

Summer 2001

# Exploring the potential usefulness of binary space partitions in architectural representations

Michael Hoon

*New Jersey Institute of Technology*

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Architecture Commons](#)

---

## Recommended Citation

Hoon, Michael, "Exploring the potential usefulness of binary space partitions in architectural representations" (2001). *Theses*. 751.  
<https://digitalcommons.njit.edu/theses/751>

This Thesis is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact [digitalcommons@njit.edu](mailto:digitalcommons@njit.edu).

## Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **EXPLORING THE POTENTIAL USEFULNESS OF BINARY SPACE PARTITIONS IN ARCHITECTURAL REPRESENTATIONS**

**by  
Michael Hoon**

There have been recent advances developed within the computer gaming industry that have made real-time first-person perspective spatial experiences feasible on the personal computer. Principally through the use of binary space partition tree structures, developers of three-dimensional gaming environments are able to convey to computer users a convincing sense of movement through space. The technology behind these advances may be termed as a particularization of Virtual Reality.

This paper will outline research intended to determine the possible usefulness of binary space partitions in the fields of architectural education and practice. The feasibility of this technology was studied by directly observing original experimentation in practical application, which was conducted primarily in the Imaging Laboratory at the New Jersey School of Architecture. In addition, this paper references existing theories and experience-based expositions on the application of computer technology to architectural design and representation, with particular regard to the use of generalized virtual reality.

**EXPLORING THE POTENTIAL USEFULNESS  
OF BINARY SPACE PARTITIONS  
IN ARCHITECTURAL REPRESENTATIONS**

by  
**Michael Hoon**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Masters of Science in Architectural Studies**

**Department of Architecture**

**August 2001**

Blank Page

**APPROVAL PAGE**

**EXPLORING THE POTENTIAL USEFULNESS  
OF BINARY SPACE PARTITIONS  
IN ARCHITECTURAL REPRESENTATIONS**

**Michael Hoon**

---

Professor Glenn Goldman, Thesis Advisor  
Professor of Architecture and Director of Imaging Laboratory, NJIT

---

Professor M. Stephen Zdepski, Committee Member  
Associate Professor of Architecture and Director of Imaging Laboratory, NJIT

---

Professor David Elwell, Committee Member  
Associate Professor of Architecture, NJIT

## **BIOGRAPHICAL SKETCH**

**Author:** Michael Hoon  
**Degree:** Masters of Science  
**Date:** August 2001

### **Undergraduate and Graduate Education:**

- Masters of Science in Architectural Studies  
New Jersey Institute of Technology, Newark, NJ, 2001
- Bachelor of Architecture  
New Jersey Institute of Technology, Newark, NJ, 1990

**Major:** Architectural Studies

### **Presentations and Publications:**

Michael Hoon and Glenn Goldman  
“Digital Design in Architecture: First Light, Then Motion, Now Sound,”  
Proceedings of ACADIA Conference, St. Louis, Mo., pp 27-38, October 1994



To my wife

*Sharing knowledge makes the world a better place.*  
Michael Abrash, id Software

## ACKNOWLEDGMENTS

I would like to thank Committee members Professor M. Stephen Zdepski and Professor David Elwell for their input, their consideration, and their perseverance. I would also like to thank the students, staff, and faculty of the New Jersey Institute of Technology who embraced the value of this work and responded in kind with support and encouragement.

In addition, I thank Urs Gauchat, Dean of the New Jersey School of Architecture, for his respect and for his support.

I wish to express my gratitude to the founders of the Imaging Laboratory at the New Jersey School of Architecture. Their legacy is the foundation of this work.

Professor Glenn Goldman, who served as my thesis advisor, is also a friend, a colleague, and above all an inspiration. I extend to him my deepest appreciation.

## TABLE OF CONTENTS

| <b>Chapter</b> |  | <b>Page</b> |
|----------------|--|-------------|
| 1              | INTRODUCTION .....                                     | 1           |
|                | 1.1 Background .....                                   | 1           |
|                | 1.2 Objective .....                                    | 1           |
| 2              | REPRESENTING ARCHITECTURAL SPACE .....                 | 2           |
|                | 2.1 Abstraction in Architectural Representation .....  | 2           |
|                | 2.2 The Artistic Component .....                       | 4           |
|                | 2.3 The Personal Computer and the Design Process ..... | 5           |
|                | 2.4 Virtual Reality .....                              | 7           |
|                | 2.5 Hyper-Reality .....                                | 9           |
| 3              | FIRST PERSON PERSPECTIVE COMPUTER GAMES .....          | 11          |
|                | 3.1 Binary Space Partitions .....                      | 11          |
|                | 3.2 Game Engines .....                                 | 12          |
|                | 3.3 Binary Space Partition Games .....                 | 14          |
|                | 3.4 The Gaming Community .....                         | 14          |
| 4              | BSP GAME FEATURES OVERVIEW .....                       | 16          |
|                | 4.1 Advantages .....                                   | 16          |
|                | 4.1.1 Fully Interrogative .....                        | 16          |
|                | 4.1.2 Tactile Solidity .....                           | 17          |
|                | 4.1.3 Integration of Audio .....                       | 18          |
|                | 4.1.4 Multiple Simultaneous Users .....                | 19          |
|                | 4.1.5 Advantageous Use of Network / Internet .....     | 20          |
|                | 4.1.6 Inclusion of Characters .....                    | 21          |
|                | 4.1.7 Simulation of Movement .....                     | 22          |

**TABLE OF CONTENTS**  
**(Continued)**

| <b>Chapter</b>                                 | <b>Page</b> |
|--|-------------|
| 4.1.8 Simulation of Site .....                 | 22          |
| 4.1.9 Low Cost .....                           | 24          |
| 4.1.10 Established User Base .....             | 24          |
| 4.1.11 Interaction with Objects .....          | 25          |
| 4.2 Disadvantages .....                        | 25          |
| 4.2.1 Credibility .....                        | 25          |
| 4.2.2 Comprehensiveness .....                  | 27          |
| 4.2.3 Complexity .....                         | 28          |
| 4.2.4 Geometry Limitations .....               | 28          |
| 4.2.5 Delivery .....                           | 29          |
| 4.2.6 Lack of Standardization .....            | 30          |
| 4.3 Summary .....                              | 30          |
| 5 APPLICATION OF BINARY SPACE PARTITIONS       |             |
| IN ARCHITECTURAL REPRESENTATION .....          | 32          |
| 5.1 The <i>Doom II</i> Project .....           | 32          |
| 5.1.1 Software Setup .....                     | 33          |
| 5.1.2 Modeling .....                           | 33          |
| 5.1.3 Texture Mapping .....                    | 34          |
| 5.1.4 Character Manipulation .....             | 35          |
| 5.1.5 Compilation .....                        | 35          |
| 5.1.6 Issues .....                             | 36          |
| 5.1.7 <i>Doom II</i> Project Conclusions ..... | 36          |
| 5.2 <i>Quake I</i> Observations .....          | 37          |
| 5.3 The <i>Quake II</i> Project .....          | 37          |
| 5.3.1 Software Setup .....                     | 38          |
| 5.3.2 Modeling .....                           | 39          |

**TABLE OF CONTENTS**  
**(Continued)**

| <b>Chapter</b>                                   | <b>Page</b> |
|--|-------------|
| 5.3.3 Texture Mapping .....                      | 40          |
| 5.3.4 Lighting .....                             | 41          |
| 5.3.5 Sound .....                                | 41          |
| 5.3.6 Site .....                                 | 41          |
| 5.3.7 Compilation .....                          | 42          |
| 5.3.8 Client Reaction .....                      | 44          |
| 5.3.9 Issues .....                               | 45          |
| 5.3.10 <i>Quake II</i> Project Conclusions ..... | 45          |
| <br>   |             |
| 6 POSSIBILITIES FOR FUTURE RESEARCH .....        | 46          |
| <br>   |             |
| 7 RECOMMENDATIONS .....                          | 47          |
| <br>   |             |
| 8 CONCLUSIONS .....                              | 48          |
| <br>   |             |
| APPENDIX A GLOSSARY OF GAMING TERMS .....        | 49          |
| <br>   |             |
| APPENDIX B <i>DOOM II</i> PROJECT GALLERY .....  | 50          |
| <br>   |             |
| APPENDIX C <i>QUAKE II</i> PROJECT GALLERY ..... | 51          |
| <br>   |             |
| APPENDIX D BACKGROUND INFORMATION .....          | 52          |
| <br>   |             |
| REFERENCES .....                                 | 54          |

## LIST OF FIGURES

| <b>Figure</b>  | <b>Page</b> |
|--|-------------|
| 4.1 Construction and Assembly of Cube Panorama ..... | 23          |
| 5.1 <i>Vis</i> Viewpoints / Views .....              | 43          |
| B1 Near Weston Restrooms .....                       | 50          |
| B2 View of Graduate Lounge .....                     | 50          |
| B3 North Hallway .....                               | 50          |
| B4 Seminar Room .....                                | 50          |
| B5 Hallway Looking South .....                       | 50          |
| B6 Main Stairway .....                               | 50          |
| B7 Studio Alcove .....                               | 50          |
| B8 Print Room.....                                   | 50          |
| C1 Mezzanine from Above .....                        | 51          |
| C2 View from Master Bedroom .....                    | 51          |
| C3 View from Terrace .....                           | 51          |
| C4 View Through Kitchen .....                        | 51          |
| C5 View from Living Area .....                       | 51          |
| C6 Mezzanine from Below .....                        | 51          |
| C7 Entry into Dwelling .....                         | 51          |
| C8 Guest Bedroom .....                               | 51          |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Background**

Over the last several years, there has been a steady stream of development by the computer gaming industry of very powerful software using binary space partitions (BSPs) to simulate spatial experiences on the personal computer. Fueled by a demanding market, and supported by advances in graphics hardware, the very competitive computer gaming industry continues to create sophisticated consumer-level products that are spurring tangential interest in many segments of the non-gaming community. The continued advancement of BSP-based software products within the gaming industry, as well as a spate of peripheral BSP software utilities being coded by a cadre of freeware developers whose work is both nurtured and availed by the internet, are of particular interest to the architecture profession specifically for the direct if yet unrefined connection to the architect's realm— the communication of spatial concepts.

### **1.2 Objective**

This paper will explore the feasibility of incorporating BSP-based technology into the tools of architectural design and presentation, vis-à-vis both the intrapersonal and extrapersonal components of the communicative design process. The potential impact, both positive and negative, that this may have on the development and presentation of conceptual spatial ideas will be explored. Information gleaned from existing works related to the subject matter will be summarized, observations based on original experimental work will be presented, and recommendations will be offered.

## **CHAPTER 2**

### **REPRESENTING ARCHITECTURAL SPACE**

The issues surrounding the representation of architecture have historically been the subject of much discourse and debate. Matters of effectiveness, efficiency, economy, artistry, and aesthetics have been deliberated within social, economical, ethical, and physiological frameworks. The following is an introductory synopsis of the current state of architectural representation, with particular attention to the important transitional forces wrought by recent technological advances.

In specific regard to the introduction of computing technology into the milieu of spatial design and representation: it must be noted that this paper builds primarily on the flurry of activity, theory, and seminal research on the subject that was principally completed and presented during the years 1986-1991. It was during these years that the first formal responses to this technology from within the architectural community were formulated. This established an intellectual datum from which future work on this subject may reference.

#### **2.1 Abstraction In Architectural Representation**

Largely due to the influence of the Ecole des Beaux-Arts, the representation of spatial concepts by means of drawing evolved into a specific and formally recognized system— what is known today as traditional architectural graphics. To facilitate the expression and communication of complex three-dimensional ideas on paper, specific abstract conventions and symbolisms were developed and widely adopted. These include well known representational conventions such as orthographic projections (plans, sections, elevations) and symbols such as section lines, bench marks, and revision clouds. As such this specialized language must be learned by prospective architects, as well as by other building professionals with whom graphic communication is necessary. For the most part, this system has proven to be useful and efficient and for these reasons it endures to this day. It can be appreciated for its economy,



for the precision with which it can be used to quantify measurable information, and for its logical order [11].

The present system of architectural graphics is not, however, without its shortcomings. One is that it is a very esoteric graphic language that excludes those not privy to its translations. This group encompasses most clients and end-users, as well as members of the construction industry, who may fully comprehend the design of a building only after the building is completed. Perhaps more serious is the possibility that the designer, particularly the journeyman, will suffer miscommunications *intrapersonally*, lacking the skills needed to accurately translate internal concepts and intents into existing architectural drawing conventions. Spatial relationships, choice of materials, lighting, acoustics, etc., are design issues that may take many cycles of concept—to drawing—to built form before a designer gains the practical experience necessary to be proficient the understanding of the language of architectural graphics and its relationship to built form.

