TEACHING DESIGN IN A VIRTUAL ENVIRONMENT WITH GAME SOFTWARE

Jules Moloney, Sam Cuttriss
University of Auckland
j.moloney@auckland.ac.nz, teanau@yahoo.com

SUMMARY

An innovative architectural design studio at the University of Auckland in which virtual reality software developed for commercial computer games was utilised as the design medium is described. The studio was proposed as a consequence of problems that students were experiencing with standard modelling and animation software – steep learning curve and lack of immediacy. The preliminary outcomes indicate that this relatively inexpensive software may address the above problems and in addition offer advantages in relation to the critique process and design emphasis. Further work is outlined for a follow up design studio in which it is intended to use a new game engine to develop additional functionality to facilitate advanced virtual studio capabilities.

INTRODUCTION

At the Auckland School of architecture there is a recognition in most design studios of the close relationship between representation and the development of design ideas - the “mark interpret mark cycle” articulated by Daniel Herbert (Herbert, 1993). While not a certainty, our experience has shown that innovative representation generally produces innovative and sophisticated architectural projects. These observations are reinforced by the work of theorist Robin Evans who in “The Projective Cast” traces the relationship between projective geometry and the generation of architectural form. (Evans, 1997) In this work he proposes that the historical development of architecture has been limited by the ability to describe form on paper, and hence related to the drafting tools and techniques available at any given period. Previous publications have extended Evan’s thesis in terms of the impact of digital media (Moloney 1998, 1999) to provide a framework for further research. This framework identifies three interrelated stages of digital craft: emergent form (using generative techniques such as cellular automata, shape grammar or genetic algorithms); immersive editing (the editing of architecture within virtual environments); computer aided construction (the automated construction of architecture using CNC machines). These three strategies are to our minds the advantages of computing for design. While the first involves computer programming beyond the curriculum of most schools and the later requires specialised and expensive machines, the facility to evaluate and alter architectural design from a point of view of occupation - immersive editing- is feasible with current 3D modelling software.

As staff responsible for computing our goal has been to encourage the use of digital media to aid student design projects. The first step was to move the computers out of the computer lab and into the design studio. A relatively simple and logical move once nervous technicians and academic staff were persuaded of the need to integrate computers with other forms of representation in the design studios. However while computers are now available there is some reluctance by students to commit to digital immersion as a means to develop projects. Our mainstream 3d modelling and animation software is studioMax which offers limited real time manipulation dependent on graphic card specification. Generally this 3D window manipulation is inadequate and designs have to be rendered and animated to test design decisions. This appears to have limited the use computers as media for conceptual design in the design studio. Students, encouraged by the success of some of their digitally fluent peers, have endeavoured to design ‘on screen’ but have ended up frustrated and disillusioned by the process. In discussion with some of these students it appears the most traumatic experience was the long anxious wait for hardware to produce renders only to find either design mistakes or technical errors such as lighting have ruined their work. 3D studio Max and similar software such as Soft Image require a reasonably steep learning curve in order to anticipate such problems. This learning curve and the time lag between computer ‘mark’ (render) and interpretation hindered the creative process and in the worst cases resulted in sub standard work. The obvious answer is to work in real time - so called virtual reality- rather than evaluate designs via rendered animations.
Indeed the concept of immersive editing originally proposed by the author was based on the assumption of a real time editing environment. Unfortunately virtual reality software and the hardware requirements are still beyond the resources of most undergraduate programmes.

GAMES

Computer game developers have huge resources to bring to their research and development and as a consequence their products offer performance that rivals commercial virtual reality software - at a fraction of the price. Recently most games have offered 'level' editing- users can design their own scenes for the game play to unfold. It was now possible to design architecture using game software. The range of 3D games, their features and modelling environments that were available in November 1999 are documented in figure 1.

