Enhancing Architectural Communication with Gaming Engines

Michael Hoon, Michael Kehoe
New Jersey School of Architecture

Abstract

This paper makes a case for exploring the feasibility of utilizing the advanced graphics and sound systems of contemporary gaming engines to promote architecturally relevant work. Gaming engines, while developed largely for the PC entertainment industry, have vast potential for application in architecture. This paper will explore the depth of this potential and will outline work demonstrating the advantages and the limitations of this technology. The supporting research and observations examine the technology and reveal its potential usefulness as an instructional or depictive authoring tool. Game engines were selected that had appropriate graphical prowess, but were customizable as to allow the removal of game-specific features (to create a “professional” user interface). Projects were authored that expressed complex building details using the engine for visual depiction. The details, which included constructional components, structural assemblies, or simple design nuances, were modeled with 3D geometry and realistically textured and lighted. The game engine allowed one user or many remote simultaneous users in the virtual environment to interactively explore the presentation in real time. Scripts were developed to encourage end-users to interactively disassemble or reassemble building components as desired. Audile and/or text-based information regarding the assembly sequence were provided by exploiting the game interface features. Furthermore, interactive object scaling was provided to facilitate analysis of component relationships.

Enhancing Architectural Communication with Gaming Engines

Michael Hoon, Michael Kehoe
New Jersey School of Architecture

1 Introduction

The growing sophistication of gaming technology merits in-depth examination. Many popular computer games, while they may be seen as “casual” applications, ironically require more computational power than most business and CAD software. Optimized code, coupled with high-performance hardware graphics acceleration through OpenGL and DirectX, currently allows for high-resolution, richly textured simulations of real-time interactive spaces. Advanced spatial sound also serves to enhance user experiences. Multi-user capabilities allow several reviewers to simultaneously explore and interact within a single project through LAN or internet connections. Customizable avatars allow for user identification in the virtual environment and permit gesturing and observation of avatar-to-avatar and avatar-to-object interaction. The multi-user environment also indicates opportunities to examine such other user behaviors as way-finding, aggregation, and condition-specific reaction (e.g., overcrowding, fire propagation, prolonged elevator wait times, etc.). The results of student projects gleaned from two semesters of coursework will be used as a basis of student reaction and possible usefulness in academia. Experiences with client reaction to actual projects will also be offered. Additionally, seven years of independent research in gaming technologies will be referenced vis-à-vis its potential application in many aspects of architecture, such as design collaboration, interactive detailing, acoustics simulation, historical analysis, and life-safety simulation.

2 Early Research

Through research conducted as early as 1995, it was determined by the authors of this paper that gaming engines have credible potential as a design and presentation tool. In experimental digital design studios conducted at that time, projects were submitted in the form of animations, VRML worlds, and game environments. Research continued into the feasibility and effectiveness of these different methods. In specific regard to the observations of the use of gaming engines, the perceived impact, both positive and negative, on the development and presentation of conceptual spatial ideas will be explained. To better understand the usefulness and limitations of gaming engines in Architecture, practical experimentation was undertaken. In all, four game engines were examined *(Doom II, Quake I, Quake II, and Quake III,* all from ID Software*)* for their capacity to represent both existing spaces and conceptual spatial designs. While a range of findings were uncovered during testing with all of the aforementioned game engines, specific noteworthy aspects of potential application to architectural modeling and presentation are briefly summarized by focusing on one early project that utilized the Quake II engine.

Figure 1. Screen captures from urban residence project.

In 1998 the *Quake II* engine was used to depict a stage in the conceptual design process of an urban residence. The designer of the project agreed to consult on the game environment development while concurrently creating still renderings in *3D Studio 4* from Kinetix. Since the design had reached a developmental plateau before game modeling commenced, the work would involve only the transcribing of a conceptual design data and would not be used for design development until after presentation to the client.

The clients of the project were presented with the final results as a supplement to a traditional design proposal. The designer opened with a presentation of traditional plans, sections, elevations, and traditional digitally rendered image. The interactive game interpretation of the design was demonstrated next. Both clients demonstrated enthusiasm for the game presentation, stating that it had clarified many personally important issues, particularly in regard to material choices, spatial adjacencies, and visual accessibility. None of these issues were obscured in the traditional presentation; on the contrary, the designer conveyed these ideas quite lucidly.

