INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6” x 9” black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600
Superspace - an exploration of architectural space in cooperative networked hyperreal gaming environments.

by:

DAVID M. Sisson

A thesis submitted for fulfillment of the requirements for the degree Master of Architecture

APPROVED, THESIS COMMITTEE

[Signatures]
Fares el-Dahdah, Director
Assistant Professor of Architecture

Lars Lerup
Dean and Smith Professor of Architecture

Michael Bell
Associate Professor of Architecture

Houston, Texas
April, 1999
ABSTRACT

Supaspae - an exploration of architectural space in cooperative networked hyperreal gaming environments.

by:

DAVID M. SISSON

In the late 20th century we find ourselves in a situation where the very nature of reality is questionable; rather than a 'virtual = being such in essence or effect though not formally recognized or admitted' supaspase questions the necessity for such distinctions, hence a 'hyper = above something; too much' to discern the paradigm shift that accompanies significant technological & theoretical innovation. Supaspase is fantastic | shared | networked | custom/ized | a literal site for interaction developed from binary code; it exists in a Krokeresque panic mode, an interior, point-to-point, baroque multi-viewer, habitable non-scripted cinema: "trans-terrafirma". It is fluid & shifting; at times cool & slow, others disorienting & dangerous.

Supaspase gaming represents one possible model event within hyperreality; it places the architect into the milieu of a quantum lightspeed amorphous middleground found on-location wherever you look for it. The author becomes the interface at the very moment of execution; multiplied & recursively scattered.
I would like to thank:

My committee:
Farez El Dahdah, Michael Bell and Lars Lerup
Without whose vision, ideas and encouragement
this project would have never have been pulled off.

My friends and untiring thesis assistants:
Itohan Osayimwese and Joe Meppelink
For excellent continual criticism and many hours
of assistance, friendship and love.

John Biln, Carlos Fighetti, Brian Heiss, Michael
Morrow, Hans & Christopher Kraus, Kindra Welsch,
John Mueller, Chris Nichols, Brian Burke and
Kerry Whitehead
For help, criticism, friendship and assistance.

Kathleen Roberts
For answering every question and her untiring
support of all of us at Rice Architecture.

Planet Quake, Rust, Quake DeveLS - Riviera,
Maj.Bitch, WarZone, Mario[RIP]
Without which, and many other people and web
sites, I could have never done this project.

Frank Lantz
For providing impetus and inspiration.

Entire Q2 Level design email list
For continual tech support and kind words.

My parents:
Gary and May Sisson
For everything.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PG.</th>
<th>TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Title page</td>
</tr>
<tr>
<td>II</td>
<td>Abstract</td>
</tr>
<tr>
<td>III</td>
<td>Acknowledgments</td>
</tr>
<tr>
<td>IV</td>
<td>Table of contents</td>
</tr>
<tr>
<td>V-VI</td>
<td>Illustration table of contents</td>
</tr>
<tr>
<td>1-21</td>
<td>Text</td>
</tr>
<tr>
<td>21-59</td>
<td>Illustrations</td>
</tr>
<tr>
<td>60-70</td>
<td>Text</td>
</tr>
<tr>
<td>71</td>
<td>Endnotes</td>
</tr>
<tr>
<td>72-74</td>
<td>Appendix A</td>
</tr>
<tr>
<td>75</td>
<td>Appendix B</td>
</tr>
<tr>
<td>SECTION</td>
<td>Pg.</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
</tr>
<tr>
<td>ICONOGRAPHY</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EARLY WORK</td>
<td>23</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPARISONS</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SUPACTF</td>
<td>34</td>
</tr>
<tr>
<td>LEVEL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
47  50: Butterfly room
48  51: Butterfly room
49  52: Butterfly room
50  53: Butterfly room
51  54: Green hall / point lights
52  55: Green hall / point lights
53  56: Disco room
54  57: Disco room / powerups
55  58: In the void
56  59: In the void
57  60: In the void
58  61: Sectional view
59  62: Sectional view
60  63: Sectional view
61  64: Sectional view
62  65: Sectional view
63  66: Sectional view
64  67: Sectional view
65  68: Sectional view
66  69: Sectional view
67  INNOVATION  70: Bots with trigger_push entity
68  71: Water doors
69  MODELING  72: Worldcraft screenshot - plan view
70  73: Worldcraft screenshot - section view
71  74: Worldcraft screenshot - section view
72  75: Worldcraft screenshot - plan view
The *Supaspace* project is fundamentally the design of a video game. With that said, the interest here is not in video gaming per se, but rather in the hypothesis that recent developments in computer-aided technologies has created a new site for architectural design.

Supa: 1> derived from jive jargon for "super" 2> something that is 'above something' or 'too much'. Supa is not virtual: virtual is traditionally defined as something that is "not real", making the concept of "virtual reality" an oxymoron. However, in it's contemporary definition, virtual means something that is computer mediated. Supaspace theory holds that it is something that transcends computer mediation, has nothing virtual (unreal) about it, and moves along beside, above, below or through the fabric of reality. Supa holds, in it's definition, a similarity to the concept of "hyper", that is, something that resembles, but is not, the same as the norm.

The argument here is not for a manifesto of virtual / online design, but rather the simple declaration that there is a virtual (supa) site for design. It is that there are people, now, doing design for online/virtual/supa architectures, but architects are not involved. It is, perhaps, a wakeup call to architects and the realization that virtual architectures need the design and spatial sensibility that architects can bring.

It is difficult to actually do virtual architec-
tures, as the tools for design there are quite foreign to the architecture community. (Architects are not programmers.) Additionally, most VR research believes that in order for something to be "virtual" it must somehow trick the mind & body into believing that it is "somewhere else" than the physical world, hence requiring mountains of "high-tech" instruments such as data gloves, shutter glasses, head tracking devices and immersive "caves".

Supaspace destroys the notion that VR is the realm of specially trained and outfitted computer scientists. Supaspace draws on commercially available software and hardware and is more the realm of junior high students than college graduate students, which is exactly the point.

