Game Engines as Social Networks

Robert F. Nideffer, 2003

Abstract: The following essay: 1) explores how the concept of the "game engine" can be applied to heterogeneous, multi-modal, distributed and pervasive network environments; 2) posits the need to situate the game engine in relation to the cultural conditions that produce it; 3) looks at the ways in which computer gaming grew out of and continue to express a strong desire to play together as part of a larger social network; 4) explores some of the conceptual, economic, and technical issues confronting infrastructure creators and end users of that infrastructure as networked gaming begins to move to the next level, particularly in the areas of interoperability and heterogeneity; and 5) asserts the need for developing network-enabled tools and services that allow people to create "context" as readily as they create "content," so that infrastructure can become a primary place of play.

Setting the Stage

Game Engines

The term "game engine" is usually applied to the software infrastructure that has been engineered to render to a display screen everything you see and interact with in the game world. From this perspective game engines provide the graphics capabilities, the physics models, the collision detection, the networking (when present), and the core functionality the player experiences during game play. Elsewhere I've asserted that one can think about the game engine as a culturally encoded "database interface" – a mechanism through which a pre-determined, relatively constrained collection of socially sanctioned procedures and protocols are used to render a world and make it navigable in context. Along these lines I've argued that it's important to look at the game engine as a cultural artifact that circulates within a specific social domain, in order to begin thinking about how to make more visible the implicit and often taken for granted assumptions operative during software development, as well as to extend the boundaries of what constitutes the game engine. I've done this in effort to move beyond thinking of the game engine strictly in software engineering terms in effort to also think about it in social engineering terms (Nideffer, 2003).

My interest in doing this is to open up the possibility for exploring the ways in which people's ideological frameworks influence the way meaning making happens during not only the gaming experience, but also during the development of the infrastructure designed to support that experience. Thus the game engine becomes not simply a piece of software, but something that reflects and embodies the cultural conditions indexical to both the developers of the system, as well as the end users of that system. The general goal is to begin to provide a platform for investigating how and why certain functionality gets consistently coded into software, and why it becomes so much more difficult to creatively modify that functionality once it's been embedded into infrastructure. In this essay I'm interested in extending this notion in order to begin looking at how we might begin thinking about game engines in relation to networked multiuser gaming environments, particularly as they begin to move toward greater heterogeneity in terms of the types of devices and interfaces available to people connecting to shared online information space.

Games have become hugely popular, and are now claimed to be the favorite form of entertainment in the United States for the third year in a row. According to a summary of recent studies compiled by the Entertainment Software Association (formerly known as the IDSA or Interactive Digital Software Association), two times as many polled stated they prefer playing games to watching television, while three times as many people would rather play games than rent movies. The average gamer is 29 years old. The average American youth spends 7-30 hours per week playing. Nearly 50% of players are now reported to be women (IDSA, 2003). In the context of the videogame industry, there is general consensus that a big part of the future of games is on the Internet. This will include anything that is able to access games through Internet capable computing devices such as desktop and notebook computers, Tablet PCs, Smart Displays, PDAs, mobile phones, set top boxes, both network specific and gaming consoles, and smart appliances, which include products such as e-mail devices, picture frames, televisions, and Internet-enabled white goods (Wolf, 2003).

According to a report from market analysts, The Themis Group, massive multiplayer games will generate roughly $1.3 billion during 2004, and increase to an anticipated $4+ billion by 2008. The bulk of this will come from subscriptions, but a growing proportion will be generated by the sale of virtual property and in-game items (BBC News, 2004). It's anticipated that the installed base of mobile game users alone will reach 100 million in 2006, and represent roughly a five billion dollar wireless gaming market (Gale Group, 2001). In 2000, Internet penetration was over 50% in the USA, with over 53 million households connected. By 2004, according to a new study released by research firm Nielsen Netratings, three out of four Americans, or a total of 204.3 million people, had access to the Internet (Gruener, 2004). Put quite simply, this growth is staggering.

Surprisingly, given the growth in online gaming, Internet connectivity, and Internet related technology hardware, software and services, though there are many examples of Net-based games, discussion of Net-based *game engines* is virtually non-existent. There is perhaps one notable exception in a short-lived commercial flop released in Summer of 2001 by Electronic Arts called Majestic, although even this instance was not framed in terms of an "engine" so much as a network-centric pervasive game run on a custom designed infrastructure that was referred to in-house as the "experience server." What I will attempt to argue a bit further on is that while Majestic may have failed in the marketplace, it remains one of the most interesting game play and development environments recently attempted, and was ground-breaking and genre-expanding in a number of significant ways. For now, suffice to say, even in the case of Majestic the notion of the "experience server" did little to advance the dialogue surrounding what might constitute the game "engine" in a networked environment, or point to where development pitfalls may lay as next generation infrastructure and content gets put into play in areas such as technical implementation, competitive economy, conceptual design, and perhaps most importantly, social interaction and engagement.

Social Networks

In the context of the social sciences, *social network analysis* is primarily concerned with people and their relations when manifest as part of a larger collective. As Laura Garton (et al) claims, just as a computer network is a set of machines connected by a set of cables, a social network is a group of people, organizations, or other entities connected by a set of social relationships such as friendship, work or information exchange. In this regard the job of the social network analyst is to describe these relations as fully as possible, teasing out their prominent patterns, tracing the flow of information and resources through them, and discovering the effects these relations have on people and organizations operating within them (Garton, Haythornthwaite and Wellman, 1997).

