Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds

A Thesis
Presented to
The Academic Faculty

by

Chaim Gingold

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Information, Design & Technology

School of Literature, Culture, & Communication
Georgia Institute of Technology
April 2003

Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds

Approved by:

Professor Janet H. Murray, Committee Chair

Professor Michael Mateas

Professor Sha Xin Wei

Will Wright
(lead designer, Maxis)

Date Approved

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

THINKING ACROSS GENRES

The computer is the ultimate pattern machine. With it, we can represent new and existing cultural patterns, such as images, video, sound, text, and behavior. Building new media forms on the computer does not mean we start from scratch, inventing new conventions for participation and representation each time we make a digital artifact. Digital media designers, when building new artifacts, reuse conventions and design patterns from many disciplines. Computer science, games, art, literature, music, graphic design, film, landscape design, and system dynamics have all been drawn upon to make digital worlds.

But how do we go about making new types of digital artifacts that recombine diverse practices of cultural pattern making? How should one go about reusing conventions from genres as diverse as games, toys, computer science, comics, and stories? How can we think across different practices, intelligently compare them, and appropriate relevant conceptual tools? How are we to make sense of digital media's procedural and participatory dimensions?

Diverse design practices don't always smoothly converge. *Hamlet on the Holodeck*, which proposes new digital genres that borrow conventions from games and stories, has drawn antagonism from ludologists eager to define the boundaries of a new theoretical practice for games. Ludologists have been quick to interpret Murray's discussion of interactive narrative, a new genre of digital artifacts that build upon game conventions, as an attempt to colonize the study of games from an alien discipline. In doing so, not only were concepts ludologists should have found useful overlooked, but the argument was appropriated for an entirely different purpose. An attempt to articulate a new genre of digital artifacts was recast as a contest between good and bad game theory: intruders from narrative theory had trespassed into the study of games, turf that requires new theoretical tools by new theorists. While some of the critique is fair, such as Murray's reading of *Tetris* [24], ludologists, to date, have been slow to fill the vacuum with a usable critical theory of games.

What we need are conceptual tools for constructively thinking across multiple genres. Janet Murray, in *Hamlet on the Holodeck*, proposes the term *cyberdrama* to describe digital things which cross genres. Espen Aarseth proposes *ergodic literature* and *cybertext* [3]. Despite their solid conceptual underpinnings, these terms fail to do justice to what designers can and do build with computers. Digital things aren't always narratively or dramatically informed, and aren't necessarily literature. Agency alone fails to capture what is unique and interesting about digital media's participatory qualities.

The signature properties of the digital medium, which Aarseth and Murray agree on, even if their terms do not, are its procedural and participatory qualities. Games, intrinsically procedural and participatory, are of special interest to digital media designers, as they are the most developed genre of expressive digital artifacts. By analyzing exemplary games and the thoughts of master game designers such as Shigeru Miyamoto and Will Wright, we can better understand the experiential and structural qualities of digital worlds.

Murray identifies space as another key property of the digital medium. Space is particularly relevant to digital games. Digital games are highly spatial, and the worlds digital games construct are often particularized in the same way as literary worlds. Game worlds articulate particularized environments, agents, and activities. The concept of game, however useful for understanding procedure and participation, doesn't take us far enough in understanding what we can make with the computer. More powerful conceptual tools are needed to account for space, and the richly reified worlds that digital artifacts often bring to life.

"World" sits at the nexus of multiple practices and discourses. Worlds can contain rules, games, spaces, characters, time, action, stories, or any number of other things. "World" subsumes categories such as space, story, and game. Defining *world* broadly is not useful for constructing a function which tells us whether something is a world or not. It does, however, provide a place to stand for leveraging tools from across disciplines, and thinking about the spatial, procedural, representational, and participatory qualities of digital media artifacts as worlds. By conceptualizing an artifact as a world, we can explore it as a space, story, or game, depending on its emphasis.

Design and analysis flows through the concept of *world* in multiple directions: designers of public parks and maps can draw upon game design to build spaces that afford easy navigation and simultaneously tease and taunt with hidden secrets; digital media designers can borrow conventions from comics, games, and stories to build new types of things altogether; game designers can draw upon stories, literary theory, and city design; web designers can leverage studies of the experience and cognitive representation of complex urban spaces.

Hamlet on the Holodeck describes the aesthetics of the digital medium as immersion, agency, and transformation. Digital media, Murray argues, are procedural, participatory, spatial, and encyclopedic. A world is something which can represent something, contain space, have moving parts, and offer immersion and agency. Immersion, agency, and transformation are useful terms for describing and thinking about digital media, but in practice these elements interact in complex and interesting ways. A twisting path, for example, teases us into imagining its continuation out of sight, and goads us into following it. *The Legend of Zelda* uses such spatial principles to lure players into exploring a compact yet mysterious world. *The Legend of Zelda* constructs a world which interweaves space, immersion, and participation. Digital worlds are the children of procedure, immersion, agency, and space.

Hamlet on the Holodeck looks at the landscape of digital media, and so the discussion of games, and their intersection with interactive narrative, is at a relatively high level. Part of the work undertaken for this thesis was designing and building *Comic Book Dollhouse*, a toy for making interactive storyworlds. Designing and building *Comic Book Dollhouse* necessitated thinking about the aesthetic, participatory, representational, and authorial conventions of games and other media forms. [1]

Understanding how games work intersects the interests of both academics and practitioners. A theory of games should provide tools for understanding how games work, a vocabulary, and generative formalisms. Such tools will help us better understand the space of possible games, and make new types of games. Building new types of games and new types of digital artifacts that incorporate game conventions are really two names for the same thing.