<table>
<thead>
<tr>
<th>3D games available 1999</th>
<th>Published</th>
<th>EAX 3d sound</th>
<th>Multi player</th>
<th>Visual Quality</th>
<th>Modelling environment notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quake 2</td>
<td>ID software <a href="http://www.idsoftware.com">www.idsoftware.com</a></td>
<td>y</td>
<td>y</td>
<td>4</td>
<td>Well established, but a little 'foreign' compared to regular cad packages.</td>
</tr>
<tr>
<td>Quake 3</td>
<td>ID software <a href="http://www.idsoftware.com">www.idsoftware.com</a></td>
<td>y</td>
<td>y</td>
<td>8</td>
<td>Sophisticated visual manipulation available, limited interactive editing capacity. Not as well established.</td>
</tr>
<tr>
<td>Halflife</td>
<td>Sierra <a href="http://www.sierra.com">www.sierra.com</a></td>
<td>y</td>
<td>y</td>
<td>6</td>
<td>Very well established community, extensive interactive editing capacity.</td>
</tr>
<tr>
<td>Tribes 1</td>
<td>Dynamics <a href="http://www.dynamics.com">www.dynamics.com</a></td>
<td>n</td>
<td>y</td>
<td>3</td>
<td>Ability to construct expansive environments.</td>
</tr>
<tr>
<td>unreal</td>
<td>Epic games <a href="http://www.epicgames.com">www.epicgames.com</a></td>
<td>y</td>
<td>y</td>
<td>7</td>
<td>Moderately well established community, but a little 'foreign' compared to regular cad packages.</td>
</tr>
<tr>
<td>Unreal Tournament</td>
<td>Epic games <a href="http://www.epicgames.com">www.epicgames.com</a></td>
<td>y</td>
<td>y</td>
<td>9</td>
<td>Same as above, improved visual quality.</td>
</tr>
</tbody>
</table>

Figure 1. Games, features and modelling environment.

After evaluating options we decided to use Worldcraft Toolkit, a 3D modelling and interactivity platform that is used to generate 3D scenes for the games 'Half Life' and 'Counter Strike'. The major advantage of Worldcraft is that the modelling navigation is identical to the game navigation interface. Hence new design ideas can be instantly evaluated edited and refined with fluency. In addition there is a well established on line community to provide technical reference and support that is inevitably required for shareware applications. While most of the modelling, texture mapping and interactivity occurs in Worldcraft a range of additional programs and utilities are used to produce the final immersive environment. The game development workflow is illustrated in figure 2.
It was anticipated that the use of Worldcraft would encourage experimentation and design modification throughout the design process. In order to test this proposition a design paper was proposed for a 2001 summer design elective. Summer schools at The University of Auckland have been utilised for 3 years to support computer intensive design papers. The normal 12 week time frame is compressed to 6 weeks with 24 hour access to an individual workstation being available. 18 students took part in the 2001 summer design paper entitled ‘Cousins: Genetic Code from the Avant Garde’.

The design paper was structured in two parts. In part 1 students were required to select an unbuilt work from a list of master architects and construct an accurate 3D model. Drawings of a un realised projects by Franco Albini, Guiseppe Teragni, Adolph Loos, Carlo Scarpa and Mies van de Rohe were scanned and bought into 3D Studio Max as backdrops. Once correct scale and accurate proportions were established the mgs translation utility was used to import these into Worldcraft where refined modelling and texture application was undertaken. The primary objective of this short project was to gain software fluency without the pressure of design. Another objective was to gain an insight into the work of their chosen architect and hence provide a precedent or referent for the next project. Figures 3, 4 and 5 illustrate some of the projects that resulted from this 2 week section.
Figure 4. Project for the Palazzo dei Congressi at E’42, Rome 1938, Franco Albini et Al.

Figure 4. Project for the Danteum, Rome 1938, Giuseppe Terragni and Pietro Lingeri.

Figure 5. Project for Villa Zoppas, Treviso 1953, Carlo Scarpa.
In part 2 students were required to produce an architectural proposal which referred to the previous model. The analogy of ‘cousins’ was presented for individual interpretation. Students were encouraged to exploit the game software to push the boundaries of architecture - the project did not necessarily have to be a realizable building as long as ideas and possibilities for architecture were explored.

