RISE OF THE VIDEOGAME ZINESTERS

HOW FREAKS, NORMALS, AMATEURS, ARTISTS, DREAMERS, DROPOUTS, QUEERS, HOUSEWIVES, AND PEOPLE LIKE YOU ARE TAKING BACK AN ART FORM

ANNA ANTHROPY

SEVEN STORIES PRESS
NEW YORK

A Seven Stories Press First Edition

Seven Stories Press
140 Watts Street
New York, NY 10013
www.sevenstories.com

College professors may order examination copies of Seven Stories Press titles for a free six-month trial period. To order, visit http://www.sevenstories.com/textbook or send a fax on school letterhead to (212) 226-1411.

Book design by Elizabeth DeLong

Library of Congress Cataloging-in-Publication Data

Anthropy, Anna.
Rise of the videogame zinesters : how freaks, normals, amateurs, artists, dreamers, dropouts, queers, housewives, and people like you are taking back an art form / Anna Anthropy.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-60980-372-8 (pbk.)
1. Video games. 2. Video games--Social aspects. I. Title.
GV1469.3.A53 2012
794.8--dc23

2011045409

Printed in the United States

9 8 7 6 5 4 3 2

For the child I was,
the book no one could write for me.

Chapter One
THE PROBLEM WITH VIDEOGAMES

I have a problem with videogames.[1]

Plenty of people seem to have problems with videogames these days. Newscasters are fond of reporting that videogames are dangerous to children, either because they teach children how to steal cars and kill cops or because they actually connect children electronically to the game-playing predators who are waiting to snatch them away. Religious leaders have wasted no time condemning videogames as a trap for children's souls, and armchair psychologists accuse them of turning kids into antisocial hermits.

People condemn videogames because videogames are pervasive in popular culture. They're on our computers and our cell phones, in our homes and purses and pockets. Even if you yourself don't play games, you have a hard time escaping their marketing. When the television isn't telling you to be afraid of videogames, it's telling you to buy them, and to drink *World of Warcraft*–flavored Mountain Dew while you play.

These are some problems people have with videogames. What's *my* problem with videogames?

As a queer transgendered woman in 2012, in a culture pervaded by videogames—a culture in which, typing on my computer, I am seconds away from a digital game, even if I

have not taken the time to buy or install a single game on my computer—I have to strain to find any game that's about a queer woman, to find any game that resembles my own experience.

This is in spite of the fact that videogames in America and elsewhere are an industry and an institution. I've already brought up *World of Warcraft*, a game about performing repetitive tasks until numbers increase. So, now that we're in the land of numbers, here are some numbers. The ESA—that's the Entertainment Software Association, who spend half their time assuring the population that videogames aren't worth being mad at, and the other half pursuing litigation against anyone who distributes games that their shareholders have long since stopped distributing or profiting from—claims that, as of 2009, 68 percent of American households play digital games.[2] In 2008 alone, people bought 269,100,000 games (the ESA word is *units*.)[3]

So digital games, by the numbers, are here, and they take up a lot of space. And almost none of these games are about me, or anyone like me.

What are videogames about?

Mostly, videogames are about men shooting men in the face. Sometimes they are about women shooting men in the face. Sometimes the men who are shot in the face are orcs, zombies, or monsters. Most of the other games the ESA is talking about when it mentions "units" are abstract games: the story of a blue square who waits for a player to place him in a line with two other blue squares, so he can disappear forever. The few commercial games that involve a woman protagonist in a role other than slaughterer put her in a role of servitude: waiting tables at a diner (or a dress shop, a pet shop, a wedding party). This is not to say that games about head shots are without value, but if one looked solely at videogames, one would think the whole of human experience is shooting men and taking their dinner orders. Surely an artistic form that has as much weight in popular culture as the videogame does now has more to offer than such a narrow view of what it is to be human.

And yes, from here on out I'll be talking about videogames as an art form. What I mean by this is that games, digital and otherwise, transmit ideas and culture. This is something they share with poems, novels, music albums, films, sculptures, and paintings. A painting conveys what it's like to experience the subject as an image; a game conveys what it's like to experience the subject as a system of rules. If videogames are compared unfavorably to other art forms such as novels and songs and films—and they are compared unfavorably with these forms, or else this paragraph defending videogames as art wouldn't be necessary—it is likely a result of how limited a perspective videogames have offered up to this point. Imagine a world in which art forms are assigned value by the number of dykes that populate them. This is the world I inhabit; this is the value games have for me. And why not? The number

of stories from marginalized cultures—from people who are othered by the mainstream—that a form contains tells us something about that form's maturity. If a form has attracted so many authors, so many voices, that several of them come from experiences outside the social norm and bring those experiences and voices to bear when working in that form, can't that form be said to have reached cultural maturity?

It should go without saying that novels and films have plenty of dykes in them, as do the media of writing and filmmaking. American comics have been around since 1896—that's over one hundred years—yet comics are still involved in a debate, as videogames are, about their cultural and artistic value. But I can think of many comics about queer women. More important, I can think of plenty of queer women who make comics: to name a few, Diane DiMassa, Alison Bechdel, Jennifer Camper, Kris Dresen, and Colleen Coover, in order of how disappointed I was when they came out in defense of the Michigan Womyn's Music Festival.[4] And those are just print comics, in a world where the majority of comics are published on the Internet.

In Alison Bechdel's *Dykes to Watch Out For*, Mo (a dyke to watch out for) explains a metric she uses to decide whether she'll watch a movie. This criteria has become known as the Bechdel Test: the movie has to (1) contain at least two women who (2) talk to each other about (3) something other than a man. So why do videogames fail my variant of the Bechdel Test? Why are there no dykes in videogames?

I know at least one of you has been itching, for several pages, to point out games like *Fear Effect 2: Retro Helix* and *Mass Effect*, both of which include scenes in which women smooch women, both on and off camera. In *Fear Effect 2*, women make out for the benefit of the male audience the game's creators

expect to buy the game. (The first scene, in fact, is of the protagonist stripping as seen through a hidden camera, which tells us something about her relationship to the player.) And the lady-sex in *Mass Effect* is just one of many branches on a tree of awkward dialogue, offering nothing resembling the actual lust, desire, and flirtation that women feel for each other. But, aesthetic failures aside, the most important distinction here is that these are stories about queer women that are generally written by white, college-educated men. These are not cases of queer women presenting their own experiences.

Why are digital games so sparse in the dykes making art department? Why are the experiences that games present, the stories they tell, the voices in which they speak, so limited?

The limitations of games aren't just thematic. When I criticize games for being mostly about shooting people in the head, that's a design criticism as well. Most games are copies of existing successful games. They play like other games, resemble their contemporaries in shape and structure, have the same buttons that interact with the world in the same way (mouse to aim, left click to shoot), and have the same shortcomings. If there's a vast pool of experiences that contemporary videogames are failing to tap, then there's just as large a pool of aesthetic and design possibilities that are being ignored. I don't believe these are separate issues, either. To tell different stories, we need different ways of interacting with games. Why are games so similar in terms of both content and design?

The problem with videogames is that they're created by a small, insular group of people. Digital games largely come from within a single culture. When computers were first installed in college campuses and laboratories, only engineers had the access to the machines, the comparative leisure

time, and the technical knowledge to teach those computers to play games. It is not surprising that the games they made looked like their own experiences: physics simulations, space adventures drawn from the science fiction they enjoyed, the *Dungeons & Dragons* tabletop role-playing games they played with their friends. As computers made their way out of labs and into homes, the games that programmers were hacking together became a salable product—and salespeople showed up to profit off of them. And so as businessmen and marketers guided videogames into becoming a billion-dollar industry, publishers installed themselves as the gatekeepers of game creation.

Commercial games have become expensive: according to a presentation at the High Performance Graphics 2009 conference, *Gears of War 2*—an industry leader in the "men shooting things" genre—had a "development budget" of 12 million dollars.[5] ("Development" refers just to the cost of creating the game—it doesn't include all the bucks that were spent marketing, manufacturing, and shipping the game.) If the game cost that much to produce, you can imagine what it would have to earn in sales in order to make any money. Hint: more than 12 million dollars. With that much money at stake, publishers and shareholders are not going to permit a game that is experimental either in terms of its content or in terms of its design. The publisher will do the minimum amount it can get away with in order to differentiate its game from all other games that follow its previously established model and that are being sold to its previously established audience.

Now we have a dangerous cycle: publishers permit only games that follow a previously established model to be marketed to previously established audiences, and only to those audiences. The audiences in question are mostly young adults,

and mostly male. And it's these dudes, already entrenched in the existing culture of games, who are eventually driven to enter the videogame industry and to take part in the creation of games. The population who creates games becomes more and more insular and homogeneous: it's the same small group of people who are creating the same games for themselves.

Videogames as they're commonly conceived today both come from and contain exactly one perspective. It should be terrifying that an entire art form can be dominated by a single perspective, that a small and privileged group has a monopoly on the creation of art. Before the adoption of the printing press, the church was responsible for the creation of books, and the books that monks hand-lettered in Latin in monasteries were largely the Bible or books that agreed with the Bible. Not to knock the Bible, but that a single institution can hold power over what works are allowed to exist within any art form should demonstrate the power that institution has over that art form, and therefore over that culture. And so the printing press, which allowed people to print their own versions of the Bible in their own languages—and eventually to print books that had nothing to do with the Bible—had a role to play in the decentralization of religious authority in Europe.

The printing press is a piece of technology. If digital games, a form that is often (and not entirely correctly) described as being "technology driven," can be compared to books, where then is the printing press for videogames?

What Videogames Need

There's a videogame about a dyke who convinces her girlfriend to stop drinking. Mainstream gamer culture by and large does not know about this game. I know about this game because I made it.

I created *Calamity Annie* in 2008. I made it by myself: I wrote the dialogue, composed the music, designed the rules, scripted the game, and drew all the characters. It was made in a couple of months. The development costs were the cost of the food that went into my belly. I made the game in a program called Game Maker, which, at the time, cost fifteen dollars.

I am nowhere close to the only person who has used Game Maker, nowhere close to the only person who makes digital games outside of the games industry's publisher model. There are hundreds, if not thousands, of such creators. A few of them have achieved some mainstream recognition, like Jonathan Blow and Jason Rohrer, who was profiled in *Esquire* magazine. But these rich white dudes were professional programmers before they came to videogames, and so they don't represent the new dynamic that I'm excited about: hobbyists and non-programmers making their first games. There are lots of tools that allow people to make and distribute games without ever having written a line of code and without having to pass through publishers' gates. In years to come, there will be a lot more tools. I hope that there will also be a lot more people.

I once heard the criticism that the phrase "what videogames need" can usually be more honestly rephrased as "what I want from videogames." In that case, what I want from videogames is a plurality of voices. I want games to come from a wider set of experiences and present a wider range of perspectives. I can imagine—you are invited to imagine with me—a world in which digital games are not manufactured by publishers for the same small audience, but one in which games are authored by you and me for the benefit of our peers.