*Supaspace* takes what I think is a pretty obvious observation: that technology has fundamentally altered the ways in which people live and interact with their environment. When I use the term "technology" I am not necessarily referring to "high technology" but rather to something more fundamental regarding the separation between people and animals. Martin Heidegger, in his essay "The Question Concerning Technology" hypothesizes that the use of objects and tools is what separates us from animals. It is from this that I draw my definition of "technology", used in a not unfamiliar form, that of the use of objects and ideas in order to leverage the functions of our minds and bodies and increase the quality of
This process of modernism can be seen as a lineage of technological advances. It was at the moment of the first tool use that mankind became more than just an animal; the curious thing about this process is some of the impetus behind it.

People are inherently ill-equipped to survive in nature. Only through cunning and bodily extensions are we able to conquer nature and turn it to suit our needs. This highlights a peculiar situation of evolution in regards to the modern project: humans do not wait for nature to evolve them to suit, rather, they evolve their surroundings, their extensions and ultimately themselves to suit.

Technology therefore changes us. As we learn to adopt different modes and methods, we begin to alter our way of living to accommodate new requirements in technological use. Examples of this include the forms and modes we now inhabit of the automobile and city.

My interest is in this change: how technologies can fundamentally alter the modes of living and how these alterations are realized in our inhabited forms. (Architectures.)

I am particularly interested in recent so-called "high-tech" innovations; information transfer and communications technologies, including but not limited to: satellite and cellular technologies, computer-based tech-
nologies, the Internet and World-Wide Web. My hypothetical assumption is that these technologies will and already have fundamentally altered the way(s) in which we live and ultimately the built form of the city as well.

This is what has lead me to my interest in virtual reality: in it’s nature of being a technological site for inhabitation and design. It’s nature of simultaneously developing a new, virtual city as well as changing the face of the built, physical city.

What is virtual reality? Jaron Lanier, the computer scientist who coined the term ‘virtual reality’, states that it is a computer-mediated environment that is “fluid like a dream and shared with others." It allows people to create their fantasies and allow others to partake. He further states that children have trouble discerning between fantasy and reality, and this is a good thing for them, as their imaginations can create their realities. There are a couple of downsides to this, the first is unbridled egotism and the second is when they realize that they are not kings of multiple universes, but rather that the only reality where there is food and other people is one in which they are a “helpless puff of pink rubber.” That is why kids love VR, because there they are kings of their worlds and can share them with other people.² For that matter, I suppose this is why we all like VR. (!)

What is trying to be achieved by VR? There are some
popular fictions surrounding the topic that can be used to illustrate the issue. The *Star Trek* fantasy of the "holodeck" is perhaps the ultimate "holy grail" of VR research – the idea of "total immersion" within a universe that, while fantastic, is indistinguishable from the 'real'. Lanier has a good argument why this "grail" is not only an unachieved goal but also an unachievable goal.

In the early days of 'virtual' reproductions there was some confusion surrounding the 'reality' of the reproduced image. In the instance of the sound recording, is documented that in blind tests people could not distinguish between the live sound and the recorded phonographic sound; even though the phonographic recording was garbled and scratchy!

Additionally, in the early days of the photograph, people used to buy "generic" images in order to remember their loved ones. If your husband wore a hat and had a beard, you would buy a photo of a man with a hat and beard, and the fact that it was an image of someone else had no bearing on the matter. It was cheaper to do it this way than to get a custom photo made!

Now of course, we can distinguish between a live music show and the CD copy of the show; and of course, a picture of "just anyone" would "just not do."

This argument sums up my point about the nature of technology: it changes us. We adapt our minds and our
modes of living to the situations we are presented with. I suppose a process commonly called learning or adapting.

In other words: we are not likely to be fooled by "virtual reality", and there is no attempt in the
Supaspace project to make one think that everything is normal, while it is in fact 'virtual'.

Nope, it goes much further and is much more fun than that.

With that said, I will commonly use the terms "virtual reality" or "VR" within this thesis in order to refer to the broader topic at hand.

As previously mentioned, Lanier proposes a definition of "good" VR, which I have chosen as a basis for the Supaspace project: it is a technology that allows people to create their fantasies and to share them with other people. It is a technology that does a very good thing, the bridging of the interpersonal gap.4

There is no single "VR", as has been previously mentioned, it could include sound recordings or photographs. However, for the purposes of this thesis, it is used in it's common understanding, that is, a computer mediated three dimensional architectural environment.

It is therefore customizable, fantastic and networked. It is also a literal site built from binary code, a middle ground where people in potentially far-flung locations can interact.

My interest, of course, as an architect, is in the
design of this middle ground, the virtual architecture. I’m holding that the design of this space is as important as the design of physical architectures, not because it could eradicate physical architectures (which it could, can and will, but is a different discussion altogether) but rather that it will have an impact on how we interact with our physical world and the mode in which we live.

This is therefore why I have chosen to design a video game. Not necessarily because I am interested in video games or gaming, although these are the things which will become important within my thesis, but rather for the things which the game represents.

I chose for the basic technology of the thesis a commercially available video game named *Quake II*. *Quake II* is not necessarily the “highest tech” game currently (late 1998) on the market, (although it’s damn close) but it is the most accessible for me, and this is partly why I chose it. It has a huge customer base and online support and it’s features are very well known. *Quake II* is written by a company named id Software, which is located in Mesquite, Texas. (Near Dallas.) Their commercial success stems from the creation of a game named *Wolfenstein 3-D*, which was the first game in the “first-person shoot-em-up” genre. This genre is simply what it sounds like, you, as the player, are presented with a first person view of the “world” that you wander around in. In early versions of the genre (*Wolfenstein, Doom*) you are pre-
sented with a series of "monsters" that you have to shoot and kill before they kill you. You are also presented with a series of environmental (architectural) problems that you have to solve in order to progress in the "level" and/or to solve "secrets".

Id Software is, by general acknowledgment, the coolest game shop in the world. Named for the instinctual part of the human psyche first identified by Freud, id’s software development team continues to make gaming history. As a renown leader in the industry and one of the world’s leading developer of best selling software, id Software has forged frenetic titles such as Wolfenstein 3-D, DOOM, DOOM II, QUAKE, and QUAKE II. With intense graphics and mind-blowing adventure, id creates frenzied demands worldwide and continues to break retail and shareware sales records. id’s advanced QUAKE II engine is leading the next revolution in 3-D interactive games with both single and multiplayer technology...

