DESIGN CRITIQUE INSIDE A MULTI-PLAYER GAME ENGINE

Jules Moloney
School of Architecture, University of Auckland
j.moloney@auckland.ac.nz

Robert Amor
Department of Computer Science, University of Auckland
trebor@cs.auckland.ac.nz

Jay Furness and Benjamin Moores
School of Engineering, University of Auckland

SUMMARY

The design critique process, used to provide expert feedback on a building design concept to students in architecture schools throughout the world, is reinterpreted within the context of an IT-based collaborative virtual environment. IT support for design critique allows new modes of participation where the experts do not have to be co-located to carry out their tasks and, through the ability to record criticisms, can join, or leave, a critique session as their time pressures allow. Students can guide tours through their virtual designs, yet those making the critique still have the ability to explore other aspects of the design which intrigue or concern them. Commentary and annotation on the design is attached to various aspects and views within the model and dialogues between the student designer and those making the critique can be built up over time. The resulting discussions recorded against the design can then be used for future reference by the student or as instructive commentary for newer students reviewing previous design approaches. A game engine provides the real-time 3D visualisation, base interactivity and multi-participant support upon which critique specific functionality has been incorporated.

INTRODUCTION

The purpose of this paper is to describe the specification and implementation of techniques within a game environment to support the design critique process.

The design critique functionality reported here is part of a larger research program initiated by the School of Architecture at the University of Auckland (UoA) into the use of collaborative virtual environments (CVE) to support architectural design teaching. The school has been experimenting with multiplayer games software as a feature-rich but low cost alternative to high-end virtual reality software. While the applications themselves are not appropriate for education usage, the underlying engines are graphically sophisticated, offer advanced multi-user network capability and are designed to perform well on standard hardware and operating systems. Given these advantages and the comparative high cost of commercial virtual reality (VR) systems there has been some interest in the use of game engines for a variety of architectural applications (Shiratuddin and Thabet 2002; Lehtinen 2002). The UoA has used the 'Half-Life' game platform (Sierra 2002) to support design programs during 1999-2000. The advantages of CVE over CAD and animation software have been reported in a prior publication (Moloney 2001) and are summarised below:

1. Iterative working: Most students, once they realised that they could fluently alter designs and examine the consequences in context, produced a series of ideas in a highly iterative and exploratory manner. Research on creativity in architectural design has indicated a relationship between creative ability and design permutations, in that the more creative solutions generally come from students who are prepared to critically examine a large number of iterations (Schoon 1992). Hence our interest in software that encourages such experimentation.
2. Design critique: The review was conducted with four critics seated side by side with independent monitor and navigation controls while the presenting student controlled a data projector. Critics were invited to experience the architectural proposal in a participatory manner in the multi player project as opposed to passive viewing and listening which is the

norm for analogue or digital reviews. This enlivened the whole process, relaxing the student and critic, and encouraging conversations about aspects of the work to evolve. Projects could be more rigorously examined given the freedom to view projects from a variety of perspective views and at differing scales. This is a marked contrast to animations typically used in architectural design where a fixed camera path can often conceal design limitations.

By mid 2001 it was decided to investigate options for developing a more educationally orientated application based on available game engines. The key factors in the choice of a development platform besides excellent graphic and sound capability were support for large scale environments, robust multi-user capability on low band-width, and of course access to source code to allow new functionality to be developed. In June 2001 Garage Games released the Torque game engine (GarageGames 2002) that met all these criteria and a decision was made to develop an application using the Torque software development kit. Basic functionality was added to enable the use of an application we have named StringCVE. It was used by a UoA design studio in January 2002 and from this a further set of requirements were established to enable asynchronous modes of communication, allowing the UoA to run joint design classes with international institutions (so called virtual design studios (VDS)). Experiments with VDS have usually involved a mix of technology and representational media. We acknowledge that different types of communication are required for different tasks and design stages but believe by accommodating these within the virtual environment we will maximize the potential of the game technology. The emphasis is on “designing within the design” (Maher and Simoff 2000), that is, communication between design collaborators and with tutors occurs within the context of the emerging architectural form.