This is something the videogame industry, by its nature, cannot give us. I like to think about zines—self-published, self-distributed magazines and books. Send me a dollar and a

self-addressed envelope; I'll send you a stapled book of some stories from my life, or some pictures I took of out-of-the-way nooks of my city, or researched accounts of historical murders, or some jokes about sea life. (What does the merman's waiter bring? He brings the MERMANATEE.[6]) I like the idea of games as zines: as transmissions of ideas and culture from person to person, as personal artifacts instead of impersonal creations by teams of forty-five artists and fifteen programmers, in the case of *Gears of War 2*.

The Internet in particular has made self-publishing and distributing games both possible and easy. Authors are able not only to put their works online, but to find audiences for them. Publishers want to be gatekeepers to the creation of videogames, but the Internet has opened those gates.

Currently, the only real barrier to game creation is the technical ability to design and create games—and that, too, is a problem that is in the process of being solved.

Digital game creation was once limited to those who knew how to speak with computers: engineers and programmers, people who could code. In the games industry of today, coders are an inescapable fixture of the hierarchy of production, since games that we play with machines need creators capable of negotiating with machines. Game creation is daunting for someone who doesn't code professionally. But more and more game-making tools are being designed with people who aren't professional coders in mind. (I describe several of these tools, and what each is good for, in the appendix.) It's now possible for people with no programming experience—hobbyists, independent game designers, zinesters—to make their own games and to distribute them online.

What I want from videogames is for creation to be open to everyone, not just to publishers and programmers. I want

games to be personal and meaningful, not just pulp for an established audience. I want game creation to be decentralized. I want open access to the creative act for everyone. I want games as zines.

It's a tall order, maybe, but the ladder's being built as you read these words.

Is What You Want Really What Games Need?

Why transform videogames, though? What do I get out of it? What, for that matter, do videogames get out of it?

In 2005, movie critic Roger Ebert infamously remarked that he does not think games can ever be considered as art. (By whom? By him, apparently.) He argues, mostly by assertion, that he doesn't feel game designers can exercise enough authorial control over the experience of a game. Ebert has gone on to make no attempt to justify or defend his remark or engage in any kind of debate, other than to allow, five years after the original remark, that he should have kept his opinion to himself.[7]

As I've made clear above, Ebert is wrong about videogames as a form. But frankly, I don't care whether Ebert is wrong or not. Achieving "artistic legitimacy" is not a good reason to transform videogames. Who legitimizes art? To cede the right to decide the value of games to an authority that has nothing to do with games—or to concede the right to decide what is and is not art to any authority outside of the artist—is a dangerous trap. Creation is art. It doesn't need validation beyond that.

What it needs is to be free. That an art form exists should be justification enough for people to be able to contribute to it, to work in it. We finally have the means to allow more than just programmers and big game publishers to create games—and the vast majority of people in the world aren't computer

engineers, or designers employed by Epic Games. What do we gain from giving so many people the means to create games? We gain a lot more games that explore much wider ground, in terms of both design and subject matter. Many of these games will be mediocre, of course; the majority of work in any form is mediocre. But we'll see many more interesting ideas just by the sheer mathematical virtue of so many people producing so many games without the commercial obligations industry games are beholden to. Remember, I'm talking about hobbyists, people who make games in their spare time with the tools they have on hand. And even if a game isn't original, it's personal, in the way a game designed to appeal to target demographics can't be. And that's a cultural artifact our world is a little bit richer for having.

To visualize this new world of games, think about network television versus YouTube. The former spends a lot of money and time creating content designed to appeal to the lowest common denominator. Because network shows need to justify themselves monetarily—they need to catch enough viewers to earn advertising dollars—they can rarely afford to be brilliant, daring, or bizarre. (Sometimes a director has enough force of will, and fights the network hard enough, to create a show that is all of these things. But it's certainly not the norm.)

YouTube: millions of videos from millions of authors. Most of them are mediocre: boring, familiar, or unwatchable. That's to be expected in an arena where everyone is allowed to contribute. But others are sublime, brilliant, valuable: Grishno's "Transgender in New York" videos,[8] wendyvainity's surreal computer animations and music,[9] or shaneduarte's *Simpsons* remixes.[10] As long as there's some sort of infrastructure, valuable works—those by both dabbling amateurs and dedicated

artists—can reach their audiences. YouTube has its own infrastructure of user ratings and featured videos, but people are just as likely to share the addresses of specific videos with the friends they think those videos will appeal to. And there's far more value in the collective content of YouTube—even given that there are more piles of trash than treasure—than in the collective content of a television network, simply as a function of the number of people contributing and the overwhelming volume of their contributions. YouTube's content is far more diverse, too, since involvement in the television industry isn't a requirement for entry. Network television shows are all made by professionals working in the field, a far smaller set of people than the set of people who own webcams. YouTube's content is made much more quickly and cheaply because it's not (usually) designed with a commercial agenda: videos can be recorded and broadcast, and their value assessed later.

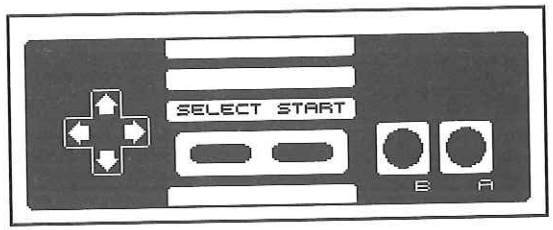
YouTube also gives people the means to make videos of themselves, their friends, their babies, and their puppies—video snapshots—not for the world at large, but for their social circles and themselves. YouTube is a means of transmitting a video directly from the author to an audience—one that can be as small and specific as the author desires. Videos become more specialized, and hence more personalized. A medium that was formerly accessible only to those with money and training can now be used by anyone for personal ends. If Internet television is in the process of reinventing television, imagine how game design tools for nonprogrammers and the free distribution of games online might reinvent videogames.

The Culture of Alienation

Limiting the creation of games to a small, exclusive group leads not only to creative stagnation, but also to the alienation

of anyone outside that group. I've described the round-the-drain cycle the games industry is in: games are designed by a small, male-dominated culture and marketed to a small, male-dominated audience, which in turn produces the next small, male-dominated generation of game designers. It's a bubble, and it largely produces work that has no meaning to those outside that bubble, those not already entrenched in the culture of games.

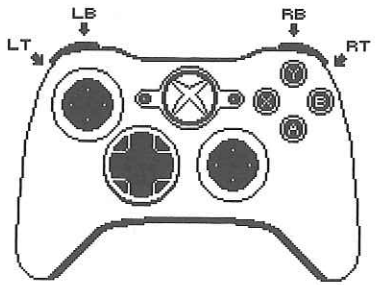
There are mechanical consequences as well. Look at how game controllers have changed as their audiences changed. The home game machines of the 1970s and '80s, which marketed themselves to large, general family audiences, had the simplest control pads. The Atari Video Computer System (or the Atari 2600) is a simple joystick with a single button. And here is the design of the Nintendo Entertainment System (NES) controller, released in the United States in 1985:



The NES controller has a four-way compass rose and two prominent red buttons. (There are also two buttons in the center for secondary functions like pausing the game, but the design of the controller clearly communicates that they're peripheral.) You use the compass to navigate your character or cursor. You use the buttons to perform actions.

After over thirty years of catering to an audience that is continuously playing and learning games—an audience that

hence requires more and more complicated games to interest it—games and the controllers with which players interact with them have become more and more complex. This is not to say *different*: layers of complexity have simply been added to the same few models of games and the same few models of controllers. Here's the controller for the Xbox 360, released in the United States in 2005:



The Xbox 360 controller is the same model as the NES controller: held between two hands, with navigational functions assigned to the left hand and manipulation verbs to the right. But instead of a single navigational pad on the left, two verb buttons on the right, and two option buttons in the center, the Xbox pad has a navigational pad plus a stick on the left, four verb buttons plus another stick on the right, four "shoulder" buttons on the top of the controller (two to each side), plus three option buttons in the center. (Additionally, some games call for the player to "click" either of the sticks in like a button, adding two more verbs.)

The means players use to interact with games guides the design of those games. A game for the NES might have a button for jump and a button for shoot, and the compass rose

directional pad for moving a character left and right. You can imagine the kinds of games that are designed for eight buttons and four sticks. Imagine introducing someone who had never seen a movie before to Matthew Barney's Cremaster films. The amount of both manual dexterity and game-playing experience required to operate a game designed for the Xbox 360 makes play inaccessible to those who aren't already grounded in the technique of playing games. And to attain that level of familiarity with games requires a huge and continuous investment of time (and money—keeping up with new games costs bucks). This means that older people—people with families and obligations, people trying to raise kids, or any people with a lack of free time to invest—have a harder time gaining access to games. At the same time, as a side effect of this unnatural selection, commercial games become longer and longer, with game covers advertising dozens and occasionally hundreds of hours of gameplay. (*Shin Megami Tensei: Persona 3*, a PlayStation 2 game from 2008, advertises a "70+ hour game" on the back of its box.) Who has that much time to invest in playing a videogame? Answer: the target audience of most of the industry's games, a mostly young and mostly male audience that has few obligations and plenty of disposable income.

The culture that this audience creates and exists within is one of in-jokes and brand worship, rituals to establish whether the participants are in or out of the tribe. It's an exclusive culture, an alienating environment that speaks only to itself. Its interactions with the outside world are decidedly hostile.

Destructoid, one of the most popular sources for videogame news on the Internet, employs a writer named Jim Sterling who once called my girl a "feminazi slut" on Twitter. This isn't some rogue nerd; this is a "journalist" whom *Destructoid*

employs to write on such topics as whether the penis is more powerful than the vagina because it can rape,[11] or on whether female *Mortal Kombat* characters have secret cocks.[12] And lest you think that such a character couldn't possibly be taken seriously, hundreds of his readers responded publicly to my open letter to *Destructoid* complaining about Sterling's behavior in an attempt to bully and shame us.[13] How is a woman, a trans person, or any rational individual expected to feel safe enough to participate in such a community?

What I want from videogames is for videogames to speak to more than just the handful of people already engaged in producing and consuming them. To de-monopolize game creation is to de-monopolize access to games.

Beyond Consumer

In an era when the Internet makes it easy to transmit and disseminate media, there's no reason for people to accept that their only contribution to the growth of an art form is as a consumer, supporting "elite" creators with money.

I've wanted to make videogames since I played Fukio Mitsuji's NES game *Bubble Bobble* as a kid. I drew characters on construction paper, cut them out, and laid out obstacle courses for them to navigate—*Bubble Bobble* stages on hardwood floors. But the technical leap to digitize my designs was beyond my reach. Programming was something mystical and arcane. I came into contact with code sometimes: the most basic BASIC examples. But something as simple as making a picture of a character move across a screen required a working knowledge of control loops, writing to video memory buffers, and advanced bit-shifting math—all of which was so inaccessible to me as a kid that I sublimated my childhood desire to make games until well into my adulthood.

It's not like it was then. There's a commercial product in videogame stores right now—*Warioware: D.I.Y.*, from Nintendo—that allows players to create their own small games.[14] What *Warioware: D.I.Y.* does is to introduce its players to the concept of designing rules, of using art and sound to communicate the state of the game to the player, of scripting the events of a game and of working cleverly within limitations. For kids today, digital game creation doesn't have to be the mystical process it was when I was little.

Kids today also have tools like Stencyl,[15] a free tool for making games and distributing them online. A website collects kits and resources contributed by the entire community, which are all made available to an individual creator for use in her game. The rules are put together in Scratch,[16] a system designed by programmers at MIT for young children to use. It involves snapping simple instructions together like LEGOs.