...have single-handedly raised the standard of excellence in gaming technology. In fact, the Computer Gaming World Hall of Fame recognized Wolfenstein 3-D as helping to shape the overall direction of the computer gaming industry. id’s products have managed to consistently break shareware and retail sales records at home and abroad while become the torch bearer of what’s hot in the game industry - immersive, frenetic and instantly-gratifying games that push user involvement to a whole new level. id writes games that are cool, games they want to play. And, because of id’s phenomenal success, its games have been ported to virtually every gaming and computer platform...

...1998 has taken id well beyond Wolfenstein 3D [sic] and DOOM to new games, gaming platforms and markets. Because nothing is more compelling
than interacting with another player, id believes that focusing on the multiplayer capabilities of its games and pushing communication technologies to their limits...

...On May 5, 1992, id’s first shareware monster-hit, Wolfenstein 3-D, was released by Apogee. Wolfenstein 3-D brought id worldwide notoriety and the game’s 3-D environment dramatically raised the standards expected for games. On December 10, 1993, id released its much-anticipated DOOM. DOOM attained phenomenal success and is still recognized as the hottest 3-D action game of all time. DOOM changed the face of computer gaming forever. Recognized as the catalyst and inspiration of what we know now as 3-D action gaming, an estimated 15 million copies have been downloaded around the world, passed from player to player by floppy disk or online networks...

...In 1996, id unleashed QUAKE, the most highly anticipated game since DOOM. QUAKE marked the next quantum leap in game technology and set a new standard in multi-player, true three-dimensional, combat action games...

...December 9, 1997, id Software and Activision rocked the continent with the release of the mind-blowing, QUAKE II. Continuing id’s legacy of bloody deathmatches, QUAKE II offered wicked multi-player capabilities allowing more than 32 gamers to compete online. Players maneuver through intense environmental hazards and annihilate evil enemies...QUAKE II exceeded expectations worldwide and has been hailed by countless reviewers and gamers alike, as the best 3-D shooter game of the year.5

These games were revolutionary. Instead of controlling an on-screen avatar (icon or picture that represents the player in the game) you now were the virtual player.
The avatar became a virtual body that the physical player controlled. While there is little theoretical difference between an icon or picture that the player controls and a 3-D model that a player controls, there is a ton of experiential difference. The physical and virtual players took a step towards merging.

Taking this concept further with the release of *Quake*, id created a game that could be played by multiple players, using networked computers. They additionally allowed certain parts of the game to be customized by enterprising users. Shareware programmers and hobbyists created programs to help create new architectural spaces to play the games in ("levels" or "maps"). Others opened up the computer code of the game and re-programmed the game’s rules. Still others made new monsters, weapons and player models. This was all possible due to the structure of the game. Instead of being a closed, inaccessible program, *Quake* was released as a set of files and programs, some of which could be changed. (This will be discussed in greater detail in terms of *Quake II*.)

*Quake* was a significant change in technology from the *Wolfenstein / Doom* set of games. *Quake* was the first true three-dimensional action game. In a *Doom* environment, the player moves through a simulated 3-D space, in *Quake*, the player moves through a real-time rendered 3-D computer model. This is critical for architecture, as now, instead of creating 3-D computer models of "real"
buildings, architects can build "real" buildings within the computer itself. This was the creation of a "virtual" site.

The iconography of these games is critical. They are, of course, about blood and guts. You kill or be killed. Like an action novel, these games use stereotype and modified historical referents (either from real events or fictional) to draw the "reader" (or player) into the story.

In the single-player version of Quake II, you are a "space marine" who is sent to another planet in order to help win the war against the Stroggs, who are an evil alien empire that are busy invading earth. Except for yourself, your squad is killed inroute and so it's up to you to battle your way through, killing Strogg goons and solving various problems and secrets in order to complete nine "missions". The final mission, of course, requires that you kill the Strogg leader.

The iconography [Illustrations 1, 2, 16-19] is militaristic and slightly futuristic. The environments compare to what you might imagine a secret underground military installation might look like, especially if it was built by folks who were something like humans, but also something slightly different. (Aliens, Jane, aliens.)

While the single-player version is certainly entertaining and interesting, it is important to note that it grows old fairly quickly. The monsters behave in a fairly
predictable way and the secrets and problems begin to follow a pattern as well. With this in mind, it included (first in Quake and with expanded capabilities in Quake II) a multiplayer mode. This allowed players to network their computers either through a local area network (LAN) or through the Internet. Players could compete against each other, creating an open-ended non-predictable game.

Quake II was released with an open architecture. Rather than being just one file, the program itself, Quake II consisted of several files. The main one was the executable program, quake2.exe, which was essentially the graphics engine and the brain of the game. This file could not be altered by the end user. Second was the gamex86.dll file. Dll stands for "dynamic linked library" which is a compiled program file that is "called" by the main executable file. This gamex86.dll file contained essentially the "rules" to the game, including the way all the weapons worked, the physics of the "world" and methods of scoring the players. It's source code was released as well, allowing enterprising programmers to re-work the "rules" of the game, create new things (such as weapons) [Illustrations 9-15] and re-compile the result into a new gamex86.dll file. These "modifications" of the game, commonly called "mods," or if heavily modified, "total conversions," could be placed into subdirectories, essentially creating new games. The Supaspace project would be considered a "mod" or "partial conversion". Ad-
ditional files are called by the quake2.exe file and gamex86.dll file, for example graphics files that make up the models of the monsters, players, and weapons, textures that cover the walls within the levels, and of course, the 3-D model, or level, itself. These additional files are simple file formats that can be decoded and reworked by the proper software. Many commercial, shareware and freeware programs have been released to do just these tasks. A complete list of software used for the Supaspase project will be discussed.

Quake II is a virtual reality. It is a networked, shared, optically simulated architectural space. It does all the good things that a virtual reality should do: it allows people to share an event in a computer-mediated middleground, create new spaces, items and games to share with each other.

The space and event of Quake II is fundamentally gory. It is a game about fighting and death. [Illustrations 3-6] The Supaspase project recognizes this, but does not seek to change it. This project is seen as a beginning and understands that this is simply one event that could happen within Supaspase, it is up to others to determine what these next events will be. Supaspase does not embrace violence per se, but works within the framework of the game as it exists, accepting the game as it is.

With that said, Supaspase changes the iconography of
the game to something other than the militaristic elements that you are presented with. It is *funky* and intended to be. [Illustrations 16-21] I feel that as an architect, I can do much better and I feel I have. *Supaspace* is cool. *Supaspace* is different.