Thus social network analysis is the mapping and measuring of relationships and flows between individuals, groups, organizations, computers or other information/knowledge processing entities. What are referred to in the literature as the "nodes" in the network are the respective people and groups that get mapped and measured, while the links between them display various relationships or flows between those nodes. Social network researchers measure network activity for a node by using the concept of degrees, which refers to the number of direct connections a node has (Krebs, 2004). Social networks can vary dramatically in their size, heterogeneity, and degree of density. Within a given network, a "tie" is that which connects a pair of actors by one or more relations. Pairs may maintain a tie based on one relation only, e.g., as members of the same organization, or they may maintain a multiplex tie, based on many relations. The more relations (or "strands") in a tie, the more "multiplex" (or "multistranded") is the tie, and the more densely knit the interconnection is perceived to be (Garton, Haythornthwaite and Wellman, 1997).

In the domain of networked games, concepts that stem from social network analysis could prove quite useful in the effort to begin articulating the conditions of an infrastructure that would assist in building community, aid in establishing an awareness of emerging patterns in information and resource exchange, friendship hierarchies and work relations, and help visualizes the ebbs and flows in the temporal density of the ties that bind individuals and organizations together. To facilitate this designers will need to be able to define their concepts clearly, attract the appropriate player population, provide means for players to come into contact with each other based on interest and experience, and encourage, sponsor and reward behaviors that reinforce the desired outcomes. Finally, and perhaps most importantly, next generation game engine makers, as well as game application makers, must also build in the flexibility for unanticipated patterns to begin to emerge.

Unsung Heroes

Though little acknowledged, computer games have been at the forefront of many core areas in the computer and engineering sciences, and have essentially supported multiple players since their inception several decades ago. The earliest and perhaps most prominent example is "Spacewar," which came out of MIT in the early 60s, and was initially implemented on a giant TX-0, the world's first online time-shared computer. What's seldom reported is the role that the TX-0 played, as an infrastructure platform, in ushering in the next generation of designers and programmers due to the way the hardware and display device had been engineered. Early experiments on the TX-0 included not only Spacewar, but other game and game-like programs such as Bouncing Ball, Mouse in the Maze and Tic-Tac-Toe as well (Graetz, 1981). Initially multiplayer meant little more than one user taking turns on the same computer at different times. However it wasn't long before multiple players were able to use different input devices on the same machine simultaneously. Eventually, with the advent of personal computing and the ability to both create local networks between multiple machines (using modem cables and the like), as well as connect to other players remotely (through IPX networks and ultimately onto the Internet), people started actually designing and playing "networked" games more along the lines of what we understand them to be today, where physical co-presence is no longer a necessary requirement.

What is particularly interesting about computing, gaming, and networking, is the way in which it spontaneously arose out of a collective desire not just to *play*, but to play *together*. Games, or for that matter computer-mediated socializing, had never been an integral part of the plan when mainframe computing, personal computing, or networked computing was envisioned and implemented. In fact, games and personal communication exchange was not on the map. However games and social networks rapidly became the quintessential unintended consequence of computing, and to date have driven forward many of the most fundamental advances in numerous technology areas. It is safe to say that gaming has propelled many of the core domains of computing forward much faster than they would have otherwise been propelled.

The contribution of games has been gargantuan, whether you look at the history of computer graphics from a software engineering standpoint, the development of hardware display devices, input devices, networked operating systems, networked multimedia delivery systems, and more recently the engineering of graphics cards, or the attempt to integrate various everyday communication modes (chat, email, instant messaging, voice over IP, FAX and GPS) into a singular and coherent entertainment experience. In fact, games and the desire to play them socially has arguably driven the design and deployment of these and other such technologies faster and further than any other singular computing activity.

So while in the early history of computing games may never have been the overt or explicit catalyst driving the development of the technology, they nevertheless often covertly or implicitly performed that function, and will continue to do so in the foreseeable future. Indeed, if anything, there is finally starting to be far greater consciousness around the pivotal role computer games have played historically, and the tremendous potential they have for future applications across an incredible spectrum of scientific, commercial, industrial, entertainment, creative, and educational sectors.

Contemplating the Divine

For young (and not so young) boys in particular, the desire to play computer games, and the often complex technical challenges one confronts in order to make them run or to be able to play them together, has had the effect of providing hands-on training in many key technology areas related to computing, operating systems, and networking. Arguably, the skill sets that emerge in relation to such activity have proven to be quite useful for many of the more adept. To date, sadly, game design, implementation, deployment, and consumption remains an exceedingly male dominated activity. However, that appears to be changing rather rapidly, at least this is certainly the case in terms of who's *playing*, though unfortunately not so much in terms of who's *making*.