But before things like Stencyl and *Warioware* existed, I made games and digital stories however I could: an old DOS shooting-game creation program that I can no longer remember the name of, the track editor in Nintendo's *Excitebike*, an editor for creating worlds made out of text called *ZZT*. People with something to say will always manage to find ways to say it, and there's a history of clever people using whatever means they can find to modify and subvert digital games and to create new ones—to engage with games in a role beyond consumer. Today, this process is easier than ever.

The Big Crunch

This same false sense that the knowledge needed to create videogames is unattainable without special institutional training is the same carrot the Big Games Industry uses to entice wannabe game artists into taking jobs within their system—and

putting up with insane hours and ridiculous working conditions. There exists within the games industry a phenomenon called "crunch mode": working sixteen-hour days, staying at work until the game you're being paid to make is finished. This isn't something you're asked to do—it's expected, a standard condition of the job. And it's likely the reason most people in the games industry, their physical and mental health fizzled, burn out and quit within a few years, forced to retrain and find a new career. According to the International Game Developers Association (IGDA), the closest thing the industry has to an advocacy group for employees, 34 percent of game developers expect to leave the industry within five years, and 51 percent—half of them!—expect not to last a decade.[17] That's lunacy.

The industry gets away with this because it's convinced its employees that these jobs are the only gateway to videogame creation. "We've graciously allowed you to fulfill your childhood dream of making games. We're even paying you for it! And what's more, we're the only way you'll ever be able to do that." Mike Capps, a former member of the board of directors at the IGDA and the president of Epic Games said that Epic expected employees to work more than sixty hours a week and in fact only hired people they expected to be willing to do so.[18] The IGDA has no official stance on the hours of unpaid overtime the people it claims to represent are obliged to do by their employers.

Since the industry sees itself as ubiquitous—as the only possible means of creating games—it feels no need to change itself for the benefit of either its employees or its art. Which is another reason why carving new paths to game creation and distribution is valuable. By undermining the industry's claim to being the only route to game creation—especially to making a living from game creation—we force the industry to reconsider its totalitarian attitude toward the people it employs. Publishers need creative people to make games for them. We have one foot in an era when creative people will no longer need publishers to distribute their games.

Creating more and better games will also challenge the industry creatively. Spending millions of dollars to remake the same seventy-hour-long games for the same small audience is no longer feasible when so many people want different experiences out of games and have the means to find them elsewhere. Games from hobbyists have the potential to change the dominant format of the videogame: instead of seventy-hour multimillion dollar games that sell for sixty bucks apiece, digital games can be short and self-contained—less than an hour, short enough to fit comfortably into an adult player's day. The focus of games could shift from features, the ways in which a game is differentiated from similar games—*thirty hours of play, twelve unique weapons, advanced four-dimensional graphics acceleration*—to ideas. Take Tarn Adams' *WWI Medic*[19] for example: a game not about chain-gunning enemy soldiers but about trying to patch them up as the bullets cut them down. Saving even a single soul—climbing out of the trench, grabbing a fallen body and lugging it back to safety under a senseless hail of bullets—is incredibly difficult. The game takes minutes to play, and communicates an idea about war that may perhaps be more valuable than space marines frotteurizing each other with chainsaws.

Smaller games with smaller budgets and smaller audiences have the luxury of being more experimental or bizarre or interesting than 12 million dollar games that need to play it as safely as possible to ensure a return on investment. Imagine what a videogames industry that wasn't fixated on hits—that wasn't required to make hits—would create.

What Are Games Good For?

But even given all of this, why worry about the accessibility of digital game creation at all when other forms—like the short story or novel—are already established and available for non-professionals to work in?

Answer: because different forms are suited to different kinds of expression, and some are more effective at communicating in certain ways than others. Broadly, films and photographs are best suited for communicating action and physical detail. Novels are best suited for communicating internal monologue and ambiguity.

What are games best suited for? Since games are composed of rules, they're uniquely suited to exploring systems and dynamics. Games are especially good at communicating relationships; digital games are most immediately about the direct relationship between the player's actions or choices and their consequences. Games are a kind of theater in which the audience is an actor and takes on a role—and experiences the circumstances and consequences of that role. It's hard to imagine a more effective way to characterize someone than to allow a player to experience life as that person.

Take, for example, a game called *We the Giants*.[20] Most people who connect to this game's website in order to play it—taking the role of a squat, block-like cyclops—will be unable to reach the game's goal, a star high in the sky. Rather, most players are given the responsibility of voluntarily dying in a position that will allow future players to use their solidified bodies as steps in a staircase leading skyward. Each player guides her cyclops to the position of its sacrifice, presses a button, types a single message to future players of the game, and watches the cyclops's eye close forever. Thereafter, the player is never allowed to play the game again; logging on to the website, she can only watch the ongoing progress of the staircase of which her body is a part.

That's a pretty compelling way to explore themes of sacrifice in a work: to ask players actually to make a sacrifice, and to show them the meaning of that sacrifice over the course of generations. This is something games are almost uniquely capable of doing, and we haven't even begun to explore the possibilities of this kind of expression.

It's also the sort of experience—a minutes-long game in which the player is asked to commit voluntary suicide and never allowed to play again afterward—that is unlikely to come out of a commercial publishing system that needs its creations to sell millions in order to justify their having been made. The author of *We the Giants*, Peter Groeneweg, is a student and created the game as part of a monthly "experimental gameplay" challenge.[21]

The ability to work in any art form with the digital game's unique capabilities for expression shouldn't be restricted to a privileged (and profit-oriented) few. If everyone is given the means to work in an art form, then we'll invariably see a much more diverse, experimental, and ultimately rich body of work. In a speech at the 2007 Game Developers Conference, Greg Costikyan—a board and videogame designer and critic of the games industry—said: "I want you to imagine a 21st century in which games are the predominant art form of the age, as film was of the 20th, and the novel of the 19th."[22]

This is what I want from videogames, and this is what I'm trying to help you imagine. Throughout the rest of this book, I hope to help you imagine how this transformation of games—and the role games will play in the art and culture of the twenty-first century—is not only necessary, but inevitable.

Chapter Four
CHANGING THE GAME

Creating a game from scratch takes time, effort, and tools—tools that haven't started to become widely available until recently. Digital games are capable of presenting video and audio, and creating a game often requires that music be composed, sounds be recorded, art be drawn, or characters be animated. That involves a variety of tools and training and a lot of work. Often it requires a team of people, each with a different specialty, to accomplish a particular design.

And while plenty of hobbyist authors work on that level, the truth is that most digital games are not made from scratch. Most tell their stories by borrowing pictures or sounds or music or code. They use kits that other people have made; they use sound effects that other people have recorded and released for their use.

And many, many game creators, budding or otherwise, piggyback on existing works, taking advantage of existing infrastructure to make their creations. A commercial game from a large studio, for example, requires thousands of 3-D models, sounds, and assets to be loaded onto the player's computer. Even more important, a commercial game comes with game logic: a means of resolving collision, of determining the effects of gravity, of moving objects around a game

world. These things are difficult for someone inexperienced to code, but here a group of paid professionals have already done all the work. The player who's bought this game has all of those resources on her computer: why not use them?

In this chapter we're going to talk about what are usually called "hacks" and "mods" (for "modifications"): changing existing games to create new stories. There are a variety of ways in which people change games, and a variety of motivations for doing so. Sometimes changes are akin to crude vandalism, such as redrawing Super Mario so he has a dangling naked cock that flops around as the player walks through the Mushroom Kingdom. Sometimes they're akin to clever, subversive vandalism, such as changing the player character into the Princess and the prisoner who waits at the end into the Plumber. And sometimes they're something entirely new, almost indistinguishable from the games that gave birth to them.

Sometimes authors want to situate their creations in the context of a particular work. For example, Jesse Petrilla created *Quest for Al-Qa'eda: The Hunt for Bin Laden* in response to the September 11, 2001, attack on the World Trade Center, which the American media attributed to Osama Bin Laden. This game is a modification of *Duke Nukem 3D*, a first-person game about ultraviolent American culture in which the stripper-tipping, alien-gunning protagonist appears to the player only as two attributes: a gun that hovers in front of the camera and a series of prerecorded movie quips. The jockish American attitude of military vigilantism totally fits the play of *Duke Nukem*, a game in which the most logical response to everything the player sees is either to shoot it or to stuff bills between its tits. It makes sense that *Quest for Al-Qa'eda* and Petrilla's 2003 sequel, *Quest for Saddam*, should take the structure of that game.

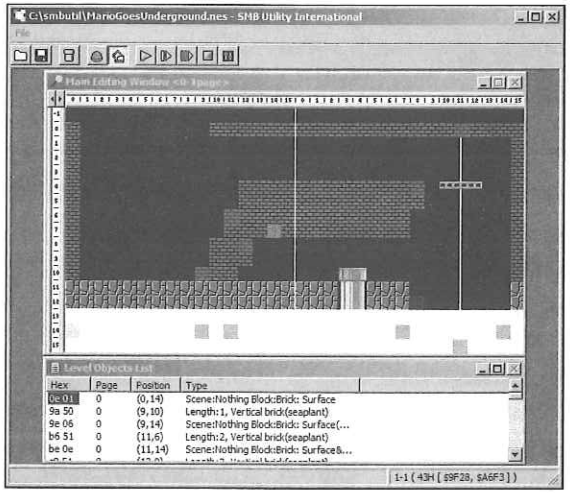
It further makes sense that the Global Islamic Media Front should, in 2006, reinvent *Quest for Saddam* as *Quest for Bush* (or *The Night of Bush Capturing*), a modification of *Quest for Saddam* in which the Saddam Husseins the player ceaselessly guns down have been replaced with George W. Bushes, an icon of American militarism. They're structurally the same game, with the faces swapped and the pictures on the wall changed in the simplest possible inversion of the original game's xenophobic aggression—a blunt "how do *you* like it?"[40] Hacks and modifications are made to subvert or comment on the original author's intentions, or to simply correct what the modder feels is an oversight on the author's part, or simply because it's an easy existing infrastructure for creating something new. Let's examine the varied motivations for hacking a game by looking at the most-hacked digital game: *Super Mario Bros.*

Another Castle

Super Mario Bros. was published on a cartridge that plugs into the Nintendo Entertainment System. Hardware has existed for a while, however, that allows the contents of a Nintendo cartridge to be "dumped" to a computer and distributed digitally as software. The digital file is called a "ROM," and "emulator" programs have been written for many different machines that will run that game software as a Nintendo Entertainment System would. In this way, *Super Mario Bros.* has made the transition from hard media to a purely digital form, available for modification by anyone with the knowledge and tools.

And tools have been readily available for a long time. *Super Mario Bros.* was written in Assembly, the machine language I mentioned in chapter 2, and reprogramming the game requires working knowledge of Assembly. But programmers cracked the code a long time ago, and plenty of tools currently exist for changing the appearance, rules and levels of *Super*

Mario Bros. without having to look at machine code. When I made my 2008 *Super Mario Bros.* hack, *Mario Goes Underground*, I used a tool called SMB Utility[41] that made it easy to rearrange *Super Mario Bros.*'s maps.



