Game Engines and Virtual Design Studios
Technology and Pedagogy

Jules Moloney
The University of Auckland
http://www.architecture.auckland.ac.nz/jules/stringcve/

Abstract. *A discussion of the outcomes from the use of a game engine based collaborative virtual environment for virtual design studios. By way of introduction the use of a game engine is positioned between the high end visualization capabilities of immersive VR, and the high dimensional accuracy of 3D CAD. Software development, which address problems related to content creation and communication lag, are reported. This is supplemented with a more general discussion of the motivations for design collaboration between architectural schools. We confer with other researchers that lack of engagement is more related to pedagogy, then as a result of technical issues. In conclusion we discuss the potential of game play to enhance virtual design studios in terms of engagement and deliberation*

Keywords. *Collaborative virtual environments, game engines, pedagogy.*

Positioning a game engine relative to VR and 3D CAD

The School of Architecture at the University of Auckland has being utilising multi player computer game technology for design teaching since 2000. This approach allows a form of design collaboration known as "designing within the design" (Mayher & Simmoff, 2000) i.e. communication between collaborators and critics occurs within the emerging architectural form and simulated environment. By way of introduction to the rationale of using a multi player game engine, we discuss immersive VR and 3D CAD. The discussion does not identify particular software attributes, but attempts to identify the relative strengths from a designer's perspective. In this comparison we consider three key factors - the potential to design in context, interact with the environment, and functionality to support distance collaboration.

1. Designing in Context: VR systems such as CAVE or head mounted displays are distinguished by the use of displays that fill the user's visual field. The aim of such hardware is to distance the users from their current surrounding to enable visual immersion within the computer simulation. Within such systems the architectural model is viewed within a simulated environment, as opposed to typical 3D CAD software where the model is viewed in the abstract context of the preview window. The distinction between a typical 3D CAD work space in which geometry is manipulated as lines against the black screen, and the context of a VR environment where design can occur in context, is a vital distinguishing feature. This can enable a desirable shift in design emphasis. Rather than concentrating on architecture as a singular object devoid of

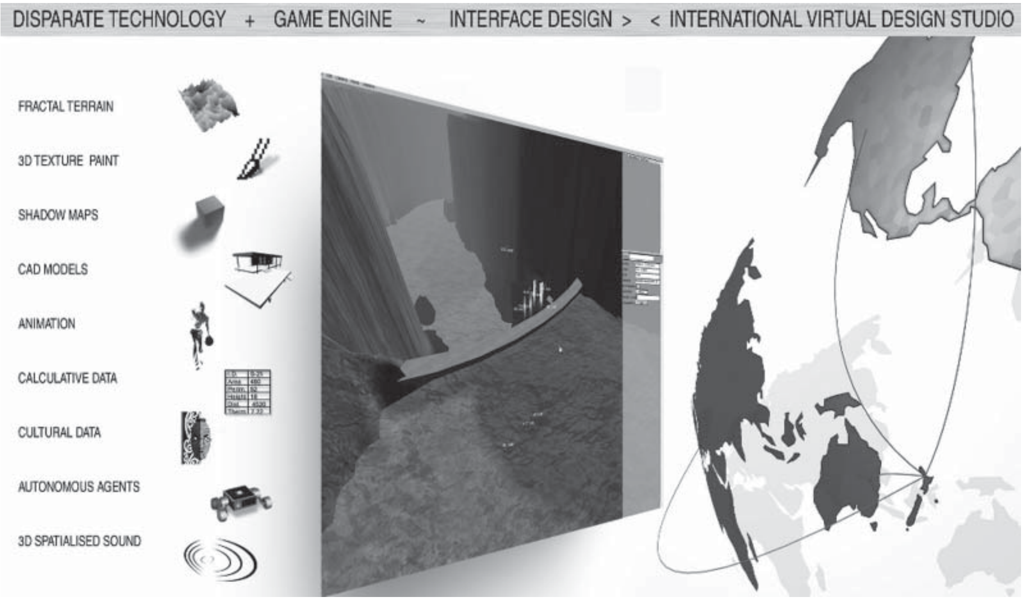

Figure 1. Overview of research agenda.

context, there is the opportunity to view the design in relationship to a simulated physical context, in relation to other design precedents, or in reference to embedded information. The potential for such evaluation in context while designing is a key distinction between VR and a typical 3D CAD workspace.

