consumer market, tremendous progress can be observed in the hardware and software. Game engines are powerful software packages that efficiently use rendering pipelines, special data-structures and speed-up techniques to visualize texture mapped 3D objects, scenes and 3D worlds in real-time⁶.

The UnrealEngine2 Runtime software is the state of the art in cross-platform, real-time 3D rendering solutions for the desktop. It is optimized for both indoor and outdoor environments and is a versatile engine.²

One of the Unreal engine's greatest strength as a design tool is its integration with the UnrealEd level editor. UnrealEd is a real-time level design tool, optimized for building real-time 3D environments. It is fully integrated with Unreal's rendering engine, offering a camera view, and immediate display of all lighting, texture placement and geometry operations. UnrealEd also offers single-click playability: even in the midst of the design process, the designer can launch the Unreality viewer and walk around their building in real-time.⁷
The aim of the research was to develop a realistic 3D architectural walkthrough using the “Unreal Engine 2” that would allow the architecture/construction science students to navigate virtual building from an ultra-realistic first person vantage point. The game engine driven models can enhance understanding of their own design in relation to time, space, environment, and scale. In addition, interactive elements, such as lighting, sound, triggered events, animations, and the realism of the 3D model can be enhanced even further.

Model Development Methodology

The step involved in making the interior and exterior of a building model using the game engine is explained in the following steps with the flowchart (Figure1):

- First step was to design a residential building.
- The next step was to develop a 2d architectural drawing that includes floor plan (Figure 1), elevation and furniture lay out using AutoCAD 2006.
- After the development of 2d drawings a basic 3D model was developed in unreal editor by subtracting the world space and then adding as few large blocks which were then carried out to form rooms and doors and windows.
- Generate a terrain in the Unreal Ed.
- Using polygonal modeling in 3ds max6, after setting up identical grid system between 3ds Max and the Unreal Editor.
- Textures were resized or created in Photoshop 7 and then saved in .dds, .bmp and .tga formats.
- Textures imported in to the unreal texture browser and applied to the geometry created in unreal. Creating the static mesh in 3ds Max like the door, windows and furniture’s to populate the basic structure that was developed in unreal.
- UV mapping and or unwrapping in 3ds Max or any third party software’s like Deep UV and texturing the Model in 3ds Max.
- Exporting the 3ds modes as .ASE and textures as .DDS or .BMP or .TGA.
- Importing the static mesh in Unreal Ed and applying material in the static mesh browser.
- Subtract space away from the level for skybox and add textured static meshes planes to act as clouds. Also add static mesh rings that act as cloud rings.
- Add different lights and sunlight and adjust them
- Use triggers for message display and lighting control
- Use movers to make auto-opening doors.
- Optimize and save the Map. Add screen shots; customize the runtime interface and test run the map.
- Add screen shot and information about the map and test run it.

The Methodology of creating a world in Unreal and adding static mesh is explained using screenprints (Figures 2, 3, 4, and 5), which were taken during development of the model. Level editors cannot create 3D content, such as static meshes and player models. For this reason 3D modeling software’s like 3ds Max or Maya was used to create things not available in the Unreal runtime engine. The advantage of using static mesh in multiples times is that it runs very fast as if only a single copy of the static mesh in the level.
Design And Develop 2D drawings
Create world space in Ued and texture
Resized, file type changed in Photoshop and Exported
Create static mesh in 3ds Max
Multi-Sub object texturing
UVW Mapping and Unwrapping
Collapse the stack and export as .ase
Generate Terrain
Add blocks and subtract to form the basic structure
Import Static Meshes
Add lighting, Triggers and movers
Optimize
Add screen shot and map Information
Save Map
Create a Sky box

Figure 1. Research Methodology Flow Chart

Figure 2. AutoCAD - Building Plan
Figure 3. Setting up units in 3ds max

Figure 4. Setting grids to match Unreal Ed
Overview of the Model

The entire model was based on the plan (Figure 1), and developed mostly using UnrealEd2, 3ds Max and Photoshop. With the help of mouse and/or arrow buttons, users can walkthrough the interior and exterior of the building with full degree of freedom. The user can open the doors and enter into the building, roam around living rooms, bed rooms, bathrooms, hallway, and other, and switch on/off lights, and so on. Some of the pictures that have taken during the walkthrough are shown in Figures 6, 7, 8 and 9.
Figure 6. The front entrance

Figure 7. Building walkthrough with the game engine
Figure 8. Bedroom walkthrough with game engine

Figure 9. Hallway walkthrough with game engine
Evaluation of the Model

In order to determine the effectiveness of the model as an instructional tool was conducted in a small sample (graduate students). Initial feedback of the evaluation of the model was very encouraging. Evaluation of the model using a large sample of population (undergraduate students) will be conducted in future. This instruction model will be tested on a Likert scale of 1 to 5 to see whether the model achieves its aim or not.

Summary and Conclusions

This paper describes an architectural model development methodology using the “Unreal Engine 2”. The Engine based walkthrough allows the students to navigate with full degree of freedom within the virtual environments. For architecture students the added realism of these models can enhance understanding of their own design in relation to time, space, environment, and scale. The interactive elements of the game engine like lighting, sound, triggered events and animations, was studied in detail to increase the level of interest of the users, and to create a near realistic space. The preliminary evaluation of the developed walkthrough visualization model using 3D Game Engines showed that it would provide better and broader understanding of planning and conceptual design, interior and exterior feel, and the aesthetic appeal of construction compared to the typical “paper based” or “movie based” models currently used by most architectural students.

References


MOHAMMED E. HAQUE, Ph.D., P.E.
Dr. Mohammed E. Haque is a professor and holder of the Cecil O. Windsor, Jr.’66 Endowed Professorship in Construction Science, Texas A&M University at College Station, Texas. He has over twenty years of professional experience in analysis, design, and investigation of building, bridges and tunnel structural projects of various city and state governments and private sectors. Dr. Haque is a registered Professional Engineer in the states of New York, Pennsylvania and Michigan, and members of ASEE, ASCE, and ACI. Dr. Haque received a BSCE from Bangladesh University of Engineering and Technology, a MSCE and a Ph.D. in Civil/Structural Engineering from New Jersey Institute of Technology, Newark, New Jersey. His research interests include fracture mechanics of engineering
materials, composite materials and advanced construction materials, architectural/construction visualization and animation, computer applications in structural analysis and design, artificial neural network applications, knowledge based expert system developments, application based software developments, and buildings/infrastructure/bridges/tunnels inspection and database management systems.

PALLAB DASGUPTA
Mr. Pallab Dasgupta is a graduate student of the Department of Construction Science, Texas A&M University.