CONCLUSIONS

This first experimentation indicates there are potential advantages in the use of game software in undergraduate design papers. In particular the **immediacy** of the interface and the low hardware requirements address problems encountered with software typically used in architectural schools. Once students realised they could alter their projects and instantly examine the 3D consequences from any point of view most produced a series of ideas that were discussed in tutorials. Students were encouraged to make file copies as they advanced these designs to provide a record of these iterations. The range and scope of iterations differed from student to student in a similar manner to non-computer based design projects. Research on creativity in architectural design has indicated a relationship between creative ability and the range and number of design permutations - the more creative solutions generally come from students who are prepared to critically examine a large number of iterations. (Schoon, 1992) Hence the importance of using software which encourages immersive editing.

As well as these anticipated benefits the use of the software resulted in other outcomes of potential importance to the use of computers for architectural design education.

The Critique

Student projects culminate in a formal critique by invited critics who have not seen the work. In previous projects we have undertaken students would present with a combination of hardcopy prints and screen animations. The experience of the game critique was radically different. As ‘Half life’ is a multi player game environment the critics manipulated avatars to follow the student around and engage **within** the architecture. Hence the critics were invited to experience the architectural proposal in a participatory manner as opposed to passive viewing and listening. This enlivened the whole process, relaxing the student and critic and encouraged conversations about aspects of the work to evolve. The process was one of mutual discovery, breaking down the 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 ESL students.

Design emphasis:

The tools available within the gamming environment affected the emphasis of the student designs. These included experimentation with surface qualities to produce in the best cases extremely seductive light and opacity effects (figure 7) - the visiting professional critic left sufficiently inspired to seek out real world materials that would achieve these effects. Other students experimented with differential gravitational fields that challenged the position of the ‘occupant’ (figure 6). Some inserted hidden triggers to animate walls or ‘teleport’ the occupant to other parts of the building (figure 8), while others engendered a sense of danger via traps or breaking building parts. In one instance the student declared to the critics that her work could only be appreciated by continually running through the project. The general outcome of student experiments facilitated by the game software was to emphasise the **experiential** attributes of the architectural proposals.
Figure 6. Winston Joyce - critical response to Alnini’s Palazzo Dei Congressi

Figure 7. Tristam Collett and Phillip Kwong - critical response to Terragni’s Danteum

Figure 8. Haniel Hong - critical response to Scarpa’s Villa Zoppas
FURTHER WORK: GAMES 02

We are preparing a subsequent design studio for the summer of 2002 in which we will be utilising an open source game engine Torque. This engine is based on the Tribes 2 game developed by Dynamics and published by Sierra. Examples of this game environment are shown in figure 9. The major advantage over Halflife is that with Torque we have the opportunity to author our own games. The primary objective will be to add functionality to enable extended virtual studio capabilities. While there have been many virtual studios in recent years the majority rely on video conferencing and non-immerse media (Dave, B. et al, 1998; Morozumi, M. et al, 1997; Rügemer, J. et al, 2000; Wojtowicz, J. et al, 1998). This will be, to the author’s best knowledge, the first studio utilising a high resolution game engine to facilitate real-time interaction and critique from within a student project. Engineering students with C++ programming skills will be collaborating with our design students in order to attempt this project. If time permits we anticipate students will attempt to develop particular ‘game play’ to suit their projects but as an initial task we will be considering the following virtual studio functionality.

**Multiple User Interaction:**
Face expressions and body language for avatar.
Point and highlight architectural features.
Post messages / comments as text or 3D graphic notation.

**Navigation:**
Multiple screens - plan view / track occupant location
Multiple modes - walk / fly / teleport to co-ordinates.
Critique mode enabling autonomous ‘follow the presenter’ attractors.

**Online Editing:**
Basic manipulation - rotate / scale/ substitute alternatives.
Zoom (picture within picture)

**Design History:**
Track user interaction and replay.
Record and provide access to previous designs.
REFERENCES:


