# Design Research

## METHODS AND PERSPECTIVES

Play as Research

The Iterative Design Process

ERIC ZIMMERMAN

**Brenda Laurel, editor**

The MIT Press
Cambridge, Massachusetts
London, England

Cover and book design: The Offices of Anne Burdick, Los Angeles
Designers: Anne Burdick, Stuart Smith, and Megan McGinley
Cover photo: © 2003 Susan L. Burdick

Library of Congress Cataloging-in-Publication Data
Design research : methods and perspectives / [edited by] Brenda Laurel.
p. cm.
Includes bibliographical references and index.
ISBN 0-262-12263-4 (hc : alk. paper)
1. Engineering design. I. Laurel, Brenda.
TA174.D483 2003
745.2–dc22 200359969
10 9 8 7 6 5 4 3

# Play as Research

## The Iterative Design Process

ERIC ZIMMERMAN

### Needs and Pleasures

Design is a way to ask questions. Design *Research*, when it occurs through the practice of design itself, is a way to ask larger questions beyond the limited scope of a particular design problem. When Design Research is integrated into the design process, new and unexpected questions emerge directly from the act of design. This chapter outlines one such research design methodology—the iterative design process—using three recent game projects with which I have been involved.

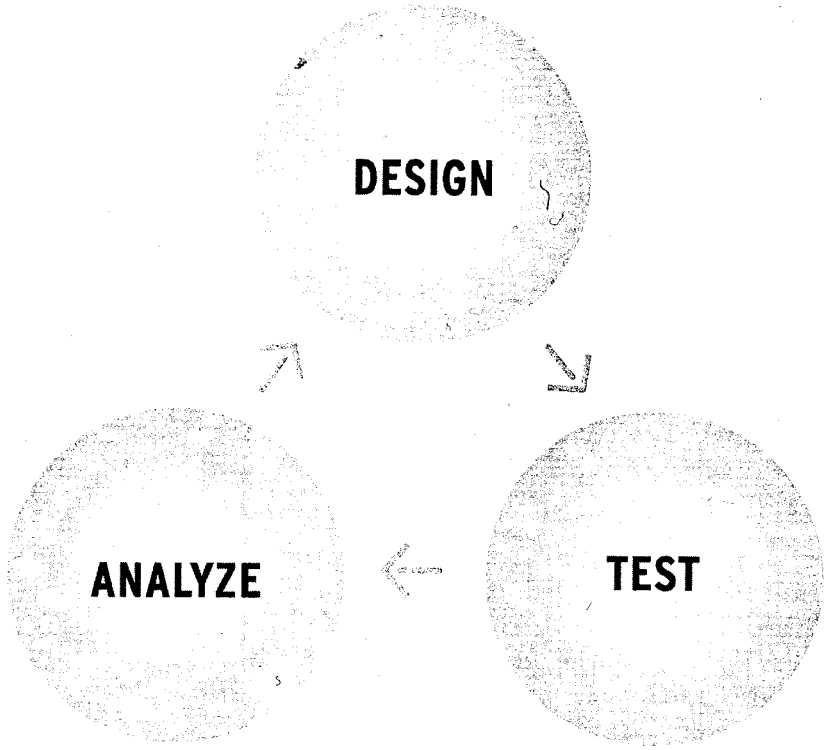
The creation of games is particularly well suited to provide a model of research through design. In this book's conclusion, Brenda Laurel alludes to the difference between designing to meet needs and designing "for delight" ©316 LAUREL. While all forms of design partake of both of these categories in some measure, game design is particularly skewed toward the creation of delightful experience, rather then the fulfillment of utilitarian needs. Although it is true that we can create and play games for a particular function (for exercise, to meet people, to learn about a topic), by and large, games are played for the intrinsic pleasures they provide.

As a form of designed "delight," the process of interacting with a game is not a means to an end, but an end in and of itself. It is this curious quality of games that makes them wonderful case studies for Design Research through the process of design. As a game evolves (through the *iterative process* outlined below), it defines and redefines its own form, the experiences it can provide for players, and the very questions about design that it can ask. Through this play of design itself, new questions come into being, present themselves to the designers, and sometimes are even answered.

### Iteration Iteration

Iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing and refining a work in progress. In iterative design, interaction with the designed system is used as a form of research for informing and evolving a project as successive versions or *iterations* of a design are implemented.

Test; analyze; refine. And repeat. Because the experience of a viewer/user/player cannot ever be completely predicted, in an iterative process design decisions are based on the experience of the prototype in progress. The prototype is tested, revisions are made, and the project is tested once more. In this way, the project develops through an ongoing dialogue between the designers, the design, and the testing audience.




The iterative design process

In the case of games, iterative design means playtesting. Throughout the entire process of design and development, your game is played. You play it. The rest of the development team plays it. Other people in the office play it. People *visiting* your office play it. You organize groups of testers that match your target audience. You have as many people as possible play the game. In each case, you observe them, ask them questions, then adjust your design and playtest again.

This iterative process of design is radically different than typical retail game development. More often than not, at the start of the design process for a computer or console title, a game designer will think up a finished concept and then write an exhaustive design document that outlines every possible aspect of the game in minute detail. Invariably, the final game never resembles the carefully conceived original. A more iterative design process, on the other hand, will not only conserve development resources, but will also result in a more robust and successful final product.

## Case Study 1: SiSSYFiGHT 2000

*Summary: SiSSYFiGHT 2000 is a multiplayer online game in which players create a schoolgirl avatar and then take part in games where 3 to 6 players vie for dominance of the playground. Each turn a player selects one of six actions to take, ranging from teasing and tattling to cowering and licking a lolly. The outcome of an action is dependent on other players' decisions, making for highly social gameplay.*
*SiSSYFiGHT 2000 is also a robust online community. You can play the game at www.sissyfight.com.*

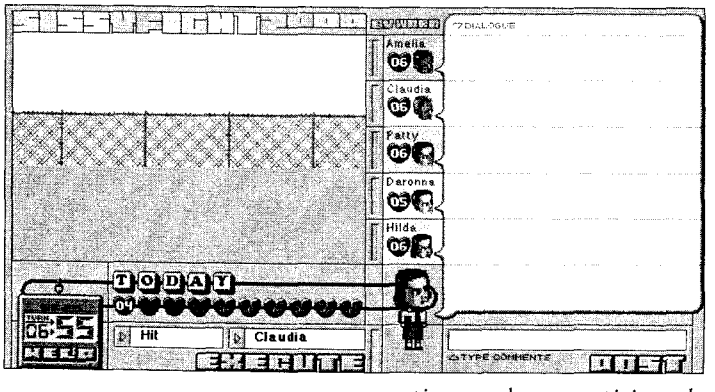
In the summer of 1999, I was hired by Word.com to help them create their first game. We initially worked to identify the project's *play values*: the abstract principles that the game design would embody. The list of play values we created included designing for a broad audience of non-gamers; a low technology barrier; a game that was easy to learn and play but deep and complex; gameplay that was intrinsically social; and finally, something that was in concert with the smart and ironic Word.com sensibility.