*Supaspace* is also cinema, and a very different type of cinema it is. Traditional cinema is developed by a team of people for an audience that is different than that team. Once developed, traditional cinema has a single story-line, and while an audience can get different things out of viewing the film multiple times, it fundamentally does not change. *Supaspace* is “habitable cinema” (Marcos Novak’s term.) It is developed by the people who view and use it, and they create the story line differently every time they inhabit it.

Several of the world’s most respected filmmakers have spoken against the notion that a film leads to a climax, and tells a single story. When Kubrick spoke of wanting to ‘explode the narrative structure of film’ in ‘Full Metal Jacket,’ I think he anticipated the new creative problems implied in the idea of Habitable Cinema. Tarkovsky makes a similar point. Compared to theater, cinema allows artificial and discontinuous environments to be woven into a single, linear experience. Image, sound, and several other cues for understanding are intertwined into one object in time. This multimodal weaving is good, but the singularity in time is something we have exceeded. Habitable cinema dislocates cinema in the same way that navigable music dislocates music. It states that the cinema of the future will be a landscape or matrix or n-dimensional manifold of opportunity. The filmmaker of the future will be a worldmaker. His or
her role will be to invent matrices of opportunity which will combine liquid architecture and navigable music and other dislocated and extended media into situations we can inhabit.6

My interest in this project, as previously stated, is in the architectural environment created, so I'll talk a bit about my discoveries about design for "virtually real Quake II".

Quake II can be seen as a baroque conception of design. The space is folded into interior always, as the game engine cannot handle the "void". [Illustrations 58-60] (Cannot handle infinite space.) Exterior spaces are therefore 'faked' and are likewise boxed in. Surfaces are created through an optical simulation of depth: the texture map. Additionally, the design must allow for progression, procession and events at different places, spaces and times.

Quake II is apparently a single-viewer perspectival space, however, in actuality it is a multi-viewer, multiperspective space. Specifically this can be described as momentum, an inherently a slippery concept. By definition, momentum is a measurement of mass and velocity, velocity referring to a motion vector, holding the qualities of direction, acceleration and speed. In physics, these linked qualities help describe the body or particle under consideration.
In *Quake II*, the situation of the player's game body depends on several variables (health, velocity (which involves trajectory, speed and acceleration) holding of objects and relationships to other objects, as well as the relationships to other momentum's (other player's game situations).

The architecture in *Quake II* must be inherently dangerous if it is to be more than background. In other words, hazards must exist.

While design here is a cinematic exercise, the space cannot be scripted for simply two reasons: firstly because of the issue of the quakebody momentum: the multi-viewer/multi-perspective space. The game space literally changes over time, due to the gameplay changing over time and player status changing over time. The story line demands multiplicity. The second reason is simply a re-statement of the first: It would be boring. In other words, there must always be alternatives within gameplay.

Likewise, the space must have sectional qualities. Here the word “section” must refer to more than architectural section, also to a sectional story-line, sectional opportunities and sectional user choices.

Obviously, it needs cues as to location within the space. A maze/field confuses and frustrates the player and can destroy the convincing nature of the immersive
environment.

I have chosen a specific modification of the traditional \textit{Quake II} game to explore: \textit{Capture the Flag}. I am interested in \textit{CTF} due to several qualities about the game: it demands multiple users, has specific rules of gameplay, demands a semi-specific architectural space and creates higher systems of organization. (Of the game and the player/player groups.)

rules:

1>> shoot enemy players in order that they do not capture your flag.
2>> defend your base (flag) against enemy players
3>> protect players on your team.
4>> penetrate enemy base
5>> capture enemy flag and return it to your base/flag. additionally, capture enemy flag in order that they cannot get your flag to theirs.
6>> shoot enemy player who is holding your flag.

sub rules:

1>> gather stronger weapons
2>> gather ammunition.
3>> stay alive
4>> gather health points as needed
5>> remain in defensive position or advance on enemy base as needed
6>> communicate with team to create teamwork within level

\textit{CTF} fits the project better simply because it re-
quires people to interact at a higher level than simply shooting each other.

*Supaspace* / *Supactf* does several things to the game. Firstly, and most importantly, it changes the architectural environment within the game. [Illustrations 24-69] This is something that is being done every day by normal, everyday people. There is nothing mysterious about it and the process to do it will be described in detail later.

For that matter, nothing within the *Supaspace* project is very mysterious, or couldn’t be done by any junior high student. In fact, it is often done by junior high students, and as previously mentioned, that is exactly the point. Secondly, there is a change of the icons that float on the screen. [Illustrations 16 and 17] I imagine that these are supposed to be part of a "heads-up display" that the "space marine" is wearing, at any rate, they have this afore-mentioned militaristic iconography about them and so have been changed for two reasons: 1) to get away from said iconography and 2) simply to show that they can be changed. I would like to note at this point that there are other icons that cannot be changed. [Illustration 2] These dogtags seem to be somehow un-changeable, while I was able to replace the files with other images, the replacement of them at times caused the game to crash. Thirdly, the majority of the textures that are applied to the walls of the new architectural environments are custom in the *Supaspace* levels. Fourthly,
the gamex86.dll file was re-coded and re-compiled to slightly change gameplay. [Illustration 9-15, 18-21] Fifthly, custom sounds were inserted into the levels. The intent was to "touch" every aspect of the game, to indicate that it could be modified by an enterprising user. The only elements of the game that remained unchanged were the "player models" or the 3-D model of the player. I wanted to leave this open, to represent an element that another player could bring to a level that I designed. I feel that this project is about the mixing of people's fantasies and therefore requires that there be an element which is designable by another user/player. To represent this, I used existing customized player models that I freely downloaded from the World Wide Web. [Illustration 38] (It should be noted that for purposes of this thesis I used a "bot" (a word that stands for robot) which is a program that inserts other "players" into the game. These are computer generated players that use artificial intelligence to operate. They are much smarter than a "monster" but not anywhere near as intelligent as a human. These bots were given the custom player models. For the final thesis presentation, I was able to arrange a small local-area network that the game was played on using a mix of human and bot players. I do not feel that an artificial intelligence can replace people, however, that is a different discussion.)

