UTILIZING 3D GAMES DEVELOPMENT TOOL FOR ARCHITECTURAL DESIGN IN A VIRTUAL ENVIRONMENT

Mohd Fairuz Shiratuddin
The School of Construction
The University of Southern Mississippi
Hattiesburg, MS 39406
mohd.shiratuddin@usm.edu
http://www.fairuz.us

Desmond Fletcher
The School of Construction
The University of Southern Mississippi
Hattiesburg, MS 39406
desmond.fletcher@usm.edu

ABSTRACT – The Architectural Engineering Technology (AET) program at the School of Construction has been utilizing 3D games development tools in assisting the teaching and learning the development of Virtual Environment specifically for architectural design. The ACT450/550 – Virtual Reality Applications 1 course is offered to both undergraduate and graduate levels. The goals of the course are to introduce students to the next level of 3D visualization that is real-time Virtual Environment, and at the same time provide students the opportunity to acquire the skills in designing and modeling using cutting edge technology. In this paper, we discuss 1) 3D games development tools in terms of: CAD compatibility, high resolution real-time 3D visualization, the deployment cost for education, and 2) student feedback on using the BuildITC4 software that is developed based on the C4Engine games development tools.

KEYWORDS: 3D game engine, 3D modeling, animation, architectural design, virtual environment

1 INTRODUCTION

Current trend in the utilization of technology in the architectural field for design visualization requires a well-rounded architectural graduate; who is able to use and adapt technologies to solve day-to-day real-world design problems. Literature and various works in virtual environment (VE) have shown that 3D modeling and animation, and VE outweigh traditional 2D building representation where viewers are given enhanced visual, and to some extent manipulative and interactive three-dimensional models of the intended buildings (Burdea and Coiffet, 2003; Maher et al, 2003; Shiratuddin and Thabet, 2002; Cory, 2001; Pantelidis, 1997). Architecture students who are skillful in producing 3D computer models and animations can greatly assist their prospective employers in presenting design ideas to clients in an easy to understand form; "a picture is worth more than a thousand words", and "a 3D computer generated model is worth more than a thousand printed pictures". However, traditional rendering process of high resolution 2D images and scenes is still time consuming.

The School of Construction at the University of Southern Mississippi has been experimenting utilizing 3D games development tools to assist teaching and learning process of 3D computer modeling and animation, and enhance the outcome produced by students. At the School of Construction, we offer courses such as the ACT348/ACT538 - 3D Modeling and Animation Applications 1 and ACT450/550 - Virtual Reality Applications 1 to our architecture students to prepare them to design buildings and facilities that meet clients' requirements, designs that are interactive and visually appealing. The courses are offered to both undergraduate and graduate levels. Our goals are to introduce students to the next level of 3D visualization, in a real-time virtual environment (VE), and provide students the opportunity to acquire skills in designing and 3D modeling using cutting edge technology. To date, we have not gathered comprehensive and quantifiable data to measure the performance of our students when using 3D games development tools in comparison to using conventional 3D modeling and animation tools. We have however gathered valuable preliminary data. We observe that when students use 3D games development tools, they are able to walkthrough their 3D CAD design in a VE. Additionally, they can produce high quality photo-realistic rendering in a much lesser time compared to using traditional 3D CAD modeling tools such as 3D Studio Viz. Currently, we are developing and enhancing our real-time design software tools called BuildITC4. Previous versions of BuildITC4 was developed based on Garagegames' Torque Game Engine (Shiratuddin and Fletcher, 2006) but as technology evolves the current version of BuildITC4 is developed based on a next generation 3D engine known as C4Engine from Terathon Software (Lengyel, 2007). Despite issues of instability and slow software responses, our students gave valuable feedback on BuildITC4 in particular and also the course in general. This paper discusses 1) BuildITC4 and the C4Engine in terms of, CAD compatibility, high resolution real-time 3D rendering and visualization, visual design approach, using VE to improve design communication and collaboration, the deployment cost for education, and 2) student feedback on using BuildITC4, and plans for improvement.