These play values were the parameters for a series of brainstorming sessions, interspersed with group play of computer-based and non-computer games. Eventually, a game concept emerged: little girls in social conflict on a playground. While every game embodies some kind of conflict, we were drawn towards modeling a conflict that we hadn't seen depicted previously in a game. Technology and production limitations meant that the game would be turn-based, although it could involve real-time chat.

Once these basic formal and conceptual questions had begun to be mapped out, the shape of the initial prototype became clear. The very first version of SiSSYFiGHT was played with post-it-notes around a conference table. I designed a handful of basic actions each player could take, and acting as the program, I "processed" the actions each turn and reported the results back to the players, keeping score on a piece of paper.

Designing a first prototype requires strategic thinking about how to most quickly implement a playable version that can begin to address the project's chief uncertainties in a meaningful way. Can you create a paper version of your digital game? Can you design a short version of a game that will last much longer in its final form? Can you test the interaction pattern of a massively multiplayer game with just a handful of players?

In the iterative design process, the most detailed thinking you need at any moment is that which will get you to your next prototype. It is, of course, important to understand the big picture as well—the larger conceptual, technical and design questions that drive the project as a whole. Just be sure not to let your design get ahead of your iterative research. Keep your eye on the prize, but leave room for play in your design, for the potential to change as you learn from your playtesting, accepting the fact that some of your assumptions will undoubtedly be wrong.

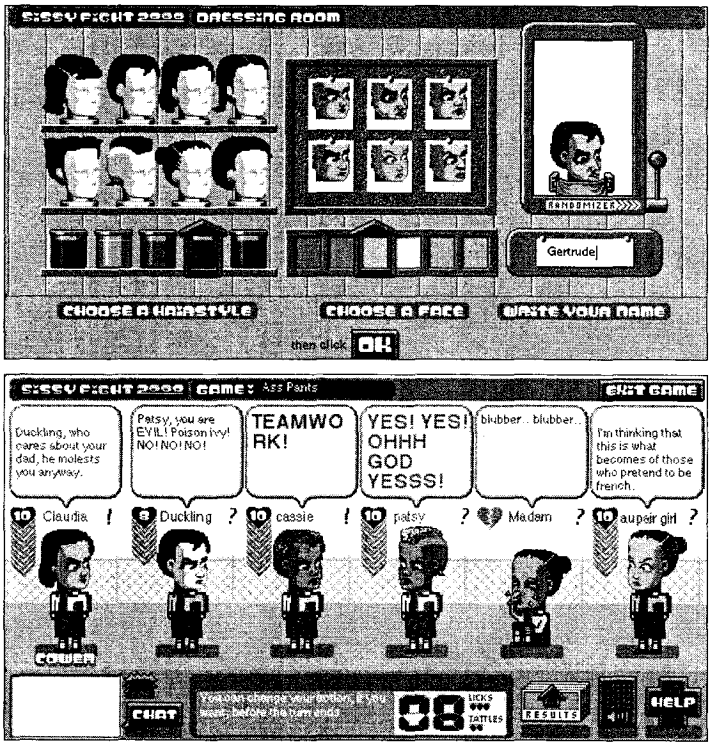



SiSSYFiGHT 2000
early prototype

The project team continued to develop the paper prototype, seeking the balance between cooperation and competition that would become the heart of the final gameplay. We refined the base ruleset—the actions a player can take each turn and the outcomes that result. These rules were turned into a spec for the first digital prototype: a text-only version on IRC, which we played hotseat-style, taking turns sitting at the same computer. Constructing that early, text-only prototype allowed us to focus on the complexities of the game logic without worrying about implementing interactivity, visual and audio aesthetics, and other aspects of the game.

While we tested gameplay via the text-only iteration, programming for the final version began in Director, and the core game logic we had developed for the IRC prototype was recycled into the Director code with little alteration. Parallel to the game design, the project's visual designers had begun to develop

the graphic language of the game and chart out possible screen layouts. These early drafts of the visuals (revised many times over the course of the entire development) were dropped into the Director version of the game, and the first rough-hewn iteration of SiSSYFiGHT as a multiplayer online game took shape, inspired by Henry Darger's outsider art and retro game graphics.




Top and bottom: SiSSYFiGHT 2000 final version

As soon as the web version was playable, the development team played it. And as our ugly duckling grew more refined, the rest of the Word.com staff was roped into playing as well. As the game grew more stable, we descended on our friends' dot-com companies after the workday had ended, sitting them down cold in front of the game and letting them play. All of this testing and feedback helped us refine the game logic, aesthetics and interface. The biggest challenge turned out to be clearly articulating the relationship between player action and game outcome: because the results of each turn are interdependent on every player's action, early versions of the game felt frustratingly arbitrary. Only through many design revisions and dialogue with our testers did we manage to structure the results of each turn to communicate unambiguously what had happened that round and why.

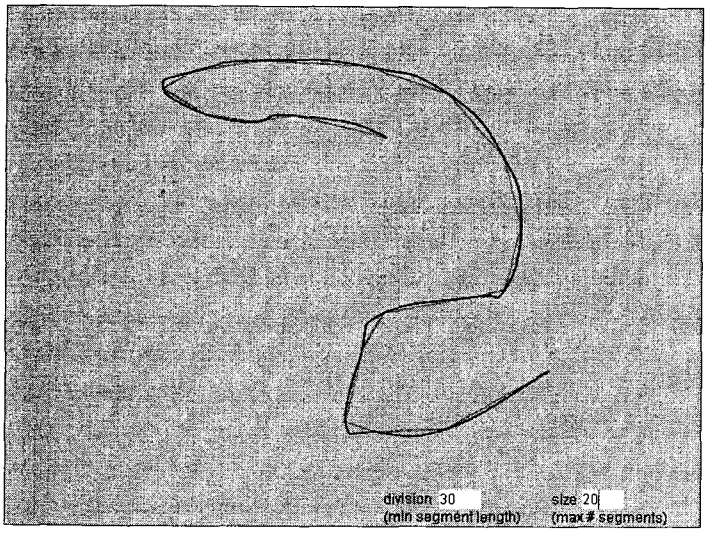
When the server infrastructure was completed, we launched the game to an invitation-only beta-tester community that slowly grew in the weeks leading up to public release. Certain time slots were scheduled as official testing events, but our beta users could come online anytime and play. We made it very easy for the beta testers to contact us and email in bug reports.