REQUIREMENTS FOR CVE-BASED DESIGN CRITIQUE

In order to support design critique within a CVE there are a range of functionalities which must be offered to the student designer alongside those required by those making the critique. These functionalities are:

- The ability to easily populate a CVE with a student's design. As a CVE tends not to be a design environment it is important that a student's design can be quickly and easily transformed from the format of the design tools they are familiar with into the specification required by the CVE. In order to encourage students to examine design permutations this task needs to be as automatic as possible.
- The ability to disseminate a design for critique. The developed design must be easily packaged and accessible by those who will undertake the critique and not so large that its transfer across the Internet is prohibitive for common equipment.
- The ability to establish a realistic environment for the design. The CVE must be able to simulate the locality, acoustic parameters, lighting parameters, and even weather conditions that are envisaged by the student designer.
- The ability to establish and run tours through the virtual design. It must be relatively easy for the student to define a tour through their design which can be replayed at any later date. Where the student is to guide a group in real time through the virtual design they should be visible as a guide and those straying from the tour should be able to rejoin the guide.
- The ability to record commentary on the design. The critiques and defence of the design should be able to be recorded within the CVE. Previous discussions should be accessible to those who may follow later and be able to be added to. Annotation of the design would be desirable in order to illustrate the point being criticised as well as to indicate solutions to design flaws.

OFFERINGS OF A MULTI-PLAYER GAME ENGINE

In examining the requirements for design critique in a CVE it seems that multi-player game engines offer a large number of the necessary base functionalities. In general they provide the following:

- A multi-player environment which supports concurrent interaction between many people in near real-time. Multi-player game engines can cope with dozens of players within the same environment. This is more than sufficient for design critique sessions.

- Realistic visualisation of the artefact that is navigated. Game engines differ in the realism of external environments and the interiors of buildings but the majority aim at high levels of realism on standard PC hardware.
- Real-time rendering on standard hardware is a goal of game engines in order to ensure that the developed products are usable on as many PCs as possible. Through many graphics tricks these engines render movement through the environment in such a way that it appears natural.
- An Internet-ready product is typical of a game engine as players interact with others anywhere in the world. To achieve this, the game engines ensure that the amount of information which needs to be coordinated across the Internet is minimised.
- A method of representing players and avatars is included in the game engine to ensure that other players and creatures are visible as the game progresses.
- Global update of all player activities is supported in game engines to ensure that all players are working in the same environment and anomalies in play are not encountered. Again, this is achieved by minimising the amount of information which must be transferred between all of the players within a single environment.
- The final product has a low cost to ensure that it is affordable by the potential market for games. This level of costing also means that the final product is also affordable within a university environment as is not usually the case with high end VR environments.

The game engine chosen for this project was Torque (GarageGames 2002) which has an added benefit of being almost free to develop upon (unlike many of the other game engines) as long as extensions are made available to the wider developer community. The low entry cost is not without drawbacks: the interior renderings are not as realistic as in many of the other game engines, there are some bugs in rendering translucent interior surfaces, and the program is lightly documented for the 300,000 lines of C++ code which are provided.

DEVELOPED DESIGN CRITIQUE FUNCTIONALITY

In this section we discuss the approaches taken to developing the required design critique functionality upon the base multi-player game engine. A full description of the development process can be found in Furness (2002) and Moores (2002). The first addition to a game engine in order that novel functionality is available is to extend the amount of data which is able to be recorded about players and the objects within their environment. Torque was extended by providing a generic database interface sitting alongside, but coordinated with, the game engine, in order to maintain design critique specific information on a design.

Design Tours

A ‘virtual tour’ would seem analogous to the way in which a student in a traditional critique takes the reviewers through the various elements of their design. A tour, directed by the student, would complement the free exploration otherwise promoted by the tool, and allow the student to showcase the various features of the design in a structured and cohesive way. There would seem to be two basic types of tour that should be offered: pre-recorded tours, which are asynchronous in that they are recorded by a student and ‘played back’ by a reviewer at a later date; and live tours, entirely synchronous, involving the interaction of a ‘tour party’ of participants in real time.

Live tours

Live tours are entirely interactive. Participants walk as a virtual tour party, visible to each other as avatars. A selected ‘guide’ leads the group to sites of interest. Participants converse via ‘chat’ (instant messaging), and are free to leave permanent annotations where desired. When wandering away from the tour group, a user may transport themselves back to re-join the other participants. This poses the challenge of determining a location close to, but not on top of, the other tour members while simultaneously avoiding terrain and other physical obstacles.

