They Must First be Imagined

Katie Salen

Game Set and Match II. *On Computer Games, Advanced Geometries and Digital Technologies.*
Kas Oosterhuis and Lukas Feireiss (ed.) Episode Publishers, March 2006.

If you can't play it, change it. If it helps, cheat.—Bernie DeKoven

When posed, the question seemed simple: What is the possible role of computer game design for research, design, and practice in architecture? When contemplated, the response—my response—seemed entirely too vast. As a game designer, I am well familiar with the issue of complexity, of the ways in which the rules of a game system combine and interact to produce surprising, and often unintended results. Players, too, contribute to these emergent effects, gaming the game by interpreting, rewriting, and transgressing rules. So when faced with the complexity of my own response to a seemingly simple question, I decided to return to my roots as a player and speak to the *experience* of games as a starting point for an inquiry into the themes of Game, Set, and Match.

My hope in this paper is to develop, however slowly, a meaningful model for ways that architects and game designers alike can identify points of intersection across disciplines, and to discover possible modes of entry into a discourse with collaboration and innovation at their core. Tactically speaking, I will move from the poetic to the practical, and will continually return to the theme of games as contested spaces, or spaces in which the negotiation of conflict drives strategic choice, movement, narrative, and reward. The concept of contested space has a long history within architecture and urban design, is inherently tied to socio-economic and political controls, and creates a bridge between the experience of games and that of architecture. Similarly, I will explore games' status as dynamic systems in a discussion of procedurally generated content, an emerging paradigm within digital games, which has significant potential to impact current architectural research and practice. There is not room to go into full detail about many of the concepts covered here, but resources have been included at the end for anyone interested in further study.

The image, the imagined, the imagination

Chrilly Donninger professes to know much less about the game of chess than Hydra, the computer chess program he wrote that is now regarded as the top player in the world. Hydra and programs like it with names like Shredder, Junior, and Fritz, have thrown into relief startling questions about the fundamental nature of play and the role technology can play in identifying new models of creativity. "They [chess programs] make a lot of counterintuitive, even absurd-looking, moves that on closer inspection can turn out to be outrageously creative," says international master John Watson.[1] Innovation, it seems, lies in the ability of the game program to go beyond hard calculation, into the realm of intuition, long considered exclusively human. Hydra, as both player and computer, elegantly illuminates the paradox of play: often simple rule-sets can lead to complex outcomes. While one can describe all the rules, one cannot describe all the products of the rules, particularly how one chooses to play.

Games are not only spaces of strategic possibility—places to battle, puzzle, explore, and socialize, but are also spaces of the imagination, of things both real and make believe. The status of games as representational systems that create spatialized, imaginary worlds for players to engage with, is one obvious tie to the concepts of Game, Set, Match:[2] games create contexts for interaction and are modified by this interaction within the limits of the system. Yet if we are to look more closely at games as imagined spaces of possibility, we soon find ourselves in a situation described by cultural anthropologist Arjun Appadurai, in a study of the cultural dimensions of globalization. Appadurai writes,

> No longer mere fantasy…no longer simple escape, no longer elite pastime…and no longer mere contemplation…the imagination has become an organized field of social practices, a form of work…and a form of negotiation between sites of agency (individuals) and globally defined fields of possibility.[3]

The imagination as conceived by Appadurai, is a site of contestation linking the individual to a transnational construction of imaginary landscapes made possible through the complex prism of modern media. According to Appadurai, the world we live in today is characterized by a new role for the imagination in social life. Citizens are no longer bound to local representations of possible

futures—we only have to see it, to begin to imagine ourselves within it. As such, the imagination is now central to all forms of agency.[4] Play, too, is a mechanic of this action. As an act of the physical, technological, and social imagination, play is a model of being with agency at its core. The negotiation of one's position within the imagined spaces of games, or architecture, can therefore be understood as both an important form of social practice, and a creative act. The creativity of player-programs like Hydra comes not only from intelligence derived via extraordinary computational power, but from an almost sublime ability to imagine and accept the improbable as possible through the act of play.

The imagination then, becomes a critical tool in relation to a central feature of games: their status as systems of conflict. Conflict arises naturally from interaction in a game as players attempt to overcome obstacles to achieve a goal. Conflict can be direct or indirect, violent or non-violent, and may require players to compete or collaborate (often in combination) against the game itself. Consider the similarities between the conflict modeled within a public space like Shibuya Crossing and that of a game like World of Warcraft. Anyone who has found him or herself in either space understands the complex dance of decision required to arrive safely on the other side.