Given this gender bias, it is perhaps no great shock that the first robust program facilitating Internet gaming should be christened "Kali" by the two guys who created it – serving as a rather unreflective homage to the Indian "mother-goddess" unconsciously heralded as the enabling device through which violent male fantasy would typically get to play itself out. Developed in the mid 1990s by two people – Jay Cotton (a Management and Information Systems analyst), and Scott Coleman (who eventually went on to work for the game company Interplay) – Kali was implemented as a software solution for connecting multiple machines together, whether they be local to each other in physical space or remote. As originally envisioned, Kali would initiate and verify connections with all players in the game, configure the game dynamically so that all machines are guaranteed to be using the same set of game parameters; support an IRC-like chat mode whereby players could communicate once connected; and provide an online player directory, so that opponents could be selected from a pick list (Cotton, 1994).

In terms of Indian mythology, Kali makes her debut in the Devi-Mahatmya, where she is said to have emanated from the brow of Goddess Durga (slayer of demons) during one of the battles between the divine and anti-divine forces. Kali is represented as a Black woman with four arms; in one hand she has a sword, in another the head of the demon she has slain, with the other two she is encouraging her worshippers. For earrings she has two dead bodies and wears a necklace of skulls; her only clothing is a girdle made of dead men's hands, and her tongue protrudes from her mouth. Her eyes are red, and her face and breasts are besmeared with blood (Kumar, 2000). To Cotton and Coleman, Kali's violent symbolism seemed an appropriate icon for the types of activities they would be facilitating during networked gameplay – hacking, slashing, and creating general chaos, death and destruction. As Cotton explained when asked about the origin of the name: "Kali is the Indian (Hindu) goddess of death and destruction... and revenge. Scott Coleman thought of the name and I agreed it would be perfect for this type of service" (NWGN, 1999).

Appropriately enough, "Doom" was the first game played over Kali. In the interest of avoiding an overly metaphysical and tangential discussion, I won't delve too deeply into the symbolic import of invoking Kali as an iconic figurehead for networked gaming culture. Suffice to say, that invocation can offer some timely connections between ancient Indian spiritualism and what will prove some of the fundamental issues confronting next generation networking technologies as applied to the design of online computer gaming infrastructure. This is particularly true with regard to the relative transparency of process, knowledge discovery and its relationship to social action, the significance of temporality in the structuring of experience, and the importance of embracing difference.

For example, a guiding principle in the literature on user interface design has been to hide as much of the core structural components of a system as possible, the nuts and bolts of the infrastructure that make it run, and which are presumed to be uninteresting or irrelevant to the end user. Ironically, this paradigm sits in fairly direct opposition to the end goals that motivate much of the social sciences and the conceptual arts, where unpacking and displaying hierarchies of power, privilege and access are seen as central to the process of understanding the cultural conditions within which we operate, and are made to perform. I would also assert that the desire for transparency can become counterproductive from an interface design process as well, as it creates an ever-expanding gulf between application development and infrastructure development, which tends to prohibit players from taking part in the construction and manipulation of their experience at a deeper and more fundamental level. For instance, I don't just want to be able to swap in different graphics, sounds, or levels in my game, I want to be able to change the physics models, determine what networking protocols get used, alter the game AI, or decide which kinds of devices get which types of media sent to them under what sorts of conditions. In this sense, I not only want to be a con*tent* provider, I want to be a con*text* provider, a critical distinction that I'll elaborate on shortly. Presumably the same desire would also hold true for those in the professional development community, while simultaneously helping to blur those rigidly entrenched boundaries between the creators of a system and the "end users" of that system.

The Problem of Interoperability

Not too long ago, another story of a couple of guys pushing at the boundaries of networked gaming captured a brief but significant amount of media attention, and served to generate a firestorm of controversy within the online gaming, open source, Internet service provision, legal, and scholarly communities. The case involved release of a software system called "bnetd" that was accused of infringing on the intellectual property rights of a hugely successful game development company, "Blizzard Entertainment."

The ongoing saga raises core issues around copyright, as well as what many consider the fundamental "nature" of the Internet – interoperability, which basically is a term referring to the need to provide things that are different with a common ground for "interfacing" with each other. It's worth spending a little time detailing some of the circumstances around this particular story, as it illustrates quite clearly what some of the key intellectual and legal challenges facing game developers and players will be as networked gaming continues to grow in scope and scale. Many of the key points that follow can be credited to Howard Wen, who wrote an excellent article on this topic for Salon.com in 2002 entitled "Battle.net Goes To War."

Quick Take

The short version is simply another unfortunate case of a *big* media company – "Blizzard Entertainment" – owned by a multi-billion dollar *giant* media company – "Vivendi Universal" – hiring an expert legal team to squash the upstart little guy that wasn't even in it for profit, "bnetd." To date, the big media corporation appears to be winning (EFF, 2004). The potential outcome is a stranglehold placed on what many perceive to be a principal characteristic of the net - interoperability. It also effectively illustrates how the Digital Millennium Copyright Act (DMCA) can be used by software companies like Blizzard to wrest control of proprietary software from independent developers.

The Playing Field

"Blizzard Entertainment," located in the corporate research park adjacent to the University of California at Irvine, is a highly successful game development company and publisher. Blizzard was founded in 1991 as *Silicon & Synapse* by three partners, Mike Morhaime, Allen Adham and Frank Pearce. The company was acquired for just under $10 million in 1994. Shortly thereafter, Blizzard shipped their breakthrough hit "WarCraft." Since its initial release of WarCraft, every title released has been a best-seller, including "WarCraft II" (1995), "Diablo" (1996), "StarCraft" (1998), "Diablo II" (2000), and "WarCraft III" (2003) (Wikipedia, 2004). Vivendi Universal the parent company of Blizzard, evolved from a water utility provider into the world's third largest media company, trailing only Time Warner and Disney (Yahoo, 2004). In 2002 Vivendi had nearly 62,000 employees (a nearly 84% one year increase), and sales growth of roughly 20% (Bills, 2004).