The main aspects of live tours are equivalent to normal game play within a game engine and hence require no further adaptation. However, the jump to user function is not common, and indeed is discouraged in many game engines in order to prevent players from cheating by escaping from deadly situations. Within the design critique tool a user is able to view a list of participants in a tour and select who they wish to re-appear beside (see Figure 1). With a selected tour guide this provides

an appropriate mechanism for getting back to the group. However, rejoining the group by appearing on top of (or inside) another player is messy, and determining a spot around a player which is free of other obstacles and terrain can be slow. To alleviate this problem, a notion of a 'breadcrumb trail' was introduced to track the previous positions of each person in the environment. Every time a user moved a particular distance in the environment their location is recorded in a short list of previous positions. Then, whenever a user wishes to jump to that participant, this list of 'breadcrumb' positions is searched to determine a good spot. We know that because the participant was previously at the spot that it can't be interfering with any of the building or terrain, so the system need only check that no other player is too close to that position. If someone is too close then another 'breadcrumb' position is found. In almost all scenarios this gets a user back to the group in a position very close to the nominated participant, and without game anomalies.

Figure 1. Selecting a design critique participant to join

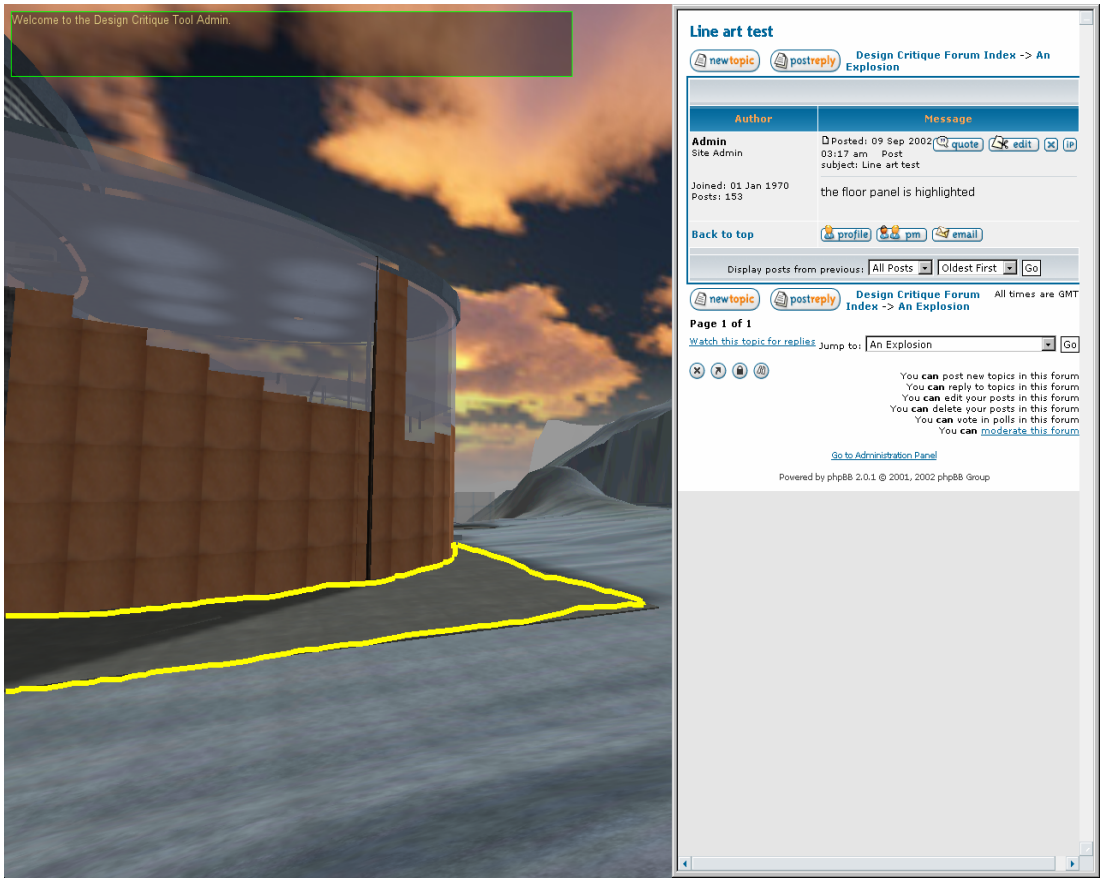

Figure 2. Forum-based discussion and mark-up on the design

Pre-recorded tours

Pre-recorded tours are analogous to animations or movies, as no interaction from the user is required. The author records a journey through the design, making comments on points of note, and stores the recording in a central location for others to view. Torque provides a journal feature which enables all