Precedent studies involving the drawn documentation of existing and accessible buildings are one tool used academically to spawn and nurture an individual's comprehension of the connection between two-dimensional drawings and physical buildings. At best such studies can only head-start a process that may take years of actual building experience to begin to master. Considering that communication via architectural graphics must also function *interpersonally*, among many different individuals spanning many different disciplines, it is not surprising that design intent can be often misconstrued beneath the many layers of abstraction necessary to document a building.

Another consideration is the intrinsic graphic quality of abstractions themselves. One need only look at a billboard or magazine advertisement to see the heavily laden messages implied by the chosen abstractions. The visual appearance of words and images may transcend literal translations; this is considered carefully as to elicit a specific response. The general quality expressed by the clean, pragmatic nature of most typical architectural plans, sections, and elevations could probably best be described as ordered but sterile. Granted,

orthographically projected drawings do not lend themselves easily to representing the spirit of a place. Whether drafting or rendering, designers have historically experimented with countless atypical means of expressing the nature of designed space by employing quasi-poetic techniques and devices, e.g., hand sketching, subjective use of color, perspective distortion, watercolor, charcoal, etc., as well as the overlaying of symbols of the ‘human factor,’ such as people, landscape, and content. While these practices can certainly lend specific aesthetic and emotive qualities to architectural drawings in their own right, the possibility of misrepresenting or obfuscating the deliverable end-product is quite real [24].

## 2.2 The Artistic Component

The philosophical nature of architectural design and the role of its practitioners merits periodic reevaluation as new techniques and technologies are introduced into the field. Presently, this is particularly applicable in regard to the relationship between *drawing* of buildings and *building* of buildings. While Architecture is a many-faceted profession, the perpetuated self-promotion as, in simplistic summary, ‘artists of space,’ seems to remain the projected core of the mission. Consider the changes to the profession since the quattrocento as the role of the so named ‘Master Builder’ spalled under the increasing pressures of external forces, i.e., systemization of structure, establishment of uniform civic code, contractual legality of construction documents, shifting social and economic climate, etc. [16]. To an increasing degree, technical acumen both in systems *and* processes is a prerequisite to practice, but few architects want this to overshadow the artistic qualities of the profession and risk being defined primarily as technicians.

Architects draw with the understanding that both the *drawing* and the *building* are both representations of the same idea. The notion that the skillful drawing of a building might translate to the skillful *delivery* of a building may encourage architects to reflect artistic sensibilities onto the graphic qualities of their drawings. This is akin to the idea that the skill of the craftsman may be judged by the order of his or her toolbox, or the skill of

the architect by the graphic nature of his or her portfolio. In regards to the use of artistic abstractions in drawing, this can be an effective ingredient in that if used skillfully can enhance the graphic quality of spatial representation and thus welcome public confidence. If a new tool appeared that allowed the complete dismissal of abstraction, depictions of reality would likely become solely a technical skill. Architects could then be emancipated from the fuzzy business of drawn depiction and instead concentrate on the logical business of designing space.

The above scenario is unlikely for two important reasons: First, design processes as now widely taught rely heavily on abstraction. The sketching process is intended to specifically avoid reality. We are conditioned to believe that premature decision making chokes evolutionary design development [29]. Second, and more formidably, tools that have historically offered improved depictions of reality (precise painting, mathematical perspective, photography, cinema) have never remained static even when their refinements reached an apex of objective reality. Rather, they have always tended to ‘progress’ into the more dynamic realm of subjective reality. This is the nature of art, and has been to a large degree the nature of architectural drawing. If architects desire to maintain the function and acceptability of an artistic component in their depictions they must preempt the *invasion* of new tools with the purposeful *adoption* of new tools.

### **2.3 The Personal Computer and the Design Process**

Typically, when a new technology is introduced to an existing process, it does not initially change the process; rather, the new technology will mimic traditional methods until developers and users become acclimated to it. During this period of incubation, the technology reveals its strengths and its weaknesses and may eventually modify the processes to which it is applied. The introduction of the personal computer into architecture followed a path not dissimilar to this. Initially it was seen as a replacement of traditional drafting. The functions of the T-square and the circle template were simply ported to a CPU. In fact, many early

digitizers featured a stylus, an electronic stand-in for a pencil. While not initially any faster than traditional drafting, electronic drafting software was soon lauded as a tool that could save the time and drudgery of modifying drawings, simplify data filing, and expand the compatibility of building information among associates. Reasons for embracing this new technology became ever more compelling.

As time progressed, the power of personal computers continued to advance and innovative programmers began to develop software capable of the high math of three-dimensional geometry. As sophisticated algorithms for shading, lighting, shadows, and material mapping trickled down from supercomputer labs to the desktop PC, the abstraction that had long been associated with the design and documentation of space began to transform, allowing designers an unprecedented accessibility to polished visual images.

Understanding the above notion of *accessibility* to images is key, however, as fully considered the end product of all these CPU cycles and ensured mathematical precision was not much more than traditional rendering, simply automated by the computer. The labor of the professional hand renderer had been supplanted, paintbrushes were replaced by pixels, vanishing points by vectors. What the quantitative prowess of the computer could accomplish, so could a large team of skilled traditional hand renderers. What had occurred was simply automation; whether by hand or by machine the qualitative manifestations were arguably the same.

This is not to say that the *process* did not yield advantages for the designer; on the contrary, the precise three-dimensional electronic modeling that is a prerequisite to the final rendering process gave designers invaluable intimacy with their designs. Spatial decisions could be made with more formidable criteria, as issues of scale, proximity, and perception now moved to the forefront. This permitted the use of alternate design methodologies, such as 'image sampling,' where imagable observations of the surrounding environment are applied to the spatial development process, allowing for greater visual and functional specificity [14]. Attempting to strip abstraction from the final visual representation had created the

by-product of diminishing abstraction from the process, forcing designers to deal not only with symbols and analytical expressions, but with perceptions and aesthetics. Because of the lack of conventions in this fast moving three-dimensional world, designers would instead examine physical systems, and choose to represent things ‘as they would be built.’

Hence, it was in the design process that computers first began to show a unique usefulness that would portend the present digital imaging revolution. While final output was still encumbered by traditional expectations, the rarefied domain of the design process was free to explore new tools and develop new and useful methodologies. (Many designers may contest these ideas, citing that the only unadulterated path from the conceptive wellspring of the mind is through the hand to the proverbial page. Many will credit the computer only as a documentation device [31]. This is perhaps symptomatic of these transitional times since the ultimate quality of design transcends methods. Nevertheless, the influence of the computer in the formative design process continues to grow, as exemplified by the architecture of Frank Gehry and Greg Lynn.)

In regard then to final output, the personal computer has not yet crossed the threshold from a tool that mimics traditional results to one that invents new ones. Our comforting affinity for the blueprint or its facsimile has maintained the status quo. This could change with the notion of ‘Virtual Reality’ and its defiance of traditional representational venues.

## **2.4 Virtual Reality**

The term ‘Virtual Reality’ is an apparent oxymoron that means different things to different people. To some it conjures up images of goggles, elaborate headsets, and data gloves connected to massive computers with thick bundles of wires. To others it can be the simple act of dreaming. Some find it in the escapism of the cinema, reading, music appreciation, or meditation. Yet others experience it by manipulating teapots on flat computer screens or by playing computer games. What all these definitions have in common is the idea of immersion, of surrendering enough of one’s senses and emotions to this alternate depiction of the physical

world to the point that true reality becomes temporarily superceded.

Virtual Reality in its purest definition will not exist, however, until one can wiggle a toe, sense a drop of water trickle down one's back, or take in the aroma of a fresh fig or the taste of Madeleine cakes— all within a *representation* of reality. As such, the notion of 'Virtual Reality' should be fully understood, and perhaps the phraseology should reflect the hard limitations among the myriad of interpretations. This paper will focus specifically on the definition of virtual reality as *a deliberate computer-driven experiential exercise allowing free movement and interaction in simulated space*. However, other possible definitions of the term will be referenced as they may help clarify the relationship of cultural and technological factors to spatial experiences and perceptions.

The distinction, by formally accepted definition, between the terms *Immersive Virtual Reality* and *Non-Immersive Virtual Reality* is noted: Immersive Virtual Reality employs a device to track some aspect(s) of the participant's body movements while fully occupying both forward and peripheral vision. Non-Immersive Virtual Reality is generally expressed on flat screens and offers no tracking of body movement; rather, a keyboard and/or pointing device are used for navigation [11]. This remainder of this paper will focus specifically on categories of Non-Immersive Virtual Reality.

What any flavor of deliberate virtual reality need do is either ignore or abstract the sensory or emotional stimuli that are beyond the scope of the particular system to adequately address. For example, an architectural plan drawing, despite its depth of abstraction, may be considered by some to be a basic form of virtual reality. To the well initiated, the difference between a dashed line and a solid line may trigger a sensory and emotive response through a visceral understanding of the correlation between a proposed or existing reality and the abstractions of a drawing. If an architect renders a space using ray-traced perspective images, much of the abstraction is diminished, perhaps allowing the uninitiated some greater access to the language of conceptual architectural space. As more layers of abstraction are stripped, the space becomes more and more understandable to the layperson.

Of course, the ultimate sloughing of abstraction in architectural representation results in the space actually being built. In a world without budgets this may be the ideal representation— Real Reality. The depiction can be perceived by all five senses, every dark corner of the space can be scrutinized, and there is absolutely no chance of being seduced by an attractive rendering or by a compelling description. This is generally fiscally impractical (with the noted exception of Josep Lluís Sert's Roosevelt Island Housing project, in which full-scale mockups of apartment units were constructed). This indicates that some amount of subjective abstraction is necessary and desirable.

The accepted balance between abstraction and the depiction of reality in architectural representations has remained steady for centuries but is shifting dramatically with the wide availability of modeling and rendering software for the personal computer. For example, progressively evolving realistic imaging software (much of it built on the foundational work of Donald Greenberg) is having a tremendous impact on the understanding and perception of the effects of light. This coupled with advancing experiential technologies such as Virtual Reality are poised to shift the balance between abstraction and reality even more.

Theoretical mediations on the potential impact of virtual reality on design and representational processes have been postulated by architects and academicians for many decades. Until now, there has been only a smattering of practical study largely because of high cost and low accessibility. We may now have the means to conduct widespread testing of mature virtual reality applications in architectural design and presentation and draw practical, learned conclusions.

## **2.5 Hyper-Reality**

One motivation for the use of abstraction in traditional architectural depictions has been in response to the difficulty of representing reality. This is abstraction for economy. In other instances, abstractions are used because they are simply better than reality for the issuance of particular information— for example an exploded axonometric drawing showing

relationships among building systems. This is abstraction for analysis. There is, however, another potent use for abstraction: abstraction to transcend reality, or 'hyper-reality.' As representational tools continue to approach an ideal of reality, abstraction will on one hand diminish, but on the other hand a new form of abstraction will flourish. In this vein Goldman and Zdepski announced the emergence of legitimate informational tools developed in the fertile valley between 'super-realistic' and 'super-abstract' imaging made possible by the computer [15]. This research is complemented by a comprehensive taxonomy of these tools, which includes but is not limited to: wireframe rendering, colorization, windowing, rescaling, separation, parts to whole, and 3D abstraction [13].

In expanding upon these ideas, the tools of hyper-reality would be overlaid on the virtual reality world to enhance the delivery of analytical information. For example, a complex construction detail could be exploded (parts to whole) in varying scales relative to the viewer (rescaling), perhaps complemented by spontaneous written or audile documentation. Walls could be instantly rendered transparent or removed and replaced (separation). Structural elements could articulate the forces that act upon them (colorization). The viewer could transfigure, becoming very large or very small in scalar relationship to the virtual architecture (rescaling), allowing anything from the manipulation of an entire city plan to the movement through ducts of an HVAC system. In addition, the designer could exaggerate his or her avatar's reaction to heat or cold, become young or old, or be confined to a wheelchair. Visual and audile viewpoints could be switched with those of fellow collaborators. Operating from a baseline of simulated first-person reality, the possibilities to enhance analytical value are virtually limitless.



## CHAPTER 3

### FIRST-PERSON PERSPECTIVE COMPUTER GAMES

The computer games that were subject to research were of the variety offering a first-person perspective interface. While there are many other specific game types, this category was chosen for its apparent close relationship to tools alluded to in much of the available discourse on Non-Immersive Virtual Reality. The technology behind these games is summarized and observations of the gaming development community are offered.