Super Mario Bros. has enjoyed popularity with ROM hackers for a variety of reasons. For one, the large number of people who've played the game made it an early target for programmers to crack, and tools have existed for a while. It's also widely played enough that most people have a sense for how the game works: move left, move right, jump, catch a mushroom to grow big. Most players have an idea of what to expect from *Super Mario Bros.*, so they can pick up a hack easily and be surprised effectively when the hack turns out to be different from the original.

Some hacks simply change the game in a superficial way. *Silhouette Bros.*, by Leon Arnott, changes all the characters

and architecture to stark black silhouettes on a solid-color background. The hack demonstrates how recognizable the elements of *Super Mario Bros.* are and the choices of color are well suited to the tones of different parts of the game. *Enigmario* by Dr. Floppy replaces the soundtrack of the game with covers of songs by the band Enigma. The somber tones of this new soundtrack give *Super Mario Bros.* a more introspective feel.

Mario vs. Airman is an interesting attempt at intertextuality: the "Airman" Mario fights is a villain from Capcom's *Mega Man 2*, and Mario is required to navigate a level from this game to defeat Airman. Mega Man, as a player character, moves very differently than Mario. Momentum does not affect his motion, he can "brake" instantly, and his arm gun makes him able to shoot his opponents, rather than requiring him to jump on them to defeat them. Trying to navigate a level designed for Mega Man as Mario is a challenge that emphasizes the importance of the nuances of motion in platform game design.

A hack like *Super Mario Forever* exists to challenge the player's technical knowledge of *Super Mario Bros.*'s rules. Mario starts mid-fall into a bottomless pit, and must instantly maneuver to land on a tiny platform. The game then requires the player to generate enough momentum on that tiny platform to make a long jump to the next, which involves jumping off of and back onto the platform in order to build speed.

Some hacks change a single rule of a familiar game in order to give the game a new experience, similar to the concept of "house rules" in non-digital games. Normally, the player of *Super Mario Bros.* must complete each stage within a limited time, which resets at the stage's end. Nanashi's *900 Time Challenge* gives the player a single time limit in which to complete the entire game: nine hundred ticks of the game's

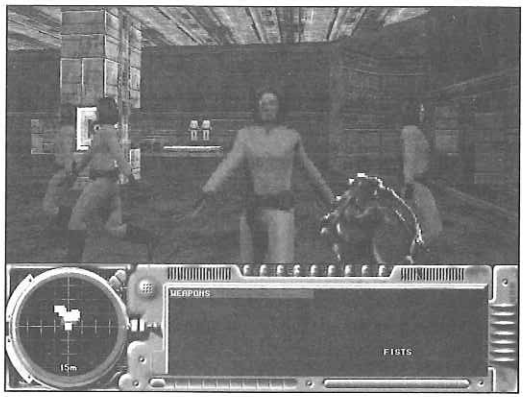
internal clock. Completing the challenge requires the player to minimize the wasted time in her play, and to exploit shortcuts and warp zones in order to get through the game in as fast a time as possible.

Then there are hacks like *Extra Mario Bros.* (or my own hack, *Mario Goes Underground*), which attempt to tell an entirely new story using the familiar framework of *Super Mario Bros.* Rather than requiring Mario to proceed from the starting point on the left to the castle on the right, *Extra Mario Bros.* contains a big world for the player to explore in multiple directions, using pipes to travel vertically from area to area. It's essentially *Super Mario Bros.* through the lens of contemporary game design, which is why it's interesting as a direct modification of *Super Mario Bros.*: the games that informed *Extra Mario Bros.* are all descended from *Super Mario Bros.*

Then there's *Super Daisy Land*, which changes the star of the Nintendo Game Boy game *Super Mario Land* from Mario to Daisy, a captured princess who normally waits at the end of the game for rescue. But that's not exactly a hack of *Super Mario Bros.*, and there's a richer subject for gender correction in game mods.

Bobs Whose First Name is Betty

Anne-Marie Schleiner, artist and writer, maintains Mutation .fem, a gallery of modifications of first-person shooting games: games where the player's character is personified as the gun she holds in front of her, like the earlier-mentioned *Duke Nukem 3D*. Each mod changes the gender of the characters in those games. The patches hosted on Mutation.fem[42] change male characters to female for a variety of reasons: either for the purposes of gender play, to correct gender bias by the games' authors, or to subvert the aggressive masculinity of this school of game creation.



The first mod in Mutation.fem that I played was Lynn Forest's *FemDOOM*. The original *DOOM*, a 1993 release from id Software, was one of the earliest games to popularize the first-person shooter, and has given tropes to that genre that are still imitated today. The game is viewed from a first person perspective, and the protagonist is given no name or background beyond being a space marine assigned to a tour of duty on Mars. Only the tiny, emoting face of a man in the status bar and the occasional baritone grunts when the player is shot gender the protagonist as male, and reinforce the tacit assumption that the player of a game of aggression must be male.

Lynn Forest, a fan and player of *DOOM*, was frustrated with the implication that she was playing a game for men. So she drew femme faces to replace the male faces that appear in the status bar, and put her patch online for other players to download. A later patch, "FEMDMSND.WAD," credited to "Amanda, Ivor B. and Rob Lord," replaces the masculine sound effects with feminine ones of equal intensity.

Gender is even more of an oversight in id Software's later *Quake*, which puts an even greater emphasis on competitive play. In *Quake*, one sees other players not as masked figures, but as hulks with a single male face. Players modded the game to introduce feminine faces and bodies. Schleiner notes[43] that later versions of *Quake* acknowledge players' desire for female avatars by including female models. She also hosts "skins" worn by PMS Clan, the Psychotic Men Slayers,[44] a band of women players who play competitively together in the game *Quake 2*.

Mutation.fem contains patches that paint a mustache on *Tomb Raider*'s protagonist Lara Croft and change Bungie's *Marathon* from a game about firing guns in dark, angular space stations to a game about fighting with egg flippers and dish towels in a checkerboard-tiled kitchen. Loren Petrich's "Tina-Bob" patch replaces the generic jumpsuited men—the "Bobs"—that solely populate *Marathon*'s space station with identically jumpsuited women.

The Hack as Sampling

The repurposing of commercial game assets can be compared to sampling in music: using part of an existing song as an instrument in your own piece of music. Hip-hop artists often rap their own material over music sampled from another source, using the found music as a background for their own words.

HyperBound was created in 2009 by Michael Iantorno as part of his thesis project at Ryerson University.[45] Iantorno's game is a hack of *EarthBound*, a Super Nintendo Entertainment System game developed by APE and the Japanese writer Shigesato Itoi and released in 1994. *EarthBound* is a digital role-playing game informed by Yuji Horii's *Dragon Quest*,

but instead of telling the story of a warrior who battles wolfmen and dragons in a fantasy world, *EarthBound* is set in a world vaguely resembling a Japanese vision of contemporary America. The protagonist is a young boy in a baseball cap who fights aliens and renegade animals using an old baseball bat and his newly discovered psychic abilities.

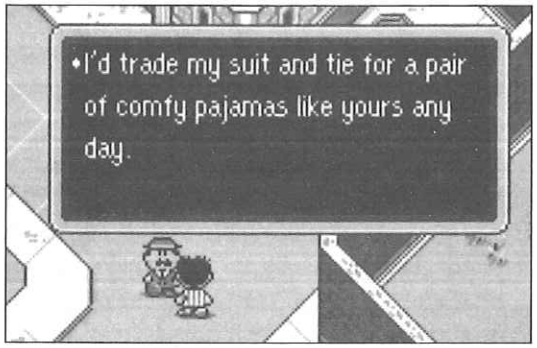
HyperBound takes its name from "hypertext," text that's arranged in a nonlinear structure. (This book is a text: it's arranged to be read from start to finish, one page to the next. A website, where you might click on a word to "link" to a page about that subject, is hypertext.) What better model for the nonlinear exploration of text than the space of a digital game, where the player moves around the world by moving her character across the screen, encountering characters, and listening to what they have to say? That's the part of the design of *EarthBound* that *HyperBound* has lifted. What it's rejected is the fighting. The hack is purely about exploring the world and discovering the text, an original script written by Iantorno and his brother.

In *EarthBound*, the protagonist typically wears a baseball cap, a striped shirt, and a backpack. There's a brief scene at the beginning of the game, though, where he appears in pajamas with ruffled bed-hair. That means that there were animated sprites of the bed-headed protagonist, seen from all angles, just waiting in the game for Iantorno to use. *HyperBound* uses those sprites for its protagonist, to give him an identity apart from that of *EarthBound*'s protagonist and to emphasize his confused state: the protagonist is amnesiac, and the information he is trying to recover is knowledge of his own identity.

Throughout the hack, Iantorno repurposes assets from *EarthBound* to fit his new story. The bearded, sunglasses-wearing criminal the player encounters in *EarthBound* becomes

the radio DJ whose show the protagonist of *HyperBound* used to call in to before he lost his memory. The boarding school that appears in *EarthBound*, with its classrooms and lockers, becomes the university that *HyperBound*'s protagonist attended, where he meets former teachers and finds valuable information on his previous life. All of *EarthBound*'s graphics are sampled and given new purposes in the hacked game.

The other assets Iantorno samples are less obvious, though just as crucial: the ways in which the player presses controller buttons to move the protagonist around a world, to engage characters who wander around of their own accord in conversation, and to advance their dialogue text on the screen. These are all things that are non-trivial to program, and that have been programmed for Iantorno by APE. To present his nonlinear story, he's taken advantage of an existing infrastructure for allowing the player to navigate a world of characters who can be spoken to.



If you can create a script but you can't create animated characters, scenery, or code to take player input and move a character around a large world that scrolls in eight directions,

why not use those existing assets to present your original script? The work has already been done and the resources already exist to be sampled.

Changing the Script

There's another obvious reason to change the script of a game: to translate it to a new language. Publishers often neglect to translate and publish a game in a new language because of the cost: if they think a game won't sell enough in a part of the world to justify paying people to rewrite the script, to manufacture cartridges for different game-playing hardware, and to market the game in another country, they won't do it. But dumped roms, which are purely digital, don't require the expense of physical publishing, and because they're distributed freely, don't depend so heavily on marketing to make up that expense. And thus, people who read languages neglected by videogame publishers and care enough about games from other parts of the world will invest their own time and effort into the unpaid translation of games.

The work of translating a game involves more than just translating the script—which is often a lot of work, given that the dynamic nature of games usually means lots of scenes and incidental text to translate. The new script also needs to be inserted into the original game. In formats where space is expensive, such as Nintendo game cartridges for example, programmers use a lot of clever tricks to store text as efficiently as possible. Cracking those codes can be tricky. And if the original game was written in Japanese, the hacker also has to insert a new font. The way the game displays text might not be suited for English letters, so screen space is usually at a premium, too. Japanese is a compact, ideographic language, and a word that takes up two Japanese characters on

screen could be an eight-letter word in English, or an idiom that requires translation as a full sentence. How do you translate the word so it fits? All game text tends to be presented in a limited part of the screen (a window that takes up the bottom half of the screen, for example, so the player can see the characters who are speaking on the top half), so all word choice is informed by just how much space is available to display those words—space that was chosen based on the structure of an entirely different language!