Battle.net is a free online game arena owned by Blizzard. Battle.net provides an arena for Blizzard customers to chat, challenge opponents and initiate multiplayer games, at no cost to the user. There is currently no hourly or monthly fee to use Battle.net, and there is no startup charge. According to Blizzard, controlling where Blizzard's software titles get played allows them to do better evaluation of product usage; but even more importantly from a business standpoint, it allows them to authenticate players using a key method so that software piracy of their games is reduced.

Bnetd, the "little guy," is essentially a group of fellows in the proverbial garage who wrote some software they decided to give away to the open source software development community with no strings attached. Bnetd is a free, GPL'ed (Gnu Public Licensed) Battle.net emulation package. With bnetd a gamer is not required to use the official Battle.net site to play Blizzard games. This did not make Blizzard very happy.

Once Vivendi and Blizzard mobilized their legal front, bnetd received the generous support of the Electronic Frontier Foundation (EFF), who became their legal representatives in 2002. The EFF is fundamentally a nonprofit group of passionate people – lawyers, volunteers, and visionaries – working to protect our digital rights. Making matters even more complicated was the relatively recent passing by Congress in 1998 of the Digital Millenium Copyright Act (DMCA), a complex and controversial bill designed to bring copyright law up to date with digital media. It effectively eliminates the long-standing "fair use" exceptions to copyright. Publishers can potentially use technology to enforce where, when, and how a copyrighted work is accessed. If the law bars consumers from circumventing such technologies – even with products they have legally purchased – then the provisions of "fair use" have become meaningless (Firooznia, 2000). Needlesss to say, the stage had been set, and the EFF had a worthy challenge.

The Backstory

It all started in 1998 whe a University of California San Diego student named Mark Baysinger reverse engineered the protocol StarCraft clients used when connecting to Battle.net. In April, Baysinger posted the first version of a Battle.net emulator he called "Starhack." It didn't do much besides allow for chat, and Baysinger soon stopped working on it. However, as was the case with bnetd a bit later, Baysinger released Starhack under the GPL. In a classic example of open source software development, it got christened the "bnetd" project. In its heyday, Bnetd had 10 listed developers at Sourceforge.net (Miller, 2002). Two of those 10, Ross Combs and Rob Crittenden, became the bnetd lead developers. A third, Tim Jung became bnetd's Internet Service Provider and systems administrator.

Like tens of thousands of others, Combs and Crittenden liked to play Blizzard games on Battle.net. The problem was that Battle.net was often buggy and slow. As a result it crashed a lot due to player volume. Battle.net also had could be termed "social malfunctions" – malicious killing and unfair advantage hacks by other players, which often made the gameplay experience frustrating for newcomers, as well as those who wanted to do things besides spend all their time killing or avoiding being killed.

So, out of their frustration Combs and Crittenden decided to revive the GNU version of Starhack that Mark Baysinger created and christened it bnetd. Bnetd was essentially a Battle.net clone. The idea was that players could freely download the open source software, install it, modify it if they wished, and run their own bnetd servers for playing Blizzard, or potentially any number of other, games. Combs and Crittenden claimed that the motive was purely to facilitate fun, bug-reduced gaming sessions within a friendly community that would be respectful of each other.

The basic method for creating bnetd involved what's called packet sniffing and interception. Packets are the fundamental unit of information carriage in modern communication networks. A packet consists of a *header*, which contains the information needed to get the packet from the source to the destination, and a *data area*, which contains the information provided by the creator of the packet (Wikipedia, 2004). In its simple form a packet sniffer is a tool used for capturing all of the data packets that pass through a given network interface. Once sniffed, the information contained within the packet can be analyzed. Combs and Crittenden did this in effort to do what's called "reverse engineering" of the core Battle.net functionality so they could replicate it in bnetd. Reverse engineering is kind of like taking apart a car engine to learn how it works so that you can put it back together again, and hopefully in the process make it run better than it did before.

The problem was that, as Howard Wen put it, for Blizzard, fun wasn't the issue - copyright infringement and the promotion of piracy was. Blizzard thus leveled a number of charges against bnetd: 1) that bnetd used stolen copyrighted code; 2) that pirated copies of Blizzard games could be played on servers running bnetd, bypassing Blizzard's proprietary authentication check (what they called their anti-circumvention technology); 3) that bnetd infringed upon Blizzard's trademark, arguing that people who used bnetd may be confused and think that they are actually using a Blizzard product because the name is so similar to Battle.net; and 4) that bnetd allows what Blizzard termed "unauthorized public performances" of Blizzard's copyrighted work, i.e., its games (Wen, 2002).