#### 3.1 Binary Space Partitions

Binary space partitions, or specifically binary space partition tree structures, are a programming method for organizing the data used to represent three-dimensional space on computer screens. Essentially, binary space partitions work by recursively dividing groupings of polygons to determine, among other things, visibility, draw order, and collision. The technique is designed to quickly cull polygons which are determined not to be visible from a particular vantage point so that computing resources consider only the PVS, or potentially visible set. To some degree, this is compatible with methods of partial precomputation [30].

The term ‘binary space partition,’ or BSP, is used throughout this paper for the correlation between the systematic ordering of digital three-dimensional data and the building of traditional architectural space. Binary space partitions are used principally to depict the static mass elements of the virtual three-dimensional world and hence the space. The term is employed in this paper to *summarize* a technology and is not intended to dismiss other aspects of game design methodologies.

### 3.2 Game Engines

Data organizing and rendering methods represent the heart of the ‘graphics engine,’ or ‘game engine.’ The game engine in turn represents the core software component of the game and as such determines game action. There are numerous types of commercial game engines, categorized mainly by how they order spatial data and how they render scenes. Those that have received the greatest attention on the personal computer include the *Quake* engine from id Software (with its numerous variants), and the *Unreal* engine from Epic MegaGames. These are true polygon engines that offer first-person perspective virtual reality-style 3D with six degrees of visual freedom (the ability to look front, back, left, right, up and down). Their success in the industry can be attributed to quality visuals, sound, gameplay, networkability, and marketing.

There is, however, a more substantial reason for the success and popularity of these graphics engines: creative response to limitations. All were designed as software-based game engines, meaning that they were designed to run on a large percentage of the then currently installed base of personal computers. To accomplish this, all aspects of game play had to be designed in response to how each best utilized available rasterizing techniques. This developmental process was described by John Carmack, principal architect of the *Quake* graphics engines as “hacks and tricks [5].” For example, static elements in a virtual world (walls, floors) are best suited to the data ordering and preprocessing techniques of binary space partitions. Dynamic elements, such as game characters, may be best represented as sprites (view-dependent 2D animations) or as z-buffered collections of polygons clipped into the static BSP tree (z-buffering is a low efficiency means of determining visibility by comparing differences in distance from viewer to all possible pixels in a line of sight). The most unfavorable result of this mix-and-match approach is that different rasterizing techniques produce different rendered image qualities. Nevertheless, finding the appropriate balance among all possible ways of rasterizing object data is still the hallmark of a visually successful game engine.

Hardware-based game engines, in contrast, are designed to run only with specific hardware, i.e., 3D accelerator cards optimized to perform graphic processing on-board. Polygons can be indiscriminately thrown at the high-speed hardware rasterizer regardless of whether they are static or moving, allowing far better consistency among visual effects. There are of course computational limits reflected by graphics performance (both of which will extend with each new generation of hardware). In addition, the hardware-only game engine risks appealing only to a very limited sector of computer users. However, even casual users are beginning to accept that the next level of graphics performance requires specialized hardware. In response, 3D accelerators are fast becoming ubiquitous, following the route taken by other once 'specialized' hardware, such as color monitors, hard drives, pointing devices, and sound cards.

Although it is clear that geometry engines offering real-time depiction of space on the PC have potential crossover usefulness in many markets (e-commerce, education, construction, etc.), the advancement of the technology is currently being driven hardest by the computer gaming industry [1]. At this writing, the gaming market is large and it is very competitive. It will continue to grow along with demand for proprietary game consoles as well as in response to the expanding base of personal computers. Users of computer games have a general intolerance for abstraction in three dimensional gaming experiences and this is forcing game developers to push the visual, audile, and network capabilities of personal computer hardware and software to absolute limits more so than virtually any other industry. While not yet enjoying a strong presence in architecture, this technology, like other examples of digital technology before it, will inevitably find its way into design and representation processes via a trickle-down path from the more highly capitalized arena of interactive entertainment. Program interfaces available for spatial designers using these tools are still relatively young and by most accounts unrefined and cumbersome. It nonetheless behooves the architectural community to examine the technology and consider for it a place in its sanctioned repertory of design and presentation tools.

### 3.3 Binary Space Partition Games

Most games built on binary space partition programming are classified by the gaming industry as ‘first-person shooters,’ so named for the player’s viewpoint and the fact that the objective of the game is usually achieved through the use of weapons. The first-person aspect of the games explains the popularity of the binary space partitioning tree; the sense of being in a place is achieved by using a virtual reality interface model requiring the optimization of BSP data for computational efficiency. To portray a virtual reality experience tolerably, a game engine rasterizer must be capable of rendering a minimum 10 frames per second, with 15–30 frames per second being an ideal range. Efficient computational processing for the feedback of simulated audile sources is also a prerequisite to acceptable experiential quality.

### 3.4 The Gaming Community

An entire subculture exists on-line which is centered around a shared interest in computer gaming. While this is not surprising given the countless special-interest virtual communities that exist on the internet, the groups that post information specific to BSP games should be of particular interest to architects and spatial designers. One observation that may generate interest, and perhaps envy, among spatial designers is the existence of websites that allow the sharing of informally authored spatial experiences. Almost all BSP games are designed to accept custom-built environments which can be constructed by almost anyone with a reasonable amount of skill and patience. Completed virtual spaces, usually with specific themes, e.g., Japanese tea house, space station, office building, warehouse, etc., are posted for the free examination and use by anyone who has access to the particular game for which it was authored.

Many games include these environment authoring tools, commonly known as ‘level editors.’ Practically all games can accept custom worlds built using freeware or commercially available level editors. The capabilities of game editing software are oriented to a broad range of programming and content-creation skills. These include manipulation of world

geometry (walls, floors, windows, stairs), modification of surface qualities (mappings, bump, transparency, reflectivity), control of lights (brightness, color, falloff, attenuation), control of sound (volume, reverberation, attenuation), customization of character appearance (male, female, young, old) and behavior (neutral, friendly, hostile), customization of artifact appearance (weapons, tools, video monitors, elevators, furniture) and behavior (movable, breakable, light emitting, sound emitting), ambient environment (day, night, fog, rain, gravity, panorama), and finally the specific objective of the gameplay itself (survival, exploration, procurement, construction, destruction).

Despite the informal undercurrents of these gaming web sites, very serious work is being produced. Free utilities for game modifications rival commercial three-dimensional modeling packages costing hundreds of dollars and more. Custom environments reflect hundreds of hours of work. Modifications to games show professional programming aptitude. Many support and tutorial references are well designed and managed. There are formal treatises posted on a vast variety of subjects related to computer gaming, for example, the influence of the architecture of Gehry, Eisenman, and Lynn on level design [3], and an adaptation of von Trier's and Vinterberg's DOGME 95 for game designers [2]. Often, commercial game developers release the source code for their products, welcoming not-for-profit modifications and improvements—evidence of the open-architecture ideals embraced by much of the computer gaming community [1].

## **CHAPTER 4**

### **BSP GAME FEATURES OVERVIEW**

The following is a summary of current first-person gaming technology features, specific to potential usefulness in architectural applications. Listings have been broadly categorized in terms of perceived advantages and disadvantages, despite the understanding that a polar distinction is inappropriate in many cases. Observations are based on detailed experience with specific games engines; therefore, this listing is not all-inclusive. (Because of crossover terminology, the words, ‘player,’ ‘participant,’ and ‘user’ will be used interchangeably.)

#### **4.1 Advantages**

##### **4.1.1 Fully Interrogative**

The category of most existing games using BSP trees is generally described as first-person shooter. Perhaps more appropriate for its portal to architecture, the interface has been termed ‘interrogative walk-through,’ the wording an example of the adaptation of existing terminologies in the face of emerging technologies. Like the well known animated architectural walk-through, this interface represents space from the visual vantage point of one moving through buildings on foot. Unlike most animated walk-throughs, however, neither the direction, the speed, nor the sequence of movement is scripted or forcefully prescribed. This effectively eliminates the passive nature of the walk-through and allows participants the complete and thorough interrogation of the rendered spaces. In attempting to cordon off spaces against requisitive examination, the BSP modeler may use only physical reality-like measures, such as locking doors or erecting fences.

Creating virtual spaces that are fully interrogative allows, for example, the discovery and subsequent enhancement of underutilized parts of buildings. This has potential usefulness in the design of museums, shopping plazas, and other public spaces [7]. Traditional design techniques may not adequately predict the habits of users in a given building. By observing

the directional tendencies of participants, as well as their propensity to linger in particular spaces, architects can tailor design changes responsively to better utilize all spatial resources.

#### **4.1.2 Tactile Solidity**

It is essential to the gameplay of BSP worlds that solids assert their solidity. This may seem elementary, but the impact that this feature has on the experience of virtual space cannot be overstated [4]. Most existing interactive three-dimensional interfaces that allow fully interrogative movement lack this property (a form of collision detection), allowing the viewer the freedom to pass through any so-called solid. While this may be advantageous during analysis studies, for example, its presence during a walk-through can have a devastating effect on the communication of the circulatory and programmatic relationships among spaces and functions. For example, the proximity between doors and the direction in which they swing reflect major design decisions that should not be glossed over in any depiction, regardless of representational methods. Such relationships reveal themselves clearly and unavoidably in BSP depictions.

In the current world of the passive animated walk-through, it is the responsibility of the scripter to avoid breaching obvious physical laws. Great pains are taken during the scripting of passive animated walk-throughs to avoid the disturbing phenomenon of walking through walls or other solid objects or moving at all in an overly unnatural way, particularly during changes in vertical orientation. BSP games build in these functions, and tend to keep closer ties to physical reality. For example, many BSP game engines allow the spatial designer to account for friction and slippage of specific materials under foot to help reinforce the sense of an interactive presence. This accounting for the behavior of materials alludes to Tobin's exposition on the value of constraint-based three-dimensional modeling [28].

### 4.1.3 Integration of Audio

The nature of sound and its relationship to visual space is another important facet of practically all gameplay. Sounds are generally tightly integrated to specific events in the BSP game world, complementing the visual stimuli. For example, the impact sounds of the footsteps of the viewer or other characters correspond to the material that is being tread upon. Appropriate sounds will issue forth from moving water or air. Doors may squeak or slam, mechanical systems may hum, crowds may roar— often with controllable levels of volume, reverberation, and attenuation. A participant's activity within a space can have implications for the interactive experience as the users find that they have control over many of the sounds created in the virtual space. The widespread use of digital stereophonic sound, 360-degree panning, and dopler shifting in many games helps reinforce the nature of space by allowing noise sources to express their locations relative to a player [27].

BSP worlds also provide rich opportunities to test and use nonobjective sound. Nonobjective sounds may be used alone or as a supplement to reality-based, or objective sounds. They can be used as audile enhancements to spatial experiences in attempts to create and nurture specific emotional dispositions in participants. While the technique is rarely used in architecture (Helmut Jahn's United Airlines Terminal at O'Hare Airport is a notable exception), it is used heavily in the film industry. In similar ways it can be used in BSP spaces to underscore, for example, grand entries, spatial contrasts, anticipatory movement, etc. Nonobjective sound may take the form of a music, or it may be decidedly unmusical. How successfully this technique works in BSP architectural depictions is largely dependent on the skill of the spatial 'director' and how well he or she is in tune with the social and physiological makeup of the audience. There are also serious ethical issues to consider as the use of nonobjective sound may be considered by some to be abjectly superfluous and emotionally underhanded [12].



#### 4.1.4 Multiple Simultaneous Users

Most current BSP games are designed from the ground up to be networkable, or, in the common parlance, multiplayer. In fact, the multiplayer function defines the essence of most BSP games. To the spatial designer, this means that designed BSP spaces can be simultaneously experienced by several participants (the current limit is 64 connections on some higher-end games). This is, of course, a boon to the possibilities within the collaborative component of architectural design. Colleagues, clients, and consultants can all meet together within a space, virtually, regardless of whether the individuals are in the same room or across the globe. In fact, the immersive nature of the game interface seems to enrich the collective comprehension and appreciation of a space even when participants in a critical evaluation are shoulder to shoulder, insofar as each critic becomes an active inhabitant. For more distant collaborations, most games feature an intercommunications function that takes place within the game interface using internet chat-like typed dialogue. Again, none of the networkability features are appended e.g., the current whiteboard-like functionality awkwardly draped over popular CAD packages; rather they are an integral part of the interface of most BSP games.

An interesting trend in BSP games is structured teamplay. Players organize themselves into teams and match skills against other teams. Often, there are personal specialties among players and individual skills cumulate to solve collective problems, usually the movement of something or someone to a specific location in the virtual world or the organized procurement of some 'prize.' This seems to indicate a growing interest in the value of collaborative efforts.