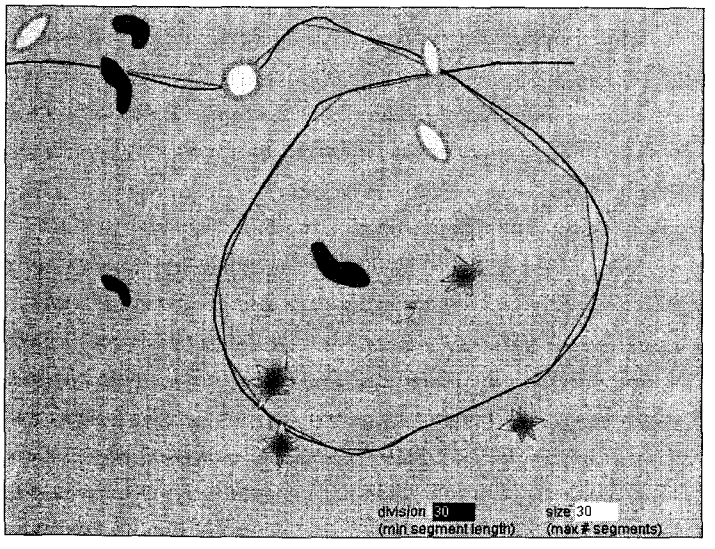
Even with this small sample of a few dozen participants, larger play patterns emerged. For example, as with many multiplayer games, it was highly advantageous to play defensively, leading to standstill matches. In response, we tweaked the game logic to discourage this play style: any player that "cowered" twice in a row was penalized for acting like a chicken! When the game did launch, our loyal beta testers became the core of the game community, easing new players into the game's social space.

In the case of SiSSYFiGHT 2000, the testing and prototyping cycle of iterative design was successful because at each stage, we clarified exactly what we wanted to test and how. We used written and online questionnaires. We debriefed after each testing session. And we strategized about how each version of the game would incorporate the aesthetic, game design, and technical elements of the previous versions, while also laying a foundation for the final form of the experience.

## Case Study 2: LOOP

*Summary: LOOP is a singleplayer game in which the player uses the mouse to catch flittering, colored butterflies. The player draws loops around groups of butterflies of the same color, or of groups in which each butterfly is a different color (the more butterflies in a loop, the more points). To finish a level, the player must capture a certain number of butterflies before the sun sets. The game includes three species of butterflies and a variety of hazardous bugs, all with different behaviors. LOOP was created by gameLab and is available for play at*




Top and Bottom: LOOP early prototypes

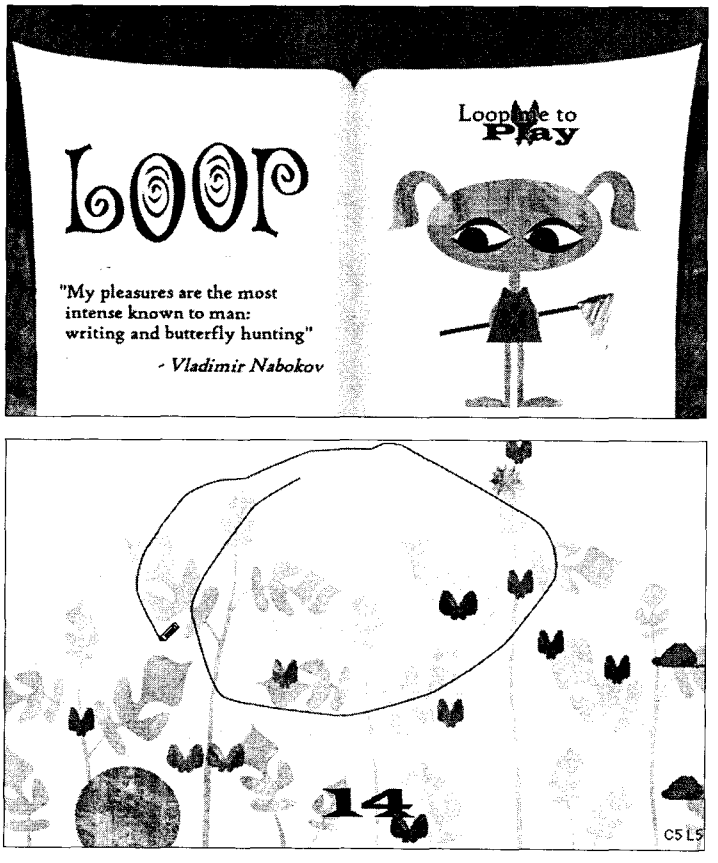
Initial prototypes are usually quite ugly. Game prototypes do not emphasize aesthetics or narrative content; they emphasize the game rules, which manifest as the internal logic of the game, tied to the player's interaction. Visuals, audio and story are important aspects of a game, but the core uncertainties of game design, the questions that a prototype should address, lie in the more fundamental elements of rules and play.

Another way of framing this problem is to ask, what is the *activity* of the game? Rather than asking what the game *is about*, ask what the player is *actually doing* from moment to moment as they play. Virtually all games have a core mechanic, an action or set of actions that players will repeat over and over as they move through the designed system of a game. The prototype should help you understand what this core mechanic is and how the activity becomes meaningful over time. Asking questions about your game's core mechanic can guide the creation of your first prototype, as well as successive iterations. Ideally, initial prototypes model this core mechanic and begin to test it through play.

LOOP grew out of a desire at gameLab to invent a new core mechanic. There are ultimately not very many ways to interact with a computer game: the player can express herself through the mouse and keyboard, and the game can express itself through the screen and speakers. Deciding to intervene on the level of player input, we had a notion to cast aside point-and-click or click-and-drag mouse interaction in favor of sweeping, fluid gestures.

The first prototype tested only this core interaction, allowing the player to draw lines, but nothing else. Our next step was to have the program detect a closed loop and add objects that would shrink and disappear when caught in a loop.

Each of these prototypes had parameters adjustable by the person playing the game. The length of line and detail on the curve could be tweaked, as well as the number of objects, their speed and behavior, and several other variables. As we played the game, we could try out different parameters and immediately see how they affected the experience, adjusting the rules to arrive at a different sort of play. This programming approach of building accessible game design tools into a game prototype is a technical strategy that takes iterative design into account.