Shibuya Crossing

World of Warcraft

The contested space of games is not only a product of the game's rules, but of its system of representation as well. The elements of a game—the players, the pieces, the rules—all have a role in generating this representation. The key to comprehending the form of conflict modeled by a game is to figure out what is being contested. In what kind of arena is conflict being held: a battlefield, a social network, or the space of language? Over what is the conflict being waged: territory, economics, knowledge, or social status? How is the conflict measured? Through frag count, inventories, high scores, or reputation? Conflict is a way games organize agency via the field of social practices that make up their play. How might architects conceive of "place-making" as sites of contestation enabling practices of social imagination? How might the form and body of architecture be translated through the rules of strategic conflict into spaces that evoke interaction, be this material, social, or psychological? The answers to these questions, while unknown, may provide points of entry in to the kinds of collaborations that are possible in the intersections between game, set, and match.

Negotiating Play

As contests of power, games model many forms of conflict, and the play of a game is in some sense, an act of *negotiation*. Players negotiate terrain, as they move through space. They negotiate representation as they parse the clues embedded within a level to develop a system of understanding about what things mean, and what elements can help or hinder their progress. Finally, players negotiate knowledge of the real vs. the make believe in order to act within the imaginary landscapes they inhabit, working to reconcile their own motivations with the stated objectives of the game.

For all of these reasons, game behavior is impossible to predict. Players do things they are not supposed to do. They are transgressive. They break rules, cause grief, and often behave very, very badly. But they also are wonderfully inventive and surprisingly generous. They share knowledge with new players. They build tools for each other, create forums, and often compete in fair and honorable ways. Usually, players do all of this without rules explicitly demanding that they do so.

These collective actions occur because the players are part of a game community, a group of individuals who all buy into a shared desire to play together. This negotiation of experience is critical to each and every moment of their experience, as users impose their agency upon a system that responds through rule-sets designed to limit their play.

In order to get to the point where social agents can act as *players* within such systems, form communities, and engender collective behavior, the rules must be defined. Game rules are directly concerned with the actions players take and the outcome of those actions. In digital games, the formal game logic (such as the way a program selects the next block to appear in Tetris) is also part of the rules. As Lev Manovich notes, "All new media objects, whether created from scratch on computers or converted from analog media sources, are composed of digital code; they are numerical representations.... A new media object can be described formally (mathematically)."[5] Noting the formal quality of game rules is important, for like architecture it is the structure underlying any material expression that makes the play of the system possible. Rules, then, particularly those enacted within computational systems, become a second point of entry into the collaborative space we seek. How might the design of rule sets be conceived to take advantage of the real-time nature of dynamic systems like games and architecture? What would it mean to generate play?

Code and Creativity

> *The system is partly a memory of its past, just as in origami, the essence of a bird or a horse is both in the nature and order of folds made. The question that must be answered when faced with a problem of planning or design of a system, is what exactly is the system? It is therefore necessary to know the nature of the inner structure before plans can be made.—Wolfgang Jonas*

The history of videogames is partially a story of the collision between computers and game players, and for many years, was a story written by programmers. Early games, including Spacewar!, Zork, Tetris, and Adventure, were designed either by a solitary programmer working out his own obsessions onscreen, or by groups of programmers collaborating to modify one another's code. In this early environment, the distinction between players and designers was often moot—games were made in order to exploit the limits of the programmers' own knowledge and the capabilities of the machine.

While much has changed in terms of the development process—video games today are no longer an act of individual authorship but the result of a coordinated effort of a large team of specialists—the residue of this history remains. We see it in many themes of this conference—real-time technologies, parametric design, generative design, genetic algorithms, and what I would call procedural programming: computationally generated content. We see it too, in games like Monster Rancher, a battle game between monsters that are bred from the data of a player's CD/DVD collection. Monster Rancher uses a serial number encoding system, which identifies any CD or DVD signature and changes it into a visual and statistical representation (i.e. a monster). The monsters that are generated can then be used in battles, and retain unique characteristics derived from the source data of the original disc. This process generates a huge number of different creatures and has resulted in many a kid riffling through their parent's music collection in search of monster DNA.

