String CVE

Collaborative Virtual Environment software developed from a game engine.

MOLONEY Jules
School of Architecture at the University of Auckland.

The University of Auckland has been experimenting with the adaptation of computer game engines for architectural education during a series of projects undertaken in specialized summer design studios 99 - 02. This paper reports on the early stages of software development of a collaborative virtual environment utilizing the ‘Torque’ game engine. The current software version (Beta 1.27) will be demonstrated and the studio outcomes discussed. The rationale for further application development is outlined with a focus on communication tools to enable experimentation with international virtual design studios.

Introduction:

Software for collaborative virtual environments (CVE) has often being aligned with military institutions or as part of long term projects within specialist computer science departments such as the DIVE project (Pettifer, S., et al, 2000). Concurrent with such institutional development there has been a proliferation of screen based virtual environment software that has been developed for the P.C. game market. 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 comparatively high cost of commercial systems there has been some interest in the use of game engines for a variety of architectural applications. (Shiratuddin, M.F. and Thabet, W., 2002). The source code for these engines is usually licensed at high cost, but recently access to the multi player product ‘Tribes 2’ has been acquired by a small independent company ‘Garage Games’, who are marketing the code as ‘Torque’. The University of Auckland has agreement with Garage Games to develop the Torque software for educational purposes. Our first application, ‘String CVE’, is being developed to support virtual design studios.

Current stage of software development.

The Torque code was received with the functionality of the tribes 2 game minus the A.I. classes. Please refer to the Garage Games web site for the feature list (www.garagegames.com) We have started development in C++ and torque script with the following functionality completed:

- Project packaging and new data file structure. A ‘publish’ application has been coded (gaxxer.exe) to package all necessary resources into a single zip file that is then uploaded to a user specified IP address.
- WWW management interface has been developed. This identifies missing files on the client and automates download to enable project connection.
- Comments recording function.
- Alpha channel support (transparency only).
- Sun location specification vector input changed to azimuth and zenith angles.
- 3D spatialised sound interface.
Studio outcomes:

A six week design studio was undertaken earlier this year (southern hemisphere summer). Twenty architecture and 6 engineering undergraduates with programming experience participated, with the author and a teaching assistant providing tutorial support. The brief for the studio was the Plutonium Memorial international competition sponsored by the Bulletin of Atomic scientists. (http://www.the-bulletin.org/contest/). The intention was to have international participation via the internet but unfortunately the software was not developed in time for distribution and testing. Online tutorial and design reviews were undertaken over the University LAN, but the primary interaction between students and tutors occurred face-to-face. As with previous studios we observed two main advantages in using game and associated level editing software as compared to CAAD applications typically used for education (Moloney, J. 2001).

1. Design Iterations. Modeling, texture application, and lighting was undertaken with ‘Worldcraft’, a shareware CSG modeler initially developed for the editing of games from Sierra Studios. A major advantage of Worldcraft is the close match, in terms of navigation and graphic quality, with the CVE. Thus new design ideas can be quickly evaluated, edited and refined with confidence right up to design deadlines. Worldcraft files are exported to the dif file format used by the torque engine to enable direct import directly into the CVE where additional editing (position, rotation and scale) can occur. The landscape can also be edited in real time with sophisticated fractal, mesh edit, and texture map tools. Most students, once they realised 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 – 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. The Worldcraft interface and examples of editing within the CVE are illustrated in Fig 1.

2. The design review. 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 – 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. The process was one of mutual discovery, breaking down the

Figure 1. Worldcraft modeling interface, geometry and texture map editing within the CVE.
normal power structure of architectural critiques – students standing in front of seated critics and endeavouring to ‘sell’ their project. This was of particular advantage to some ESL students. As an aside it was noticeable that the less ‘CAAD fluent’ reviewers appeared to require standard orthogonal views to engage with the work. This was to some degree overcome by panning the camera to simulate ‘mobile sections’ or ‘dive bombing’ through levels to indicate plan relationships.