Top and Bottom: LOOP final version

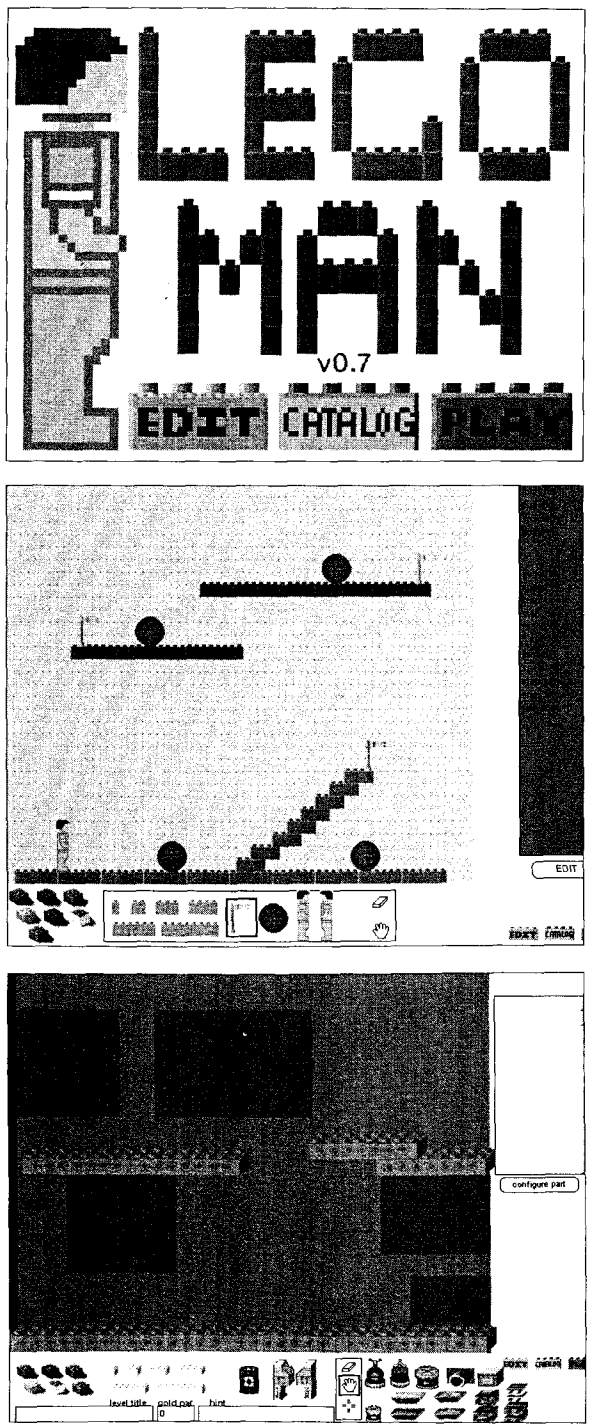
As the butterfly content of the game emerged, so did debate about the game's overall structure and victory and loss conditions. Did the entire screen need to be cleared of butterflies or did the player just have to catch a certain number of them? Did the butterflies gradually fill up the screen or did their number remain constant? Were there discrete levels or did the game just go on until the loss conditions were met? Was there some kind of time-pressure element? These fundamental questions, which grew out of our core mechanic prototyping, were only answered by actually trying out possibilities and coming to a conclusion through play.

As the game code solidified, the many adjustable parameters of the game were placed in a text file that was read into the application when it ran. These parameters controlled everything from the behavior of game creatures to points scored for different numbers of butterflies in a loop to the progression of the game's escalating difficulty. Thus the game designers could focus on refining game variables and designing levels, while the rest of the program—screen transitions and help functionality, the high score system and integration with the host site—was under construction.

LOOP followed a testing pattern similar to that of SiSSYFiGHT, moving outward from the game creators to include a larger circle of players. During the development of LOOP, gameLab created the gameLab Rats, our official playtesting "club," to facilitate the process of testing and feedback. In the end, LOOP managed to achieve the fluid interaction we had first envisioned. An entire game evolved from a simple idea about mouse control. That is the power of iterative design.

## Case Study 3: LEGO Junkbot

*Summary: LEGO Junkbot is a singleplayer game in which the player helps the robot character Junkbot empty trash cans throughout a factory. The player doesn't control Junkbot directly but instead uses the mouse to move LEGO bricks around the screen, deconstructing and reconstructing his environment brick by brick, building stairways and bridges that help Junkbot get where he needs to go. A variety of helpful and hazardous objects and robots add variety and complication to the game's 60 levels. Junkbot levels can be solved in multiple ways and the game structure encourages players to go back to previously solved levels and complete them using a different method.*

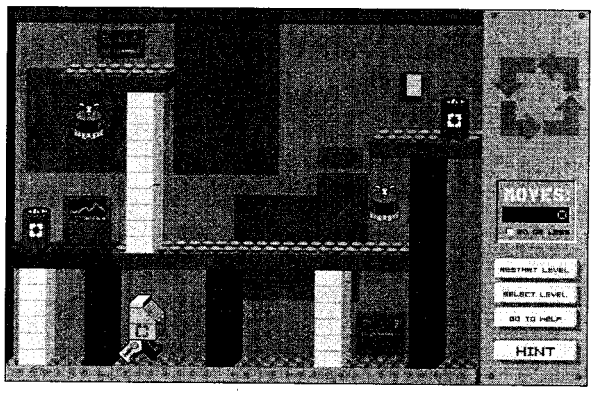


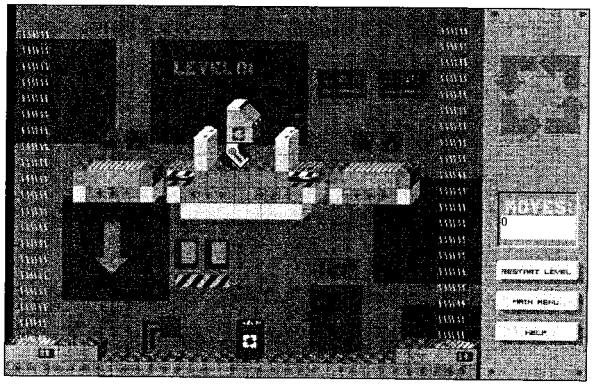

All: LEGO Junkbot early prototype

The conceptual starting point for the creation of LEGO Junkbot came from gameLab's client, LEGO.com. LEGO wanted a game about brick construction with a target audience of 8- to 12-year-old boys that that could also be played and enjoyed by adults. Here was our challenge: gameLab had been tasked with creating a web game in which real-world LEGO play was the clear referent. Yet in no way could we ever hope to recreate the sublime interactivity of plastic LEGO bricks. How could we translate LEGO play into a digital game?