The architectural environment of Quake II, as previ-
ously mentioned, is called a map or level. There are many programs designed to edit or make levels, all are similar. They are essentially a 3-D model editor, missing some of the more advanced functions of a commercial modeler such as Form-Z or 3D Studio Max. The architectural environments of Quake II use only primitive modeling elements. (3-D primitives are defined as simple shapes: cubes, prisms and other n-sided objects.) Quake II does not allow curved or spherical surfaces. (Although they can be approximated with n-sided objects.) This reduction in complexity helps allow the Quake II engine to render the architectural space in real-time, it is ironically this reduction in complexity that allows something very complex to happen. (The real-time rendering.) I chose a program called Worldcraft to use for the Supaspace project. [Illustrations 72-75] Worldcraft is a commercial program, although there is a shareware version available. I chose Worldcraft as it had the most intuitive interface and proved to be the most crash-free. The process for building a map is fairly simple: one builds a box, places a light and a player start entity inside it, puts textures on the walls, compiles it and loads it into the game.

The model has to be a box. There cannot be any "leaks" to the outside void. (As previously mentioned, the graphics engine cannot render infinity.) The lights are considered a point entity, and do not show up in the
game. (Only the light cast from them shows up, not the lights themselves.) [Illustrations 46, 54, 55] There are many entities that may be placed throughout the map; weapons, ammunition and health, [Illustration 8] player starts, lights, spotlights, monsters, doors and many other things that go bump in the night. The textures are a two dimensional image file that are applied to the walls, stretched, rotated and aligned as necessary. The compiling is a three step process involving three programs: qbsp3, qvis3 and qrad3. Qbsp3 checks the map for problems, qvis3 determines what the player will see from each point within the map and qrad3 does the light casting. (Of course these are somewhat simplified explanations of what each of these programs do.) The compiled map is put into the proper directory, and is loaded into Quake II by typing "map nameofmymap" at the Quake II console. If all goes well, you will be playing your game in no time. It can take from 10 seconds to 40 hours to compile a map on a fast computer, longer times, of course, being for more complex maps.

I used several customized programs to do the compilation process, including a bug-fix for qbsp3 and a different version of qrad3 called arghrad. Arghrad allowed for better manipulation of the lights, simply allowing the final level to look better. I also used a program called Mapspy to check the level for errors before compilation. All of these programs are freeware.
Illustration 1: Typical id Software level (iconography)

Illustration 2: Dogtag icons
Illustration 3: Early level - action

Illustration 4: Early level - action
Illustration 5: Early level - action

Illustration 6: Early level - action
Illustration 7: Early level - note misaligned textures

Illustration 8: Health pack
Illustration 9: Normal laser - sparks

Illustration 10: Supalaser - no sparks
Illustration 11: Normal laser - pain

Illustration 12: Supalaser - no pain
Illustration 13: The hyperblaster
Illustration 14: Normal hyperblaster

Illustration 15: Supagreen laser hyperblaster
Illustration 16: Normal death screen / normal icons

Illustration 17: Supadeath screen / supa icons
Illustration 18: id Software console

Illustration 19: Supaspace console
Illustration 20: id Software / Eraser Bot entry screen & brag

Illustration 21: Supaspace entry screen w/supaspace icons & supabrag
Illustration 22: id Software credits screen

Illustration 23: Supacredits screen
Illustration 24: Blue hall / floating elements [SUPACTF]

Illustration 25: Spiral stair entry [SUPACTF]
Illustration 30: Knife edge hall to red base [SUPACTP]

Illustration 31: Water tunnel up to middleground [SUPACTP]
Illustration 36: Blue base [SUPACTF]

Illustration 37: Red base [SUPAQUOTA]
Illustration 38: Red base with flag [SUPAQUOTA]

Illustration 39: Entry to red base [SUPAQUOTA]
Illustration 40: Tiles [SUPAQUOTA]

Illustration 41: Hallways [SUPAQUOTA]
Illustration 42: Outside [SUPAQUOTA]

Illustration 43: Water - up! [SUPAQUOTA]
Illustration 44: Rocket launcher room [SUPAQUOTA]

Illustration 45: Signing the level / quad damage secret [SUPAQUOTA]
Illustration 48: Blue base near lava traps [SUPAQUOTA]

Illustration 49: Blue base [SUPAQUOTA]
Illustration 50: Butterfly room [SUPAQUOTA]

Illustration 51: Butterfly room [SUPAQUOTA]
Illustration 60: In the void [SUPAQUOTA]

Illustration 61: Sectional view [SUPAQUOTA]
Illustration 62: Sectional view [SUPAQUOTA]

Illustration 63: Sectional view [SUPAQUOTA]
Illustration 66: Sectional view [SUPAQUOTA]

Illustration 67: Sectional view [SUPAQUOTA]
Illustration 70: Bots with trigger_push entity [SUPAQUOTA]

Illustration 71: Water doors [SUPAQUOTA]
Illustration 74: Worldcraft screenshot - section view

Illustration 75: Worldcraft screenshot - plan view
The on-screen icons were changed using Photoshop, a commercial graphics editing program. These icons were in the .pcx file format. In order to get them to show up properly, I had to use a couple of Photoshop "actions" (macros) that were given to me by a member of the Quake2 level editing email list. (Of which I am both a member and the moderator.) These actions essentially remapped the color palate into the one that Quake II uses.

The textures were created using a combination of Photoshop and Wally, a freeware editor for making and editing Quake II textures. These images are saved in a .wal format (hence the name of Wally) for which I had to obtain a Photoshop plug-in in order that Photoshop could read and write the .wal format.

The source code for the game was downloaded from the Eraser Bot website. Eraser Bot is the bot program I used for testing the maps and I wanted this functionality within my game. I modified the Eraser source code (which is itself a modification of the original source code from id Software) to do two things. Firstly, I changed how the laser operated. The original laser threw "sparks" when it hit a surface (such as a wall.) It also caused pain to the player if they walked into the beam. I reworked the laser so it was simply a light, that did not throw sparks or cause pain. [Illustrations 9-12] Secondly, I changed how a weapon called the "hyperblaster" operated. The code to change this weapon came from the tutorials board at
the Quake Developers Library site, and was written by Mario[RIP]. (His name (Mario[RIP]) is his online handle. The [RIP] portion of it means that he belongs to the RIP clan, clans being groups of players that join together to play and challenge other clans to play.) The weapon originally shot bolts of yellow light. The new weapon shoots a green laser. [Illustrations 13-15] I additionally changed the introductory screen to give credit to myself and Mario[RIP] as well as say "supactf". There is also text that appears on the screen when starting the level that says "supactf". Both of these were hardcoded into the game\texttt{x86.dll} file. [Illustrations 20-23] I used Microsoft Visual C++ (also called MSVC) to compile the file, LCC to edit and debug the source files. Both LCC and MSVC are programming environments, but only MSVC will compile the Eraser Bot modification. LCC is freeware and MSVC is a commercially available program.