In conclusion, the game model had acted to substantiate the same information that was given via traditional methods, which included both orthographic line drawings and detailed high-resolution renderings. It is not known how the clients would have reacted to the game presentation alone. Although the scope of these findings is limited, it was concluded that most laypeople will find value in interactive virtual presentations because of the diminished abstraction and the experiential nature of the depiction. Feasibility, however, was an unresolved question which needed be subjected to cost/value analysis.

The documentation of the urban dwelling design was extremely useful with regards to the clear, accessible representation of a complex conceptual design solution. Complemented by traditional tools, the game walk-through added an experiential dimension to the presentation and gave the clients valued confidence in the elapsed process, the current results, and the potential of continued design development.

Another documented experience with the *Quake II* engine is offered by Richens and Trinder, who used it advantageously in the design development of the new Computer Laboratory at the University of Cambridge.

3 Coursework

An elective course using gaming engines for the design and depiction of architectural space was developed by the authors and first offered in spring 2002, when the gaming engines and ancillary editing tools were judged to have reached an acceptable level of graphics capability and technical maturity. The purpose of the course was to introduce design students to the power of the game engines within an architectural context. While the structure of the course was based on the digital design studio paradigm, the additional capabilities of the game engine were used to exploit the potential for participant immersion, spatial interaction, and multi-user design collaboration.

Successfully negotiating the course necessitated a many-layered infusion of new skill sets for both students and instructor. Beyond new methods of 3D modeling and texture mapping, there was a need to learn new scripting conventions for object and surface behavior. While more mature 3D software can be expected to include high-level interfaces for the editing of scripted textures, or “shaders,” the script manipulations in the game interface had to be executed using text files.

There is only a cursory relationship between game engine tools and “industry-standard” architectural visualization tools. For example, there is a very limited offering of primitive geometry. The basic building block for game engine geometry is the hexagon (generally referred to as a “brush”), and higher-level geometry (including spheres) generally requires a skillful distillation of complex shapes and curves to constituent

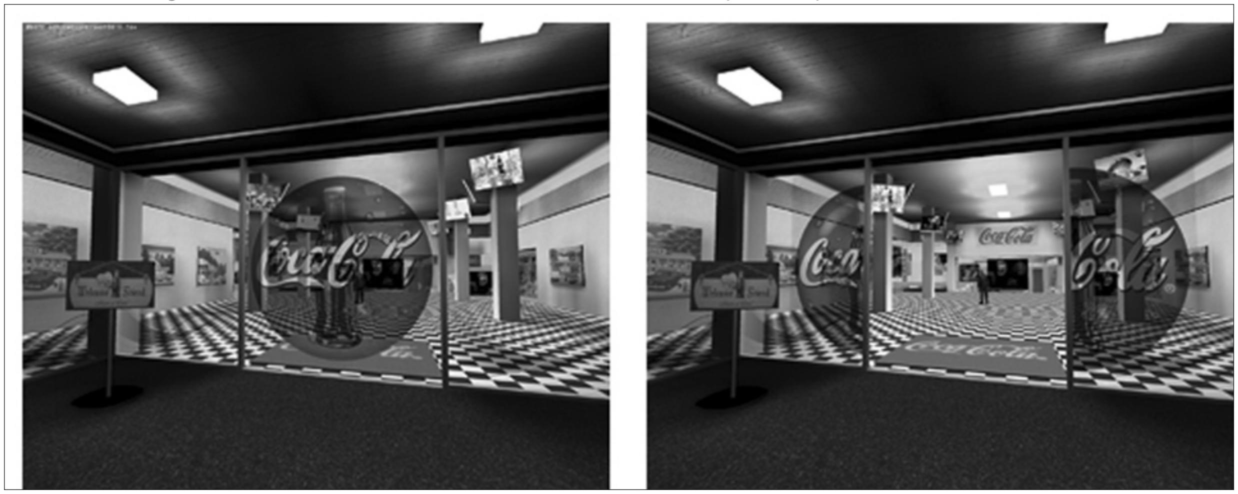

Figure 2. Screen captures illustrating avatar and functional doors.

hexagons. Increasingly, game editors are supporting macros and plug-ins that will deliver complex shapes. However, the judicious designer must stay ever mindful of the true nature of the geometry in use, as a successfully executed wireframe model does not guarantee a successful final compilation or acceptable in-game performance. Furthermore, proper texture mapping generally requires painstaking refinement on a face-by-face level. For example, a texture which spans the multiple hexagons constituting a complex curve generally requires adjustment at the individual polygon faces for acceptable results.