Our first step was to purchase and play with a whole mess of LEGO bricks, as a way of analyzing and understanding the subtleties of LEGO play. Then, as with most gameLab projects, we began to design by identifying the project's play values. These values, which embodied the material and experiential qualities of LEGO as well as the cultural ethos of the LEGO play philosophy, included concepts like modularity, open-ended construction, design creativity, multiple-solution problem-solving, imaginative play and engineering. Using these play values as our limiting parameters, we brainstormed a number of game concepts.

The concept LEGO selected was called LEGOman (the character and storyline of Junkbot had not yet emerged) and it centered around moving bricks to indirectly help a character move through an environment. The first playable prototype was the simplest possible iteration of the core interactive idea: the player could use the mouse to drag bricks on the screen; there was a single, autonomously-moving protagonist character, there were goal flags to touch, and there were rolling wheel hazards to avoid.


Top and Middle: LEGO Junkbot level editor tool
Bottom: LEGO Junkbot

We played the prototype. And it was not very fun. Because gameLab projects often try to invent new forms of gameplay, we sometimes find that our initial prototypes are just not that enjoyable to play. At such an early juncture in the iterative design process, we could have scrapped the design altogether and started fresh, building on insights learned from the unsuccessful prototype, or we could dig in and push on through. We chose the latter. Gradually we added elements to the game, refining the interaction, expanding the level possibilities, putting in new kinds of special bricks and robot hazards.

Each new element addressed something that was lacking in the experience of the previous prototype: it was monotonous to move bricks one by one, so we implemented code that let players stack bricks and move them as a group. We needed a way to move the main character vertically on the screen, so we added fan bricks, which float Junkbot upwards. The game obstacles all felt too deterministic, so we introduced robot hazards that responded to Junkbot in real time. And as these interactive embellishments deepened the game (which was actually becoming fun to play), the character and storyline of Junkbot emerged.

Throughout the process, we utilized a level editor, a visual design tool that let the game designers create and save levels. The editor allowed them to experiment with game elements and level designs, refining the overall experience and planning features for the next iteration of the prototype.

Playtesting continued with the gameLab Rats, using a web-based form to collect and collate testing data about the difficulty and enjoyment of each level. However, our main concern was whether the basic brick-construction core mechanic would be understood by our target audience, so we visited an elementary school computer classroom, sat kids down in front of the game, and let them play cold. This testing was invaluable, and confirmed our fears: too many of the testers had trouble picking up basic game concepts, such as how to make a stairway for Junkbot out of bricks. This testing directly influenced the design of the game, and we slowed down the overall learning curve, designing the first several game levels to more clearly communicate the essential interactive ideas.

A good rule of thumb for iterative testing is to err on the side of observation rather than guidance. While it may be difficult to keep your hands off the tester's mouse, sit back and see what your audience actually does, rather than telling them how it is supposed to work. What you observe can sometimes be

painful to watch, but it will help you design more successful play. Part of iterative design is simply learning how to listen.

## Conclusion

Iterative design is a process-based design methodology, but it is also a form of Design Research. In each of these three case studies, questions emerged out of the process of design—questions that were not part of the initial problem but that were nevertheless answered through iterative design and play.

To design a game is to construct a set of rules. But the point of game design is not to have players experience rules—it is to have players experience *play*. Game design is therefore a second-order design problem, in which designers craft play, but only indirectly. Play arises out of the rules as they are inhabited and enacted by players, creating emergent patterns of behavior, sensation and interaction. Thus the necessity of the interactive design process. The delicate interaction of rule and play is something too complex and emergent to script out in advance, requiring the improvisational balancing that only testing and prototyping can provide.

The principles of this process are clearly applicable beyond the limited domain of games. Rules and play are just game design terms for structure and experience: a designer creates some kind of structured system (a typeface, a building, a car), and people encounter, inhabit, explore and manipulate the system, using it, experiencing it, playing with it.

In iterative design, there is a blending of designer and user, creator and player. It is a form of design through the reinvention of play. Through iterative design, designers create systems and play with them, but only in order to question them, bend them, break them and re-fashion them into something new. This process of iteration, of design through play, of discovering the answers to questions you didn't even know were there, is just another form of what this book is about: Design Research.

DEMO

# CREATING A CULTURE OF DESIGN RESEARCH

ERIC ZIMMERMAN

As this book abundantly demonstrates, Design Research can come in many forms, from quantitative market research to personal interviews to experimental design explorations. But Design Research is more than a set of strategies and procedures. It also represents a particular attitude about design, a willingness to look beyond the immediate concerns of crafting a specific project, an openness to integrating ideas and insights from the outside world into the design process itself. Successful Design Research in a commercial firm requires a company culture that embraces research in concert with design.

This demo outlines some of the strategies taken at gameLab, a game development studio founded by myself and Peter Lee, to foster a culture of Design Research. At the heart of the way we run gameLab is a relentless drive to connect the experience of working at our company to larger cultural spheres. Whatever it is that your firm designs, emphasize the links between your daily design practice and related cultures outside the company walls.

### 1. Create a space that encourages design research

gameLab designs and develops computer games, and the office space we inhabit is filled to bursting with games, toys, and other play objects. Company staff are encouraged to spend time playing every day, whether that means surfing online games, spending lunchtime playing a boardgame, or taking work breaks interacting with one of the many game consoles in the office. Work tasks always take precedence over this kind of play research, but generally staff end up spending several hours a week just playing. This activity serves many purposes for us, including competitive market analysis, technological research and general design inspiration.

Any office is a nexus for the exchange of ideas, and at gameLab we encourage staff to share the insights from their informal play research. A section of our intranet is reserved for posting links and thoughts about new games and game sites. Furthermore, the open physical layout of our office lets us see what is happening on each others' screens, reducing the potential stigma of "playing at work" and encouraging discussions about games even as they are being played.

**2. Build a design research library**

One concrete strategy we've undertaken at gameLab is the development of a research library. Our library includes retail game titles, books and graphic novels, DVDs and videotapes, magazines (we have many subscriptions), board and card games, and toys of all kinds.

While some of our research library has come from the research needs of specific projects, most of it has evolved over time through staff purchases. Every month, each gameLab staff is encouraged to contribute to the research library by spending up to $50 (reimbursed, of course) to buy something for the office. Thus the library is a reflection of the tastes and interests of the staff, giving them a way to express themselves and to shape the mindspace of the company as a whole.

The library fosters research by encouraging gameLab staff to seek out cultural objects and bring them into the company from the outside world. And of course the library itself is an ever-present opportunity for formal or informal research investigations as well. A sign-out sheet lets staff check out items from the library for use outside the office.