actions to be recorded and played back at any later stage (useful for game demos). As this function was not of major importance for our requirements no further investigation was made for this functionality.

Critique and Commentary

Critique and commentary represents the major additional functionality required to be added to a game engine. While most game engines offer an online chat functionality, so that players may converse, this is not a persistent recording and is not tracked against particular locations within the design. To support design critique it is necessary to have a record of discussions attached to the aspect of the design from which they were initiated. There are three aspects to this which are discussed in further depth below. One aspect is the ability to record discussions, another is to denote the location of commentary within the design, and the third is how to handle non-textual mark-up from a critique.

Discussion Forum

To record critique of a design and allow response and discussion to a critique requires functionality analogous to that offered by standard discussion forums. Whether this discussion appears in the CVE or as a separate window then drives the technologies which may be employed. We believe that a separate window, which can be hidden when not required, is the correct approach for design critique. Therefore, a standard threaded discussion forum was linked to the Torque tool to record discussions (phpBB 2002). This tool provides for many discussion threads to be maintained independently with all contributors uniquely identified (see right-hand window in Figure 2). A critique or comment is used to start a new thread in the forum and anyone may respond to the thread. This provides a record of the discussion, but doesn't link it to the CVE. The critique and commentary can be perused independent of navigating through the design, however, if the design is open then it is useful to be able to navigate to the position from which the discussion was initiated. The forum was extended to interact with Torque so that when a user selected a discussion thread the user was transported to the appropriate position within the design.

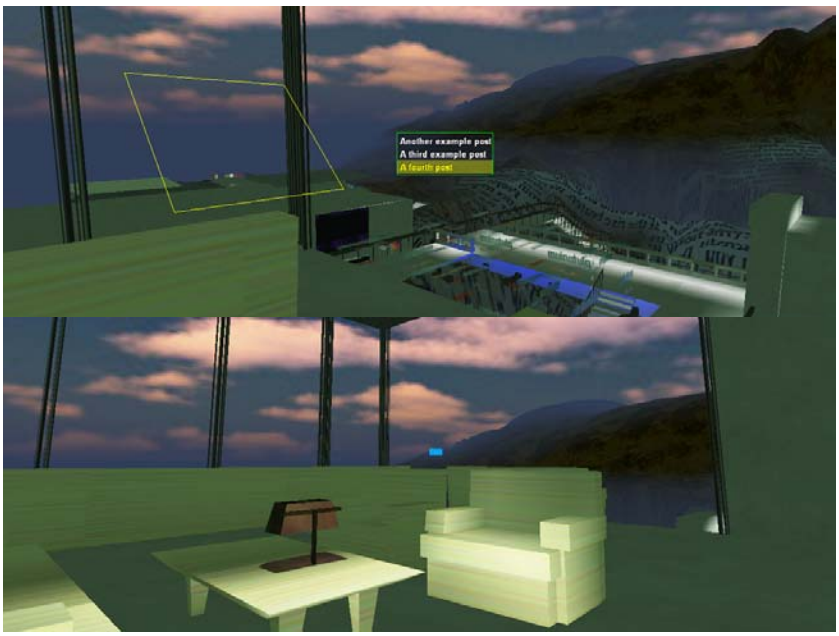

Figure 3. Comment locations as a frame (upper) and as a symbol (lower)

Location of Comments

When someone is navigating through the design it is necessary to indicate where comments and critiques were made, so the user can navigate to that position. Two approaches to indicating the location of designs were incorporated into the system. In one a frame is drawn which shows the angle and position at which a user was standing when the critique was made. The user can navigate to this position and access the forum from that spot (see Figure 3). In the other a small symbol is overlaid in the design space (the brick in Figure 3, lower snapshot) that the user can click upon and be transported to the correct location and facing for the critique in the forum. When a user clicks on a location that contains a discussion thread then the appropriate thread is brought up in the forum

window. In the case that many comments are made in about the same location, which could look confusing for users navigating the design, then a single symbol or frame is drawn and when the user clicks on this location then the various individual threads are listed for the user to select from (see top snapshot in Figure 3).