The Fallout

A cease and desist was successfully issued by Blizzard in February 2002. And as of this writing, it appears that the legal wrangling is still underway (EFF, 2004). In the meantime, the bnetd.org site is nothing but a 404 – meaning it is no longer accessible. As many of the past and present architects of the Internet have argued, interoperability - enabling software to work with other software - is a core principle of how the Internet and computers work. What Blizzard's charges imply is that bnetd infringed on their copyright, since it "interoperates," or is otherwise compatible, with Blizzard's products. bnetd's counter-claim is that it doesn't infringe, it just allows both non-infringing and infringing code to run.

Funny thing is, Blizzard still profits from games sold whether used on Battle.net or not. Ironically, Blizzard appears to be suing customers for making better product servicing their own content. Moreover, under the guise of protecting IP, Blizzard has demonstrated that it can force its customers to use its software with its services alone. Thus producers could actually demand that you use product in a certain way, or risk prosecution. Clearly, in this sorry saga it appears that the right to protect intellectual property has managed to erode the rights of consumers.

As has been pointed out, the outcome has broad ramifications for the free-software and open-source communities. Open source, seeks to empower consumers, giving them the right to use, modify, and distribute on their own terms. Many of the most significant innovations in Internet technology have happened in an open source fashion. In the bnetd case, the very concept of compatibility as a feature in software and hardware gets brought into question. If it stands, it appears plausible that any company could create a mini-monopoly on network communications; thus bringing the interoperability of the Internet to a halt (Wen, 2002). In this sense, the freedom to use content whether legally acquired or not, across multiple services, is severely curtailed, forcing a structure of participation that may very well happen at the expense of the majority, and in the interest of an increasingly homogeneous, controlling and vertically integrated minority.

The Need for Heterogeneity

As networks expand, both technologically and socially, there is going to be a growing need to embrace diversity, even when it appears to run counter to the dominant software and social engineering paradigms that perceive an increasing need for assimilation and standards compliance. From a networking standpoint, as well as a cultural standpoint, nothing poses a bigger challenge for a whole host of reasons, not the least of which has to do with maintaining control over circumscribed political and economic boundaries in light of increasingly global competitive capital flows, and the desire to enforce technical, linguistic and cultural uniformity, and hence conformity, on an international scale.

I would assert that the future success of network games is in their ability to run in heterogeneous environments that combine a range of devices, such as cell phones, PDAs, desktop and WIFI mobile computers, handheld game devices and game consoles into continuous and contiguous multi-player game experiences. A fundamental challenge in the transition to increasingly networked gaming has been how to synchronously communicate or represent the actions of players to each other in effort to ensure that everything was happening at the same time on all computers involved, in spite of network latency, or differences in machine speed. However, this concern over synchronicity, manifest in the desire to create a consensually coherent reality, only begins to address one small piece of the much larger puzzle around where networked gaming environments are heading. Other important pieces of the puzzle involve developing compelling strategies for how best to take advantage of pervasive and ubiquitous technologies like phone, fax, email, and the Web as part of an overall game environment in ways that make sense given the capabilities of the devices. It also involves thinking about how to design compelling social interaction through that multiplicity of devices, how to open up authorial control to allow players to have more open-ended and flexible play spaces, how to allow players to modify those play spaces within an existing domain, and how to provide customizable services enabling them to build entirely new play spaces if so desired.

The Case of Majestic

The most compelling recent example of a game that was designed from the ground up as a network-centric experience is Majestic by Electronic Arts (released in Summer of 2001). Majestic's marketing hook was the tagline that "the game played you." Conceived as an episodic Net-based adventure, players created accounts, logged in, downloaded the first installment for free, and then were charged for subsequent installments if they chose to continue. Majestic was spawned in the government conspiracy and cover-up genre, and incorporated a number of complicated and convoluted plot twists and turns, experienced through a variety of interfaces.

The brilliant conceptual move made by the designers was to design the game in such a way that it could take meaningful and contextually specific advantage of everyday communication devices like the Internet, phones, email, instant messaging, and FAX to create a narrative experience that blurred the line between lived space and game space. Upon registering players were able to determine the parameters of in-game communication. But in theory, depending upon what sort of access was granted to the game, players could be contacted at any point during the day or night by game operatives, either giving them vital pieces of information to aid in moving them to the next stage in the drama, or providing them with misinformation in effort to take them off track. The other big idea was that the episodic structure would appeal to an older generation of gamers that didn't have time to sit in front of their computers for hours on end, but wanted to periodically drop in and spend an hour or so to try and decipher a clue, or check up on who had been attempting to contact them.

According to scattered bulletin board postings, random online articles, and some personal conversation I had with the game's brainchild and producer Neil Young (a VP at Electronic Arts and General Manager at Maxis) in late Fall of 2003, Majestic failed for a couple main reasons. The public claim is that the subject matter of the game became too controversial after the World Trade Center bombings and the subsequent War on Terrorism. Interestingly, and just as an aside, Young was very reluctant to even talk about Majestic, feeling it was a failed project. He was far more eager to discuss the recent "Return of the King" title he'd just produced. Our conversation quickly petered out and I was promptly shown the elevator when I said I believed history would prove Majestic a far more significant contribution to game culture and technology than any of the Lord of the Rings titles.

The other part of the failed Majestic story, from my perspective, has to do with the way that commercial concerns over people freaking out if family members answered phones or received faxes from strange and unsavory characters claiming that their loved ones lives were at stake prevented the game from really pushing more intensively at the lived space/game space boundary. It became "too safe" due to all the disclaimers and legal protections necessary to make it palatable to a general public.