Monster Rancher

We see it in software development of products like Processing, an open source programming language and environment created by Ben Fry and Casey Reas. Processing is a programming language that allows for the dynamic manipulation of images, animation, and sound. It has been widely used as a platform for developing and exhibiting generative art—where a system of rules, defined in code, interacts with a degree of randomness and autonomy to create a complex output. Many designers have also taken up Processing as a prototyping and visualization aid—a kind of software sketchbook and professional production tool. ART+COM, a company made up of designers, scientists, artists, and technicians currently use Processing as a key part of their process, modeling computational animations which are later integrated into architectural forms. Architect-turned-new media designer Evan Roth recently employed Processing in the design of customized software that translated the gesture of graffiti writers into 3D models, which could then be projected, and manipulated in real-time. The movement from street graphics to data model to animation in Roth's work shows the way that any information, once it is converted into code, can take on a range of expressions. Together these projects show the kind of radical interdisciplinary practices emerging from the application of procedural thinking to the design of inhabitable environments.

floating.numbers

Garden, ART+COM

Graffiti Analysis, Evan Roth

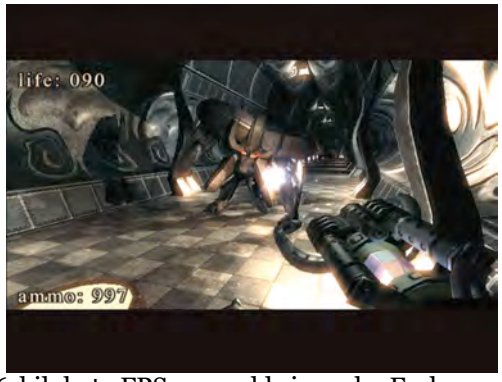
Last, and perhaps most importantly, we see it in a paradigm shift in the way digital games are being made. Sparked by Spore, the newest project from publishing giant Electronic Arts and game designer Wil Wright, this shift is away from content that is pre-rendered toward that which is authored dynamically. The shift takes advantage of the real-time status of games, and the fact that the representation of the game space must accommodate the moment-to-moment choices of players: games are spaces of possibility that respond to the actions of players in real-time. Even though the practice of architecture, as it conventionally relates to building design, is in that respect limited by materiality, one can find examples in the history of architecture where game-playing and toy-playing influenced the practice of architecture directly. For example, noted early 20th century architects—such as F.L. Wright, Buckminster Fuller, and Le Corbusier—influenced by Friedrich Froebel's theories, had been obsessed with creating toy-like systems that were then translated into building concepts. On the other hand, one can imagine a more aggressive parallel that would orient the practice of architecture towards understanding buildings as contexts for user interaction, or "sandboxes," which create contexts for user creativity. Dynamically authored spaces borne from the collaboration between architects and users is an idea with historical precedent. Early works of Lawrence and Ann Halprin in the 1960s, for example, the work of Behaviorists, the Design Methods group, and Christopher Alexander, point t a possibility of configuring the practice of architecture. To what extent architecture can parallel game design in terms of including players in such processes of technological reasoning will depend on a myriad of variables, physical as well as socio-cultural and political.

As the designer behind such games as SimCity and SimEarth, Wil Wright has long been obsessed with creating such sandboxes, or systems that are more like toys and less like games. The Sims was one of the first of his titles to acknowledge that players enjoy making their own content, and that this process could be made simpler through the inclusion of in-game editors. Character and level editors in games as diverse as Half-Life, World of Warcraft, Roller Coaster Tycoon, and Second Life have since led players to establish an incredibly rich culture of production that Wright believes is the key to a new model of game development predicated on user created content. The character editors in Spore automate the process of modeling, skinning, and animating a creature, allowing players to customize to their heart's content. Players produce and exchange content that is dynamically integrated back into the game, leading to greater and greater forms of emergent complexity. And because every creature in the game is defined and animated using procedural techniques, the data files are small, around 1kB each. This makes storing, exchanging, uploading, and sharing content easy. When things are easy, Wright found, players are willing to do even more.

The Demoscene

In order to understand the revolutionary nature of Wright's thinking, we must travel to Scandinavia, where a computer subculture known as the demoscene traces its roots back to the 1980's and the era of 8-bit Ataris. Coders in the demoscene create a lot with very little—a fully animated 3D environment can be as small as 64kB (think floppy disk)—producing complicated computer graphics directly from lines of code. A method known as procedural programming, it allows for sound, textures, and graphics to be created in real-time, through the use of generative algorithms.