2. Interaction with Environment: A second factor that distinguishes VR systems from 3D CAD is the degree of interaction within the designed environment. The interaction can be simple visual cues, such as animated openings, day / night light cycles, or 3D located sound in order to simulate, for example, external noise levels in various parts of the design. Complementing this quasi experiential interaction is the possibility to embed data relative to the model, and its interaction within the environment. A 3D CAD system can of course provide similar functionality by rendering animations with sound, and attaching data to objects – the distinction is that a VR system is dynamic, allowing interaction between user, 3D model, environment, and embedded information in real time.

3. Collaboration. The third issue we consider important from a designers perspective is the degree of online collaboration. Distributed VR systems allow the opportunity for multiple users to collaborate - synchronous collaboration can occur via text chat, voice communication and through interaction with shared design components. In this way multiple users can evaluate and edit designs in a real time virtual simulation. 3D CAD systems do not allow for such real time collaboration, but there are possibilities for users to collaborate asynchronously. Typically these are text annotation and the marking up of designs via whiteboard type functionality.

The comparisons above relate directly to the needs of the user for which the software was specified. For CAD users the 3D functionality is a development of their primary role as drawing tools for architects to produce construction information.

High accuracy and detail are required, but there is little need to visualize the full context in which the detail is located. There is also no need for the model to be interactive, other then manipulation of viewpoint. Collaboration is limited to a supervisory activity - hence the development of drawing 'mark up' functionality to enable the project leader to monitor design decisions. Motivations for VR users are harder to identify, given the range of applications, but in general, users require a fully immersive experience with high levels of detail and some interaction.

In contrast to VR and 3D CAD, game users require highly interactive environments with good collaboration possibilities, while there is less need of full graphic immersion, as the focus is on game play and interaction. Conversely when compared to 3D CAD, game engines are not designed to support high levels of detail or dimensional accuracy. We suggest that in terms of the above discussion - design context, environmental interactivity and collaboration - an application based on a multiplayer game engine can be positioned between the high end visualization capabilities of VR, and the high dimensional accuracy of 3D CAD. While a game engine can be seen as 'budget' VR, it is arguable that full immersion and highly detailed representations is a hindrance to collaboration and design thinking, particularly at the early formative design stages. We propose that the most suitable use of game engine-based virtual environment is during these early design stages, when designers require a relatively low level of detail.

Motivations for Using Game Technology in the Design Studio

As a result of 8 design studios undertaken using various game engines, we have reported elsewhere key points of advantage in the use of a virtual environment, in comparison to typical software used in design education (Moloney & Harvey, 2004). In summary these are:

- Design iterations. Fluency of editing in real time, and the subsequent scope to make changes up to the last minute encouraged experimentation in a manner similar to the mark - interpret - mark cycle of sketching by hand.

- Sound as notation. Sound was an effective way to communicate intent in terms of spatial characteristics or materiality, and to convey a sense of occupation and functional use. The sound samples are not acoustically correct but are used primarily as a form of notation – in effect a 3D sound sketch.

- Context over object. The ability to explore projects in real time reinforced the importance of context, and challenged object based thinking where architecture is conceived as geometry devoid of occupation.

- Participatory critique. Reviewers can actively explore the design, rather than the passive observation of pre-rendered animations.

We are committed to further exploring the use of game engine based virtual environments for design education, and anticipate that due to the massive investment driven by the entertainment sector, game technologies will continue to provide state of the art computer graphics, sound, and interactivity at low cost. In particular, we believe that the current focus by developers on physics and artificial intelligence will provide great opportunities to embed performative feedback. While for some time there has been software available which allows evaluation of environmental and physical performance, this had tended to be used at the developed design stages. As discussed earlier our focus is on the early sketch design stages, and we anticipate that integration of aesthetic considerations and basic measures of performance will be invaluable in the context of design education. A precedent for this approach is Pangea, software developed by researchers at the Bartlett School of Architecture, which linked computer models to a range of techniques (Neural Nets, Genetic Algorithms, Fuzzy Logic) in order to communicate design performance to the designer while manipulating 3D form.

Virtual Design Studios based on Multiplayer Game Technology

We have been adding management and communication tools to the Torque multiplayer game engine to support Virtual Design Studios (VDS). Initial outcomes were presented at eCAADe 2002. (Moloney, 2002) The further work outlined in that paper - the implementation of asynchronous communication via 3D browsing and a PHP chat forum - has been implemented, and tested via two collaborative design studios with RMIT University. Subsequent in house studios have been undertaken at Auckland.

The virtual design studios demonstrated mixed results. Students at Auckland produced consistently good design and collaboration outcomes; the first studio with RMIT was deemed excellent by both institutions and outcomes have been exhibited internationally; while the second RMIT studio was less successful. Current thinking in regard to virtual design studios, is that success depends more upon the studio pedagogy then the technology (Cheng & Kvan, 2000). We will discuss these crucial implementation issues later, after first reporting problems of a more technical nature.