Other outcomes
The primary differentiation between this and previous studios where we have utilized game engines is the support for large scale context. ‘Half-Life’ for example, constrained projects within a claustrophobic environment cube in which the physical backdrop is reduced to ‘wallpaper’ textures – an overwhelming emphasis on interiority pervades this game space. In contrast String CVE supports a 1.8 sq.km landscape that tiles indefinitely and includes editing tools that allow detailed geometry and texture manipulation of landform and textures. In combination with animated sky, water maps and fog effects this has allowed the evaluation of landscape to building relationships and discovery of interstitial spaces within projects. In most cases this lead to successful articulation of architecture in relation to context – an aspect of architectural design that in our experience is difficult for students to grasp and evaluate in terms of spatial sequence and multiple perspective. Fig 2 contains snapshots of completed projects.

One other aspect of note was the studio methodology of obliging student designers and programmers to interact. The application development and design speculation occurred side by side in a large, airy summer studio when students had no distractions from the task at hand. As new software functionality was developed this could be instantly evaluated by design students who subsequently would suggest improvements (in some cases design students started to edit at scripting level) and the programmers could directly experience the end user feedback. The shared workspace and the intensive aspect of the summer school resulted in high productivity compared to our subsequent supervision of programmers that is less focused without the momentum generated by collegial co-operation with designers.

Further software development:
The objective of the next phase of development is on improving communication tools to enable experimentation with international virtual design studios. 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 (Cheng, N.Y. and Kvan, T. 2000) but believe by accommodating these within the virtual environment we will maximize the potential of the game technology. Our emphasis is on “designing within
the design” (Mayher, M. & Simmoff S., 1999), that is, communication between design collaborators and with tutors occurs within the context and emerging architectural form. This may necessarily be abstract and gestural during the early phases of design. The physical analogy is working on site with spray cans and heavy machinery while colleagues and clients look down from above or drop in to experience the actual scale. In relation to synchronous communication we will add voice functionality in the near future. Craig and Zimring (2002) discuss the importance of unstructured verbal communication in CVE but note that it is “more a question of how to stimulate verbal interaction, rather than how to specialize it.” To this end the laser pointer function will be extended to provide marks on 3D surfaces; avatars will include a range of body gestures; text input will be extended to associate text blocks with avatars; and multiple camera and diagrammatic plan views will be available on screen. While synchronous communication is valuable, especially for tutorials and design reviews, recent reports stress asynchronous modes for design reasoning (Craig D.L. and Zimring, C., 2002). At present we have a basic method of recording comments and images that can be accessed via an external WWW forum. Current work on a relational database server accessed from within the CVE will supercede the WWW forum. Text comments will be browsed either by ‘walking’ near the location to which they relate, or by selecting from a complete list. Comments may be threaded by commenting on a comment and, if this relates to a recorded camera position, selection of the text will transport the user to the relevant location. In addition to threaded text forums we propose to allow participants to create and share design walkthroughs or ‘tours’ by recording user movement and interaction such as laser point highlighting or manipulation of the sun position. These will be able to be re-enacted by subsequent users and again threaded comments on the work attached to these tours.

Besides this technical development two critical issues figure uppermost at this stage – notation and play. The problem of appropriate notation when designing with (in) virtual environments has surfaced often when tutoring design students. Is this merely the placing of traditional orthographic drawings and the ubiquitous ‘concept sketch’ within the CVE or should alternate approaches be developed to include, for example, references to movement (mobile subject and deforming object), materiality, and acoustics. Can precedent from temporal art forms such as music, dance, film, be engaged with the sophisticated new genre of game space? A second issue we will address is the incorporation of play itself. How may we enliven learning and stimulate creativity by using appropriate gaming in the context of design education? If we accept the work of Malcom McCullogh (2000) on the link between play, craft, and creativity much research is needed in this area to define and test design relevant game play.

References