One potential impact on the design process made possible by multiple-user spatial interactions is worth noting: as a spatial designer invites other users into a virtual world during design development, their input is likely to effect the design process more forcefully than traditional collaborations. Spaces designed for public use, for example, can be experienced by other design professionals acting as surrogate users. Opinions about perceived

successes and failures in the design can be well communicated because the game interface allows the articulation of the total spatial *experience*, rather than a perception of the cumulative effect of spatial elements. The tendency for some designers to embrace design solutions because of some personal or idiosyncratic ideal will likely be diminished within the facilitated consensual framework [7]. In many ways this reinforces the paradigm of the studio peer critique; nonetheless the potential impact on individual creativity should be duly considered.

#### **4.1.5 Advantageous Use of Network / Internet**

Another important implementation of network resources pertains to the sharing of spatial data files. Because most BSP games are by design optimized to perform well under various network speeds and connection qualities, centrally accessing complex BSP maps live is an impracticality on all but the fastest networks. During network play, sent and received data packets control only the behavior or predicted behavior of existing game entities [26] and cannot create new entities (this should not be confused with visibility and invisibility). Therefore, a prerequisite to gameplay requires that up-to-date copies of scene maps be locally accessible on the computer of each participant. Some games require the manual downloading of the BSP map as well as other support files, e.g., specialized textures, sounds, and characters.

However, the latest solution to this compatibility issue has been the automatic download feature. When a participant hosts a BSP collaboration, invited participants automatically receive the required updates when they attempt to connect. While the length of this operation is largely dependent on the speed of the connection as well as the size of the map, the feature nonetheless facilitates the complex business of ensuring compatibility among participants by utilizing unattended background processes.

#### 4.1.6 Inclusion of Characters

A functionally and philosophically significant feature experienced while playing networked BSP games is the presence of moving human characters. Considering that most architecture is designed for human use, representing spaces without the human presence for which they were intended may be something of a deception. For example, a performing arts company may advertise a concert hall by featuring photographs of an empty auditorium despite the knowledge that most of the targeted patrons will never experience the space devoid of people. This makes something as visually trustworthy as an actual photograph little more than a sketchy abstraction of a probable spatial experience. In order to properly represent an architectural space meant for humans, designers must accept that humans are design entities that impact space as powerfully as walls and columns.

Understandably, humans are not walls or columns— they sit, they stand, they move from place to place, they change their clothing, they interact, and they express free will. Perhaps the very complexity of this ‘design element’ accounts for its omission from most architectural depictions (although human presence *is* generally considered during thermal analysis of spaces). The popular paste-on abstractions of human form used ostensibly for the explication of scale in architectural renderings can not properly represent human integration into designed space.

As mentioned earlier, networked implementations allowing multi-participant interactions provide an enriched simulation of the human/spatial dynamic. Unlike the drone-like models that move robotically from place to place in the typical passive animated walk-through, multiplayer entities in BSP games are dynamic, purposed characters that are very much the avatar of their embodied user. Participants can choose how their virtual character appears to other players by choosing from libraries or by creating custom models; this can help afford immediate visual recognition of participants in virtual space. To varying degrees game characters can gesture or otherwise interact with other participants. This gives BSP spaces a sense of human presence and utility not available in most other spatial simulations.

#### **4.1.7 Simulation of Movement**

The importance that movement plays in our perception of the physical world has been demonstrated by Davis [7]. In fact, perception of movement has been shown to be more important in animals than the perception of color [20]. BSP games are predicated on the fact that things move. Whether it is the movement of ourselves through space or the movement of other characters, the dynamic relationship of occupants to spatial boundaries helps us to understand the ergonomic characteristics of a space. Movement also introduces a valuable temporal quality to spatial depictions, demonstrating the time it would take to move across a single space or among multiple spaces in particular regard to obstacles to free movement, line-of-sight versus available path of movement, and separation of visual and tactile accessibility. This feature can simulate wait times for elevators, predict delivery times for goods, and test effectiveness of locations for emergency means of egress. The movement of inanimate object through space also helps to gauge the changing characteristic of a space, e.g., a closed door containing space versus an opened door admitting space.

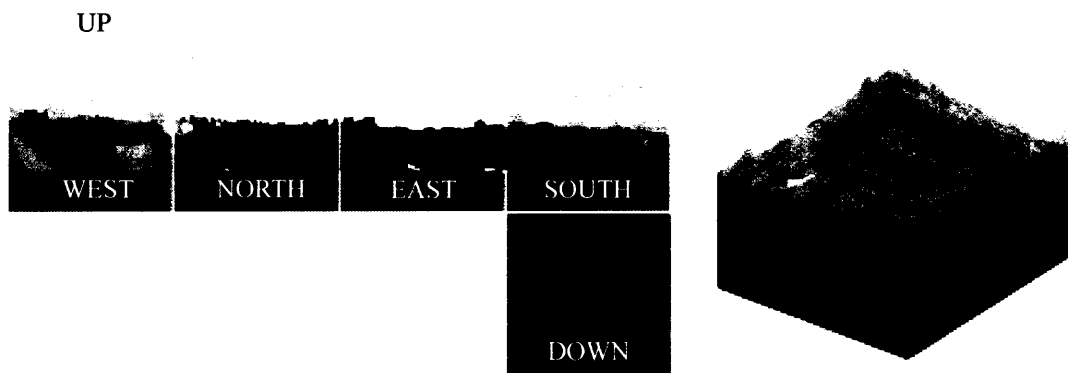
Movement also plays an important role in the application of the notion of ‘hyper-reality.’ For example, the elements that comprise a three-dimensional construction detail may translate, rotate, and scale as they are exploded. The sequence of movement among constituent parts can describe the proper order of assembly. Ancillary wording may move across the screen. In addition, the participant may break from traditional self-motion (In Davis’ term ‘egomotion’ [7]), by ‘flying’ to examine hard-to-reach details and connections on high ceilings, cornices, rooftops, etc.

#### **4.1.8 Simulation of Site**

Dealing with the expansive world that exists beyond the addressable radius of a sited building design presents a challenge for all forms of architecturally descriptive modeling. To enforce contextual implications, models are often represented within surrounding site elements, but it is usually beyond the scope of any given project to model and texturize all elements

within visual range. Physical models may represent adjacent buildings and other site elements that lie within a distance threshold. Digital models may use site photographs with perspective correction to achieve the illusion of site placement. Fog, depth of field, and distance cueing are common tools used to diminish distant details with varying degrees of success.

Since BSP models are interrogative, views of the surrounding site must be dynamically adaptable to the viewer's constantly changing position. To address this, buildings with views of the exterior world that are modeled with BSP tools use panoramic texture-mapping to simulate distant views. Photos are taken of all views surrounding a site and processed into the six square images that each describe one of the  $90^\circ$  horizontal and vertical cones of vision as projected on the inside faces of a cube. The properly oriented cube then envelops the depicted BSP spaces preserving the visual continuity of six degrees of visual freedom. In the viewer's perception the panoramic effect is made to appear spherical by a visual smoothing of corners and edges. This technique is somewhat difficult to accomplish without proper photographic equipment and obviously does not allow participants to interact with the objects pictured on the panorama bitmaps.



**Figure 4.1:** *Construction and assembly of a cube panorama*

#### **4.1.9 Low Cost**

As consumer level products aimed at a large, nonprofessional market, the cost of most BSP games is in the vicinity of \$35–\$50 US (2001). Most level-editing utilities are free. Commercially available game editing software generally costs \$20–\$30 US (2001). These estimated costs should not be confused with those of game development software, which can be thousands of dollars.

#### **4.1.10 Established User Base**

A notable and potentially exploitable phenomenon is the mass popularity of BSP games among recent generations of computer users. The user interfaces of most popular BSP games support generalized standards for mouse and keyboard input which are user customizable. Many users possess, through practical experience, the hand-eye coordination and the conditioned expectation required to easily navigate the virtual three-dimensional worlds of BSP games. Many will understand usefulness of artifacts and the meaning of abstract symbols within the BSP world, even as they move from game to game. Building on a substantial level of familiarity with the interface can help direct the use of spatial simulations as venues for information distribution both within and without the profession of Architecture.

Specific aspects of the study of architectural history provide insight into the opportunities to exploit interface familiarity. BSP depictions of architecturally significant buildings in context, both existing and destroyed, whether realized or forever conceptual, can be maintained in a database complementary to more traditional data sets. The availed experiential study of such spaces can serve as surrogate site visits when actual visits are impractical, impossible, or incomplete. This could afford, for example, walks through reconstructions of the Larkin Building, Hadrian's Villa, or constructions of any of the works of Bouleé or Ledoux. Students' willingness to embrace these alternate avenues of historical research may be facilitated by their preconditioned comfort with the interface.

#### **4.1.11 Interaction with Objects**

The competition to enrich gameplay has led to some innovative strides in the gaming industry with regard to player interaction with inanimate objects set in the virtual world. In early BSP games, most inanimate objects in space did not respond to user interaction. Because most inanimate scene objects (walls, furniture, appliances) were part of a precomputed visible set of polygons in the BSP tree, gameplay performance goals dictated that they remain static. Eventually, doors and windows began to open and shut, lights responded to switches, and elevators could be operated. Each successive version of these games brings some new level of interactivity with scene objects. At this writing, there are movable objects (e.g., furnishings), breakable objects (e.g., glass panes), and operable objects (e.g., video monitors, soda machines). While there are clearly shortcomings in the physical behavior and visual quality of effected objects, these will only improve with time. What may be potentially more useful to spatial designers is Woodbury's idea of 'designerly virtual reality' in regard to inanimate objects. In such a world, even 'non-movable' entities would be modifiable during the spatial experience. For example, walls could be moved or reconfigured, ceiling heights could be adjusted, openings could be shifted or resized, etc. [32].

## **4.2 Disadvantages**

### **4.2.1 Credibility**

As can be expected, architects and architecture educators may be reluctant to embrace gaming technologies as tools suitable for professional work. They are, after all, games. Exacerbating this potential prejudice is the typically violent nature of most BSP games, unabashedly categorized by the gaming industry as first-person shooters. Currently, the potential usefulness of the interface is being overshadowed by implied connections between this type of game and disturbing trends in youth violence and this possibility certainly merits further investigation. It should be noted, however, that it is the intensity of the game objective that drives advancements in spatial representation.

The virtual space as communal venue takes its form through agenda. The agenda of the participants in most first-person shooter games is to locate and incapacitate opposing characters. This is essentially no different than the objectives of the game of chess. While there are undercurrents of violence in both games, the degree to which they are abstracted seems to determine social acceptance.

The simulation of violence may be a popular part of these games because there is presently no other interaction has its clarity or its immediacy. Even the ostensibly simple interpersonal act of shaking hands is difficult to represent convincingly in the virtual world. Current levels of detail and weaknesses in the transmittal of both verbal and nonverbal nuances make capturing and holding the attention of fellow virtual participants difficult. Thus the simulated and exaggerated violence; it makes others keenly aware of our presence, holds their attention, and possibly changes their immediate behavior. Some may see it as a digital counterpart of engaging conversation— although such sublimations may be beyond reach of many and perhaps these games should be discriminately censored. But to dismiss the attendant technological advancements because of a potentially unsavory connection is tantamount to dismissing the Great Pyramids or the Roman Colosseum. Rather, efforts should be made to distill from this technology its positive essence.

There are, of course, computer games that do not rely on violence. The original game *Myst* and its sequel *Riven*, both from Cyan, are praised for their successes in depicting space and environment while omitting violence. These are adventure games, or ‘interactive fiction.’ They are not fully interrogative; rather, they rely on a series of well sequenced still images to give a limited sense of interactivity with space. There is no multiplayer mode. In many respects, these games have closer similarities to passive cinema than to architectural experiences in that the outcomes are essentially predetermined. Nonetheless, such games offer hope for more refined agenda in computer games.

Most BSP games freely allow modifications that can remove the violent characteristics of the gameplay. This has not been a common practice, however, as there seems to be little



interest among general game players in the form without the function. The intervention of spatial designers may help bring digital intercourse to a more refined plane— specifically by assertive communication through spatial manipulation. The most successful players in BSP games are typically those that comprehend the specific characteristics of the virtual environment. They use spatial mnemonics and logic to locate nourishment, shelter, and tools. They develop proficiency with tools, mastering their usefulness under various situations as related to space, terrain, environmental condition, and opponent. In time, players' role in the virtual environment may shift from reactive inhabitant to proactive inventor.