Subsequent Software Development

Two key software problems were identified in the studios undertaken with RMIT - content creation and communication time lag. The creation of design content was too laborious, with the process used being typical of game applications, in that a basic ‘level’ editing authoring application is available which is limited and crude in comparison to typical CAAD software. Compounding the problem were restrictions on texture formats, and the need to recompile scenes to evaluate design changes. Time lag in asynchronous discussion also became a problem. By the time students received feedback via the linked PHP forum the design had often changed, which would lead to frustration on the part of the design collaborators. There was a need to easily update design changes while maintaining the history of the linked text discussion.

Before implementing changes in the torque engine application which would address these two problems, a review of alternate software development platforms was undertaken. While there are a multitude of computer game engines and associated authoring applications, we restricted our trials to those with inherent networking capabilities, state of the art graphics and sound, and access to source code or a SDK. The Unreal Runtime engine was trialed in a design studio, and while the authoring environment was a substantial improvement over torque, we later found that access to the SDK would cost in excess of $ US 600,000. Other platforms considered were the Jet and Quake engines, but they were not substantially better than Torque in terms of content creation. In mid 2004, Deep Creator, a new content creation and virtual environment authoring environment came onto the market with an extensive software development kit. The primary advantage of this platform was the integrated authoring environment, which allowed geometry, graphics, sound, animation and interactivity to be authored in one application, and instantly tested without compiling the scene. The Deep Creator engine incorporates DirectX 9c graphics and audio within an authoring application very similar to the 3D Studio MAX interface. This includes sophisticated selection sets, numerical input, and other interface features common to CAAD, but seldom found in game authoring applications. Recently the software has been acquired by Right Hemisphere Ltd, who have added extensive file interoperability, thus enabling users to leverage existing software skills. The vastly superior authoring tools and excellent file support, convinced us a change of development platform to

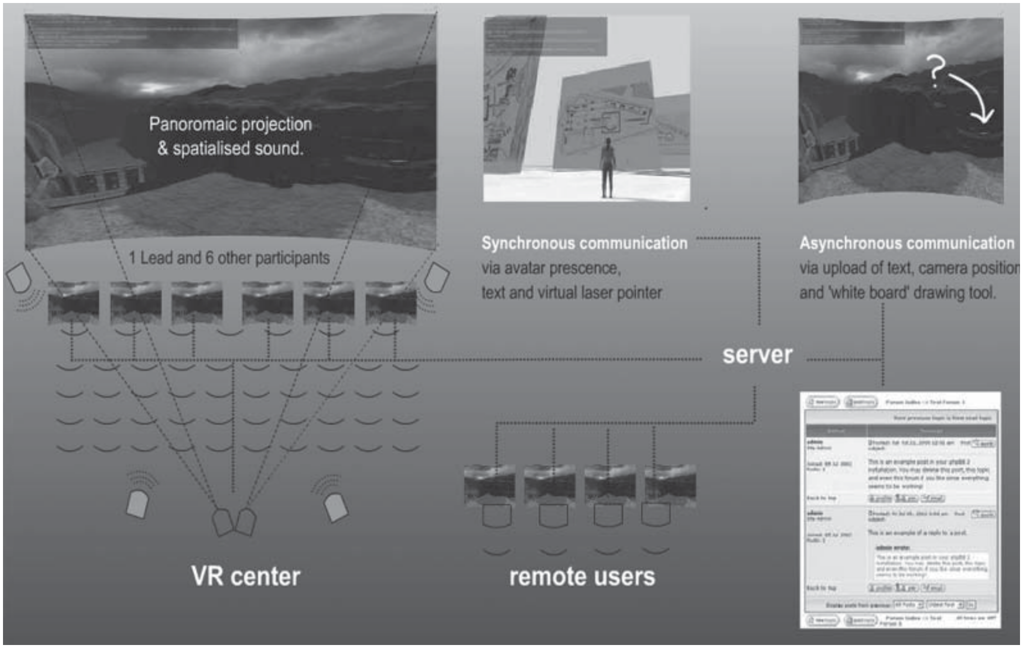

Figure 2. Communication functionality with the Torque Engine, Implemented 2003.

the Deep Creator game engine would address the problem of content creation.

The second problem of information lag, has been addressed by using the Deep Creator API to develop an integrated project database with the following features: multiple users can access a database server and add or modify design components; on client login new design components and design edits are detected and the client updated; implementation of a design history function allows users to step through previous design iterations. These initial functions have been planned to facilitate tutor input and peer to peer learning throughout the duration of the design class. While there are commercial applications with a similar approach, these predominately focus on the representation of geometry, with little integration of real time animation, interactivity and sound. As outlined above, it is the facility to combine VR like functionality with geometry, which attracts us to game technologies. Continuing our strategy of adapting existing technology we have implemented the integrated project database using available shareware products: CORBA Naming Service Installation; Apache HTTP Server; MySQL database; and Subversion CVS for version management.

Virtual Design Studio Pedagogy

It is ten years since the 1995 CAAD Futures conference 'The Global Design Studio', in which William Mitchell, in a key note presentation, predicted "development of geographically distributed virtual design studios is likely to be speedy "(Mitchell, 1995). This has not come to pass, and as leading researchers Tom Kvan and Nancy Cheng have discussed, problems with distance collaborations are more of a pedagogic and social nature then directly related to technical issues.

"People who are motivated to interact will workaround technical difficulties. Often technical systems support ancillary and non-beneficial activity. The success of a virtual studio depends upon the definition of the task, the expectations of the participants and methods of engagement." (Cheng & Kvan,, 2000)

In subsequent work Kvan goes on to suggest that the problems of VDS accentuate more general deficiencies within architectural education.

"The advent of the VDS appears to raise promising opportunities for reconsidering the way we teach design. It changes the relationship between teacher and student and student and the rest of the world. In this way, it opens up numerous opportunities."

(Kvan, 2001)

As discussed by others reviewing Kvan's research, these opportunities include two contentious issues in design studio pedagogy (Bennet & Broadfoot, 2003). The first is the focus on individual creativity, when in practice design is a collaborative activity, requiring social skills to enable excellence. The second issue is that by concentrating on end results - the packaged design presented at the final review - there is less time for deliberation and reflection as part of the design process.

Observations of the Auckland / RMIT studios confirm these problems. After been immersed in an architectural education that stresses individual creativity, most students were resistant to collaboration, or even co-operation between students. There was also a marked difference in engagement between the Auckland and RMIT student groups. This in my opinion can be related to the difference in computer and studio resources. Auckland has integrated computers into the studio spaces, with students having a workspace and a shared computer. These spaces are open 24 hours and include kitchen facilities and secure storage. Students effectively live in the studio when deadlines are due, helping each other out with both technical and design problems. By contrast RMIT have no design studios, with comparatively limited computer resources made available in labs. Tutor presence in Melbourne for the first studio enabled workshops to be undertaken in the lab, with the students 'occupying' the lab throughout the studio. However the second RMIT group had less group sessions, and in general were individualistic and worked primarily from home. While it is difficult to draw firm conclusions, our experience of teaching digital design studios have indicated that social cohesiveness, and a culture of sharing technical knowledge is crucial. The demands of a virtual design studio highlight that design teaching is predominantly a social process.

The second issue of studio pedagogy - a concentration on end results - was magnified by the use of a game engine. Design reviews of non game engine based studios typically focus on discussion of still images and orthogonal views, supplemented by animations often in the form of the seductive 'fly through'. The student presents these rarefied outcomes to a group of critics who observe the performance of the student. In most of the game based studios critics who were used to manipulating themselves within a multi player game environment, and the reviews involved them following the student through the design proposal. The critics were actively engaged with exploring the design project with the student, breaking down the often intimidating of critic to student. However designing a project that can reviewed in real time is far more revealing of design weakness - it is very demanding on students to resolve the whole project to the same level of detail, so inevitably critics will find substantial areas where the project is unresolved. This seemed to put students at a disadvantage compared to students in other studio groups utilising renders and animations. Our students could of course take selective screen grabs or pre record animations from their game files, but to my mind this negates opportunities for deliberation and reflection on negative outcomes as well as design strengths. The outcomes of a game engine design studio highlight the pedagogic requirement for final outcomes packaged for quick review, rather then considered deliberation and evaluation throughout the design process.

Game Play: An Agenda For Virtual Design Studios

Before embarking on future virtual design studios it is apparent that more work needs to be undertaken to address the problem of facilitating social interaction between students, and the problem of the design review. To this end I would like to introduce an agenda for virtual design studios which embraces the pedigree of the appropriated technology - design as a form of game play. There is some history to the use of play as a strategy to teach design in architecture, predominantly based on a shape grammar approach (Radford, 2000, Woodbury et al, 2001) in which students are given a vocabulary of architectural form, compositional rules, and design objectives. Qualitative evaluation by researchers at the University of Adelaide report three positive outcomes from the application of this approach.

“Play seems to capture essence of designing - it is intrinsically engaging, both bounded and free, and open ended, Second, play can breed confidence and build new skill.....Third...form making can be learnt and refined through self-directed, structured play with form making games” (Woodbury et al, 2001)

These architectural precedents indicate there is pedagogic value in such game play, although I would suggest that shape grammars are best suited to design in the early years, where students are not experienced at a range of design vocabularies and approaches. As students become more proficient and explorative, a tutor defined shape grammar may seem too prescriptive in terms of design possibilities. An alternative to this approach more suitable to latter design years, would be for students to contribute their own formal vocabularies as parametric objects.

There are numerous experiments with game play based pedagogy in other disciplines, but few have engaged with the sophisticated interaction available within current 3D video games. This range of video game genres are a potentially rich source of scenarios for virtual design studios. While first person shooters may not be appropriate, adventure and role playing games based on way finding, or strategy games which focus on careful planning and skillful resource management in order to achieve victory, may be appropriate genres on which to base a scenario for a design studio. Studio pedagogy could also address the essential characteristic that defines current video games and makes them so popular: the high degree of interactivity and the pleasure of competition. A design studio could be planned around design and play modes in which design elements are integral to subsequent game play, enabling player / designers to compete within the evolving virtual environment.

Over and above the potential of the video game genres as scenarios for design studios, the successful distance based design collaborations within the independent game community may form another useful precedent. Ad hoc design teams creating game ‘mods’, or complete games, by design members spread across the globe are prolific. The home of the torque engine, Garage Games, is the leading independent game site and includes numerous examples of teams forming and communicating across the internet. The basic premise of such teams is that the production of a prototype game requires multiple team members with a range of skills. The motivation for the individual being the possibility of being part of a project, which may eventually make the market. A similar motivation drives level ‘moders’ who jostle for the prestige of having their scenes picked up for such popular games as half life and doom. Design studios with the aim of producing a public outcome, and based on collaboration with students with complementary skills such as narrative, programming and sound design, could be a possibility. In a similar approach collaboration with departments of history and archaeology could result in meaningful outcomes similar to the virtual heritage project undertaken at the University of Melbourne (Champion, 2003).

Conclusion and Further Work

The use of game engine technology will continue to provide state of the art computer graphics, sound, networking and interactivity at low cost. We have reported the adaptation of the Deep Explorer game engine which addresses two technical problems encountered in virtual design studio trials - content creation and communication lag. However we concur with other researchers that problems encountered with virtual design studios are more related to the fact that design is a social process. We suggest that studios which are based on game play offer the potential for student engagement, which may address the inherent problem of social interaction among design collaborators. There are problems related to the tradition of the design review as the mode of assessment for game engine based projects, and further work needs to be undertaken on architectural pedagogy which allows deliberation and reflection throughout the design process. We intend developing virtual design studio which investigate the potential of game play to enhance virtual design studios in terms of social engagement and deliberation. On the technical side we intend developing functionality for the Deep Explorer engine which will extend asynchronous communication functionality to mobile devices.

References

Bennett, R., Broadfoot, O: 2003, “Design Studios: Online?,” Apple University Consortium Academic and Developers Conference Proceedings 2003, Wollongong, pp 9-21

Champion, E: 2003, Applying game design theory to virtual heritage environments, 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Melbourne, pp 273-274.

Cheng, N. , T. Kvan: 2000, Design collaboration strategies, Design and Decision Support Systems in Architecture – Proceedings of the 5th International Conference, Ampt van Nijkerk, , pp 62-73.

Kvan, T: 2001, The Pedagogy Of Virtual Design Studios, Automation in Construction, 10(3), pp. 345-353.

Maher, M.L., Simoff, S.J. : 2000. Collaborative design within the design: A case study. International Conference of the DesignResearch Society. Coventry.

Mitchell, W: 1995, CAD as a Social Process, The Global Design Studio, CAAD Futures 1995,Singapore, p. 9

Moloney, J., Harvey, L: 2004 ‘Visualization and Auralization of Architectural Design in a Game Engine Based Collaborative Virtual Environment’ IEEE 8th International Conference on Information Visualisation, London, pp 827- 832.

Moloney, J: 2002 ‘StringCVE: Collaborative Virtual Environment software developed from a game engine’, Proceedings of the 20th Conference eCAADe , Warsaw. pp 522 - 525.

Woodbury, R. F., Shannon, S. J., & Sterk, T. d. E.: 2001. What works in a design game?, CAADRIA’2001, Sydney.