Because of the limitations listed above, many experienced digital designers were easily frustrated; gratifying discoveries were often followed by difficult limitations. Comparisons to familiar modeling and rendering software were unavoidable, as students struggled to adapt to low-level techniques. Additionally, spatial designers needed to invent interfaces within the virtual environment to indicate to users the presence of “triggerable” events such as the deconstruction of a wall detail. Despite the difficulties, however, the open exploration of these issues forced designers to move beyond a static-image paradigm and consider their decision-making beyond a superficial level.

The game engines chosen for use in the course were versions of *Quake* engines. The original engines were modified to remove irrelevant gaming-specific features. Through a series of assignments within the course, students were asked to iteratively use and test the game engines for the collaborative exploration of a designed environment. After completing theoretical readings on virtual reality, the students were asked to design and create spatial environments using the 3D graphical editor *QuArK* (a freeware game level editor not to be confused with *QuarkXPress*). Students made their environments available for others to navigate in real-time and offer comments. Three design reviews were conducted during which critics were asked to enter the designed environments, explore at will, and interact with the student as well as others present in the same virtual spaces. Both local and remote reviewers participated. Communication was established through an IP voice channel and through avatar gesturing. An LCD projected image in the main critique space was periodically switched to display the first-person view of the currently speaking critic. Critics could allow their avatars to point or otherwise gesture in accordance with their voice comments.

Three major projects were required to complete the course. The first, an art gallery design proposal, accustomed the students to the interface and forced them to concentrate on circulation, lighting, and the development of custom texture libraries. In the second project, students used the scripting functions of the game application to explode a construction detail within a virtual space for abstract analysis. This also required careful consideration of how an exploding detail event should be announced to, and triggered by, those navigating a virtual space. The purpose of the final project was to express the cumulative knowledge of the semester in a student-devised project, in which each student was asked to demonstrate an accomplished understanding of the media. The projects submitted in the most recent offering of the course (which utilized a version of the Quake III engine) included a virtual museum for soft drink memorabilia, a recreation of Escher’s infinite stairs, a modern day interpretation of Plato’s Allegory of the Cave, several virtual spaces which adjusted their configuration according to user behavior, an enormously scaled section of a four-stroke internal combustion engine, and a poetic exploration of the threshold between real and unreal featuring disparate spaces, atmospheric soundtrack, and ambiguous movement.