**3. Attend and create events**

Be on the lookout for cultural events relevant to your company's design work. In the past, gameLab has attended films, exhibits, conferences, and

other events connected to games, design and popular culture. These group activities (always optional and always paid for by the company) serve double-duty as research opportunities and as occasions for team-building. We keep outside field trips somewhat infrequent, so that they maintain their status as special events.

In addition, we host our own Design Research affairs. Approximately once per month, game developers in the New York City area come to the gameLab offices for an evening of beer, pizza and boardgames. These get-togethers not only let our staff learn about fundamental game design principles through non-computer game play, but also help foster the local game development community, giving us an opportunity to network, share industry gossip, discuss technical dilemmas, and even show off game prototypes—all forms of Design Research.

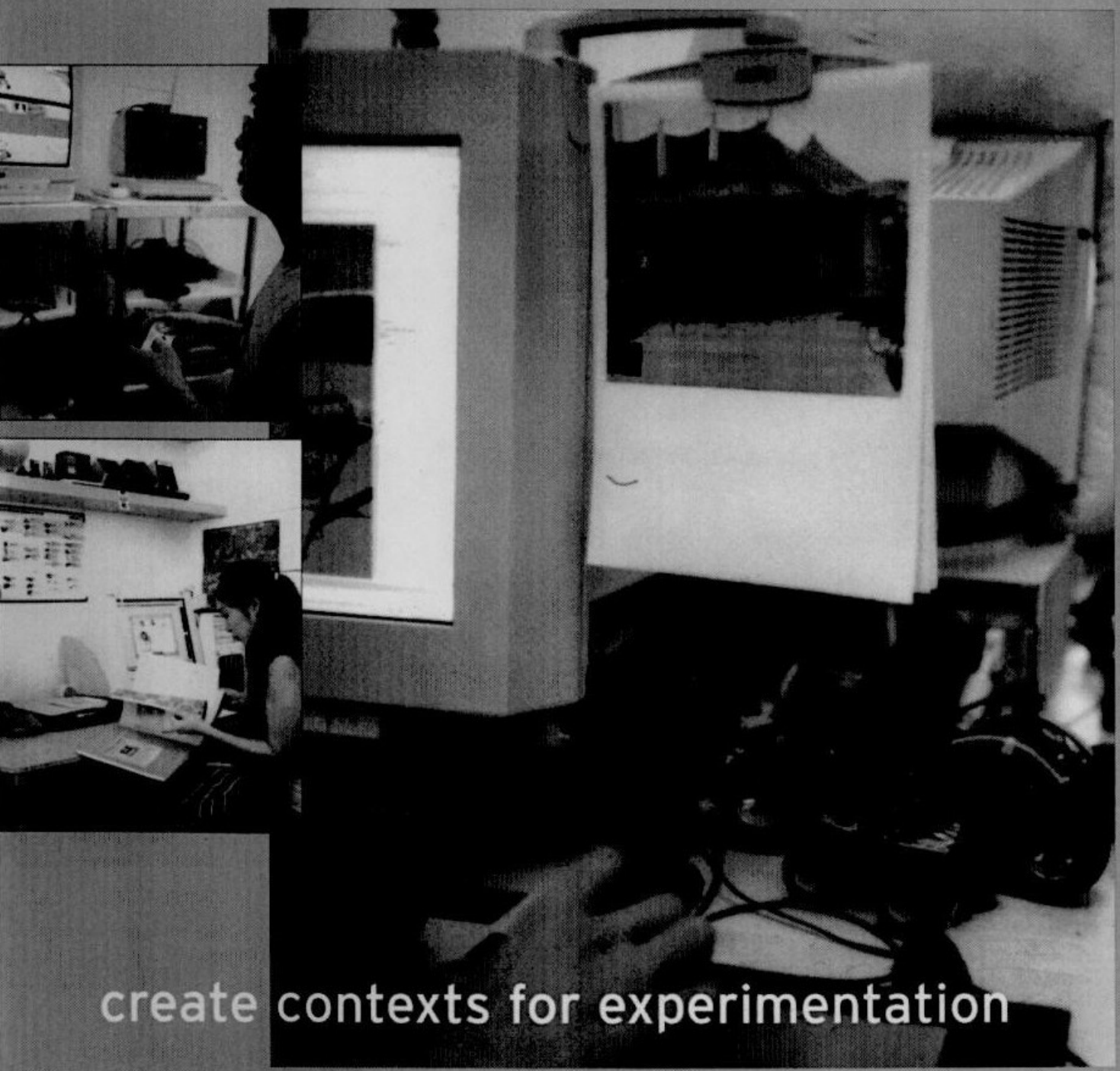
**4. Let them teach**

A majority of gameLab staff teach. Peter and I encourage them to teach courses, attend critiques, participate on panels, and give talks and workshops. Teaching is a profoundly challenging and effective form of research, and gameLab staff have taught everything from game design and game programming to Masters thesis seminars and interactive narrative design.

Having your staff teach also builds bridges between your company and the local academic community. Teaching creates contact with students and faculty. As a result, gameLab has constant access to qualified interns and freelancers, potential new staff hires, and legions of enthusiastic student game testers. Our academic relationships have also fostered unique project opportunities as well, such as a studio class at Parsons School of Design in which students work with gameLab to research, design and implement an experimental game.

# create a space


# build a design research library that encourages design research

create contexts for experimentation

**5. Encourage side projects**

We encourage our staff to pursue personal projects. These can take the form of articles, essays, and books (many of our staff are published writers); experimental design projects (our Director of Technology Ranjit Bhatnagar regularly exhibits his robot artwork); and the creation of non-computer games (such as gameLab Game Designer Nick Fortugno's self-published role-playing game).

As long as the side projects of full-time workers do not compete directly with gameLab's core business of making commercial computer games, Peter and I urge our staff to have creatively rich lives outside their work at the company, and we celebrate their accomplishments in such endeavors. Games are culture, and the success of our collective work at gameLab is dependent on the cultural sophistication of our staff. By engaging with culture productively on their own terms outside gameLab, our staff bring insight into the projects they complete within the company.

**6. Create contexts for experimentation**

Although gameLab is a commercial studio, from time to time we create opportunities for our staff to undertake experimental, noncommercial projects as a form of design research. For example, over the last three years, gameLab has designed and implemented a large-scale social game for our annual industry gathering, the Game Developers Conference. These massively multiplayer off-line games are events played by thousands of players, and they vary in form from year to year.