Thus, the current state of BSP games may be seen as a necessary evolutionary step. As more complex interactions with virtual participants are made possible, the nature and quality of interpersonal exchanges will very likely evolve. The next major advance in BSP games will likely involve greater interactivity with the objects that define the virtual world. In the current gaming model this may involve the erection of barricades or the building of bridges. The more refined manifestations will likely be called something like architecture.

#### **4.2.2 Comprehensiveness**

The distribution of BSP worlds for critical evaluation is not for the feint of heart. Giving others an interrogative 'run of the place' is essentially relegating control over what is viewed and how it is viewed, an important dominion of traditional architectural presentation. As such, neglect and omission become glaringly obvious. Minor spaces and details may receive as much scrutiny as major ones. As abstraction diminishes so will the benefits of its economy. Details will invite more details; if a door to a room opens, should not a door to a cabinet? If a fountain issues forth water should not a faucet? While users may come to accept limitations and abstractions, ambitious BSP level designers will always expand experiential possibilities and thus expectations.

Comprehensiveness can of course be considered an advantage, even within the context expressed above. Differing approaches to design development will derive differing value

from a representational paradigm that demands completeness. Nevertheless, this technology will likely force a reevaluation of the acceptable balance between abstraction and reality.

### 4.2.3 Complexity

BSP level construction software currently lacks turnkey functionality and professional polish. Setup can be frustrating and lines of support are usually informal at best. Both freeware and commercial BSP modeling software packages, as well as their authoring companies, appear and disappear with unnerving volatility. There are still very few industry standards or conventions regarding features and user interface. Data exchange options between BSP applications and industry-standard CAD and modeling software are minimal with geometry and practically nonexistent with textures and mapping. Familiarity with traditional CAD and modeling software is often more a hinderance than a help.

However, developers of high-end modeling software are beginning to take notice of the needs of gaming content creators. Polygon-friendly data exchange formats are beginning to appear in high-end modeling software that allow the exchange of geometry with BSP level-editing tools. In fact, Discreet calls its recently released *3DS Max version 4*, “the premiere 3D content creation tool for next generation game development..[8].”

### 4.2.4 Geometry Limitations

One of the most difficult aspects of working with BSP modeling in its current state is geometry limitations. Objects in any possible scene must be rendered in real time. Binary space partition trees can be used to construct a predetermination of the visibility and draw-order of objects from any given viewpoint, but the dynamic depiction of the world must occur instantaneously. As such, current non-accelerated graphics computing power begins to compromise frame rates at about 10,000 visible polygons [19]. Most BSP worlds necessarily avoid excessive geometric detail opting instead for textural detail, with mixed results.

On the positive side are the continuing advances in graphics hardware and software

application programming interfaces (APIs), i.e., OpenGL and DirectX. Furthermore, manufacturers of consumer level GPUs (graphics processing units) are responding to the gaming industry's appetite for rich, complex graphics with powerful and sophisticated hardware solutions that optimally handle off-loaded graphics operations that were previously accomplished through software and CPU, i.e., lighting, shading, shadows, mapping, fog, antialiasing, etc. Game designers are in turn leveraging this newfound computational potency to create more detailed and expressive virtual environments.

Ignoring for a moment the promise of the above mentioned advances, the geometric limitations that now exist in BSP scene construction are no more formidable than were the obstacles facing pioneers of digital architectural graphics in the early 1980s. A host of potential shortcomings certainly existed— geometry, color, resolution— but this did not detract from the intrinsic quality of the work that was produced or the profound advances in learning that occurred. In BSP modeling, as well as in traditional CAD modeling, there will always be the possibility of improvement and always the rationale for waiting. Waiting, however, carries with it the same risks that it did in 1985.

#### **4.2.5 Delivery**

The development of most current interactive 3D object viewing software is geared to World Wide Web access. Many existing 3D geometry viewers, for example, have been designed to be used as plug-ins to internet browsers. This is possible because viewing is generally limited to single isolated objects in space, therefore requiring relatively small amounts of data. Accessibility via the Web, while ideal, is unfortunately beyond the scope of current BSP technology. To experience a BSP world, either singly or in a collaborative mode, all participants must run the game exclusively. The demand on CPU and GPU cycles, I/O, networking resources, and audio subsystem are such that the BSP experience precludes the option of running as a background or secondary process.

In contrast to fully-immersive virtual reality apparatuses, however, the cost and

accessibility of BSP software and BSP compatible platforms favorably promote wide availability. Furthermore, as hardware becomes more robust, internet browser-based BSP implementations will be possible and will likely become commonplace [25]. While the viewing of BSP worlds on flat screens will never compare to the sensory fullness of immersive virtual reality, the extensively compatible operating platforms and ubiquitous presence of desktop PCs nonetheless grant BSP technology an impressive reach.

#### **4.2.6 Lack of Standardization**

As with any fledging technology, there can be expected to be many similar but competing products. As the race to create better game engines ensues, the advances are certainly a benefit to all but the volatility of the industry is frustrating to those using level-editing software. While there is some compatibility of game geometry files among game engines, the finer points of the virtual depictions (textures, behaviors, sounds) do not translate very well among the various BSP content creation packages. At this writing, the *Quake* engines from id Software has proven to be the most versatile. The *Unreal* engine, from Epic MegaGames, has made calculated strides towards compatibility by offering architecturally relevant modeling techniques in its included level-editor, perhaps seeding crossover appeal [19]. The *Quake III* engine, released in 2000, is a hardware-only iteration of the *Quake* line, requiring OpenGL or DirectX support. It can be expected that most new commercial game engine development will be directed toward hardware specific platforms.

### **4.3 Summary**

While many of the features of BSP technology were designed for the needs of game designers rather than for the needs of spatial designers, there is clearly potential for usefulness within the architecture profession. In fact, many of the tools apparently created for the development of engaging and interesting game content may have an adaptive usefulness as tools of spatial analysis. It should be noted that many of the existing digital tools now widely used by

architects were initially conceived for other disciplines (in fact, advertising literature for Discreet's *3D Studio Max* and Adobe *Photoshop* make very little mention of a usefulness to architects). Nevertheless, any tool that demonstrates utility should be examined for consideration. Wide adoption, if it occurs, it will most certainly be followed by the development of a more specific tool set. The degree to which this tool set will be shaped from within the architecture profession will determine its nature and its usefulness.

## CHAPTER 5

### APPLICATION OF BINARY SPACE PARTITIONS IN ARCHITECTURAL REPRESENTATIONS

In an effort to better understand the usefulness and limitations of BSP modeling as applied to Architecture, practical experimentation was undertaken. In all, three BSP game engines were examined for their capacity to represent both existing spaces and conceptual spatial designs. Specific aspects of potential application to architectural modeling and presentation are outlined and summarized.

#### 5.1 The *Doom II* Project

*Doom II* was released in 1994 by id Software as a first-person perspective shooter that offered four degrees of visual freedom (no ability to look up and down). Observation of its mass popularity among Architecture students at the New Jersey Institute of Technology during 1996 revealed the game's impressive graphics and network performance. Its offering of real-time spatial movement far surpassed any of the commercial design and drafting software then available for the personal computer. A decision was made to test the feasibility of the *Doom* game engine for possible usefulness in the depiction and virtual sharing of architecturally relevant spatial experiences.

Initially, a critical examination of id Software's native gameplay spaces revealed that there was no inclination by the game's designers to create 'traditional' space; for example, there were few attempts to create spaces and spatial experiences that mimic those of a typical dwelling, office building, shop, etc. Rather, spaces depicted in the game expressed fantastic, futuristic, foreboding environments with a heavy emphasis on metal panels, large-block stone construction, bulky stair elements, and pass-through window openings. To test whether the game engine was compatible with the depiction of more traditional space, a BSP model was undertaken based on accessible sections of the interiors of two connected

buildings then part of the New Jersey School of Architecture. This area was chosen specifically for its easy access, which facilitated continuous measurement for accurate documentation. Its close proximity also allowed an ongoing comparison of spatial qualities between the actual space and the BSP depicted space. Interested students were enlisted as volunteers for the this documentational phase.

### 5.1.1 Software Setup

Using freeware utilities to perform the actual modeling proved to be a challenge. Setup required a meticulous interaction between the manufacturer's software and the required configuration of third-party level editing utilities. Chosen for level geometry editing was the program *Windeu* (coded by Raphaël Quinet) and for texture manipulation, *Wintex* (coded by Oliver Montanuy). *Doom II* was originally designed for the DOS platform and the editing utilities ran under Microsoft *Windows 3.1*. To facilitate BSP development *Doom II* was ported to Windows and several scripts were written to automate the compile-run process.

### 5.1.2 Modeling

The modeling phase was surprisingly straightforward, but at the same time very limiting. *Doom II* is actually a two-dimensional game engine that simulates 3D by vertically projecting 2D texture information. As such, level editing in *Doom II* allows creation of spaces from a two-dimensional basis. Each entire level is derived from a single plan drawing; there is no section or elevation information. The final level is essentially extruded straight up from the plan, similar to surface generation from single lines. Each line drawn in the plan is assigned a top and bottom Z value, as well as a texture. Connected lines form sectors, which are given floor and ceiling heights, floor and ceiling texture assignments, and global lighting values. Because of its 2D basis the *Doom II* engine is geometrically restrictive, precluding sloped surfaces, bridges, and spaces over spaces. This meant, for example, that there could be no true multistory structures. Also, the ceiling above stairways would mirror the form of

the stair (but not necessarily the texture). In the *Doom II* world, each pixel is roughly equivalent to one inch. A grid option with a minimum tolerance of 2 pixels was used for precise alignment, often at the expense of precise object measurement.

### 5.1.3 Texture Mapping

A requisitive search of custom *Doom II* levels posted on internet sites revealed that most level design hobbyists apparently shared the same spatial sensibilities as the authors of the original game. In spite of the optimism that the internet crowd would demonstrate originality and inventiveness, all native and custom *Doom II* levels looked essentially the same. This can be attributable to three major reasons: 1) Everyone followed the same functional program, i.e., make a series of spaces that is suitable for entertaining gunplay, 2) The game engine restricted all colors to a static 8-bit palette. The 256 colors used for *Doom II* strongly enforced the game designers' intent of foreboding environment, and 3) Nobody wanted to make new textures. New textures were painstakingly difficult to make and even more difficult to distribute. While custom level geometries could be packaged as relatively small files for easy distribution, textures were locked into *Doom II's* WAD file, an enormous (50+ megabyte) file which contained all native levels, characters, actors, weapons, sounds, sprites, and textures. Modifying WAD files is a precarious operation that carries with it the risk of corrupting the native gameplay. Nevertheless, the overuse of native textures had the same unpalatable effect that it does when designers conveniently use the hackneyed sky and brick textures that ship with popular rendering packages— everything looks the same. It was determined that the textures used in the attempted level would borrow not at all from the *Doom II* texture library but would be built entirely from scratch.

The first step that was taken toward creating a new texture library was the documentation of the principal textures that comprised the architectonic spirit of the target space. Since digital cameras were rare in 1996, each wall, floor, ceiling, door, stair, and window was filmed using a VHS video camera. Attention to lighting conditions was crucial



as the *Doom II* engine lights sectors globally; lighting variation in source textures compromises visual consistency. Still frames were captured from the VHS tape using a DPS PAR system and saved as 752x480 pixels 24-bit targa images. These images were then cropped, resized to a maximum 128x128 pixels, and resampled into the 8-bit *Doom II* palette. This procrustean manipulation obviously stressed the original image a great deal and necessitated several cycles of judicious doctoring and field testing before the mapped surfaces rendered satisfactorily.

#### **5.1.4 Character Manipulation**

To fulfill the mission of creating a multi-user, architecturally relevant virtual space, it was decided that the soldier-like character sprites in *Doom II* should perhaps be replaced with something decidedly more civilian. A volunteer student was photographed in various states of stillness and movement from a multitude of different angles. These approximately 36 images were then masked, resized, and resampled to the *Doom* palette before being renamed to replace the constituent frames of an existing character sprite. This proved to be very difficult to complete satisfactorily as it became clear that the artists at id Software had exaggerated the bulk and cartoon-like characteristics of the original character to help mask the flaws inherent to the display of sprite-based characters. This experiment with photography-based images did not bring favorable results.

#### **5.1.5 Compilation**

Because the *Doom II* engine uses a simplified lighting model and is not a polygon-based engine, the time required to compile editable maps to BSP files was relatively short (<5 minutes on a dual processor 90 Mhz Pentium). This was useful in that it facilitated the detached wireframe modeling process by offering almost immediate feedback. In comparison to still-image rendering time requirements of the day, the *Doom II* engine offered more experientially gratifying results, despite overall deficiencies in visual richness.