From an infrastructure standpoint, what Majestic attempted to do, on the one hand, was not, according to Young, particularly complex – though the same could be said about many of the more compelling scientific and technical innovations. What made it difficult, on the other hand, was dealing with all the proprietary issues and third-party companies that got in the way of what should have been relatively simple integration. Doing voice over IP, instant messaging, and sending faxes all had to be worked out with service and technology providers. Negotiating such partnerships was no small task, and required considerable financial investment, reportedly in the neighborhood of $10 million US dollars, and several years in the planning and implementation.

The end result of all this time and investment is what the developers called "the experience server," which consisted of a robust (Oracle Enterprise class) database server, Web servers (running on Sun Solaris), and a handful of BEE WebLogic application servers. According to Rich Moore, Majestic's director of technology, "those three major components – the database, the Web servers, and the JAVA server code – implement what is Majestic." (Bonsor, 2001). In addition to an Internet connection, a computer and a Web browser, to be fully experienced Majestic required a custom application download, Flash, RealPLayer, AOL, pop-mail, a phone, and a FAX machine.

Working with Infrastructure Versus Reworking Infrastructure

The current network milieu is still one of fragmentation and divergence – continually increasing types of devices are connected to through separate and discrete networks. In the future, all network devices can and should be able to interact with one another through a single network infrastructure that can deliver a contiguous experience that is optimized for the unique properties of each device. Such an infrastructure, or "engine" if you will, would enable players to participate in the same game world from whatever device they chose, while maintaining persistent identities and accounts – allowing them to easily carry their game personas, assets and social networks between platforms, to scale media in sensible ways, and to have anywhere anytime access to their shared communities of interest. Creating such an infrastructure, particularly if freely available, and able to circumvent the current necessity of third party partnering and exorbitant service costs to take advantage of basic communication modes, would be a huge step in a promising direction. Implementing that infrastructure so that it is relatively easily customizable would be a giant leap in an even better direction, as it puts content and context creation back into the hands, hearts, and minds of a far more diverse, and potentially innovative, population of "players."

To date, the experience server infrastructure and the game application that ran on top of it, in my mind, most closely approximates the networked game engine of the future. But equally important is how the game application was developed directly in tandem with the "engine" that drove it, and how it took conceptually appropriate advantage of the various communication devices and platforms that created the overall experience – in ways that made sense given the distinctive qualities those platforms possessed.

Majestic still represents the most heterogeneous online gaming environment yet attempted, though some of the research and development in heterogeneous networking currently underway in the Game Culture and Technology Laboratory which I direct at UC Irvine, and which is currently sponsored by Sun Microsystems, hopes to change a bit of that.

There are different levels at which creative work happens. If you think about work methodologies in the context of Net-art practices, an important distinction can be made between work that is made using available tools and technologies (i.e., pre-coded software applications such as image, sound, and video editors, word processors, 3D modeling programs and the like), and work that is made by retooling the tools, or by mucking around with the infrastructure that the tools run on (like custom-coded Web browsers, image processors, sound generators and so on).

As already alluded to, another way of making this distinction is to frame it in terms of con*tent* provision versus con*text* provision. In the first case, artists could be seen as providing content for and working with existing infrastructure, while in the second case they could be seen as reworking that infrastructure, or providing new infrastructure, in the interest of creating alternative contexts for interaction and experience. In this interest, hacking, cracking, poaching, sampling, mixing, appropriating, misusing, abusing, reverse engineering are all legitimate creative methodologies that in the history of technology and the arts have at best led to significant innovation, and at worst been a lot of fun to witness and participate in.

In a nutshell, what's at stake is turning over ownership and control of the game environment, not just the game application, in order to more explicitly expose the tools of creation so that they also can become a primary place of play. Things like Modding (using commercial game products to create custom levels in existing titles) and Machinima (using 3D game engines as real-time movie-making platforms) signal important steps in this direction. What we need to be consistently doing is figuring out how to promote those strategies in a distributed networking environment.

Into the Aether

To take just one sector of the games industry, mobile phones, for example, things are looking pretty bright. Games are already the number one downloaded application on mobile devices, representing upwards of 90% of total download requests. According to Datamonitor analysts, by 2005 more than 200 million people in the United States and Western Europe – 80 percent of all wireless phone users – will play online games using wireless devices. Four of the major mobile phone manufactures, Nokia, Siemens, Ericsson and Motorola, recently set up the Mobile Games Interoperability Forum (now consolidated into the Open Mobile Alliance, or OMA), which aims to define an open-standards mobile games interoperability specification that will let game developers create and deploy products that can be distributed across multiple game servers and wireless networks to be played using a variety of mobile devices. The forum represents a joint commitment by the companies to the mobile games industry as a whole and to their work on creating APIs for game developers (Wrolstad, 2001).