Although the primary purpose of our conference games is to research forms of social game play, they have a number of side benefits as well. The games serve as potent publicity stunts for our industry peers, game publishers and the game press, highlighting our profile as an innovative game company. The games also feed other research efforts: some of our staff are

currently writing a paper about the design insights we have gained from these conference projects.

In sum, there are innumerable ways of incorporating Design Research into the everyday experience of your firm, thereby fostering a company culture that embraces research. The informal guidelines outlined above are not meant to replace more formalized research techniques, but instead to foster a company context in which Design Research is tightly integrated into design practice. A company that engages in Design Research on a daily basis will be much more open to bring research into specific design projects when the opportunity arises.

While all of these Design Research activities may seem like a drain on your company resources, in fact the benefits of fostering a Design Research culture far outweigh the costs. gameLab is a small design studio of about a dozen staff, and if we can do it, you can too. All that is required is a willingness to open your company to the world beyond its walls, to imaginatively mix cultures and contexts inside and outside the company. Design shapes the world: shouldn't we let the world shape our design?

# Interdisciplinary Design Research

PATRIK SVENSSON

## Intersections

It is becoming increasingly critical to be able to relate ideas and experience from many different fields. The world is not neatly categorized, and this chapter is about intersections of thinking and doing that you will not find shelved in a bookshop or distinctly pigeonholed by search engines. Being at the forefront of any kind of field means that you have to have a sense of what is going on, and where new ideas, products and research emerge. For designers this is particularly important, and in this chapter we will consider interdisciplinary Design Research as a process and tool.

Let us use a fairly concrete example to illustrate the concept of interdisciplinary research. I am a frequent bookshop visitor, and have often reflected on how bookshops in fact reflect some kind of categorization of the world. Over time you might observe changes in the way books are shelved and categorized; for instance, how new fields slowly make their way into the established scheme of categorization. An example would be the field of cognitive science, established in the late 1980s in the form we now know it. It encompasses research in areas such as linguistics, biology, neuroscience, computing science and philosophy. For a long time, the relevant literature was spread out all over bookshops, and the field might best have been described as multidisciplinary. Over the last decade, many academic bookshops have introduced cognitive science sections. This development has been accompanied by growth of cognitive science journals, conferences and professional organizations [Klein 1996, 57]. At least in some parts of the world, cognitive science has not only moved from being multidisciplinary to interdisciplinary, but is also on its way to become a new discipline in its own right. In the following, we will mostly be concerned with exactly the stage at which something new is coming into being, and indeed, how we as designers can both benefit from and contribute to such processes.

Being truly interdisciplinary is rarely easy, as it is all about fuzzy boundaries and being in between established categories [Nissani 1997]. First of all, we have to be willing to break out of discipline-specific structures; as we all know, walls between disciplines or departments do not exist only in the academic world. Second, we need to be open to the ideas and language of people who have different backgrounds and ideas about the world (for example, the different "languages" used in the first and second sections of this book). As a consequence, we need to be willing to change the way we think about the world and what we do. Third, in order to be open to new possibilities we also need to know something about current

themes, relevant fields and people in relation to our own field. Fourth, we need to maintain a strong sense of who we are and where we come from. Interdisciplinary work sometimes has a tendency to be shallow as it brings together people who know a bit about many things—we should take care not to lose ourselves in the interdisciplinary process. But what is absolutely vital is being passionate about exploring new arenas and not giving in to standard solutions.

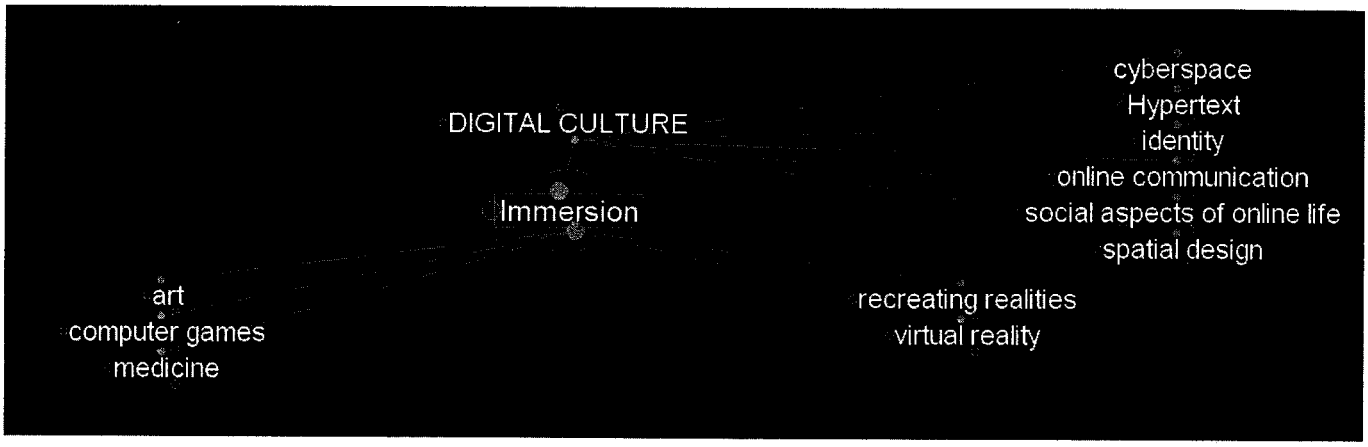
The process of carrying out interdisciplinary design research and using it is probably best described through examples, and I will discuss the field of digital culture as well as two case studies set within this field. The examples focus on technology and its cultural context, but many of the basic ideas apply to other fields and contexts as well.

Many of us spend much time on the computer, on the mobile phone, and engaged with new kinds of activities emerging in computer games, online forums and text messaging. Technology here serves not so much as a tool as an arena and our world is to some extent "technologically textured" [Ihde 1990, 1]. The cultural, societal and commercial potential of this arena is enormous. The two case studies focus on specific aspects of digital culture; namely, innovative learning environments and the interrupt culture. We will look at a project where students have created graphical virtual worlds instead of traditional paper essays as learning environments. Graphical virtual environments have a great potential as collaborative and creative spaces that can be used, for instance, in supporting life-long learning in industry. The second case study, the Interrupt Culture Project, deals with increasing information flows and time fragmentation. How do we as humans with our evolutionary past manage multiple information flows such as email, pagers, PDAs and blogs? And how can we design new products and thinking that make it easier for humans to mange the emerging interrupt culture?