### 5.1.6 Issues

The experience with the *Doom II* BSP level manipulation uncovered the following obstacles to feasibility in professional-level architectural representation:

- The faked 3D nature of the 2D *Doom II* engine is not compatible with true geometric modeling.
- The static 8-bit palette is too limiting.
- Screen playback resolution of 320x200 is not adequate.
- A minimum opening width for character pass-through of 33 inches is unrealistic.
- The 2D BSP disallows unresolved endpoints, thus barring swinging doors (objects move only along the Z axis).
- There is the possibility of complete void transparency but no differing degrees of transparency for the depiction of glass.
- Inability to look up and down is too restrictive and unnatural for interrogative users.

### 5.1.7 *Doom II* Project Conclusion

Despite the formidable obstacles, the virtual place that was created with the *Doom II* engine captured some hitherto untenable aspect of spatial representation. Many students who were invited to ‘play’ the level for extended periods reported fleeting moments of confusion and panic while walking the physical space, unsure if a repetitive experience in the virtual world could be expected in the physical world. This could be attributable to an interaction of space, function, and mnemonics more so than to a realistic and accurate depiction of a place. Nevertheless, a revisit of this virtual world many years after the irrevocable reconfiguration of the actual one upon which it was based can enliven the memory of that place in ways different than any photograph, floorplan, or still rendering.

## 5.2 *Quake I* Observations

*Quake I*, backed by a true three-dimensional polygon-based game engine offering six degrees of visual freedom, was released in 1997 by id Software. At the time, *Quake I* created an abundance of excitement in the gaming community for its radical new graphics engine and its polygon based characters, but discernible shortcomings in its relevance to architectural depiction caused reservations about its potential usefulness. Experience with *Doom II* level editing provided an evaluative sensitivity to the potential for gameplay versus the potential for Architecture. Screen resolution was still poor, the palette was still limited to 256 colors, and glaringly absent still were swinging doors. This diminished the possibility of gleaning any value from large scale testing. Nevertheless, two intrepid student who had had involvement in the *Doom II* project decided to make use of the *Quake I* engine in exploratory study of architectural design and graphics. Fortunate to have had open-minded instructional guidance, the students produced projects that were innovative in their representational scope, if somewhat inconclusive in architectural value.

## 5.3 The *Quake II* Project

In 1998 id Software released a sequel to *Quake I* with *Quake II*. The *Quake II* engine offered major improvements which included 3D graphic hardware support through the OpenGL API, better screen resolution, and more sophisticated actor behavior— including swinging doors. In keeping with the overall genre of ID's offerings, *Quake II* also featured native environments based on a futuristic, quasi-military theme. Interest in testing the feasibility of this more mature BSP application in an architecturally relevant project was abetted by access to the conceptual design of a New York City residence. The designer of the project agreed to consult on the BSP work while concurrently documenting her work in *3D Studio 4* from Kinetix specifically for the production of still renderings. Since the design had reached a developmental plateau before BSP modeling commenced, the BSP work would involve only the transcribing of conceptual design data and would not be considered for use in

design development until after presentation to the client.

The translating of the residence into a digital format would present formidable challenges for any 3D modeling package. In the design, the proposed residence was contained within a new five story building to be erected on the lower east side of Manhattan on a lot of approximately 23' average width by 100' length. The building itself was 70' long with a 20' wide front façade and a 23'-6" wide rear façade. The 2.86° shift from normal for a side wall would test the usefulness of the software when encountering difficult angles. The overall height of the building was approximately 60'.

Programmatically, the ground floor of the building contained a community facility, the second floor contained a rental unit, and the third, fourth, and fifth floors encompassed the owner's dwelling. At the time, design was focused primarily on the owner's living quarters. Within the residence was a mezzanine creating a double-height space over the main living area. The centralizing design element was an open riser stair crafted of diamond-plate steel in a housed stringer with cable railings above. Most floors were exposed concrete. There was extensive use of glass on the front façade to give views of the city and grey-smoked glass panel partitions were used throughout the interior to express ambiguity of spatial confinement and to offer views of the stair from otherwise enclosed spaces. Among other challenging elements were a fifth-floor terrace and a kitchen ceiling expressed in sheet copper and configured as triangular in plan and curved in section. Overall, the heavy use of transparency would ultimately manifest itself in the contrast between visual and physical accessibility with particular regard to circulatory paths and rich materiality. This and the urban siting, with its attendant sights and sounds, made the project a good fit for the perceived capabilities of the *Quake II* engine.

### 5.3.1 Software Setup

In preparation of the project, *Quake II* level editing software available both commercially and as freeware was researched and evaluated. Ultimately, *qED2* by 3D Matrix, a commercial

product, was chosen for its clean interface, object grouping tools, and good lines of support. Chosen for texture manipulation was *qWall*, a sister product from the same software manufacturer which was available as freeware.

Both *qED2* and *qWall* were designed only for high-level manipulation of game map components and as such additional utilities were needed to compile data to the BSP format. *qEd2* could convert its propriety data files to lower-level map files; compilation of map files would then be performed by three utilities available directly from id Software: *Bsp*, *Rad*, and *Vis*. *Bsp* compiled raw map data to a binary space partition format. This was actually all that was needed to view the geometry within the game interface. Running *Rad* would calculate all designated light sources and the distribution of light energy using radiosity lighting. If *Rad* were not run, the resultant level would display with all pixels at maximum brightness, which was useful for the testing and analysis of geometry. Finally, *Vis* would do precomputation on the BSP map to determine visibility of polygons from all possible visual vantage points. If *Vis* were not run, polygons in the BSP world would erratically appear and vanish as the player moved about— but only in the software implementation of the graphics engine. When running hardware implementations (specifically with the OpenGL API), polygons displayed correctly whether or not *Vis* were used.

### 5.3.2 Modeling

The primitive polygons used in modeling *Quake II* geometry are known as brushes. The most basic and usable brush is the hexahedron. Efficient and effective modeling techniques in most BSP boards revolves around the idea of using only hexahedrons to model all geometry. While *qED2* offered other brush configurations, these were essentially higher-level assemblies of hexahedron brushes. Since economical use of polygons was central to the flow of walk-through simulations, efficient modeling methods dictated the judicious use of brushes.

The use of only hexahedrons to model spatial geometry was not foreign; such techniques were commonly used many years earlier to bridge the limitations of early iterations

of ray-tracing software. Nevertheless, it was deemed advantageous to explore the possibility of importing geometric data from higher-level modeling packages, specifically *Autocad version 14* from Autodesk and *3D Studio Max version 2* from Discreet. While achieving success in importing limited geometry, the third-party converter used was not able to efficiently order the assemblies of hexahedrons that make up complex shapes. Most of the necessary complex shapes would need to be created using only native tools. The most useful native tool functioned using boolean subtraction and could create complex shapes by carving groups of simple shapes with other groups of simple shapes. Resultant hexahedrons were then properly ordered into group describers for further manipulation. Ultimately, knowledge of mainstream modeling software was both helpful and detrimental in that while experience in good modeling practices was essential, the difficulty of working with a low-level tools invited constant comparison to more sophisticated modelers.

### 5.3.3 Texture Mapping

As in the *Doom II* project, the texture libraries and palette provided with the original *Quake II* product demonstrated clear bias for a narrowly defined visual set, again incompatible with most traditional architectural design. Custom material libraries would need to be built from scratch. Through close work with the designer as she continued her *3D Studio* documentation it was determined that the same base bitmaps were serviceable for the creation of mapped materials in either modeling program. The greatest visual compromises were precipitated by the resizing (256x256 pixel maximum) and the integration into the static 8-bit color palette. It must be noted that the palette limitation is diffused somewhat in OpenGL implementations as the hardware accelerator translates compiled maps to 24-bit color allowing for more subtle gradients in lighting and shading.

*Quake II* employs PAK files, which are very similar to the WAD files in *Doom II*. Performing texture manipulation (as well as sprite, character, sound, and panorama manipulation) requires editing the PAK file and distributing changes to it with dependent

maps for coincided multiplayer use. Whereas the *Doom II* WAD file was approximately 50 megabytes in size, the *Quake II* PAK file had grown to nearly 200 megabytes.

#### **5.3.4 Lighting**

Unlike the *Doom II* engine which globally lit each sector, the *Quake II* engine used light-emitting objects to light BSP spaces. The sky texture also emitted light at a controllable level and angle. This was a welcome feature in that it emulated true lighting design somewhat more closely than most conventional renderers. Light sources were constructed faithful to the lighting concept in the original design. The radiosity model of lighting used by *Quake II* was extremely useful as it considered light energy reflected from objects and allowed control over the number of bounces of reflected light energy. Nonetheless, many lighting iterations were required before the settings could be adjusted to give satisfactory results.

#### **5.3.5 Sound**

To reinforce the fact that the building was sited in a busy urban environment, selected ‘objective’ sounds were integrated into the interrogative spatial experience. On the street level, the participant would hear a city soundtrack, replete with the sounds of crowd chatter, footsteps, passing automobiles, and the occasional honking horn. Upon entering the building, the sound would attenuate as the participant gained distance from the street. The street sounds could still be faintly heard if the participant moved close to the windows of the front façade, but would become less discernible as one moved to the upper floors. On the fifth floor terrace the street sounds were barely audible, now replaced by the sounds of light wind and occasional bird chirps.

#### **5.3.6 Site**

The site panorama (known as a sky texture) surrounding the building was set to a generic cityscape to demonstrate the urban setting. This was deemed adequate for the purpose of the

initial presentation. Photographs of the area surrounding the actual building site were obtained and plans were made to use them in a later iteration of the BSP model.

### 5.3.7 Compilation

After *qED2* had saved its propriety data format as a MAP file, *Rad* and *Vis* were run for evaluative model testing. When the model reached about 60% completion (in terms of total number of polygons), running *Vis* became impractical because of protracted compilation times. At this point, testing was moved to a hardware-only solution, essentially a 3D Labs *Permedia II* video card running OpenGL. This eliminated the need to run *Vis*. When the BSP model reached its final state, compilation times on a dual processor 300 Mhz Pentium II were approximately 12 minutes to convert MAP files to BSP format and approximately 3½ hours to light the BSP file using *Rad*. A *Vis* compilation was attempted on the final version of the model, but was aborted after approximately 200 hours of computation. The final BSP model was thus deemed a hardware-only solution.

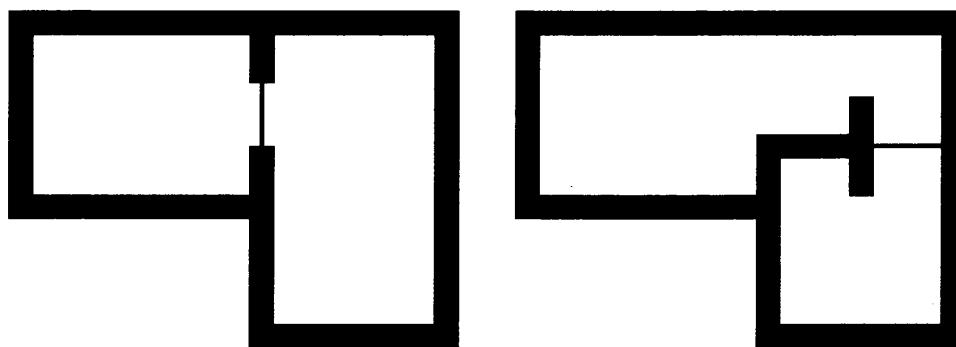
Exploratory research into the cause of the extensive *Vis* compilation times revealed some important issues regarding game design and how it differs fundamentally from architectural design. An uncovering of the processes behind the *Vis* computations revealed that unlimited reference views of large numbers of polygons had quickly overwhelmed the compiler. Game designers avoid this pitfall by carefully segmenting areas (called *portals*) of the BSP world so that the number of views and viewpoints remains manageable for the compiler. For example, if a doorway allows a line of sight from one room into an adjacent room, then all viewpoints possible in *both* rooms must consider all possible views in *both* rooms. If the rooms are oriented at 90° to each other, severely limiting the line of sight, then all possible viewpoints in one room need consider the possible views *of only that room*. This exponentially reduces calculation times for the two rooms.

Game designers understand the advantages of portals and tailor their architecture accordingly. Such measures, of course, must be dutifully repelled by the architect. The



technique that had been used of enveloping the building in a large cube that acts as a site panorama would never be considered by a serious game designer since the number of polygons visible through windows from vantage points outside would quickly become burdensome to the graphics engine and detrimental to the smoothness of gameplay. To the architect, however, such a technique would be a natural impulse and a necessity in that there is no other reasonable alternative.