2 BUILDITC4 FOR TEACHING AND LEARNING

2.1 CAD compatibility

In the early years of 3D games development, there was compatibility issue between 3D CAD models and the 3D games development tools. Many early 3D game engines were unable to support 3D objects created in CAD software. Due to limitations in hardware and software technologies, 3D objects in the game's virtual environment had to be created from scratch either through specialized tools or through hand-coded programming languages. Limitations in 3D hardware technology disallowed performance of real-time rendering of complex high polygon count 3D CAD geometry. However, the past few years have shown dramatic improvement in the ability of games engines to utilize 3D objects created using commercial 3D CAD software (Shiratuddin and Thabet, 2003). Commercial software such as 3D Max, Maya, XSI etc. is now an accepted component of the game's content creation pipeline.

For the past two semesters we have been using Google SketchUp, Artlantis and BuildITC4. SketchUp is used for development of 3D architectural models, Artlantis for the traditional photorealistic renderings and animations, and BuildITC4 for the creation of a VE.

BuildITC4 is currently under development and it is based on the C4Engine. We licensed the C4Engine and have access to the engine's source code. Access to the source code allows us to make changes and accommodate our students' needs. The C4Engine came with variety of tools and one of them is a visual level editor known as the World Editor. The World Editor (Figure 1) allows for the assembling and creation of a real-time VE scene. The scene assembling may involve the placement of imported 3D CAD objects, creation of basic 3D primitives (cube, sphere, cylinder etc.), placement of lights, defining interactions and physics properties, and any special properties that may be needed for the scene. In terms of importing 3D CAD file, presently the C4Engine only supports the Collada file format.

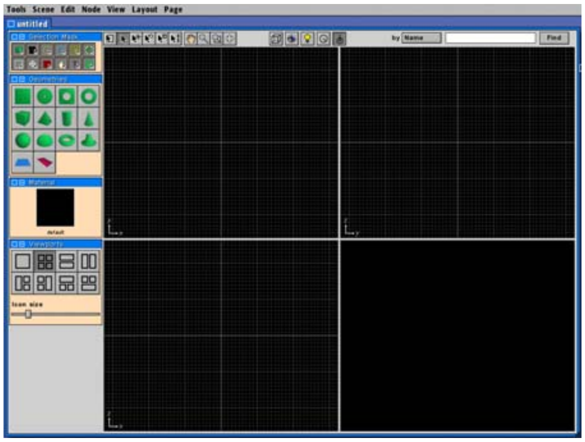


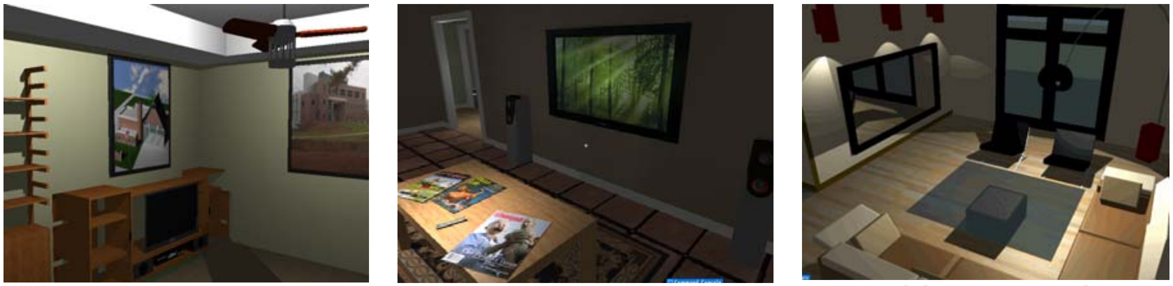
FIG. 1: The World Editor

Collada stands for COLLAborative Design Activity. Collada is developed by The Khronos Group and consortium members as an open standard Digital Asset schema for interactive 3D applications. To date, Collada has successfully provides a neutral zone where it becomes possible for competitors to work together in the design of a common specification, and creates a new paradigm in which the schema is supported directly by the Digital Content Creation (DCC) vendors. Each vendor will be able to write and support their own implementation of the Collada importer and exporter tools. Varieties of Collada implementations can be found in many commercial and open-source 3D CAD software such as 3D Max, Maya, SketchUp, AutoCAD, ArchiCAD, XSI, Blender, Milkshape 3D etc. The Collada schema supports all features of modern 3D interactive applications which includes programmable shader effects and physics simulation. The Collada XML Schema document can be accessed publicly on the Internet for online content validation (Khronos Group, 2007).