The OMA represents a significant and very positive move toward interoperability in at least one sector of the industry. Unfortunately, other sectors of the industry aren't quite as forward thinking. The leading console makers each have their own proprietary game development suites and protocols, as do personal computer manufacturers, hand-held device manufacturers, other mobile phone providers, etc, etc. The list could go on and on. Making these things talk to each other in meaningful ways is no small task. At the moment, the only thing this ever-expanding multitude of devices *will* share is an IP address. What's needed is a framework for creating seamless interoperability between devices in contiguous and integrated gaming experiences, enabling network access at anytime, with anyone, anywhere. Unfortunately, this is not likely to emerge from within an increasingly competitive corporate climate anytime soon. However, it is precisely for this reason that academia may perhaps be able to play an important role.

Anywhere anytime access is the mantra of Cal-(IT)2, the California Institute for Telecommunications and Information Technology, one of four newly established (proposed in 2000, and established in 2001) California Institutes for science and innovation. Cal-(IT)2 is largely the brainchild of Larry Smarr, an astrophysicist by training and now a Professor of Computer Science and Engineering at UCSD. Smarr is renowned for founding the National Center for Supercomputing Application (NCSA), which made major contributions to the development of the Internet, the Web, the Access Grid, and scientific visualization.

Cal-(IT)2 is a partnership between the University of California San Diego (UCSD) and the University of California Irvine (UCI) campuses. Cal-(IT)2 receives several hundred million dollars in funding from a combination of state and federal government sources and industry partnerships. The ideas is to extend the reach of IT throughout the physical world to enable anytime/anywhere access, and to complement R&D in related IT sectors in effort to help provide new capabilities to important market segments poised to be transformed by the new Internet. Well over 100 faculty are involved in the institute. New buildings and a campus-wide broadband infrastructure are under construction at both the UCSD and UCI locations for prototyping purposes. Cal-(IT)2 is divided into distinctive layers – Environment and Civil Infrastructure, Intelligent Transportation, Digitally Enabled Genomic Medicine, and New Media Arts (Cal-(IT)2, 2004). The vision was to integrate the arts and humanities, as well as the social sciences into the core of the institute. This was done because it was recognized that involving the arts and humanities at the beginning phase of technological implementation can play an important part in helping design systems in a more self-reflexive social context.

Within the New Media Arts layer of the institute sits the UC Game Culture and Technology Laboratory, a facility designed to support faculty, graduate, and undergraduate research into games and game culture, and to help lay the foundation for the emergence of gaming studies as a legitimate academic field of study. Also within the New Media Arts layer, and working in partnership with the Game Lab, are the Advanced Visualization Lab (which focuses on research in visualization, virtual reality, geometric modeling, computer graphics, simulation based design), and the Action Lab (which explores new installation, performance and interface paradigms).

The Game Lab is an open source development environment comprised of an eclectic and interdisciplinary community interested in using game metaphors, design principles and technologies for creating and analyzing next generation multiuser work and play environments. It is concerned with issues of cultural and artistic production, software and hardware design and implementation, and advancing the practice of game-making and critique with an emphasis on multi-person, networked interactivity. A core premise of the lab is that appropriation, misuse, and hacking of technologies is not only a legitimate form of R&D, but is central to the process of innovation.

Recently the lab established a partnership with the San Diego Supercomputer Center and Butterfly.net (a third-party software provider) to implement a game research test grid for experimental purposes. The grid is a networking infrastructure research environment designed to provide a real-world context for looking at grid-based computing in massively multi-user applications. The lab is also in partnership with Sun Microsystems on several projects. The most relevant to the discussion here is the heterogeneous networking project, which is being developed in conjunction with a group of artists, engineers, computer scientists and theorists in effort to start chipping away at some of the problems around interoperability and heterogeneity as discussed this essay.

The first prototype project coming out of the lab exploring fundamental issues around the idea of application and infrastructure design in heterogeneous and distributed computing environments is one of my own entitled "unexceptional.net." With unexceptional.net the goal is to build an arts-driven game project that demonstrates pervasive, multi-modal interoperability between various client devices. Currently we have a system prototyped that supports bi-directional communication between location-aware GPS phones and the Web. This means players can be doing things that change game-state in Web-space and have it reflected for players using mobile phones, and vice-versa. We're in the process of extending this to a Grid system running at the San Diego Supercomputer Center that can theoretically scale well beyond one million concurrent users. The Grid will initially run a client version of unexceptional.net as a 3D MMORPG (Massively Multiuser Online Role Playing Game). However, and this is the salient point, the Grid will also allow players in unexceptional.net to connect to each other on the Grid via GPS phones and Web portals. In my mind, what we're doing with unexceptional.net, is starting to lay a foundation for a modular, flexible, and highly extensible online gaming "engine." Toward that goal, as much as possible, all the low-level capabilities we develop are not only open-sourced, but can be exposed, manipulated, and modified by ourselves, and eventually others, through easily accessible Browser-based administrative and creative interfaces.

In an interesting and timely article, author and game developer Crosbie Fitch playfully makes a pitch for a new protocol (used loosely) that he terms the "Distributed Internet Operating System" or "DIOS." From Fitch's perspective, DIOS will facilitate the equitable pooling and exploitation of all resources and information around the world. According to Fitch the most suitable candidate for a DIOS is the field of interactive entertainment technology, specifically massive multiplayer games and the engines that support them. He asserts that the ones to survive will be scalable, distributed systems, and that it is likely we would see the entertainment industry making more intensive use of DIOS than financial, military, and other industries. For Fitch this all adds up to the conclusion that massive multiplayer games are not only where games are going, and not only will they be the primary form of entertainment for this century, but they will also end up defining the future of the Internet itself (Fitch, 2000). In a nutshell, this is what we believe in the Game Culture and Technology Lab as well.