## Digital Culture

A very clear example of intense interdisciplinary activity today is the area that might be called digital culture. In the western world, and increasingly elsewhere, we are becoming more and more digitized. This is true of processes, activities and communication. Many of us find that we spend a fair share of our lives on the computer, and that computers are also moving out of the gray boxes and into our everyday lives in the form of PDAs, mobile technology and wireless networks. Some aspects of our lives are becoming virtualized. Clearly, this is an area that is inherently multidisciplinary as it involves so many different academic disciplines, industries and competencies. In the past, much industrial activity has been technology-based, but increasingly, there is a common acceptance of the necessity for teams with multiple competencies and interdisciplinary knowledge. Design competence is especially relevant to interaction design, participatory design and experience design and, in this context, it makes good sense to have an overview of the field. How can we achieve that?

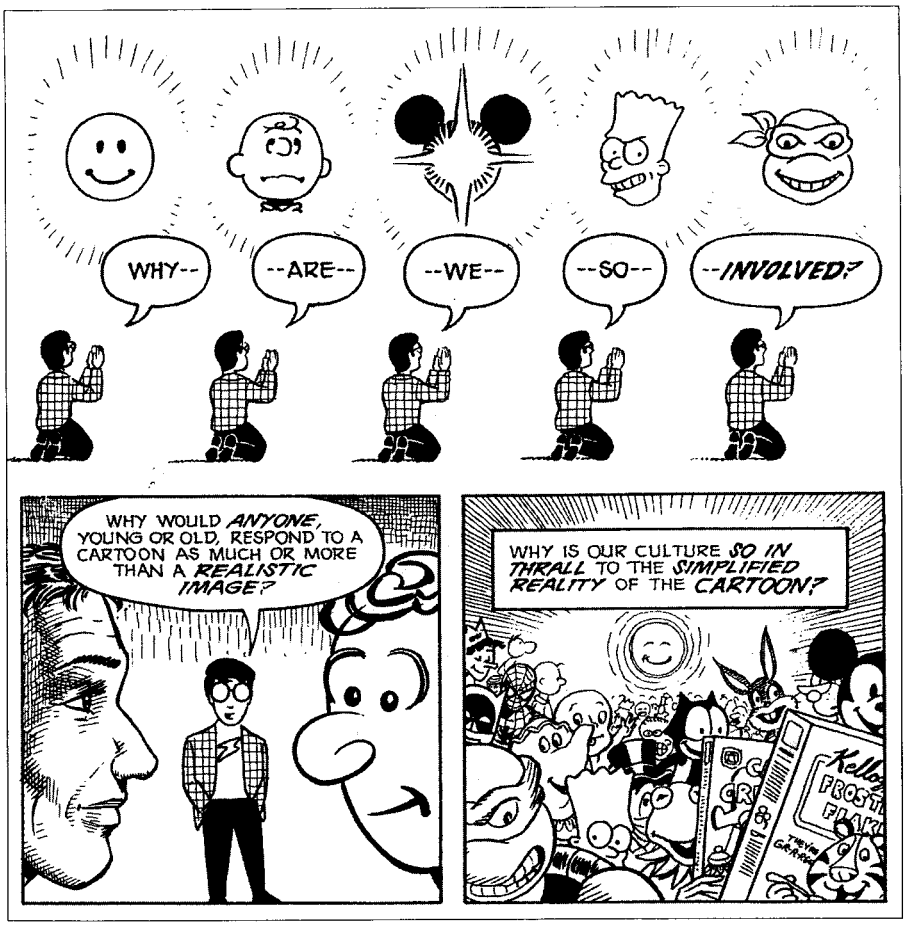
First of all, it is a good idea to identify some of the salient aspects of digital culture. My personal list would probably include cyberspace, spatial design, virtual reality, online communication, virtual communities, online environments, social aspects of online life, embodiment (or its converse), socialization, identity, reputation, immersion, interfaces, hypertext, narratives, blogs, digital art, computer games and emergence. This is a long list that at first might seem a bit cluttered, but it does help us to approach the field. Looking at just one or two of these aspects will not be enough for a real understanding of what we are dealing with here. If you are involved in designing computer games you need to know about social aspects of gaming worlds, reputation systems, identity and immersion (among other things). Even though these issues might seem new and particular to digital media, we rarely find that is the case. Sociologists have been concerned with social behavior for a long time now; anthropologists work with reputation and social structures; and researchers in literary reception studies have thought about how young people get immersed in the worlds that are created through texts.




Digital culture brain

We are usually not equally interested in all the nodes in an interdisciplinary network. We tend to have a theme, project or question that provides a starting point or focus. The nature of interdisciplinary work requires us to follow connections and relate things in unexpected ways. This is not something that is easily learned, but having a holistic understanding and being open to "cross-thinking" certainly help. There are also tools that allow you to visualize thought structures and links, and these might be useful in laying out a network. In the illustration above the field of digital culture is represented on one such platform, and here the focus is on "immersion." The full network can be found at , where you can investigate my personal interpretation of digital culture in an associative, interactive way.

Let us look at the immersion node a bit more carefully. In techno-industry, immersion has become closely associated with expensive technology that gives you a wide visual field filled with computer-generated graphics, sometimes




Scott McCloud on immersion [McCloud 1990, 30]

employing spatialized sound, through CAVE environments or head-mounted displays. There is much focus on the visual environment. People who work with 3D graphics design frequently seem to be concerned with recreating reality in the computer. I know many people who work with 3D graphics and animation who are obsessed about recreating "real" water in the computer—a particularly difficult challenge ⊙268 DAVIS ET AL. Computer games are becoming increasingly photorealistic, and an important design concern relates to how we best create immersive environments. Naimark states that entire industries work with these issues, and that "[t]he goal is less about creating metaphor and poetry as about not creating headaches" ⊙109 NAIMARK. It could be argued that creating immersion has become far too involved with high-end graphics at the expensive of social and narrative depth ⊙276 DAMER. Also, humans are very good at detecting faults with visual presentation that is highly photorealistic. Such representations tend to be non-symbolic in nature as the audience is given such a "filled out" presentation. Immersion is not only visual, of course, and people may be perfectly well immersed in text environments (such as MUDs) and literature.

From the point of view of design, this is an interesting and highly relevant area. How do we create truly immersive environments? What features are important? How much should we rely on photorealistic graphics? Can we create real social depth? Evidence and research come from many different spheres. One such sphere is admirably represented by cartoonist Scott McCloud in the illustration on this page.

For someone concerned with the creation of immersive environments it certainly makes sense to listen to experts on visual communication like McCloud. Other sources would probably include literature on 3D graphics, presence, online communities and interaction design. It would probably also be worthwhile looking at literature concerned with representation and magical realism in literature [Walton 1990] and immersion in art [Grau 2003]. Moreover, we might be helped by