The Japanese Game Boy Advance game *Mother 3*, a sequel to *EarthBound*, was translated into English over the course of two years by a two-man team who refer to themselves as "Jeffman" and "Tomato."[46] The script for the game was written by the Japanese writer and journalist Shigesato Itoi. Nintendo's American branch decided that translating the game would be too much work, and too expensive a project, given the limited audience they expected the game to have. Nine years before, Nintendo of America put a lot of money into translating *Mother 2* into *EarthBound* and manufacturing copies, only to have their marketing department completely mishandle it. Physical distribution is a big investment, after all: Nintendo of America was unwilling to make the investment a second time. American players who had encountered and liked *EarthBound*, though, were eager to play the sequel and were upset that it wasn't to be published in English. Tomato, who works as a professional Japanese-to-English translator, played the Japanese game and decided that English-reading players deserved to experience the game. He undertook the translation of the game, and Jeffman did most of the coding to get around the memory limits of the original Japanese game: a two-year project, undertaken for free. The English version of *Mother 3* was released in July 2009. You can go on the official

translation site's forums to read how many players cried during the ending[47] (which is, incidentally, Stephen Spielberg's metric for when games will become art,[48] not that tears are anywhere close to the only metric for judging the value of an experience).

But don't be misled into thinking that the majority of translation work is in translating Japanese games to English. As of this writing, people are working to translate *Mother 3* into Spanish and Latin American Spanish, Portuguese and Brazilian Portuguese, French, German, Italian, Dutch, and Malay. Whenever I visit the ROM hacking database at romhacking.net, I see translations of English games into German, Polish, Russian, Korean. Hackers are engaged in correcting the oversights of profit-oriented thinking and making the games they care about available in their native tongues.

Machinima

The product of "machinima"—from *machine* and *cinema*—isn't games, but the source material is. It's the same kind of sampling I've described: machinima uses the resources and infrastructure of commercial games as the basis for creat-

ing animated movies. All the resources are there in most 3-D games: the ability to move a camera through a world full of 3-D objects, and to move and animate those objects. Game hacking provides an immediate avenue to 3-D animation, one that's far more accessible than dedicated 3-D modeling (with a specialized—and costly—modeling program like Maya or 3D Studio Max) because there are already scripts in place to move things around and to operate a camera.

The origins of the community surrounding machinima can be traced to that of the game *Quake*, a fully 3-D first-person shooting game that offers players the ability to record, distribute, and play back "demo" recordings of their games. Players began to manipulate and position the camera during these playbacks, introducing cinematography to the demos. Soon they were directing story scenes to frame their demos.

Most machinima stars the characters of the games they sample, because that's what's available: usually armored soldiers who crack jokes for the benefit of players familiar with the game. The 1996 *Diary of a Camper* by Matthew Van Sickler, Heath Brown, and company,[49] for example, follows a band of *Quake* characters as they deal with an opponent who is hiding and picking off their comrades one-by-one. (And who is ultimately revealed to be John Romero, one of the founders of id Software.) *Diary of a Camper* is widely remembered as the first game demo interested purely in storytelling, rather than presenting a sample of play footage.

But the subject matter of machinima has become increasingly diverse. The 2003 machinima *Anna*, produced by Katherine Anna Kang's (wife of *Quake* programmer John Carmack) Fountainhead Entertainment, follows the life of a single flower on a forest floor.[50] *Anna* bears little resemblance to *Quake* at all, set in the woods with a cast of plants and animals.

As more tools become available, machinima looks less and less like videogames. A few years ago I encountered a movie of Sherwood Anderson's short story "The Dumb Man" filmed in *Second Life*,[51] a game that gives players the tools to construct anything they like, be it movie sets or actors or other games.

While machinima is not games, it's demonstrative of the ways that game tools can be used to create things radically different than the originals: including new games altogether.

Something Borrowed

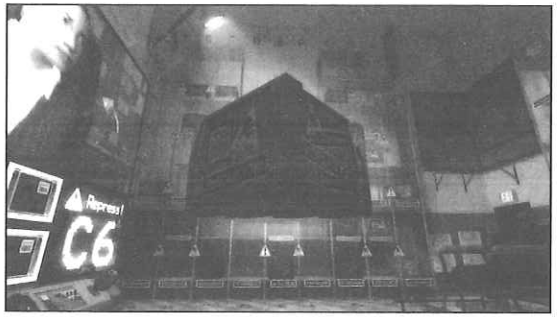
Few commercial games, these days, are made entirely of original resources. The creators of the *Quake* and *Unreal* "engines," the rules and code that drive the games, license their engines to other developers so that they can save time and work by modifying an existing framework rather than building an entirely new one. The *id Tech 3* engine—the engine that runs the third *Quake* game—was used to create games like *American McGee's Alice*, *Call of Duty*, *Star Trek: Voyager—Elite Force*, and *Star Wars Jedi Knight II: Jedi Outcast*.

The publishers of these games paid for licenses allowing them to sell and distribute their modifications, but plenty of players, using their knowledge of how their favorite games work, changed the games into something new and distributed their modifications for free.

Brandon Chung's *Citizen Abel: Gravity Bone*[52] was built in a modified *Quake 2* engine, but the finished game bears little resemblance to a game about space marines blowing up aliens. *Gravity Bone* is a spy pastiche that opens with the player riding a fancy gated elevator into a masked costume party. Once there she must make contact with a fellow agent. This sets forth a rapidly accelerating chain of events that lead to an unexpected outcome.

What *Gravity Bone* takes from *Quake* is logic for moving a player around a three-dimensional world, looking around with a first-person camera, and collecting and manipulating items. Everything else—the game's appearance and presentation, the locations and scenario, the player's goals and the events they trigger—is original, designed and inserted by Brandon Chung. *Quake*, here, is just a vehicle for the delivery of Chung's original story.

Or look at Robert Yang's *Radiator* series,[53] which is built in the *Half-Life 2* engine. The second episode of the series, *Handle With Care*, involves the player visiting a marriage counselor. What *Half-Life 2* does well is allow the player to manipulate and move objects using a rudimentary physics simulation: in *Handle With Care*, the protagonist, James, withdraws to a "Repression Facility" while his husband speaks with the counselor. The Repression Facility looks like the kind of grimy steel structure common to *Half-Life 2*. In the facility, the player carries crates containing repressed memories (if there's one thing *Half-Life 2* does well, it's letting the player carry crates around). If they're put on the facility's shelves, they're locked away forever; if they're broken, Jason revisits the memory he was trying to repress. The therapy session, seen through a monitor, progresses depending on whether Jason confronts or seals away his memories.



Both of these games look significantly different than the games they're based on. Nevertheless, we can identify rules they've borrowed from their mother games. It's apparent why it's easier to change existing games into new games rather than creating them from scratch: particularly when it comes to 3-D games, solitary creators often don't have the resources that the programmers of *Half-Life 2* had. It's unlikely that either game could've existed without such sampling—it's far less likely that Valve would make a game about the roles individual memories and pasts play in negotiating modern queer relationships than a *Half-Life 2* level involving building a wall of crates to stop drone planes from killing you.

Scripting cameras and collision, as well as creating 3-D architecture and a way of storing it conveniently in computer memory, are all big projects. The Valve Hammer Editor, which edits *Half-Life 2* levels, gives a creator immediate access to all the design tools a team of paid programmers put into place. It also gives her access to the resources *Half-Life*'s art team created, which would otherwise be another huge investment of time, effort, and skill. Editors like Hammer allow creators to create without having to build an infrastructure to manage their creations.

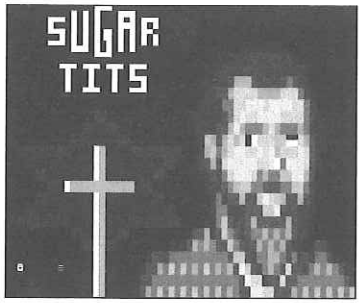
New Worlds

Many game designers have anticipated their players' desires and created editors to accompany their games. The first level editor I ever encountered was for the Nintendo game *Excitebike*. *Excitebike* is a stunt bike race. The included EDIT MODE allows players to mix and match the obstacles from the game—ramps, bumps, gravel pits, and boost pads—to create new stunt tracks to race on. When I was little, I spent a lot of time with the instruction book open in my lap to the list of pieces, moving my racer through the track and rearranging the obstacles. The appeal of an editor like this to a child who's creative but lacks experience is that everything—the rules, the code, all of the art and sounds—are provided. The only thing I had to worry about, as a kid, was the track itself.

When we were discussing shareware, I mentioned Tim Sweeney, who founded Epic MegaGames—which later became Epic Games, the publisher of the *Unreal* engine used by many commercial and independent designers to build 3-D worlds. The first game Tim Sweeney sold, as "Potomac Computer Systems," was released in 1991 and is called *ZZT*. (He picked this name so it would always be the very last game listed on alphabetically ordered shareware CDs.) The shareware part of *ZZT* was a series of four self-contained adventures, the first of which came with the game and the other three of which had to be purchased. (They were later released for free.) The free part of *ZZT*, however, was the editor that Sweeney used to make those adventures, and the means to play "worlds" created with it.

ZZT is a text-mode game: it takes advantage of DOS computers' native ability to display text. In addition to letters and numbers, DOS has a special extended sheet of characters for drawing basic pictures: smiley faces, playing card suits, Greek

letters, and a variety of borders, lines, and patterns. *ZZT* co-opts these: the player is a smiley face, Greek characters such as pi and omega are monsters who roam around, the diamond from the playing card suits is a gem that the player can collect and use to buy items. This set of 255 characters comprises everything that any *ZZT* world will ever contain.



And that's incredibly liberating. It means that *ZZT* is self-contained: there's no need to go outside the editor to find another tool to draw and animate graphics to import into the game. Authors get 255 characters in sixteen colors. And creative authors can do a lot with those: in that set there are patterns of varying density and lines of varying thickness, and I've seen some pretty impressive environments and portraits drawn with those text characters.

Sound and music are similarly self-contained: *ZZT* uses the PC speaker, which contains a range of simple notes and instruments. *ZZT* has a simple scripting system for playing PC speaker tunes, which again obviates the need to go outside the world editor for anything.

But oh, yes: let's talk about scripting. *ZZT* has a nice library of game pieces, which appear in Tim Sweeney's shareware *ZZT* worlds. There are pushable boulders, "rotators" that

move objects around and can form conveyor belts if a series of them is placed in a line, stock monsters, weapons, and power-ups, and several kinds of walls (invisible walls, secret passages, walls that can be shot and destroyed). But the most interesting game piece is the Object, a programmable creature that can take on any appearance. Each Object contains a script written in a language called ZZT-OOP (ZZT Object-Oriented Programming), a simple programming language.

Here's an example of ZZT-OOP:

```
/n/n/n/n/n/n
#shoot e
Haha, I shot your favorite PRICELESS VASE!
```

That tells an Object to move north (or up) six times, then shoot a bullet east, then display a haughty message on the screen. That's pretty simple, something someone who's never coded before should be able to understand. Though it's simple, ZZT-OOP is also surprisingly robust: each Object has its own script, and can receive messages from the player and from each other. Here's a more complicated example of two Objects interacting:

```
@Priceless Vase
#end
:shot
#send Collector: broken
#die
```

```
@Collector
#end
:broken
You broke my favorite PRICELESS VASE!
#char 1
/i
#char 2
/i
#char 1
/i
#char 2
```


When the vase is shot (the colon indicates a message the Object can receive) it sends a message to the object named Collector (the @ indicates an Object's name), which causes the Collector to display a message and briefly animate (#char is the "change character command"; character 1 is a hollow smiley face and character 2 is a filled smiley face. The direction "i" means "idle"; it tells the object to wait a moment between flipping its character).