We must expand the notion of what constitutes the networked game "engine" to include not only hardware and software infrastructure, but also the interpersonal and culturally driven social networks that emerge in relation to the applications written for that infrastructure. At the same time we need to recognize the crucial importance of heterogeneity and access to and ability to create and to play, not simply at the level of the application – the content – but at the level of the infrastructure – the context – a kind of "leveling *down*" if you will.

One way or another, things are bound to get interesting.

References

BBC News. 2004. "Online games make serious money." Internet Publication. http://news.bbc.co.uk/1/hi/technology/3403605.stm

Bills, Larry. 2004. "Vivendi Universal S.A." Hoover's Online: The Business Information Authority. http://premium.hoovers.com/subscribe/co/factsheet.xhtml?COID=92356

Bonsor, Kevin. 2001. "How Majestic Works." How Stuff Works. http://entertainment.howstuffworks.com/majestic3.htm

California Institute for Telecommunications and Information Technology (Cal-(IT)2). 2004. http://www.calit2.net/

Cotton, Jay. 1994. "History of Kali & iFrag." http://games.igateway.net/kali/history.html

Electronic Frontier Foundation. 2004. Blizzard v. bnetd Archive. http://www.eff.org/IP/Emulation/Blizzard_v_bnetd/

European and US Internet Usage. 2000. http://216.239.53.104/search?q=cache:FctkUqXwqoAJ:banners.noticiasdot.com/termometro/boletines/docs/audiencias/netvalue/2000/10_2000/NetValue_001000_europa-usa.pdf+internet+usage+usa&hl=en&start=5&ie=UTF-8

Firooznia, Hoss. 2000. Hackers vs. Hollywood, round 1: A very brief history of DeCSS and the DMCA. Lodestone.org. http://www.lodestone.org/people/hoss/politics/DeCSS-history.html

Fitch, Crosbie. 2000. Cyberspace in the 21st Century: Mapping the Future of Massive Multiplayer Games. Gamasutra. http://www.gamasutra.com/features/20000120/fitch_01.htm

Garton, Laura, Caroline Haythornthwaite, Barry Wellman. 1997. "Studying Online Social Networks." http://www.ascusc.org/jcmc/vol3/issue1/garton.html#What%20is%20Social%20Network%20Analysis

Graetz, J.M. 1981. "The origin of Spacewar." This article is reprinted from the August, 1981 issue of *Creative Computing* magazine. Copyright © 1981 by Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. Sample issue $2.50, 12-issue subscription $20. http://www.wheels.org/spacewar/creative/SpacewarOrigin.html

Gruener, Wolfgang. 2004. "Internet penetration in the US passes 75 percent." http://www.tomshardware.com/hardnews/20040319_130922.html

Interactive Digital Software Association (IDSA). 2003. Media Center. http://www.idsa.com/pressroom.html

"Introduction to Replication." The Unreal Engine Documentation Site. 2004. Uncredited. http://wiki.beyondunreal.com/wiki/Replication

Krebs, Valdis. 2004. An Introduction to Social Network Analysis. http://www.orgnet.com/sna.html

Kumar, Nitin. 2000. "Mother Goddess as Kali - The Feminine Force in Indian Art." Exotic India. http://www.exoticindiaart.com/kali.htm

Lea, M, T. O'Shea, and P. Fung (1995) "Constructing the Networked Organization: Content and Context in the Development of Electronic Communications," *Organization Science*, 6:4, 462-478.

Miller, Ernest. 2002. "Analysis of BNETD and Blizzard." LawMeme at Yale Law School. http://research.yale.edu/lawmeme/modules.php?name=News&file=article&sid=149

NWGN Interview. 1999. Nuclear Winter Gaming Network. Uncredited.
http://www.nukewinter.com/articles/december1999/kali-interview.shtml

UC Game Culture and Technology Laboratory. 2004.
http://www.ucgamelab.net

Wen, Howard. 2002. "Battle.net Goes To War." Technology and Business. Salon.com.
http://www.salon.com/tech/feature/2002/04/18/bnetd/

Wikipedia. 2004. Blizzard Entertainment. Online Encyclopedia.
http://en.wikipedia.org/wiki/Blizzard_Entertainment

Wikipedia. 2004. Packet. Online Encyclopedia.
http://en.wikipedia.org/wiki/Packet

Wireless Internet. 2001. "Strategy Analytics Forecasts $5 Billion Wireless Gaming Market by 2006."
http://www.findarticles.com/cf_dls/m0IGV/7_3/77379482/p1/article.jhtml

Wolf, Cindy. 2003. "Forecast and Analysis of the Worldwide Internet Access Device Market, 2002-2007."
http://ealibrary.bitpipe.com/detail/RES/1070672414_890.html

Wrolstad, Jay, 2001. "Forum Pushes Open Standards for Next-Generation Wireless Gaming."
http://www.newsfactor.com/perl/story/11746.html

Yahoo Finance. 2004. "Vivendi Universal S.A. Company Profile."
http://biz.yahoo.com/ic/92/92356.html