Custom sounds were edited using Goldwave, a shareware sound editing program.

All of the files, except for the map file itself, were compiled into what is called a .pak file. The .pak file is created and edited using either Qped (Quake pak editor) or Quark, both of which are freeware. The .pak file, like the .wal file, are proprietary formats developed by id Software. The .pak file allows for easy distribution of the software, in order that the user deal with only one file.
Frank Lantz, in his article (quoted in entirety)

Panic Quake Servers:

Quake is id software’s much-anticipated follow-up to Doom, the most successful computer game of all time. Addictive, immersive, and hyperviolent, Quake has already established itself as the ultimate productivity black hole. Now, across the network, puzzled system administrators are discovering more and more machines that have been transformed into secret servers dedicated to managing the shadow traffic of non-stop multiplayer deathmatches.

Panic Quake Servers are the avant-garde of a parasite nervous system grafting itself onto the corporate backbone. Bandwidth scavengers hosting the endless recombinant congregations of vapourwar.

Forget virtual reality and cyberspace. The ultimate synthesis of architecture and cinema is already being coded up around you. Forget about soaring over gleaming spires of data in a weightless universe of pure information. Quakespace is claustrophobic, scatological, prepubescent, and very, very dangerous.

And forget about leaving the meat behind: Panic Quake is nothing but bodies. Bodies splattered, pulverized and exploded. The body fragged and multiplied, becoming pure speed in a point-to-point network of ammunition flows and tactical lust. All-sucking, all-spewing, the Quakebody is projectile and target, monster and hero, author and interface, key, switch, and trap. It is the body with nothing but organs, interrupting and transmitting, and always forever the barricaded global variable in an infinite cascade of lightspeed calculations: surface, perspective, and line of sight - the baroque codes for subjectivity in the digital space of deathmatch culture.
It is, of course, impossible to talk about VR without discussing the body. There seems to be a particular worry in our culture about the relationship of our "self" to our "body", as if perhaps it is wrong for them to be one singular thing or to be two separate things. The worry seems to arise when one determines that if you can create a virtual body, you can create a virtual self as well. The relationship of these virtual bodies and selves to the physical bodies seems to be at the center of the matter. Many popular fictions and concerns highlight the fact that we have a morbid fascination/disgust with the biological apparatus that we are given. The virtual body is a doubling and removing of the "self" from the physical body, at times a re-grounding in the biological and at times a transcendence of it. The emerging science of nanotechnology promises to create a host of machines that can live and work within the physical body, as technology has reshaped our relationship with our physical environs (urbanisms & architectures) so will technology reshape our relationship with our bodies. Urbanism is often an issue of the body versus the environment, and it is becoming more obvious that the two are not separate, in fact are part of the same thing. Likewise, the 'virtual city' is a process that flows through the physical city and physical body. Architecture and urbanism can no longer consider the physical without considering the virtual. Rosanne Stone highlights this issue:
"In virtual systems an interface is that which mediates between the human body (or bodies) and an associated "I" (or "I's"). This double view of "where" the "person" is, and the corresponding trouble it may cause with thinking about "who" we are talking about when we discuss such a problematic "person," underlies the structure of more recent virtual communities."^8

Frank Lantz, in a personal e-mail^9, applied this thinking to Quake, stating that the physical player controls a "marine body" or pawn, maneuvering it through an optically simulated architectural environment. The status of this virtual body / pawn depends upon identifiable factors within the game such as a (virtual) physical position of the pawn, the health and weapon status, and the position of the other player's pawns.

As previously mentioned, design for Quake II is interior always. This created a peculiar situation for an architectural design, because you find yourself creating a sort of twisty cave of a space. It should be mentioned at this point that there are several limitations to Quake II design. All of these limitations exist to allow the real time rendering of the space to happen in a smooth and liquid fashion. Each of these is a reduction in complexity in order that the computer can stream the information more quickly. The first of these limitations is a 256 color palate. These colors are applied to every surface (textures and model's skins, the textures applied to the weapons, player models and etc.) within Quake II, and
the Supaspase project was no exception. When a computer
graphics program uses a limited color palate, it has less
to do, as there are fewer elements to deal with and
therefore less lookup time. The second is the previously
mentioned necessity for the use of 3-D modeling primi-
tives. The simpler shapes require less computing power to
render. The third is imposed by the graphics engine it-
self and is called "r-speeds". R-speed stands for "ren-
dering speed" and is a measure of how many polygons are
in the view of the graphics engine (and essentially the
view of the player themselves) at any time. These numbers
should be as low as possible, especially for a map that
will be shared online (as any information that has to be
transferred through a network takes time and in order for
a smooth multiplayer game to take place online there must
be a maximal reduction in complexity). Typical "good" r-
speed numbers are 300 or less, although they can go to
500. Spikes to 800 are not uncommon in id Software cre-
ated maps. The creation of low r-speeds is one of the
most daunting tasks facing a mapper. There are many
tricks to lower r-speeds, however, they fundamentally
boil down to using fewer polygons to build the level with
and reducing the distance (and therefore the number of
polygons in view) that the graphics engine can see. What
this results in is simple, small spaces. The creation of
large, complex rooms usually results in an unplayable
map. There is a delicate tradeoff between creating inter-
est ing spaces to look and explore and creating interesting maps to play the game in. You have to limit the view of the graphics engine, one way of doing this is using a "vis blocker" (visibility block) or wall in between two rooms so that there is no direct line of sight between the two. This is accomplished by placing a "doughnut" hallway or door set between the two rooms where you have to take a couple of 90 degree turns to move between the two rooms. I also developed, for the SUPAQUOTA map, a vis blocker that I call a "water door". The water door uses a thin brush (it should be mentioned that in Quake II terminology each modeling polygon is called a brush) that has a plain texture applied to it, that uses the water property. It looks like a solid wall, however, the player can walk through it as it is "made from water". The nice thing is that non-transparent water blocks vis in Quake II. I am certain that this trick has been used before in one context or another, however, I have not seen it used as a replacement for a door specifically. The SUPAQUOTA map uses this trick extensively. [Illustration 71] It should be mentioned that a normal door will not block vis, as it is considered "open" by the graphics engine. There is a door called an "area portal" which does block vis, but is difficult to use and requires such a specific architecture to accommodate it that I determined that I did not want to use it.