CAD compatibility with BuildIT is thus achieved by using 3D CAD objects created in SketchUp (through the use of the Collada file format). We use Google SkecthUp for 3D modeling and animation purposes as it is a more mature 3D design tool with the ability to create complex 3D geometry and better in texturing as compared to BuildITC4's World Editor itself. Using SketchUp, our students are free to develop complex 3D designs and they can easily bring in their designs into BuildITC4 for further real-time interactions and 3D visualization.

2.2 High Resolution Real-Time 3D Visualization

Many computer games environment nowadays have reached a high level of realism, not only in terms of gameplay but most importantly 3D visualization. Compared to current commercial VE development tools, computer games support the latest technologies in computing and graphics. Many of the 3D objects shown in computer games VE are more realistic and life-like. This is mainly due to support for high resolution and realistic images that are used as textures for the 3D objects in the game's VE. Features like support for 24-bit, 32-bit and 64-bit color images, multi-layered texturing, volumetric environmental effects, shaders, advanced dynamic lightings and dynamic shadows etc. are now available in many games development tools. Even though these features were solely created by games developers to be competitive in the games industry, we believe and have observed that utilizing games development tools greatly enhance the design produced by our architecture students. Our students are able to create believable VE to represent their designs in real-time and the use of high resolution images greatly enhances the appearance of 3D objects in a VE (Figure 2).



Work by David Roberston

Work by Jeremy Wright

Work by Rion Snowden

FIG. 2: Examples of students' work

In a real-time VE generated using 3D games development tools, students can freely navigate as desired, and examine 3D objects from different positions, angles, and orientation; either by walking-through or flying-through the scene. Interaction and changes can be made in real-time to various 3D objects in the VE. The student's first (or third) person viewpoint generated in the VE by the games rendering engine is continuously updated at 30 fps or more (with many reaching between 60-120 fps). The higher the frame-rate, the smoother (lagging is greatly reduced) the real-time images are presented; therefore, allowing students to experience the VE to a greater level of immersion and interaction, than it is possible for a passive observer.

2.3 Visual Design Approach

Early games development tools were more programming oriented. The C4Engine development tools being one of the next-generation engine, employs a more visual graphical user interface (GUI) approach in designing a VE, and yet, it still allows for flexibility for new features to be programmed through its' programming environment. Many of the recent games development tools have similar visual design approach such as the Unreal Engine's UnrealEd. Our students are visual learners and with BuildITC4 they are able to demonstrate the ability and potential to explore their imagination and creativity in designing real-time 3D VE scenes. Many 3D objects in the VE can be modified with ease using buttons, icons and menus available in BuiltITC4, such as:

- dynamic lightings can be applied and changed instantly
- built-in special effects such as water, flares, lens and fire effects can be applied easily,
- textures can be changed and oriented accordingly to any surfaces of the 3D objects, and
- interaction can be defined through a visual script editor

These features enable our students to learn to design in a much more direct way by using a more GUI oriented software. Using BuildITC4, our students are more focused in designing rather than being distracted and weighed down to learning how to program.