Figure 5.1 uses exaggerated wall thicknesses to demonstrate two approaches to connecting spaces. A game designer would more likely favor the configuration on the right, since the non-grayed area represents limited line of sight between spaces and the potential for increased polygon usage without diminished graphics performance.



**Figure 5.1:** *Vis Viewpoints / Views.*

*Gray areas must calculate viewpoints and views from adjacent room.*

This design condition explains the binding force that makes so many of the commercially created BSP worlds appear similar and decidedly non-architectural— game performance is the primary goal. Proficient game designers carefully control the amount of game data visible from any given viewpoint. When a game engine is expected to adequately display dozens of polygon-based characters, projectiles, weapons, and explosions, it cannot be preoccupied with resource-hungry architectural details. Of course, spatial designers are concerned with just the opposite. Fortunately, the absence of pyrotechnics in most architectural depictions circumvents many of the above performance issues, and increasingly ubiquitous

display acceleration technology is averting the remaining limitations.

The current proliferation of video adapters supporting acceleration through OpenGL or DirectX is making the entire *Vis* issue moot. id Software understood this when they released *Quake III* with a hardware-only game engine. Many architects understand a similar concept when they upgrade their graphics cards to better run *Autocad* or *3D Studio*. The potential for wide adoption of BSP gaming technology remains reasonably on course.

### 5.3.8 Client Reaction

The clients for the project, a real estate developer and a banker, were presented with the final results as a supplement to a traditional design proposal. The designer opened with a presentation of traditional plans, sections and elevations before moving to her rendered images. The interrogative BSP walk-through was then demonstrated. The clients did not initially use the BSP game interface as they were unfamiliar with the navigation controls. They did, however, verbally direct the nature of interrogative movement through the space. Due to their backgrounds, both clients had had differing experiences with traditional architectural graphics, yet both demonstrated an equal enthusiasm for the BSP 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 (although the observers' perceptions may have been reverse-handicapped because several have backgrounds in architecture).

In conclusion, the BSP 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. Although the scope of these findings is limited, it is probably safe to conclude that most laypeople will develop an affinity for BSP presentations because of the diminished abstraction and the experiential nature of the depiction. Feasibility, however, is a yet unresolved question which must be subject to cost/value analysis.

### 5.3.9 Issues

The experience with the *Quake II* BSP level manipulation uncovered the following issues which were deemed potentially problematic in specific application to architectural modeling.:

- Low-level modeling using only hexahedrons is painstaking but not impossible.
- The static 8-bit palette limits color rendition, but this can be somewhat improved with OpenGL 24-bit translation.
- Maximum screen playback resolution of up to 1600x1200 pixels is excellent but inversely effects graphics performance.
- A minimum opening width for character pass-through of 33 inches is unrealistic.
- Lack of transparency texture-masking forces overuse of polygons.
- Precomputed software solutions are feasible only for elementary architectural depictions.

### 5.3.10 Quake II Project Conclusions

The complex geometric manipulation made possible by the polygon-based *Quake II* engine had made multidimensional architectural depictions feasible, but not yet universally practical. To some degree, the interface needs to be more transparent to the user lest it impede design flow and decision making. Designers who choose to use these tools must understand the impact that the technological limitations have on design and must maintain the ability and willingness to massage the technology to fit the design— rather than compromise the design to fit the technology. Clearly, the software-only implementation of BSPs for architecturally relevant depictions is not feasible due to realistic computational limitations. The hardware-only solution makes the technology usable and therefore useful.

The results of the Manhattan dwelling documentation were extremely salient with regard to the clear, accessible representation of a complex conceptual design solution. Complemented by traditional tools, the BSP walk-through added an experiential dimension to the traditional presentation. This gave the clients valued confidence in the elapsed process, the current results, and the potential of continued design development.

## CHAPTER 6

### POSSIBILITIES FOR FURTHER RESEARCH

As BSP implementations, software engines, and graphics hardware continue to improve, their integration into architecture— in some form— appears imminent. While the mimicry of reality and the concretion of spatial concepts is now plausible using current BSP technology, the full value of these tools will not be realized until they purposefully transcend reality. Presently under exploration is the potential usefulness for architectural hyper-reality of three game-based graphics engines: the *Half-life* engine from Valve Software (a heavily modified *Quake I* engine), the *Unreal* engine from Epic games, and the *Quake III* engine from id Software. The following projects are being considered for further exploration:

- the usefulness and standardization of textural notation overlays.
- the articulation of architectural details in virtual reality.
- the analytical uses of shifting user-to-object scale.
- the development of abstract visual conventions as triggers to analytical events.
- the role of nonobjective sound in interrogative walk-throughs.
- the potential impact of improving consumer-level graphics accelerators.
- the feasibility of virtual reality depiction as legal contract document.

## CHAPTER 7

### RECOMMENDATIONS

Although numerous recommendations can be offered to the gaming industry on ways to make their products more usable for architects, this hardly seems a reasonable course given that there has yet been very little acceptance of gaming technology within the architecture profession. However, many designers appear to be discovering the usefulness of these tools and, guided by the groundwork of early digital pioneers, they will continue this relentless evolution. The following recommendations are most appropriately considered by them.

- Spatial designers should explore the advances in gaming technology as a means of furthering their own goals. The tools have reached a level of maturity that allows their use in informal design development and, with the proper modifications, formal presentation. Rather than let this valuable graphic tool continue to fledge only within the gaming community, the spatial design profession should extract the architecturally relevant components of the technology to develop and nurture a specific tool set.
- Architects should seek to develop new abstractions for use in the BSP world that will supplement the information availed by increasingly accurate depictions of reality. Using the experience and knowledge gained through the use of traditional conventions, architects can complement the weaknesses of the BSP systems by judiciously borrowing from established tools.
- Slavish attention to the details of reality should be pursued carefully. The cost of this in both computational resources and man-hours must be justified by project goal requirements. Rather, synergies with more traditional methods of depiction should be established and standardized.
- Educators should exploit the BSP tools both for the analytical and representational values as well for the apparent affinity for this informational interface displayed by a new generation of students.

## **CHAPTER 8**

### **CONCLUSIONS**

The technological advances in computing during the last 15 years are precipitating a fundamental reevaluation of the tools that define Architecture as a profession. The maturation of graphic engines which use binary space partition tree structures is clearly a watershed in the present digital graphics revolution. Like cinema before it, this technology will inexorably insert itself into public consciousness.

With or without the participation of the architectural profession, a new specialist will emerge who will understand and harness the vast potential of BSP virtual reality tools. He will present complex spatial ideas to the public in forms that they will find accessible, engaging, and attractive. Unless the representational tools of the architect evolve in kind, this new authority may appear to possess greater spatial and technological expertise. To presume that this new specialist and his domain will remain forever within the arena of interactive entertainment, distanced from public confidence, is possibly to relinquish the perception of architects as society's singular and premier designers of space.

Much of the scholarly discourse on the nature of Architecture and Virtual Reality submitted over the past three decades opines that the technology needs further development before issuance of formal conclusions. This technology threshold has been recently breached from without the Architecture profession. The computer gaming industry has developed new tools that allow architects access to mature and useful technology that applies directly and germanely to their spatial, pedagogical, and physiological sensibilities. The trajectory that these tools will take as they continue to advance into the realm of wide public acceptance should be directed by those who best understand the great potential of spatial mastery, be it real or virtual.

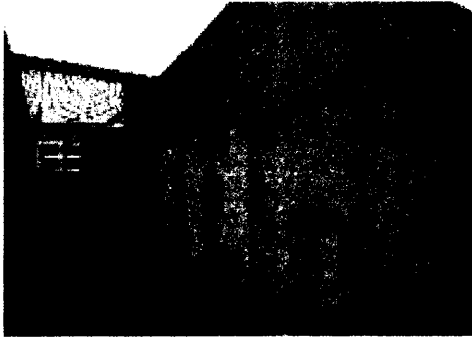
## APPENDIX A

### GLOSSARY OF GAMING TERMS

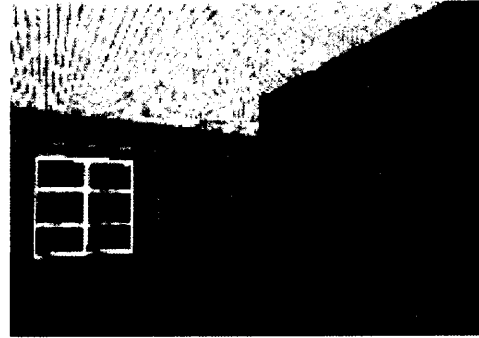
- Actor:** Any object in a virtual gaming environment that can be scripted.
- Bot:** A game player that is not directly controlled by a human.
- Board:** A game level used for multiplayer competition.
- Brush:** The primitive shapes used in modeling BSP geometry.
- BSP:** Binary Space Partition. A data organizing technique for 3D information.
- Frag:** The scoring unit representing the successful incapacitation of a opponent's avatar.
- Freelook:** A mode that allows six degrees of freedom by using only mouse movements.
- Level:** A completed, self-contained game environment.
- Map:** The raw, editable data that comprises a level.
- Mipmap:** A resolution graduated, multiple image texture-map file for distance cueing.
- Mod:** The low-level modification of a commercially available game.
- Model:** The geometry that describes polygon-based characters.
- Pak File:** Similar to Wad File.
- Portal:** The connective visual relationship between or among parts of a level.
- PVS:** Potentially Visible Set. Determination culls nonmember polygons from the BSP tree.
- Rad:** A precomputation performed to determine the effect of light on static objects.
- Screenshot:** A captured still image of a specific view within the game environment.
- Skin:** A descriptive image mapped onto model (character) geometry.
- Sky:** A 2D cube-panoramic depiction of the surrounding site rendered in six images.
- Sprite:** A 2D view dependent animation overlay.
- Teleport:** The instantaneous movement of a character to another location within the level.
- Texel:** A pixel in a texture bitmap. Used for color and lighting determinations.
- Vis:** A precomputation performed on maps to determine visibility of static objects.
- Wad File:** A file comprising a self-contained subdirectory structure of game elements.

APPENDIX B

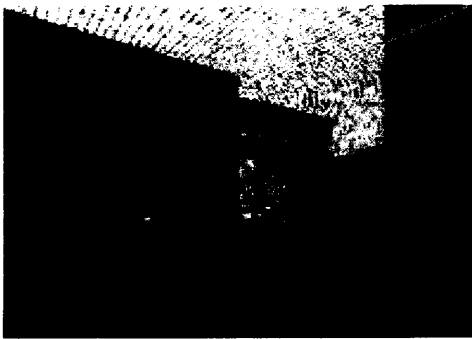
*DOOM II* PROJECT IMAGE GALLERY



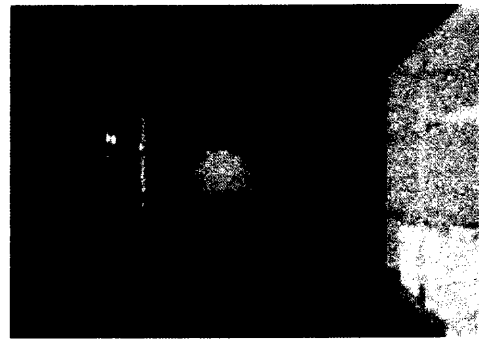
**Figure B1:** *Near Weston Restrooms*



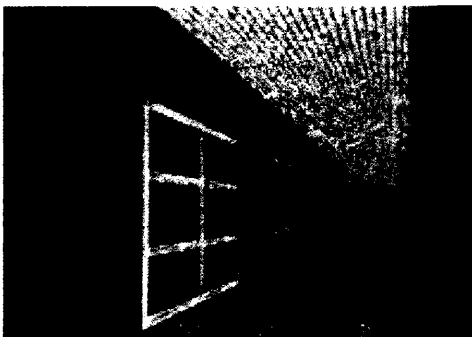
**Figure B2:** *View of Graduate Lounge*