There were several maps created for the Supaspace
project. The first attempt was to determine how "real" space (or real/built architectures) corresponded to the Quake II spaces. With that in mind, I constructed a reasonably accurate version of Anderson Hall, the Rice University architecture building. I determined very quickly that there is only a small correlation between a real building and a virtual one. The virtual Anderson Hall felt narrow and vertical and had many rooms that were not constructive to game play. (Such as offices and classrooms.) I also determined early on that there is no direct correlation of units of measurement between real (feet and inches) and virtual (the unit of measurement in Quake II is called, ironically enough, the Unit.) A "unit" is approximately one inch, however, you often need a lot more inches horizontally or vertically depending on the activity in the space. (For instance, a room designed for a lot of fighting will have to have a lot of horizontal space for the players to move around in.) I discovered that a "unit" has a direct correlation to the computer measurement of a pixel, for example, if the texture you make to put on a wall is 32 pixels by 32 pixels in Photoshop, it will fit neatly onto a wall that is 32 units square. It becomes an intuitive process to determine how large to make things. Therefore I determined that design for Quake II had to be completely different that anything we as architects were used to. I decided early on that I ought to exploit some of these differ-
ences within my designs.

The first successful level I designed was an exercise in doing just that: designing a working level. [Illustrations 3-7] It used textures designed by id Software and therefore had much the "look" of an id level. It was small and very fast. A CTF game played out very quickly and it was actually a very fun map to play. It's layout was symmetrical, with each base being a mirror image of each other. There were two sets of tunnels [Illustration 7] connecting the bases, each a sort of doughnut with two routes each around a central barrier. (Vis blocker) There were additionally water tunnels connecting the two bases.

I decided at this point that I knew enough to dive straight into the project and so began developing the SUPACTF level with custom textures. [Illustrations 24-36] This level is essentially a loop of rooms and hallways with several "short circuits" in order to create sectional opportunities within the design. I am pleased as punch to announce that this level received great reviews on the Quake2 level editing list [Appendix A] and additionally won the first round of the S3 Quake II level editing contest. [Appendix B]

The final level for the Supaspace project was the SUPAQUOTA map. This map was intended to be more of a massing of spaces with multiple routes between them. In order to create what I determined was a more vertical space, while using less horizontal space, I used the
afore-mentioned water doors to block vis and a series of trigger_push entities to throw the player to exits to rooms. This map is designed for the advanced player, as it requires the use of the grapple to completely navigate it. (The grapple is a grappling hook that the player can use to get to otherwise inaccessible locations within the map. It is only used in CTF.) The trigger_push entity acts as a sort of trampoline that pushes or throws the player. While it in itself is not an innovation, I believe that the extensive use of it within this map, as it is able to rework the architecture within it, is in itself an innovation. There was one bug with the trigger_push. While it worked fine for me, as the player, it seems that the Eraser Bot has troubles with it. The bots will get on the trigger_push (which is an invisible brush) and will jump up and down. They are not, however, thrown up by it, and so they will stay on it jumping up and down essentially forever. It is funny to watch, but not very conducive to game play. (See illustration 70 to see two bots stuck on a trigger_push.) I was able to train the bots to largely ignore the trigger_pushes. It should be noted at this point that it is necessary to "train" the bots to use each new map. You do this by wandering through the map, as you do, the bots learn from you how to navigate the space. A very curious thing for architects to be sure. In order to make them ignore the trigger_pushes, I simply ignored them myself during the
training sessions, and used the grapple to get through the sections that would normally be navigated with the trigger pushes. It worked.

The design for the two SUPA levels was decidedly different. There was extensive use of angular elements, physics-defying stunts, floating and flying platforms, bright and bold colors, lighting effects, and ambient and specific sound usage. The idea was to create maps that were quite different than the average Quake II map being made, and the result was very successful, in my humble opinion.

If there would be any manifesto created in this thesis, I would like it to be regarding computer usage in architecture. I think it’s time we begin considering it more than a tool for creating traditional types of architectural artifacts such as drawings, models and photos. It’s time we consider it a site for design, and begin to understand the impact it is having on our profession and the built forms that surround us. I also believe that this process should be fun and exciting, as this thesis was for me. As they say: all work and no play makes David a dull boy.
ENDNOTES


2 ibid

3 ibid

4 ibid


9 Frank Lantz (frank@rga.com). (1998, October 18). RE: panic quake + ?. E-mail to David Sisson (dms@rice.edu).
APPENDIX A

Received: by marsh.owlnet.rice.edu for dms
(with Cubic.Circle's cucipop (v1.21 1997/08/10) Tue Dec 1 18:02:53 1998)
X-From: bounce-q2mapediting-29283@list.criticalmass.com Tue Dec 1 17:58 CST 1998
Received: from list.criticalmass.com (list.criticalmass.com [204.182.161.114])
    by marsh.owlnet.rice.edu (8.9.0/8.9.0) with SMTP id RAA06190
    for <dms@owlnet.rice.edu>; Tue, 1 Dec 1998 17:58:46 -0600 (CST)
Received: from mail.wes.army.mil ([134.164.38.12]) by list.criticalmass.com with SMTP (Lyris Server version 3.0); Tue, 01 Dec 1998 15:56:58 -0800
Received: by mail.wes.army.mil
    with Novell_GroupWise; Tue, 01 Dec 1998 18:01:55 -0600
Received: from hyded ([134.164.68.30])
    by mail.wes.army.mil (GroupWise SMTP/MIME daemon 4.1 v3)
    ; Tue, 1 Dec 98 18:01:40 CST
Message-ID: <LYR29283-211570-1998.12.01-15.57.03—dms#owlnet.rice.edu@list.criticalmass.com>
From: “David Hyde” <rascal@magnolia.net>
To: “Quake2 Level Design” <q2mapediting@list.criticalmass.com>
Subject: [q2mapediting] Re: supa map
Date: Tue, 1 Dec 1998 17:56:47 -0600
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MMMail-Priority: Normal
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.3110.3
List-Unsubscribe: <mailto:leave-q2mapediting-29283S@list.criticalmass.com>
Reply-To: “Quake2 Level Design” <q2mapediting@list.criticalmass.com>
Precedence: bulk
Content-Type: text/plain;
    charset="iso-8859-1"
Content-Length: 1622

David Hyde <rascal@magnolia.net> posted the message below

David,

You’ve GOT to be an architecture student... am I right?
No one else has this kind of funky imagination. Wow.
You’ll certainly win any “unique Q2 map” contest hands
down. Aside from some very dark textures, I liked the hell
out of this map. I especially like the hallway with
animated textures - though you probably need to
distribute barf bags with your map. The lighting effects
with the cubes in several places was very cool
and unique - surprised we haven’t seen something
like that before.