2.4 Deployment Cost for Educational

The deployment cost of setting up a VE course in an academic environment to the very least involves acquiring computers and software. In support of using games development tools in an academic setting, the games development tools in general can be characterized as being low-cost-high-performance and their availability off-the-shelf from many computer stores or online sources. Although fully licensing a commercial game engine is non-economical, recent development shows that game engine developers are making their technology available for free or at a very affordable price for non-commercial and educational purposes. In our case, we purchased licenses for the C4Engine. A C4Engine license only cost $200 and can be used not only for educational and non-commercial, but also royalty-free commercial applications development. Due to low-cost and readily availability of the C4Engine games development tools, we are able to provide our students with the games development tools executables installed on all the computers in two labs. We also allow for our students to download a copy of BuildITC4 so that they can use it with their own personal computer.

In past semesters and prior to using the C4Engine, we used the Unreal engine from Epic Games and GarageGames's Torque Engine. Epic Games allows their Unreal engine technology to be used freely in an academic and non-commercial setting, whereby educators and students can either download the freely available Unreal Runtime version, or simply purchase one of the Unreal Tournament games. GarageGames also allows the use of their Torque Game Engine free of charge in academic settings. In addition, the Torque Game Engine software development kit can be licensed royalty-free for $150 for independent developers and commercial users for less than $500.

Another cost factor that can be reduced is the computer ownership cost. Many games and games developments tools are designed to be scalable as the market audience are broader with many end-users owning only low-end to mid-range home personal computers. Scalability means that the games development tools have support for low-end personal home computers to the very latest high-end graphics workstations. We believe scalability allows many academic institutions to be able to acquire, deploy and use games development tools in currently established computer labs for teaching and learning the development of VE. In summary, by utilizing games development tools, the deployment cost for a teaching laboratory will be greatly reduced as licensing can either be free or at a very affordable rate, with low-cost computer ownership.

3 STUDENTS FEEDBACK ON BUILDITC4

With the two courses we offer; one in 3D modeling and animation, and the other in development of VEs, our students adapted readily to the basic VE scene building process. The teaching and learning concepts are built upon the principles of CAD, 3D modeling and animation, and also digital imaging. We find that students are motivated to learn to use the tool. Our unconventional approach of utilizing games development tools (that usually lies within the domain of computer graphics and computer science) for design is exciting for our students. We believe that this partnership can work well and is beneficial for students. We collected feedback from twenty-seven of our undergraduate students. As much as we would like BuiltIT (in specific) and the C4Engine (in general) to be a perfect tool the first time it is used in a classroom setting, results depicted from Table 1 shows that more work and improvements need to be incorporated into BuildITC4. Improvements based on our students' feedback ensure that the software truly assist and support the instructor in teaching the course, and students in learning the principles of VE and its creation from the course.

The following are some of the questions that were asked to the undergraduate students, and the feedback they provided. Further detailed analysis is underway and planned for future work. Despite the not too high ratings given by the majority of the students, a number of students have produced high quality work using BuiltIT.

Question	**Rating**	**0**	**1**	**2**	**3**	**4**	
a) Performing tasks is straightforward	never	1	10	13	3	0	always
		4%	37%	48%	11%	0%	
b) Organization of information	confusing	5	3	14	3	2	very clear
		19%	11%	52%	11%	7%	
c) Software speed	too slow	8	8	6	3	2	fast enough
		30%	30%	22%	11%	7%	
d) Software reliability	unreliable (always crashes)	10	10	4	2	1	reliable (no crashes at all)
		37%	37%	15%	7%	4%	
e) Learning to operate software	difficult	7	8	9	3	0	Easy
		26%	30%	33%	11%	0%	

Table 1: Student feedback on using BuildITC4

a) Performing tasks is straightforward

The World Editor is an integrated tool that allows for the scene assembling of a real-time VE. Varieties of tasks can be performed in the World Editor and they range from texturing, defining different types interaction in the VE, 3D real-time modeling etc. Table 1 shows that more than 50% of students are able to perform tasks with minimal complications. This corresponds closely with the results in (b).

b) Organization of information

When we first utilized the C4Engine, we realized that the organization of information such as the menu layout, icons used, the terms used in dialog boxes etc. is foreign to architecture students. However, we decided to go along with majority of the C4Engine's conventions. We modified terminologies that may confuse the students. Some of the terms that we replaced are "World" with "Scene", "Import Model" with "Import Scene" etc.