Authors have taken advantage of ZZT-OOP to do things author Tim Sweeney never envisioned. One can create a *Super Mario Bros.*–style jumping game by creating button Objects that, when touched by the player, send messages causing a Super Mario Object to move left or right or jump, and that falls when there's nothing beneath it. Recently, someone built a machine that graphs sine waves in *ZZT*. Creative authors have bent the tools they were provided with to entirely different ends than they were intended for.

What's interesting about *ZZT* is not only the robustness and versatility of ZZT-OOP, but the simplicity of the game's overall presentation. Because all of the graphics and sounds that a *ZZT* world can produce are already inside it, it frees authors from having to worry about asset creation. They can just design with the tools they have in front of them. This makes it an ideal introduction to scripting and design, and I

know a handful of game designers who got their start making *ZZT* worlds.

House Rules

There's a concept in non-digital games called "house rules." Because players keep the rules in board and card and physical games, and because all the players have to agree on what rules they'll play by, it's easy for the players to change the rules or invent new rules to suit themselves. Maybe they feel as though their modification will correct an oversight on the part of the designer: part of the game may not quite work as it's presented. Or perhaps they've thought of a variation on the rules that's more interesting to them as veterans of the game. Maybe they need the game to be easier or harder, or to change it so it better suits the circumstances in which they're playing it: a game of Tag where the goals change to suit the space that's available, for example.

In digital games, the rules are kept by the computer and thus are somewhat hidden from the player. Adding house rules to a digital game seems more far-fetched. There are plenty of ways, though, to change games: changing the rules, inventing new rules, inventing new scenarios and whole new games out of the same content. Telling new stories!

What's really valuable about hacking and modifying games is the realization that there are ways of interacting with games other than just playing them: roles beyond consumer. Inventing rules is, after all, inventing games. The knowledge that games can be changed and remixed is the knowledge that games can be created: it's a small conceptual leap from the one to the other. While game creation tends to be more interdisciplinary—using the Valve Hammer Editor allows the creator access to all of the 3-D models, textures, sounds, characters, and such from *Half-Life*, letting her focus on design—there are game creation tools now that are as easy to approach as the typical level editor.

I said that the first level editor I encountered was *Excitebike*'s. I've played with plenty through the years: *ZZT* and its sort-of successor *MegaZeux*, *Dezaemon 3D* (a spaceship shooter kit for the Super Famicom), *Knytt Stories* (a jumping-and-climbing story creator), *Tombs of ASCIIroth* (a text-mode puzzle game kit), *Ragdoll Masters* (rubber dolls fighting each other in the air), *The Elder Scrolls IV: Oblivion* (a fantasy computer role-playing game). I've used whatever tools I could get my hands on to tell stories.

You can, too. And once you've realized that videogames—even if they're made by big corporations with teams of hundreds and budgets of millions—are mutable, and can be reinvented by a single person, you can start to imagine what's possible when you have the means to create games that are entirely your own.

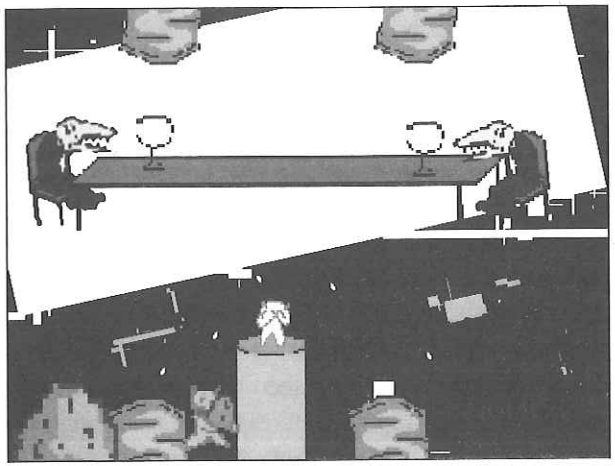
Chapter Five
THE NEW VIDEOGAME

The *La La Land* series was created by Matt Aldridge under the alias "The Anemic." It's a series of five games, each supposedly based on one of the author's dreams. The games are populated with recurring scrawl-doodle characters whose features are abstract enough to be sinister. The protagonist, Biggt (pronounced like "bigot"), has a smile that's Glasgow-wide and a pair of eyes that don't line up. The music is often seemingly incongruous: the climax of *La La Land 5* happens with the anti-evolution song "I'm No Kin to the Monkey" playing in the background. And the arbitrary goals the games give the player are almost always revealed to be against the protagonist's best interest.

The first one I stumbled upon was *La La Land 2*. In this game, Biggt is confronted by a towering snaggletooth fish head, baby pink and jittering, which tells him: "it's ok to steal from the rich cos they havelots [sic] to spare!" A counter—which in most games would be nailed to a fixed point on the screen, but in this game jiggers and jumps from point to point—reads "$0.00."

Biggt travels left to a screen where crow-like "nobles" with fish heads sit at opposite corners of a long table. A figure that resembles Biggt is on a pedestal, face in his hands. As he cries, some of the tears pass through pipes and fill the nobles' wineglasses. At the far left is a hay-colored, shapeless pile that

represents gold. Biggt scoops up an armful, making the pile smaller, and carries it back to the initial fish head, adding money to the counter. The fish head says, "Oh thanx biggt i want MORE."



The player runs back and forth between the fish head and the nobles, redistributing wealth. When the player has stolen enough money, the fish head rises up to reveal a gold medallion hanging from one of its bones. "Look at the shini JEWELS I bougt thanks. a lot." Returning to the nobles, the player finds them dead at their table. "without money how were, the nobles to surivve?" The game then closes.

Personal and abstract, The Anemic's *La La Land* games were the first games of their kind that I encountered, and they would never have been made if not for Game Maker.

I'm not sure when *La La Land 2* was created, but in 2003, when I was seeking out the other games in the series, I discovered a game called *Seiklus* by an admirer of the *La La Land*

games. The author, cly5m, had made worlds for Tim Sweeney's *ZZT* that were as dreamlike as *La La Land*, if a bit larger in scope. (One, *Kudzu*, involves making a collect call to heaven at a melting payphone and visiting a hall filled with imprisoned invisible people.) American by birth, cly5m had visited the country of Estonia on a religious mission, then returned to America and discovered Game Maker. *Seiklus* is the Estonian word for "adventure."

Seiklus begins with a boy and girl watching the stars on a cliff. A meteor hits and causes the boy to fall into a valley below. The game ends when he has found his way back to the girl. To return, the player has to collect colored tokens, which are found in a giant hollow tree, on a quiet, snow-covered landscape, in a dark land full of bones, and inside the stomach of an enormous creature. All of these scenes were hand-drawn by cly5m himself; the thick line art style is as unmistakably his as the collection of pastoral-alien dream scenes.

To the West

From my current perspective, I can say that what I always wanted to do was make videogames, but as a child, making videogames was something mystical that I never thought I'd be capable of it. So I funneled my creativity into other channels. I was better at writing than drawing, so I was a creative writing major in college.

After playing *Seiklus*, I figured that if cly5m could make the transition from making add-ons for *ZZT* to making self-contained games in Game Maker, then I could do the same.

I had taught myself to program a little, particularly in a language called Blitz Basic, in which I had recreated a few games that I knew. I had made a board game–like version of *PAC-MAN* where the player and the ghosts took turns moving rather than

moving at the same time. Making objects move and react to each other all at once was something that was beyond me, and objects' placement on the screen had to be hard-coded, making it difficult to experiment with actual design. Game Maker was refreshingly similar to *ZZT* to me: every individual object had its own instructions to tell it what to do and how to react to the other objects. I could draw every object, give it instructions, put it wherever I wanted to on the screen, and let the program play out and resolve interactions based on how I'd told the objects to behave.

My first Game Maker game was called *Jaywalker*. In *Jaywalker*, the player controls a disembodied head named Marjorie with a HotHead Paisan–style crown of orange hair and a stud in her nose (I think I made the game around the time I first got my nose pierced.) Marjorie is fiercely about the rights of pedestrians, and is committed to destroying as many cars as possible. The player moves her back and forth across a busy intersection, causing cars to swerve around her and crash into each other. Cars will crash into cars that have already been wrecked, creating huge pile-ups and causing Marjorie to stick out her tongue triumphantly. If she's ever hit by a car, the game is over.

I dropped out of college some time after creating *Jaywalker*. When I quit, I was in my fourth year of school with no end in sight. I didn't see myself as a writer, I hated school, and I thought (and think) that "higher education" is bullshit.

I spent another year in New York, where I was born. I made some more games during that time. Terrified I'd be stuck in the same spot for the rest of my life, I decided I needed a reason to move somewhere far away and a way to earn rent, a trade that I could learn while I was there. The thought of earning my living by making games was exciting: why not go to a

games college in another part of the country, perfect my craft, and find a way to pay my rent by making videogames?

Specialized videogame schools had been around for a while at this time. I had known of DigiPen, in Seattle, since I was a teen, and had recently heard of Full Sail, a movie school in Florida, but both of those seemed to me like game programs tacked on to visual arts schools. I chose the Guildhall at Southern Methodist University because it was the only school I discovered that had a level design focus. (A Game Maker game I made after quitting creative writing school, *Invader*, which was about an alien from the game *Space Invaders* crash-landing on a weird planet and having to find her way off, was part of my portfolio.)

The Guildhall was in the middle of Plano, Texas. Plano, Texas, is brown and not much else. They have a Frito-Lay factory, parking lots, and a videogame school. At the time, I kept a strict vegan diet and didn't drive. There was nothing to eat and nowhere to go.

But the latter didn't matter; when you were at the Guildhall you had no life outside the Guildhall. I remember the first day of orientation, sitting in a lecture hall with my future classmates and the spouses they'd brought with them to this wasted brown land. One of the other level design students had his wife and their year-old child with him. "Give her a kiss and say good-bye," the director of the school told him in front of the assembly. "You're not going to see her for two years."

I was in Plano, Texas, for six months.

You're at school from nine to five. You stay after and do your work with the teams they've assigned you to. Late at night you drag yourself home and do your actual homework. Maybe you get a few hours of sleep. The idea behind the school is that you're always in what the Big Games Industry

calls "crunch time": unpaid overtime. Your masters want the game done by Christmas, so you don't leave the office until it's done. This is why people in the industry aren't healthy; this is why they burn out and quit games within a few years. This is why you miss the second year of your daughter's life. This is their scheme: you put up with crunch time all the time while you're in school, so when you work for a big publisher—or, rather, a studio contracted by a big publisher—you won't complain about being told you can't see your daughter until the game's done. The Guildhall boasts an over 90 percent employment rate, and it's true: they will get you a job in the games industry. That's because they will make you into exactly the kind of worker the games industry wants. It's that kind of school.