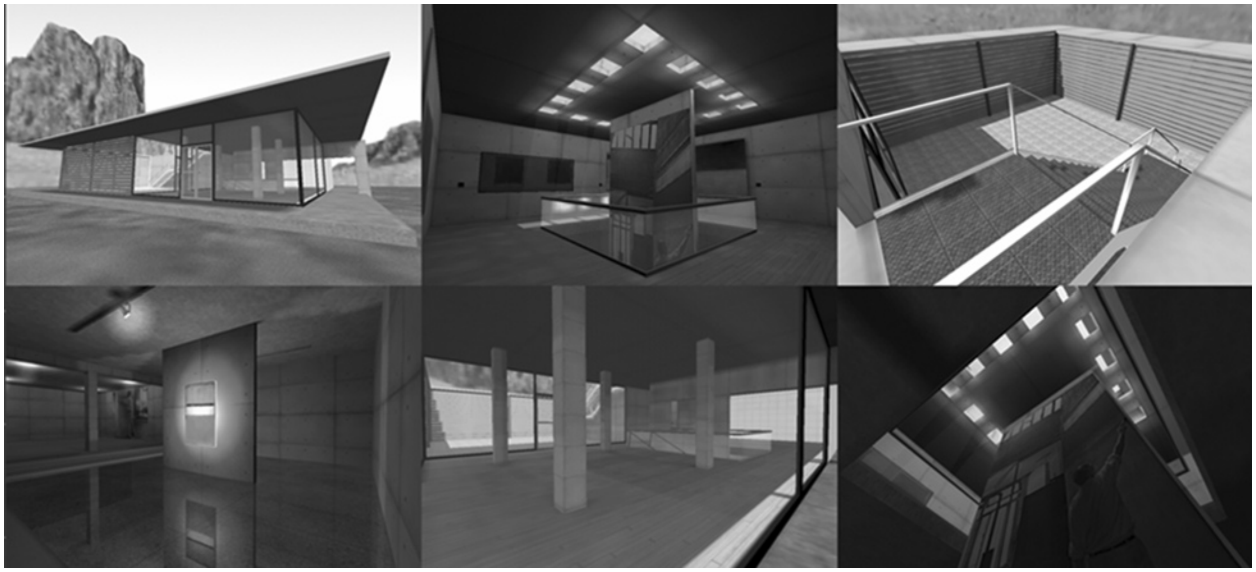

Figure 3. Screen captures illustrating virtual gallery.

4 Observations

The following is a summary of current gaming technology features observed during periodic research outlined by the mentioned projects and assignments. They are intended to be specific to potential usefulness in architectural applications. Listings have been broadly categorized in terms of perceived advantages and limitations.

4.1 Advantages

- Fully Interrogative. Movement is not forcefully prescribed or scripted. By observing the directional tendencies of participants, as well as their propensity to linger or aggregate in particular spaces, architects can tailor designs changes responsively to the more efficient use of all spatial resources.
- Tactile Solidity. Solids in game environments assert their solidity. Through collision-detection, gaming engines allow for proper navigation through spaces bounded by solid elements such as walls as well as realistic navigation of staircases and ramps.
- Integrated Audio. Sounds can be ambient, or triggered by specific events in the virtual game world, complementing the visual stimuli.
- Multiple Simultaneous Users. Designed game environments can be simultaneously experienced by multiple remote participants.
- Inclusion of Avatars. Networked implementations allowing multi-participant interactions provide an enriched simulation of the human/spatial dynamic.

Figure 5. Screen captures illustrating object/avatar scale.

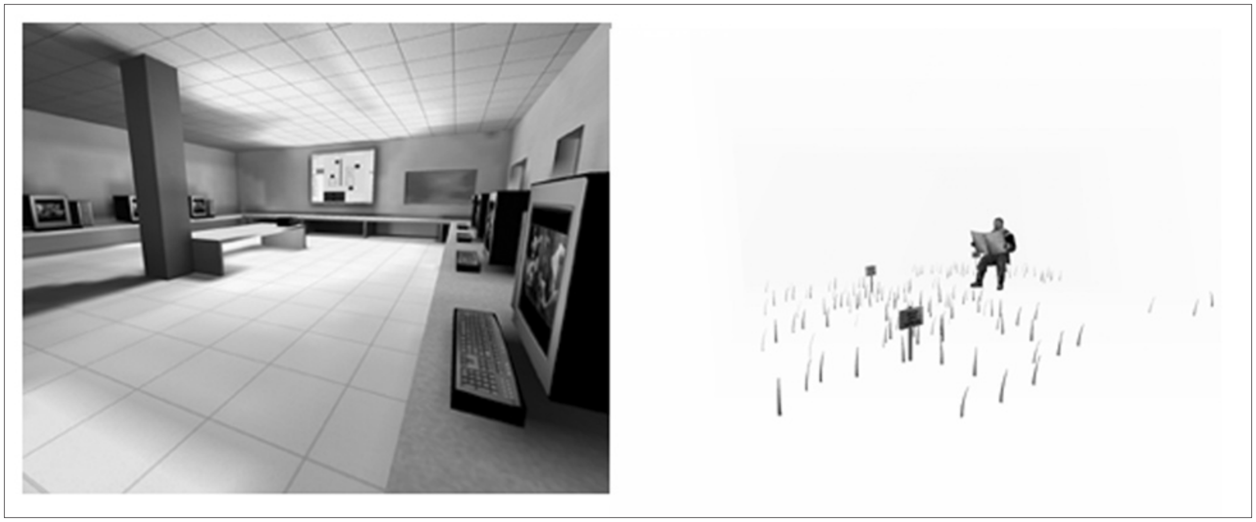

Figure 4. Screen captures demonstrating experiential juxtaposition.