I’d suggest a few strange ambient sounds to fit your
theme.

I imagine you’ll hear a lot of whining about how this
thing doesn’t fit into the Q2 scheme - please ignore it if it occurs. I’m very impressed. Please let us know when you finish this thing up.

David Hyde

> >
> >David Sisson <dms@owlnet.rice.edu> posted the message below
> >
> >ok: i put on a pak9.pak file, which includes the custom textures.
> >
> >(there’s only 8 of them, so the pak is small)
> >
> >trance
> >
> >
> OFFTOPIC posts will get you kicked off the list. Don’t do it. Email the person who sent this message personally at
> David Hyde <rascal@magnolia.net>

Brought to you by PlanetQuake: http://www.planetquake.com
Entity Questions? Go to RUST...NOW! http://www.gamedesign.net
To unsubscribe, forward this message to leave-q2mapediting-29283S@list.criticalmass.com

Received: by marsh.owlnet.rice.edu for dms
X:From: bounce-q2mapediting-29283@list.criticalmass.com Wed Dec 2 16:19 CST 1998
Received: from list.criticalmass.com (list.criticalmass.com [204.182.161.114])
    by marsh.owlnet.rice.edu (8.9.0/8.9.0) with SMTP id QAA26998
    for <dms@owlnet.rice.edu>; Wed, 2 Dec 1998 16:19:52 -0600 (CST)
Received: from smtp2.andrew.cmu.edu ([128.2.10.82]) by list.criticalmass.com with SMTP (Lyris Server version 3.0); Wed, 02 Dec 1998 14:10:49 -0800
Received: from JOSH-B.RES.CMU.EDU (JOSH-B.RES.CMU.EDU [128.2.94.52]) by smtp2.andrew.cmu.edu (8.8.3/8.8.2) with SMTP id RAA24830 for <q2mapediting@list.criticalmass.com>; Wed, 2 Dec 1998 17:11:12 -0500 (EST)
Date: Wed, 02 Dec 1998 17:11:51 -0500
From: “Josh Binder” <jmbinder@andrew.cmu.edu>
To: “Quake2 Level Design” <q2mapediting@list.criticalmass.com>
Subject: [q2mapediting] Re: Things
Message-ID: <LYR29283-211970-1998.12.02-14.11.07—dms#owlnet.rice.edu@list.criticalmass.com>
X-Authenticated: jmbinder by cyrus.andrew.cmu.edu
X-Licensed-To: site license
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
List-Unsubscribe: <mailto:leave-q2mapediting-29283S@list.criticalmass.com>
Reply-To: “Quake2 Level Design” <q2mapediting@list.criticalmass.com>
Precedence: bulk
Content-Type: text/plain; charset=us-ascii
Content-Length: 1035

Josh Binder <jmbinder-q2map@andrew.cmu.edu> posted the message below
> yep. i think this was the problem: i posted a zip file, which should be all
> you
> need as well...and "should" work. if not, someone plz let me know!
> >
> > tranc

Just for your info, it DID work with the zip. Phr33ky level man. I especially like the spiral stairs-on-the-wall looking part underwater. Some of the darkness made it look like a bunch of thrown together crap though, but I used my blaster for a flashlight and saw all the pretty shapes. =]

-=/ DinK

OFFTOPIC posts will get you kicked off the list. Don’t do it. Email the person who sent this message personally at Josh Binder <jmbinder+q2map@andrew.cmu.edu>

Brought to you by PlanetQuake: http://www.planetquake.com
Entity Questions? Go to RUST...NOW! http://www.gamedesign.net
To unsubscribe, forward this message to leave-q2mapediting-29283S@list.criticalmass.com
APPENDIX B

Received: by marsh.owlnet.rice.edu for dms
X-From_: Developer_Relations@s3.com Mon Jan 11 19:46 CST 1999
Received: from gateway.s3.com (gateway.s3.com [208.202.167.134])
    by marsh.owlnet.rice.edu (8.9.0/8.9.0) with ESMTP id TAA02110
    for <dms@owlnet.rice.edu>; Mon, 11 Jan 1999 19:46:29 -0600 (CST)
From: Developer_Relations@s3.com
Received: from s3.s3.com (loghost.s3.com [172.19.8.4])
    by gateway.s3.com (8.8.8/8.8.8) with ESMTP id RAA08014;
    Mon, 11 Jan 1999 17:36:07 -0800 (PST)
Received: from cc-smtp.s3.com ([10.10.8.10]) by s3.s3.com (8.6.12/8.6.12) with SMTP id RAA12523; Mon, 11 Jan 1999 17:35:49 -0800
Received: from ccMail by cc-smtp.s3.com
    (IMA Internet Exchange 2.12 Enterprise) id 001976CF; Mon, 11 Jan 1999 17:33:07 -0800
Mime-Version: 1.0
Date: Mon, 11 Jan 1999 17:30:04 -0800
Message-ID: <001976CF.1504@s3.com>
Subject: Congratulations!!! (Finalist)
Content-Transfer-Encoding: 7bit
Content-Description: cc:Mail note part
Content-Type: text/plain; charset=US-ASCII
Content-Length: 693

Congratulations! You have been selected as one of the finalists for
our S3TC Game Level Contest. We need to verify your eligibility and
get some additional information before we can send you a
Savage3D-based graphics board and other tools for the second phase of
the contest.

Please reply to this email as soon as possible with your full name,
mailing address, and whether your system is AGP-compatible. Please
make the subject line of your reply “S3TC Contest Finalist
Verification”.

Again, congratulations on being a finalist. We’re looking forward to
seeing your final 3D scene.

S3 Incorporated