We provide our students with printed step-by-step tutorials, hands-on experience in the lab and also lecture. Table 1 show that more than 50% of students find that the information presented to them is comprehensible and clearly understood.

c) Software speed

Each computer in our labs has the following configurations; Intel Pentium 4 2.4 GHz processor with 1 GB RAM, a 40 GB hard drive and an nVidia GeForce 6200 LE graphics card. The computer's specifications conform to just about the minimum requirements to run the C4Engine. Even though the C4Engine can handle a high number of polygon count geometry, the bottleneck to render complex scene came back down to our computer's hardware limitations. More than 50% of the students felt that once 3D terrain geometry and a high number dynamic lightings were introduced into the scene, the interaction with the software began to slow down and at times seemed to be unresponsive. We referred to the online forum and found out that many of the C4Engine's users recommend utilizing no less than a 2 GHz processor with the nVidia 8000 series graphics card with at least a 128 Mb dedicated video memory.

d) Software reliability

In its current form, the C4Engine is robust. However, it is still in its early stage compared to the Unreal Engine or Torque Game Engine. The C4Engine is at times very sensitive to improper execution of tasks, and when this happened it will crash. Based on the results shown in Table 1, more than 50% of the students experienced software crashing. We are continuing to improve BuildITC4's and the C4Engine's reliability.

e) Learning to operate software

Some of the students' comments regarding using BuildITC4 include, "….too many steps to achieve a task…", "…to many copying files and folders…" and "…terms used in dialog boxes are hard to understand…" During the first few weeks of using BuildITC4, majority of the students had difficulties learning how to use the software. This was mainly due to the fact that many of the terms used are different than what they were used to, the menu layout is different than many commercial CAD software and the design concepts in a real-time VE is unfamiliar to them. We also noticed that many students simply lack practice or spent less time learning the software. However, we remedied this by giving them in-class exercises, quizzes and more frequent weekly deliverables. As Table 1 depicts, by the end of the semester, almost 50% of the students felt that operating BuildITC4 became less complicated and they were able to complete all the assigned exercises and projects.

4 CONCLUSION

We are continuing to improve BuildITC4 and the C4Engine to assist our students in learning VE. Our in-house Architecture and Construction Visualization research and development lab (ACV-rl) is currently utilizing 3D games development tools to further simplify tools for more rapid end-product creation, and reduce the training and learning requirements for students and professionals. Future research is planned in areas such as obtaining quantifiable data in measuring the learning performance and productivity of the students when using 3D game development tools versus the traditional 3D tools.

Other challenges that we observe are (1) teaching students to understand which tools are most appropriate for the task and (2) developing the programming/scripting skills and understanding for design students and/or developing the organizational skills to work in scene building development teams. There are numerous online communities available on the internet to provide educational support for both instructors and students (in the forms of tutorials, forums, chats and video courses). An example of a website that provides video courses on how to use games development tools is 3DBuzz.com. Besides the online communities, reference books such as from Finney (2004) who authored "3D Game Programming All in One" and "Mastering Unreal Technology : The Art of Level Design" by Jason Busby et al (2004), provide additional support for the students depending on the student's depth of interest.

The use of VE is a natural extension of world representation beyond traditional 2D rendering and more recent 3D modeling and animation techniques. A few years ago, the software tools used to develop real-time VEs of substantial content (more than what VRML was capable of) were extremely costly and did not provide access to the underlying code. However, more recently, a number of tools have become available primarily through the impetus of the online computer games community that provides such access at affordable costs.

We believe that by exposing our students to cutting edge technology such the 3D game development tools, the students can develop rich understanding of architecture, architectural 3D modeling and also 3D VE. Our students are at the advantage in terms of their acquired skills, competitiveness and employability. Not only will they become competitive in the design and construction engineering industry, they can also explore the possibility of working with the video and computer games, and movie industries. These multi-billion industries are always looking for creative and talented people in producing photorealistic imageries, animations and movies. Our students have been successful in securing employment in other industries for example, forensic animation and visualization. Visualization for real estate development, civil emergency response, and military applications has also provided employment avenues for our students.