- Simulation of Movement. The dynamic relationship of occupants to spatial boundaries allows better understanding of the ergonomic characteristics of a space. Observation of travel time further enhances the understanding of spatial juxtaposition, hierarchy, and scale.
- Simulation of Site. Environments with views of the surrounding world can be modeled using panoramic texture-mapping to simulate distant views. Support for alpha-channel texturing further allows decreased-overhead inclusion of simulated site elements such as landscaping, fencing, skylines, etc.
- Interaction with Objects. Although it is currently limited, newer engines allow for increasingly higher levels of realistic player interaction with inanimate objects set in the virtual world.

4.2 Limitations

- Comprehensiveness. The entire environment must be addressed and modeled. 'Minor' spaces and details may receive as much scrutiny as major ones. As abstraction diminishes, so do the benefits of its economy. (This can be considered an advantage of the tool.)
- Complexity. Game environment creation software currently lacks turnkey functionality and professional polish. Familiarity with traditional CAD and modeling software is often more a hindrance than a help.
- Geometry Limitations. Objects in any possible scene must be rendered in real time. Most game environments necessarily avoid excessive geometric detail opting instead for textural detail, with mixed results.
- Lack of Standardization. The finer points of the virtual depictions (textures, behaviors, sounds) do not translate very well among the various game content creation packages. File exchange with industry-standard modeling and rendering software, while improving, is still very limited.

5 Summary

Most students who used the software displayed energetic enthusiasm for the game engine as design tool. Many were willing to learn programming languages to gain better control of the appearance and behavior of textures and moving objects within the space. While undertaking the complete design and modeling of even a small building proved daunting, many students found gratification and freedom in the move away from the still-image representational paradigm. Interestingly, many of the projects deemed successful by students and critics alike were generally more experiential than spatial in nature; keen observation of circulation tendencies and an understanding of the importance of spatial sequencing became desirable skills as still-image refinement skills were shelved.

6 Future Research

Research to further refine avatar behavior within the virtual environment is planned as the next step in the evolution of this technology as it applies to architectural design. As avatars act more realistically, it is believed that designers can more confidently create spaces that better respond to their movements and behaviors. (For previously accomplished research, changes to avatar behavior were affected after observation of students walking through hallway spaces, specifically their behavior when approaching oncoming students and their need for personal space.)

Another planned research pursuit is the successful translation of game character artificial intelligence to building components within the virtual environment. In theory, this would allow building components to respond intelligently to changes in climate, lighting, structural loads, or inhabitant behavior. The potential benefits of this research were positively demonstrated by scripted object behaviors observed during coursework.

7 Conclusion

Architecture is a profession which relies heavily on the clear and accurate communication of design intent. The sophistication of recent gaming engines brings to light many capabilities that have potential within architecture to meet these needs by leveraging the power of the modern personal computer. Through practical experimentation and observation, the research outlined by this paper has attempted to determine the nature and the direction of this technology.

References

Campbell, D., and M. Wells. (1994). A critique of virtual reality in the architectural design process.

Technical Report: R-94-3, University of Washington. Retrieved March 20, 2001 from the World Wide Web: http://www.hitl.washington.edu/publications/r-94-3/

Chronis,G., and W. Knibbe. (1998). The game engines of the not-too-distant future. IDG.net Retrieved March 20, 2001 from the World Wide Web: http://cgi.cnn.com/TECH/computing/9808/21/gameengines.idg/

Davis, S. B., A. Huxor and J. Lansdown. (1996). The design of virtual environments with particular reference to VRML. A Report for the Advisory Group on Computer Graphics. Retrieved March 18, 2001 from the World Wide Web: http://www.agocg.ac.uk/reports/virtual/vrmldes/title.htm

Isakovic, K. (1997). Commercial 3D graphic game engines. Technical University of Berlin. Retrieved March 18, 2001 from the World Wide Web: http://cg.cs.tu-berlin.de/~ki/game_eng.html

Richens, P., and M. Trinder. (1999). Exploiting the internet to improve collaboration between users and design team.

Cambridge University. Retrieved October 21, 2002 from the World Wide Web: http://www.arct.cam.ac.uk/research/pubs/pdfs/rich99a.pdf

Student work courtesy of Akekarach Paibulkulsiri, Mark Lagunzad, Aaron Fredericks, and Brandon Amodeo.