Mark-up in the Design

While the above extensions allow textual comments there is no ability to indicate traditional mark-up on the design. In order to support this form on annotation users are able to draw onto the screen from the point at which they are making a comment (see mark-up in Figure 2). This set of vector art is stored with the comment and when a user selects a comment to view then the associated mark-up is re-drawn onto the screen at the same time.

Design Lighting Conditions

A game engine uses the placement of a nominal sun, and designer specified lighting objects, to pre-calculate (render) the brightness of objects and the shadows cast by them. The lighting and shadow casting algorithms are not as accurate as those provided by rendering packages, but they allow the lighting to be recalculated quickly (typically 5 to 20 seconds) which is of more importance in the real time environment. A game engine also has no concept of latitude or longitude, nor the direction of north. However, for a design critique it is vital that the natural lighting conditions closely resemble those that would be expected at the site of the design.

In order to provide this functionality a lighting control has been developed on top of Torque which allows site specific conditions to be manipulated. With this control the user can specify the location of north and hence highlight the impact of the major orientation of the building. Two methods of determining the sun's position are supported. In one the user can select a location along with date and time to allow the sun's sky position to be calculated (see Figure 4). In the other the elevation and compass position of the sun can be set manually. The second approach will allow impossible sun positions, but it does allow a sun position to be specified for locations not available from the database.