And it works; that's the horrifying thing. My classmates were all self-identified gamers and game fans and were willing to put up with anything in order to live their dream of making videogames. That's the carrot the industry dangles, and it's what we take away from the industry when we create a form to which anyone can contribute. As long as the industry is allowed to continue acting as the gatekeeper to game creation, people will continue to accept the ways in which the industry tramples the lives and well-being of the creative people who make games, rather than challenging the insane level of control that publishers ask over developers' lives.

Needless to say, I was not at the school long. I butted heads with lecturers too many times, I asked too many questions, and my *Oblivion* mods were too experimental (I didn't include messages that popped up and told the player she'd achieved her goal because I didn't think the player was too stupid to realize that.) Eventually I was pulled into the director's office and asked to leave. "If your unit is marching to Fort Worth, you don't ask

'what if we go to Austin instead.'" That's a paraphrase of what the school's director told me.[54] I've had a while to think about this, and I've since decided that if you know there's an army waiting to ambush you at Fort Worth, it's fatally irresponsible not to try to dissuade your comrades from marching there blind.

I said that I was in Texas for six months. Each semester at the Guildhall was just over two months long; they canned me the first day of the third semester. I spent the last month or so getting my stuff moved back to New York and making a game called *Calamity Annie*.

Calamity Strikes

Stranded in Plano, Texas, with everyone I knew occupied with school all of the day and most of the night, working on *Calamity Annie* is what kept me from going stir-crazy until I flew back to New York.

The protagonist and namesake of *Calamity Annie* is a brash young dyke who rolls into the Old West (from the New West, naturally) with her breasts bound and a pistol in her holster. She tears across the countryside, dropping hombres in one-on-one shootouts. Along the way she meets a lady named Valentine with a different past but the same breed of loneliness. If the player's sharp enough, Annie and her Valentine ride off into the sunset together at the end.

The whole game is played with the mouse, by pointing to aim and clicking to shoot. A duel consists of "holstering" your cursor at the bottom of the screen, waiting for the call of "DRAW!" and quickly aiming and firing before your opponent can do the same. Interactions with Annie's love interest use the same controls. The first time you meet her, Valentine will ask for a light for her cigarette. Aim at the cigarette, click, and PEW! Flirtation has begun.

It's a hard game—the quick-draw contests get quicker and quicker, and eventually the player has just milliseconds to react. But even when the player gets GAME OVER and Annie spills some blood, Valentine patches her up, and their relationship goes on. What's between them outlasts any individual firefight. Every play of the game isn't self-contained but is—in a small way—fitted into the ongoing tale of love and connection, which the gunfights are just a means to developing.

Calamity Annie is a lot of things to me. It's about being an angry young woman in a hostile land (that just so happens to be Texas), trying to prove your worth. I certainly was trying to do that after they kicked me out of the Guildhall. I was making a game, and I was doing it my way: I built every piece of the game myself. I drew all the pictures; I wrote all the music; I scripted all the events in Game Maker. If the Guildhall taught me anything, it was that videogames needed

to be saved from the industry, that a creative form deserves better than an assembly line production process.

But the game is also about love, and how passion finds love. I had started seeing a loud-ass submissive named Daphny Drucilla Delight David shortly before starting at the Guildhall, and the workload had kept me away from her except for one week of vacation between each semester. The time we got to spend with each other while I was at school was mostly limited to me calling her and crying from the stress of having to split my energy between standing up to teachers, finishing my work, and doing good work, when all I wanted to do was make games.

I'm writing this, three years later, from the apartment where I live with her in California. The theme of *Calamity Annie* is that being driven will drive you to love, and that passionate people are attracted to passion. *Annie* was one of the first in a series of games that led to my being able to pay for a home and food by making games, which is what I went to Texas to seek. But it didn't come from the Guildhall. It came from me, from my stubbornness, and the stubborn friends who helped me.

Later games have made me more money—I asked players who wanted to support *Calamity Annie* for a donation of at least a dollar, in exchange for which they got a bunch of secret bad guys to shoot (I made a few hundred total)—but I still think

of *Annie* as my most important game. It's everything I realized I wanted games to be. It's personal (Annie has my name, after all) and it has a clear, unmuddied mechanical idea that translates perfectly to a storytelling idea. I want games to tell stories, and *Calamity Annie* tells my story, or at least the story of my stay in Texas.

Authors

My friend Lamar Williams is working on a documentary about videogame creators called *You Meet the Nicest People Making Videogames*. In one of the project's trailers, he points out that the idea of digital games as the product of teams is a myth: many of our most important games, and the ones that have been the most widely duplicated, are almost entirely the projects of impassioned individuals.[55] The examples that appear in the trailer are: *Another World*, a game about finding friendship during a struggle for survival in an alien world, created by Eric Chahi (who even painted the cover art for the game himself!); *Karateka*, a kung fu game with strong visual storytelling by *Prince of Persia* creator Jordan Mechner; *Defender*, a seminal arcade game by Eugene Jarvis and Larry DeMar; and *Berzerk*, almost as seminal an arcade game by Alan McNeil. He goes on to show contemporary games by individual authors, some of them made in Game Maker.

There's nothing unnatural about a digital game by an individual creator (or a pair of creators). It is, in fact, much harder to keep the idea behind a game coherent when the designer is managing a team of many people who are each working on one aspect of the game separately. That's part of the reason why contemporary big-budget games have so much clutter and so few strong ideas. The games are all over the place because the creators were all over the place. It's hard to have a strong singular vision when the process of creation is spread so thin.

Digital games contain video, audio, animation, design, and rules. You can parcel out these roles, but the closer they remain to each other, the more cohesive the work you create. If I'm the designer and I'm also drawing the spaceship that appears in the game, I know exactly how I intend to use that spaceship in terms of play, what its place is in the larger story, and what its appearance should express. I have a vision, in other words.

Books are written by single authors or by author-editor teams. Visual art is typically made by an individual artist. It makes sense for creators to be close to their work and to own their work completely, and that's something that the big teams that big-budget games demand can't have. When an individual or pair is solely responsible for a work you can watch an individual style develop: you can trace themes, both mechanical and otherwise, across a creator's work. (The Anemic's *La La Land* games, though each game is very different from the rest, have a strong singular style that persists throughout the series.)

And being able (or learning to) identify the individual style, and growth, of individual authors leads to better criticism and a critical understanding of games. Not to mention, like I said, more personal games, more relevant games, more games with something to say. I want a world where everyone is capable of sitting down at a computer and making a game by herself. This is not to say that all games need to be made that way, but as a paradigm, I think the individual author has more to offer us than the team, especially at a time when videogames are so seemingly creatively bankrupt.

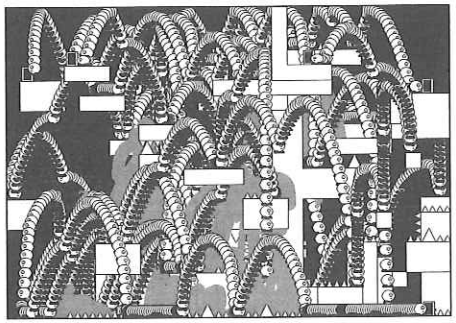
Death from Overwork

I first became aware of Jesse Venbrux, a game designer from the Netherlands, in 2008 because of his game *Execution*. When the player starts this game, she sees, through the sight of a gun, a prisoner tied to a stake. If the player shoots the prisoner, the prisoner dies, and the player is told, "You lose." If the player starts the game a second time to try to avoid losing, she again sees the same prisoner, still riddled with bullet holes.

Execution (a direct inspiration for *Calamity Annie*) points out something about causality in games: in particular, it points out how, as players, we expect to be allowed to undo and redo our choices at whim. The game takes just a few minutes to make this point: it's the kind of game that could never be a commercial product, but nevertheless has something essential to tell us.

Execution would probably never have been made without Game Maker. Jesse Venbrux might never have dug into game creation were it not for Game Maker. Though he's more recently gotten involved in Flash game creation and released games for the iPhone and its permutations, his website lists Game Maker–created games going as far back as 2004.[56] In Venbrux's body of work we can clearly see ongoing themes, both mechanical and contextual. His 2007 *Frozzd*, which was created for a Game Maker theme competition (entrants were asked to design games with a "winter" theme), involves navigating free-floating, irregularly shaped planets that all have their own gravity. That is, the player can walk all the way around the surface of one and arrive where she started, before jumping off and flying to another one. His 2010 game *Maru* features the same kind of irregular planet navigation, as does his most recent game (as of this writing), *They Need To Be Fed* for the iPhone.

His 2008 game *Deaths* remembers every time any player playing the game anywhere in the world meets death. Players will see the corpses of the last fifty players to have played the game, indicating particularly difficult areas and hidden traps. There is one screen in which players must continually die in order to build a bridge of corpses for future players to reach a high plateau (this is where the earlier-mentioned *We the Giants* gets its seed). His 2008 and 2009 diptych, *You Made It* and *You Probably Won't Make It*, also remember the player's death, painting patches of blood wherever she encounters a deadly spike. In the former game, this is especially important because the blood will paint over the visual information from the previous screen, showing what the terrain actually looks like. (In this game, nothing the player sees is ever erased from the screen. It is only overwritten by new information.)



The theme of protagonist death as narrative progress is an important one in Venbrux's work. His *Karoshi* series of games (*Karoshi* can be translated from Japanese as "death from overwork") require the player to guide the protagonist, a suit-and-tied salary man, to his death in a series of increasingly convoluted screens. Sometimes, in a similar vein to *Execution*,

the levels of *Karoshi* force the player to make use of something she thought was external to the game in order to proceed. For example, there's a stage in *Karoshi 2.0* called "What's On the Menu Today?" In this stage, the player must quit to the game menu—the screen where the player selects what stage she wants to play. The protagonist follows and falls to his death. Another stage requires the player to put a CD in her computer so that a boom box on the screen can play loudly enough to shunt a safe onto the protagonist.

Sure enough, Venbrux's iPhone game *They Need To Be Fed* requires the player to feed the protagonist to a monster at the end of every stage in order to complete the stage. (His 2008 game *Pazzon* requires the player to lead characters to the dinner tables of other game inhabitants.) It also features some of Venbrux's recurring "characters," such as the guided missile launchers that appear in his 2009 *Focus*, which in turn features the protagonist of *You Made It*.

We can notice and trace details and themes like these from game to game because each of these games is almost entirely the work of a sole, identifiable author. It makes us better, more literate critics of games to be able to see and discuss the progress of Jesse Venbrux as a designer. And it makes games more richly personal if we can play them in the context of the ongoing work and growth of a knowable author. This is a more useful paradigm than viewing games as the work of nebulous teams of hundreds.

The Author in the Games marketplace

At the time that Jesse Venbrux started making games, there was no place for him to market his work. Six years later, he's released a game for sale through the Apple Store. While this book isn't about making games for money—it's about making

games for the sake of having more games by more people, and there will always be more hobbyist authors than commercial authors—I think it's worth discussing the changing videogames marketplace in the context of the changing videogames paradigm. Just as digital games are starting to be thought of as the work of an author rather than a corporation, there is a growing place for solo game authors in the marketplace, and people can make a living doing this (though it's not easy).