Farbrausch, a German demoscene group, has been one of the most influential groups to emerge from the culture, along with the Finnish group Future Crew, who founded Remedy Entertainment, known for the Max Payne series of games. .theprodukkt's (a subgroup of Farbrausch) 96kb FPS game, .kkrieger, owes its small file size to the use of procedurally generated content and generative algorithms popular among veterans of the demoscene, but largely unknown outside this community. .werkkzeug is an authoring application created by Farbrausch, which applies procedural methods to graphics. With .werkkzeug, one is able to create textures, 3D meshes, camera-movement, animation, and sound synchronization. Because of the introduction of speed and efficiency into the design process, .werkkzeug along with the tools kkrunch and lekktor,[6] which help with the compression of code, may be a valuable resource set for those interested in the design of dynamically authored spaces. The tools created by these demosceners traffic in the language of procedural content and allow for a range of generative concepts to be quickly explored. Rapid prototyping is a method already used by practitioners from both games and architecture; the procedural capacity of programming therefore provides a likely bridge between the two.



96-kilobyte FPS game .kkrieger by Farbrausch

.werkkzeug1, editing tool by Farbrausch

Connecting Practices

Unlike in architecture, technology *always* creates a context for digital game design and it is often the limits of technology, rather than its promise of possibility, which drives innovation. When Warren Robinett sat down in 1979 to write Adventure for the Atari 2600, the first graphical adventure game, he not only spawned a new genre but demonstrated that deeply meaningful game play could be born from technological constraint. The single button at the base of the Atari joystick severely limited player interaction: rather than storing objects in an inventory and toggling between items (the standard for any game designed today) players could only carry one object at a time. As Robinett notes, "The limitation of being able to carry only one object gave the player some interesting strategic choices: which object should he carry—the treasure or the weapon?" This choice, as it turns out, was the only choice that mattered, spawning hundreds of hours of game play from players intent on mastering its form.

Thus technology conceived as both enabler and constraint can provide a powerful context for considering the role of computer game design for research, design, and practice in architecture. Good game designers, just like outstanding architects, are masters of the craft of interactivity, of seducing players to interact and getting them to stay in the game, of designing systems bursting with strategic choice and agency, by creating an innate sense of control over the system in their players/users/inhabitants. Digital game designers, just like architects, are also fluent in using technology as a tool to shift their frame of reference, as in the case of Monster Rancher or .kkrieger. In the case of games like Spore, technological innovation is a step toward the design of opportunities for players to become designers too. Taken in sum, these games point to procedural programming as a paradigm poised to change the way interactive experiences from games to buildings, are designed and conceived. Whether procedural programming, when applied in the realm of conventional architecture, will produce algorithmic predictability of a kind one finds in Walt Disney theme parks or the work of Jon Jerde Partnership, remains a question of modality rather than ethics. What games and architecture de facto share in that respect is, as Clive Dilnot puts it, "the discovery of configurative possibilities" that in order to be realized first have to be imagined. It is through such an act of social and technological imagination that both propose alternative "models of being".

As computer game-worlds continue to evolve, they are pushed not only by advances in technology, but also by the players themselves. Players are constructing, inhabiting, and negotiating complex spaces of the imagination. The contested spaces of games have a true affinity to the forms of agency afforded by architecture. Negotiating the play of these spaces requires a delicate balance between design and technology, form and function, the practical and the poetic—all issues that have been at the core of architectural discourses for centuries. The design of experiences is above all the design of contested spaces of possibility and the imagination. How we work (together) to define the nature, quality and accessibility (both physical and social) of these spaces is still to be seen, but as a chess program named Hydra has shown, we can begin first by imagining all that is possible.

Notes

1 Mueller, Tom. "Your Move. How Computer Chess Programs are Changing the Game." *The New Yorker,* December 12, 2005.

2 Architecture becomes a *game* played by its users. By playing the game, users *set* the parameters. New configurations are then *matched* to the desired conditions. (Kas Oosterhuis)

3 Appadurai, Arjun. *Modernity at Large: Cultural Dimensions of Globalization*. Public Worlds, Volume 1. Minneapolis: The University of Minnesota Press, 1996.

4 *Ibid*.

5 Manovich, Lev. *The Language of New Media*. Cambridge: MIT Press, 2001.

6 kkrunchy, is .theprodukkt's own executable packer, which generates a comprehensive report of the quality of compressibility of different pieces of code. lekktor is a small tool that can, very rudimentarily, parse C++ source code and output an instrumented version. It can be used to find out which code passages have been executed, and automatically remove everything that hasn't been executed.

7 Dilnot, Clive. *Scapes,* Number 4, Fall 2005, p. 6.

Many thanks to Miodrag Mitrasinovic and Douglas Diaz, for their invaluable comments on this paper.

Resources

The Sims: http://thesims.ea.com
Spore: http://spore.ea.com/
Demoscene: http://www.scene.org/
Processing: www.processing.org
http://www.artcom.de