**Figure B3:** *North Hallway*



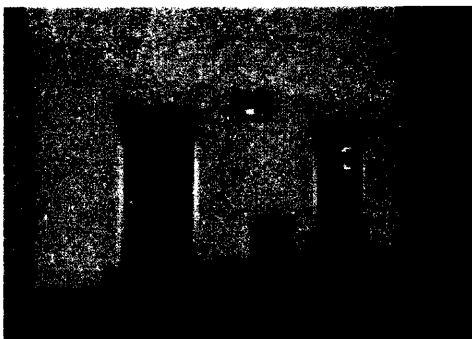
**Figure B4:** *Seminar Room*



**Figure B5:** *Hallway Looking South*



**Figure B6:** *Main Stairway*



**Figure B7:** *Studio Alcove*



**Figure B8:** *Print Room*



APPENDIX C

QUAKE II PROJECT IMAGE GALLERY

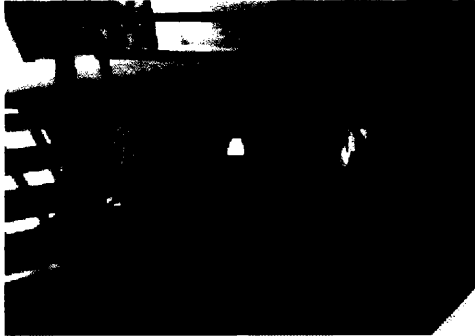


Figure C1: Mezzanine from Above



Figure C2: View from Master Bedroom



Figure C3: View From Terrace

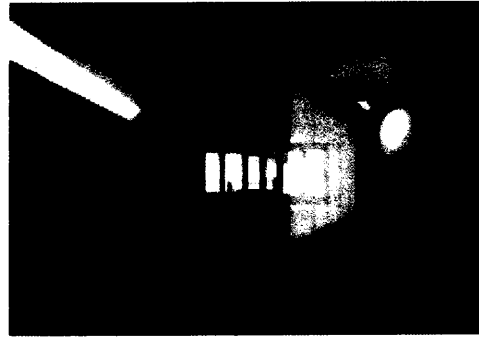


Figure C4: View through Kitchen



Figure C5: View from Living Area



Figure C6: Mezzanine from Below



Figure C7: Entry into Dwelling

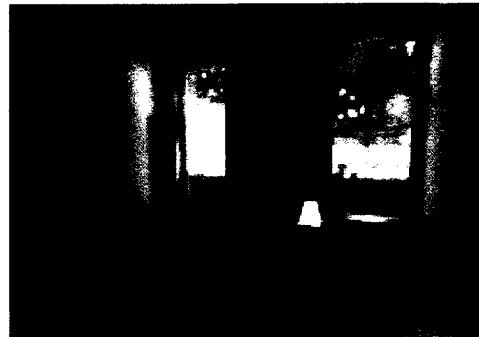


Figure C8: Guest Bedroom

## APPENDIX D

### BACKGROUND INFORMATION

All of the practical projects referenced in this paper are part of original experimental work personally completed within the Imaging Laboratory, New Jersey School of Architecture at the New Jersey Institute of Technology, Newark. The following acknowledgments are noted with deep appreciation:

- There are seven individuals who are credited with formulating the initial response to pervasive computerization in architecture (pp. 2). Noted alphabetically with their then respective affiliations: Elizabeth Bollinger, University of Houston; Glenn Goldman, New Jersey Institute of Technology; Genevieve Katz, California College of Arts and Crafts; Malcolm McCoullough, Harvard University; William Mitchell, Harvard University; Richard Norman, Clemson University; and M. Stephen Zdepski, New Jersey Institute of Technology.
- The work of Dr. Donald Greenberg (pp. 9), Director of Cornell University's Program of Computer Graphics, led to the creation of *Lightscape*, a radiosity renderer from Discreet.
- The useful and appropriate phase, 'interrogative walk-through,' (pp. 16) was, by all available accounts, first assembled and used by Professor Glenn Goldman.
- The *Doom II* project (pp. 32) was completed with the help of the following students: Stephen Vitale, Matthew Gosser, Amado Batour, and Sherri Osowski. The BSP model which was constructed was based on the third floors of Colton Hall and Weston Hall, then part of the School of Architecture, NJIT.
- The *Quake I* observations (pp. 37) describes projects completed by Sherri Osowski and Matthew Gosser, who both worked under the guidance of Professor Glenn Goldman.

Sherri Osowski's project, part of independent study coursework, sought to depict the yet unbuilt New Jersey School of Architecture by translating available building plans into the *Quake I* interface. The project was intended to compare both late-phase conceptual line drawings and BSP depictions to the delivered building product. Largely because of the limits that were described in this paper, she prematurely finalized the project soon after the completion of model geometry.

Matthew Gosser's project was completed within the guidelines of studio work directed by Professor Goldman which was titled, 'Imaginary Worlds: Architecture and Literature.' The onus of the studio was to explore the blurring distinction between architectural design, movie set design, and game design. Matthew's final project, inspired by the book, *Labyrinths: Selected Stories and Other Writings*, by Jorge Luis Borges, was presented in an interactive forum intended to draw guest critics into the designer's interpretation of the author's intent.

- The *Quake II* project (pp. 37) was made possible by the willing collaboration and inquisitive spirit of Kim de Freitas, a NJIT School of Architecture alumna, who was charged with the design of the Manhattan dwelling by her employer, Cutsogeorge and Tooman Architects, New York, NY. Inclusion of the BSP work was agreed to by both the principals of the firm and the clients in the name of research; there was no compensation involved and no promise of further work on the BSP representation. The building, now many times removed design-wise from the original BSP modeled depiction, is slated to be built near the corner of First Street and First Avenue.

## REFERENCES

1. Abrash, Michael 2000  
“The Quake Graphics Engine,” id Software  
Retrieved March 20, 2001 from the World Wide Web:  
<http://www.gamers.org/dEngine/quake/papers/mikeab-cgdc.html>
2. Adams, Ernest 2001  
“Dogma 2001: A Challenge to Game Designers,” Gamasutra  
Retrieved April 25, 2001 from the World Wide Web:  
[http://www.gamasutra.com/features/20010129/adams\\_pfv.htm](http://www.gamasutra.com/features/20010129/adams_pfv.htm)
3. Brown, Duncan 2001  
“GDC 2001: Current Architecture and Potential Approaches to Level Design,”  
Gamasutra  
Retrieved April 18, 2001 from the World Wide Web:  
[http://www.gamasutra.com/features/20010324/brown\\_pfv.htm](http://www.gamasutra.com/features/20010324/brown_pfv.htm)
4. Campbell, Dace and Wells, Maxwell 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/>
5. Carmack, John 2001  
Guest Speaker at Macworld, Tokyo 2001  
Keynote Speaker: Steve Jobs
6. Chronis, George and Knibbe, Willem 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/>
7. Davis, Stephen Boyd, principal author and Huxor, Avon and Lansdown, John 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/vrml-des/title.htm>
8. Discreet 2001  
from Discreet website describing 3D Studio Max 4, Discreet  
Retrieved April 27, 2001 from the World Wide Web:  
[http://www2.discreet.com/products/d\\_products.html?prod=3dsmax](http://www2.discreet.com/products/d_products.html?prod=3dsmax)

9. Dobson, Daniel and Forbus, Ken 1999  
 "Towards Articulate Game Engines," Northwestern University  
 Retrieved March 21, 2001 from the World Wide Web:  
<http://www.cs.nwu/~wolff/aicg99/dobson-forbus.html>
  
10. Duvall, Hayden 2001  
 "It's All in Your Mind: Visual Psychology and Perception in Game Design,"  
 Gamasutra  
 Retrieved April 9, 2001 from the World Wide Web:  
[http://www.gamasutra.com/features/20010309/duvall\\_pfv.htm](http://www.gamasutra.com/features/20010309/duvall_pfv.htm)
  
11. Goldman, Glenn 1997  
*Architectural Graphics: Traditional and Digital Communication*,  
 Prentice-Hall, Inc.
  
12. Goldman, Glenn and Hoon, Michael 1994  
 "Digital Design in Architecture: First Light, Then Motion, Now Sound,"  
*Reconnecting*  
 Proceedings of ACADIA Conference, St. Louis, MO. pp. 27-38, October 1994,  
 Anton Harfmann and Mike Fraser, editors
  
13. Goldman, Glenn and Zdepski, M. Stephen 1988  
 "Abstraction and Representation: Computer Graphics and Architectural Design,"  
*Computing in Design Education*  
 Proceedings of ACADIA Conference, Ann Arbor, MI. pp. 205-215, October 1988,  
 Pamela J. Bancroft, editor
  
14. Goldman, Glenn and Zdepski, M. Stephen 1988  
 "Image Sampling,"  
*Computing in Design Education*  
 Proceedings of ACADIA Conference, Ann Arbor, MI. pp 21-28, October 1988,  
 Pamela J. Bancroft, editor
  
15. Goldman, Glenn and Zdepski, M. Stephen 1990  
 "What am I Looking At?"  
 Text from Lecture,  
 ACSA Technology Conference, Los Angeles, CA. February 14, 1990
  
16. Harfmann, Anton, and Chen, Stuart 1990  
 "Building Representation Within a Component Based Paradigm,"  
*From Research to Practice*,  
 Proceedings of ACADIA Conference, Big Sky, Mo. pp 61-77, October 1990,  
 Peter Jordan, editor

17. Isakovic, Karsten 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](http://cg.cs.tu-berlin.de/~ki/game_eng.html)
18. Melax, Stan 2001  
“BSP Collision Detection As Used In MDK2 and Never Winter Nights,” Gamasutra  
Retrieved March 26, 2001 from the World Wide Web:  
[http://www.gamasutra.com/features/20010324/melax\\_01.htm](http://www.gamasutra.com/features/20010324/melax_01.htm)
19. Miliano, Vito 1999  
“Unrealty: Application of a 3D Game Engine to Enhance the Design,  
Visualization, and Presentation of Commercial Real Estate,” Perilith Industrielle  
Retrieved March 19, 2001 from the World Wide Web:  
<http://www.unrealty.net/vsмм99/>
20. Nakayama, K 1985  
“Biological Image Motion Processing: A Review,” Vision Research,  
as cited by Davis, Stephen Boyd, and Huxor, Avon and Lansdown, John 1996
21. Norman, Richard 1991  
“Real and Un-real Color,” *Reality and Virtual Reality*, pp. 45-52  
Glenn Goldman and Michael Stephen Zdepski, editors,  
Association for Computer Aided Design in Architecture
22. Perron, Richard and Miller, Deron 1991  
“Landscape of the Mind,” *Reality and Virtual Reality*, pp. 71-86  
Glenn Goldman and Michael Stephen Zdepski, editors  
Association for Computer Aided Design in Architecture
23. Reynolds, Craig 2001  
“Stylized Depiction in Computer Graphics,  
Non-Photorealistic, Painterly and ‘Toon Rendering,”  
Retrieved April 2, 2001 from the World Wide Web:  
<http://www.red3d.com/cwr/npr/>
24. Seebohm, Thomas 1990  
“Deconstructing the Constructivist Drawings of Iakov Chernikhov,”  
*From Research to Practice*,  
Proceedings of ACADIA Conference, Big Sky, Mo. pp 61-77, October 1990,  
Peter Jordan, editor

25. Squacky 2000  
Vic DeLeon – Digitalo Design Inc.  
Interviewed by PlanetUnreal  
Retrieved April 2, 2001 from the World Wide Web:  
<http://www.planetunreal.com/interviews/vicdeleon.shtml>
26. Sweeney, Tim 1999  
“Unreal Networking Architecture,” Epic MegaGames, Inc.  
Retrieved March 20, 2001 from the World Wide Web:  
<http://unreal.epicgames.com/network.htm>
27. Sweeney, Tim 1999  
“Unreal Technology Features,” Epic MegaGames, Inc.  
Retrieved March 20, 2001 from the World Wide Web:  
<http://unreal.epicgames.com/UnrealFeatures.htm>
28. Tobin, Kenneth 1991  
“Constraint-Based Three-Dimensional Modeling as a Design Tool,”  
*Reality and Virtual Reality*, pp. 193-209  
Glenn Goldman and Michael Stephen Zdepski, editors  
Association for Computer Aided Design in Architecture
29. Vigo, Libertad and Martinez, Alfonso 2000  
“Before the Digital Studio,” *Arquitectura Digital*, May 2000
30. Wade, Bretton 1997  
“BSP Tree Frequently Asked Questions,” Silicon Graphics Incorporated  
Retrieved March 19, 2001 from the World Wide Web:  
<http://reality.sgi.com/cgi-bin/bspfaq/bsp?1.txt>
31. Wilford, Michael 1996  
“Essay for the Presentation of the Exhibition of the Works of Stirling,”  
RIBA, 1996,  
as cited by Vigo, Libertad and Martinez, Alfonso 2000
32. Woodbury, Robert 1991  
“‘Realities’ of Design,” *Reality and Virtual Reality*, pp. 171-187  
Glenn Goldman and Michael Stephen Zdepski, editors  
Association for Computer Aided Design in Architecture