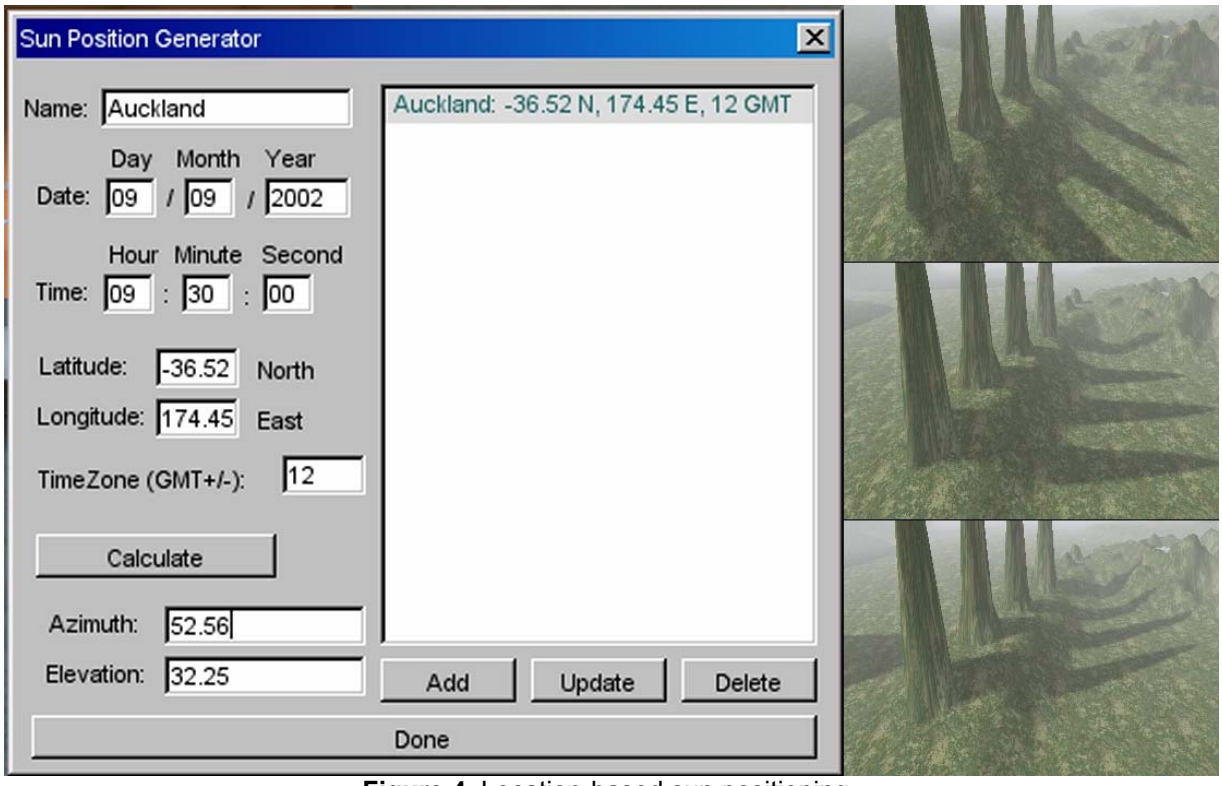

Figure 4. Location-based sun positioning

Figure 5 shows the interface for positioning the sun, two panels are shown the left-hand one shows north, the sun's azimuth, and the player's facing. The right-hand panel shows the elevation of the sun. This allows easy positioning of the sun and compass relative to the users vantage point.

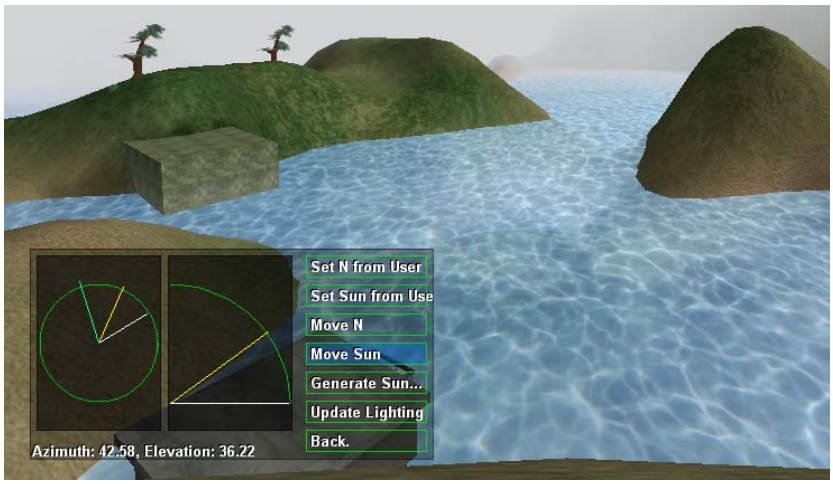

Figure 5. Azimuth and elevation-based sun positioning

Design Distribution

There are two main aspects to design distribution. One is the provision of a known repository of available designs for critique to which designs can be posted by students. The second is for a system which can bundle all the required components of a design into one archive for storage on such a repository.

Design Repository

When starting the design critique system a user must select a design to load (see Figure 6). While a design may be available on the local machine it is only designs joined up to on a game server which may be critiqued by several participants. The design critique system presents a list of known designs, both local and on known servers, which the user may join. Designs have titles, an author, and brief descriptions to provide information about their content. When a design is selected its geometric and rendering information is downloaded to the local machine and, assuming the user joined the design on a server, the user joins all others online taking part in the design critique.

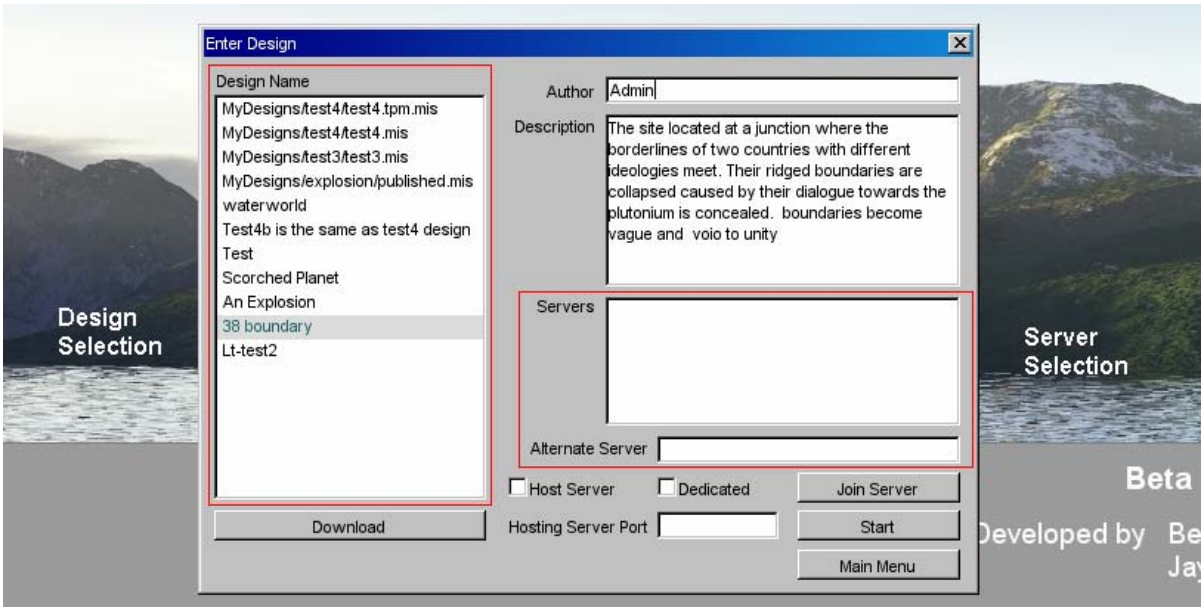

Figure 6. Accessing an available design

Bundling Designs

When a design is translated from its original design environment (e.g., a CAD system) into the format required by a game engine there are many files created. These files represent the geometry of the design, terrain, texture maps for objects, sound files, etc. In order that a design is available for others

to critique all these files must be able to be transferred to their local machine. In order to simplify this process a tool was developed which determines all the files which comprise a single design and places them into a ZIP archive file. This file is then able to be transferred to a design repository and downloaded by users who wish to critique the particular design as described in the previous section.

FUTURE WORK AND CONCLUSIONS

StringCVE with the additional design critique functionality reported here will be used to facilitate a design studio involving students and staff from the UoA, the Architectural Association School of Architecture London, and the Technical University of Graz, Austria in January 2003. This will no doubt help to further refine the approach.

Feedback from beta testers has already confirmed the asynchronous design critique functionality is a valuable addition to the StringCVE research program. Further work in the short term will concentrate on real time updating of new project data to avoid the time consuming upload and download of the full data set. In the longer term it is planned to investigate the incorporation of parametric approaches to the editing of building, infrastructure, landscape and environmental effects.

REFERENCES

Furness, J. (2002) Design Critique Inside a Multi-Player Game Engine, Part 4 Project Report, Department of Electrical and Electronic Engineering, University of Auckland, New Zealand.

GarageGames (2002) GarageGames and Torque Game Engine, http://www.garagegames.com/, last accessed 13/12/2002.

Lehtinen, S. (2002) Visualization and Teaching with state-of-the-art 3D Game Technologies, Proceedings of the 20th Conference on Education in Computer Aided Design in Europe, Warsaw, Poland, 18-20 September, pp. 538-541.

Maher, M.L. and Simoff, S. (2000) Collaboratively Designing Within the Design, Proceedings of Co-Designing 2000, Coventry, UK, 11-13 September, pp. 391-399.

Moloney, J. (2001) 3D Game Software and Architectural Education, Proceedings of the 18th Conference of the Australasian Society for Computers in Learning in Tertiary Education, Melbourne, Australia, 10-12 December, pp. 121-124.

Moores, B.T. (2002) Design Critique Inside a Multiplayer Game Engine, Part 4 Project Report, Department of Electrical and Electronic Engineering, University of Auckland, New Zealand.

phpBB (2002) phpBB: Creating Communities, http://www.phpbb.com/, last accessed 13/12/2002.

Schoon, I. (1992) Creative Achievement in Architecture: A Psychological Study. Leiden: DSWO Press.

Shiratuddin, M.F. and Thabet, W. (2002) Virtual Office Walkthrough Using a 3D Game Engine, International Journal of Design Computing, 4, http://www.arch.usyd.edu.au/kcdc/journal/vol4/.

Sierra (2002) The Official Half-Life Web Site, Sierra, http://www.sierra.com/games/half-life/, last accessed 13/12/2002.