I've been making my living in recent years by selling Flash games to "sponsors." Flash is an infrastructure for embedding movies and games in websites, and sponsors will pay for the privilege of including good games in their websites because games attract players who look at ads, and then explore the rest of the site to play more games and look at more ads. The more traffic a website gets, the more valuable that site is to advertisers, so traffic from popular games is important to site owners. Flash is much more complicated to use than something like Game Maker (although that's changing in a small way with tools like Stencyl, Flixel, and FlashPunk), and requires the kind of dedication that someone who just wants to make games in her spare time isn't likely to be able to spare. It's an avenue, though, for solitary authors to make the games they want and to find money for them.

The first Flash game I sold is about a flying pig. I sold it to *Newgrounds*—the "Everything, By Everyone" portal I mentioned earlier—for enough money to help me move from America's East Coast to the West. Later I sold a game called *REDDER* for twice as much, and while working on this book I sold a game called *Lesbian Spider-Queens of Mars* to *Adult Swim* for enough to keep me out of trouble for a while. I like this model because it means that although I get paid for my work, the game remains free to players. That's important, because

I want as few barriers between my work and its audience as possible.

Like Jesse Venbrux, I've also released an iPhone game, although mine is totally free. It's also based on a game I made in Game Maker—a friend of mine with greater technical knowledge, Bennett Foddy, programmed the iPhone version. The game's called *Chicanery*,[57] and it's about the interactions that go on between players outside of the digital components of the game (the stuff that's on the screen). Each player's goal is to keep a finger on the screen as long as possible, and to hinder the other players' abilities to do that. Usually the players do this by punching, shoving, or tickling each other. A guy I met who worked at WayForward Technologies later showed me the dent that playing the game had made in his iPhone.

When they're not free like *Chicanery*, iPhone games are usually sold for a dollar or two. And Apple has total control over its marketplace and what games can be sold within it, and is willing to wipe out games without a second word, as they did in 2010 with thousands of games they thought were too "sexual."[58] But the fact that authors are selling games there, and sometimes making a profit, shows that the digital games marketplace is making room for a new paradigm. Note that the Apple Store sells games exclusively through digital distribution—there's no publishing or manufacturing cost whatsoever.

Similar markets are Android phones, Steam for the computer, and the Xbox Live Marketplace. These are all digital distribution stores tied to specific technologies, and each is run by a corporation that exercises total control over what content is available on which device. It's a situation that leaves creators at the mercy of corporations, but it's a sign, at least,

that avenues are appearing for solitary authors to make money by creating games. And, hopefully, there will soon be more avenues, and decentralized ones.

Some authors take it upon themselves to sell and distribute their own games. My friend Edmund McMillen sold a CD with a bunch of his games and drawings on it—his attempt to create a game-as-zine. One of the things I used to do was to release a game that was free to play from start to finish—again, free is important—but to offer a secret password to anyone who donated at least a dollar. The password unlocked additional characters that would show up in the game as neat surprises. I got a few people who donated exactly one dollar, but most people gave five or ten: the invitation to donate was all they needed.

But that's enough about money. What I'm interested in is game creation as a goal in and of itself.

Crap Games

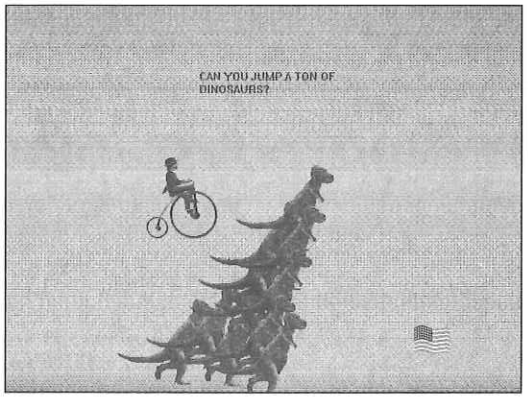
> *Glorious Trainwrecks* is about bringing back the spirit of postcardware, circa 1993. It's about throwing a bunch of random crap into your game and keeping whatever sticks. About bringing back a time when you didn't care so much about "production values," as much as ripping sound samples from your favorite television shows to use in your game, or animating pictures of yourself making goofy faces on your webcam. Where every ridiculous idea you had, you would just sit down and code. When you would make up a "company name" to legitimize dorking around on the computer with your friends. . . .
>
> Together, you and I will bring the true spirit of indie gaming back. Yes, you! For this site is about nothing, if it is not about getting off your ass and *creating*. Wikipedia claims that they used to stage trainwrecks (with

> empty trains, of course) for the amusement of the general population. Would the world not be a better place if we brought this tradition back?

So states Jeremy Penner on his website, *Glorious Trainwrecks* (at glorioustrainwrecks.com), founded in 2007. The site is interested in a videogame application of the "Crap Art" philosophy[59]: that professionals aren't the only ones who should make art, and that creation is a meaningful goal in and of itself. Games are more wonderful, more creative, and more inventive when they're thrown together around an idea with little regard for production values or painstaking creative choices. There was no place to explore the videogame as an act of raw creation, so *Glorious Trainwrecks* was built to provide one.

On the third Saturday of every month, *Glorious Trainwrecks* holds an event called Klik of the Month Klub, after Klik & Play, the simple, ugly game creation system (designed for children and people who have never made games before) that most of the participants prefer. But there are no enforced rules about what tools people can use, what they should make, or how much time they should take, though it's in the spirit of the thing to finish the game, from start to end, in two hours.

The experience forces participants to get past their egos and their meticulous plans for future epic games, to stop focusing on details and CREATE. Klik of the Month is about doing, not planning. Having to finish a game in two hours keeps authors from getting hung up on lesser decisions about their games. This is why Klik & Play is so well suited for the exercise: it's full of clip art, stock sound effects, and design shortcuts. All the resources are already there: just get your hands dirty.



The results are usually sloppy and haphazard. That's what the glory is: having made something raw and improvised. During different Kliks of the Month, I made a version of *Pong* where touching the ball makes you lose points (*Dodgeball*), a game about escorting vulvas back to their home planet (*Box Pushing Game*), a game about pushing sumo wrestlers into basketball nets with your belly (*Sumo Dunk*), and a pile more. In 2007, Glorious Trainwrecks ran a two-day event (the 100-in-1 Klik & Play Pirate Kart) that produced one hundred games. We did it again in 2010 and produced 529 games by over a hundred people. People have the desire and creativity to make games; they're only stopped by the roadblocks that get put in the way. Tear down the roadblocks and we'll have so many more games—and more important, more authors.

Game Sketches

What will help to tear down those roadblocks and allow games to be more creative—in the sense of, "Let's create something!"—are tools that allow more spontaneity. Klik & Play comes with a bunch of clip art that the author can easily appropriate toward

other ends. Game Maker allows the author to draw sprites right in the program before giving them rules and actions to perform. Stencyl allows instant access to an online database of resources other creators have already made. That's the solution, I think: to put as little distance as possible between the idea and a playable game. Certainly, many authors will spend a lot of time adding details to their games and developing their ideas. But there's value in ideas, especially in a form so young, and those ideas need to be put out there.

There's no way to sketch in games. In half an hour, I can sit with a pencil and paper and draw some dumb little comic strip. Videogames have no easy equivalent to that. And yet dumb little games are important because they enrich our vocabulary of ideas. The ideal game-making tool, I think, would look like a sketchpad: I could draw a landscape with a character in it, then give that character rules about how she could move across that landscape.

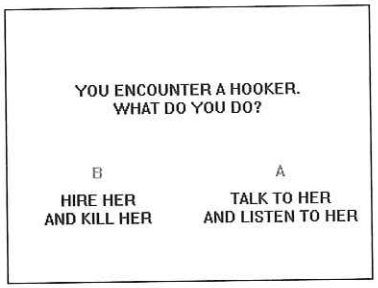
The editor that Fred Wood made for his game, *Love*,[60] comes close. *Love* is a simple little game about a stick figure who navigates a world by running and jumping; if she touches anything dangerous, she dies. To create a *Love* level, all an author has to do is draw it. The game will figure out how the stick figure interacts with the level. What stops the editor from being perfect is the fact that the author has to split the drawing into a few different images: a layer for things the player can touch and a layer for things that will kill the player on touch. But this is the right direction: if level design can be as simple as drawing a sketch, then game making can be as quick as putting an idea onto the screen.

And someone that doesn't know how to script or code probably knows how to draw—at least, to sketch. All of the most successful game-making tools for non-professionals I've seen

have looked like this. *Warioware: D.I.Y.* is probably the most mainstream commercial game-making software I've encountered, in that it was developed by a big publisher (Nintendo) for sale on store shelves. And while Nintendo makes it hard for people to actually distribute games they've made with the software (if the tool had a YouTube-like gallery where people could browse and play other peoples' creations, it would have changed a generation), the process of creating a game is brilliant.

Warioware is for the Nintendo DS, which has a touch screen and plastic stylus. Player-authors can draw sprites on the screen like they're drawing with a marker. Then they assign the sprites simple rules to tell them what to do during the game. But there are also a number of constraints that keep the author from being too ambitious: each game is only seconds long, the author only has a handful of objects to work with, and there's an equally limited number of sprites. This is a tool that's designed specifically to create very small games, and that's good, because small games are exactly what we need more of.

The proliferation of small games also serves another goal: it'll make games more personal. Ian Bogost, writing for *Gamasutra*, discussed the potential of games by way of talking about snapshots. The introduction of cheap portable cameras like the Kodak Brownie in the 1960s allowed people who were not professional photographers with access to expensive cameras and lenses to document anything, including things that held little or no value for anyone other that the taker:



> [Consider] You're Invited to Go to Heaven, a simple quiz game. . . . You're Invited is a rudimentary example of Christian evangelism.
>
> The game poses just a single question, "Who is the Lord of your life," and offers four answers: Chris Brown, Orlando Bloom, Zac Efron, and Jesus Christ. The "correct" choice is obvious, and it's tempting to write off this game as trite, even worthless.
>
> Its single question would seem barely to qualify it as a quiz game, a genre itself on the very fringes of the medium.
>
> But there is something deliberate and honest about its simplicity: this is not a game meant to inspire conversion or even head-scratching; it's just a little touchstone in someone's day for reinforcing what's really important to the believer.[61]

It might be hard to see the value in a thousand versions of *You're Invited to Go to Heaven*, but the real value of people producing digital games as quickly and easily as photographs is more subtle:

> The Brownie teaches us that snapshots aren't just good pictures created easily thanks to simple tools. They are also good pictures—or games—created for different purposes. The future of video game snapshots will require platform creators to show their potential users how to incorporate games into their individual lives.
>
> The result could be very important. The snapshot didn't just popularize photography as chaff, it also helped more ordinary people appreciate photography as craft. The successful game creation platform will be the one we can say the same of, someday.[62]

Making games more pervasive—not just games, but game

CREATION—will help us to better appreciate games and think about the craft and design of them. It'll also, I think, demystify game creation. We won't think of games so narrowly because we'll understand that they're capable of telling many different stories: stories about dyke cowgirls getting their girlfriends off booze, for example, or a record of a dream.

The ideal game-making tool, the game sketchpad, isn't here yet, but there are many tools that are close—or at least much closer than what engineers had to work with back in the day. But even given that, how do you, with little to no game making experience, get your feet wet and your hands dirty? How do you make a game?