Many design schools and college programs in the USA have not taken advantage of such features available and offered by games development tools. We believe that there are still perceptions such as, computer games are for entertainment and not for serious academic use, the development tools are expensive, and too programming intensive. Our program however has successfully utilizes games development tools in teaching students to design buildings and facilities, and view them in a VE, without exposing the students to intensive coding and programming. The BuildITC4 software that we developed based on the C4Engine game development tools, provides state-of-the-art techniques for rendering real-time VE scenes. The engine provides both instructors and students a low-cost solution to develop and explore VEs, and provide students with a marketing edge upon graduation. We hope that in the near future, many more schools and programs would adapt games technology in their design and construction programs as tools to assist teaching, learning, and also research.

ACKNOWLEDGEMENT

The authors would like to sincere thank Eric Lengyel and all C4Engine's online community members for their assistance throughout.

REFERENCES

Burdea, G.C. and Phillippe Coiffet, (2003) "Virtual Reality Technology", Second Edition, John Wiley & Sons, Inc., 2003.

Busby, J., Parrish, Z., and VanEenwyk, J., (2004) "Mastering UnrealTechnology : The Art of Level Design", Sams, 2004.

C4Engine, Terathon Software (2007). Website: http://www.terathon.com/index.php

Campbell, D.A., (2000) "Architectural construction documents on the web: VRML as a case study", Automation in Construction 9, pp. 129–138, 2000.

Cory, Clark A., (2001) "Utilization of 2D, 3D, or 4D CAD in Construction Communication Documentation", Fifth International Conference on Information Visualisation (IV'01).

Dunston, P.S. , Wang, X., and Lee T., (2003) "Benefit/Cost Analysis of Desktop Virtual Reality Application in Piping Design", Proceedings of the Conference on Construction Applications of Virtual Reality, CONVR 2003, Virginia Tech, Blacksburg, Virginia, pp. 22-31, 2003.

Epic Games (2007), Website: http://www.epicgames.com

Finney, KC (2004) "3D Game Programming All In One", Thomson, Boston, 2004.

GarageGames (2007), Website: http://www.garagegames.com

Khronos Group (2007), Collada - 3D Asset Exchange Schema. Website: http://www.khronos.org/collada/

Maher, M.L., P. Liew, and J.S. Gero, (2003) "An agent approach to data sharing in virtual worlds and CAD", Proceedings of the CIB W78 Conference on IT in Construction, Auckland, New Zealand, 2003.

Pantelidis, V. (1997) "Virtual Reality and Engineering Education", John Wiley & Sons, Inc., 1997.

Sabelli, N., (1994) "On using technology for understandings science", Interactive Learning Environments, 4(3), pp. 195-198, 1994.

Shiratuddin, M.F and Thabet, W. (2002) "Virtual Office Walkthrough Using a 3D game Engine", Special Issue on Designing Virtual Worlds, International Journal of Design Computing, Vol 4., 2002.

Shiratuddin, M.F. and Thabet, W. (2003) "A framework for a collaborative design review system utilizing the Unreal Tournament (UT) game development tool", Proceedings of the CIB W78 Conference on IT in Construction, Auckland, New Zealand, 2003.

Lengyel, Eric (2007). "C4Engine.", Terathon Software. Website: http://www.terathon.com/c4engine.

Shiratuddin, M.F. & Fletcher, D. (2006), Southern Miss' Innovation and Commercialization Park: Development of a Large Scale Real-Time Virtual Reality Environment, Proceedings of 6th International Conference on Construction Applications of Virtual Reality, August 3-4, 2006, Orlando, Florida, USA.

Shiratuddin, M.F., Fletcher, D., Breland, J. and Kitchens, K. (2007), "Collaborative Design Virtual Environment", Video Proceedings of the IEEE VR 2007, North Carolina, USA.