World, space, and game are key concepts that emphasize the properties and possibilities of the digital medium, and link it to other genres and practices. This thesis proposes conceptual tools for thinking across genres, and is an example of how one can study game structure and aesthetics through close analysis. Some language for thinking about games is proposed, which has been called

[1] Storyworlds are digital worlds which emphasize the kinds of dramatic materials found in the worlds constructed by stories.

for by both game designers and theorists.

Doug Church makes an excellent case for game analysis, design theory, and language in his articulate essay, "Formal Abstract Design Tools" [13]. Markku Eskelinen argues the same point from the opposite direction in the first issue of the academic journal *Game Studies* [24]. The call has been repeated in both academic circles and game design communities, as evidenced by proliferating sessions at the *Game Developer Conference* dedicated to the topic, and the recent establishment of the *Digital Games Research Association* and *Game Studies* journal. *Hamlet on the Holodeck* implicitly calls for a theory of games in its charting of the digital media landscape. This thesis contributes aesthetics, vocabulary, and analysis to this endeavor. Understanding how games work will help digital media designers leverage the wealth of game design knowledge towards other practices.

Chapter 2 describes an aesthetic of miniature worlds. This miniature world aesthetic is analyzed both structurally and experientially, and is derived from the work and thoughts of Shigeru Miyamoto, Will Wright, and Seymour Papert. The construction, representation, and interpretation of digital worlds by programmers, authors, and players is discussed in Chapter 3, and a category of authorship tools called *magic crayons* is identified. Magic crayons are computational languages for representation that integrate conventions of artistic practice. Chapter 4 proposes a principle of *ludic playability* isomorphic to Marie-Laure Ryan's principle of *narrative tellability.* Both ludic playability and narrative tellability seek the diversification of possible worlds, and diversification of possible worlds is analyzed with respect to the aesthetics of both games and stories. Point of view in games is a complex topic, and conceptual tools are currently lacking. Chapter 5 offers conceptual tools for analyzing and describing the operation of point of view in games. Chapter 6 describes the design of *Comic Book Dollhouse*, and draws upon the conceptual tools and aesthetics articulated in the previous chapters.

Hamlet on the Holodeck introduces many concepts that are elaborated here. A variety of terms are used in place of *scripting the interactor,* such as inviting, prompting, calling, teasing, luring, goading, or scripting play. *Hamlet on the Holodeck* also identifies space as particularly important to digital media. Digital worlds can be understood as things which integrate space, immersion, behavior, and participation. *Agency* is filled out with respect to games through possible worlds theory, which gives us tools for understanding why winning, losing, consequences, and empathy are interesting, and how they are structurally constructed by games and stories. The pleasures of agency and transformation are also elaborated. Transformation manifests itself as worlds in flux, player character transformations, and compound worlds. *The fourth wall* is elaborated through specific structures that build player safety and encourage participation.

A growing body of work inhabits the space between games and stories identified by Murray. Michael Mateas and Andrew Stern's *Façade* uses state of the art procedural animation and artificial intelligence to create the first real time interactive drama. Chris Crawford has worked for years on the *Erasmatron*, an interactive storytelling architecture. Players have appropriated *The Sims* and *The Sims Exchange* as a storytelling medium, a practice with a huge and active community. An unoccupied niche in the space of interactive storyworld software is a storyworld construction kit. *Façade* reserves authorship for Mateas and Stern, and *The Sims* is designed to support the construction of houses, not stories or storyworlds. The lengths to which users go to create stories with *The Sims* indicates a desire for storyworld construction software.

Digital artifacts which combine conventions from games and other media are edge cases that test the power of our conceptual tools. Building *Comic Book Dollhouse* required thinking about the conventions, structure, and experience of games, spaces, stories, and worlds. *Comic Book Dollhouse*, a magic crayon for making and playing interactive comic book storyworlds, is designed and evaluated with the conceptual tools proposed for thinking about games, spaces, and worlds.

Comic Book Dollhouse merges conventions from games, comics, and stories. With *Comic Book Dollhouse*, authors and players can make characters, props, worlds, and stories which come to life. *Comic Book Dollhouse* demonstrates a productive merging of conventions from separate practices, and the usefulness of cross-discipline analysis for thinking about digital media aesthetics, participation, representation, and authorship.

Comic Book Dollhouse is a toy in two senses: it is a magic crayon for making and playing with storyworlds; the software is robust, usable, and rewarding to play and make storyworlds with. *CBD* is also a toy in the sense that it doesn't attempt to do everything interactive storyworld theorists and makers have talked about. *CBD* is a miniature, internally consistent, and complete storyworld authoring tool. An interactive storyworld toy which is usable, interesting, and malleable is meant to entice people into making and playing storyworlds, which will in turn create demand for better tools. *CBD* is simple so that it will easily develop in response to user feedback. Encouraging a practice of making and playing interactive storyworlds that maximizes the aesthetics of the digital medium is a specific design goal of *CBD*. In both senses, storyworld construction toy, and toy interactive storyworld tool, *CBD* is unique.

CHAPTER II

AESTHETICS OF MINIATURE WORLDS

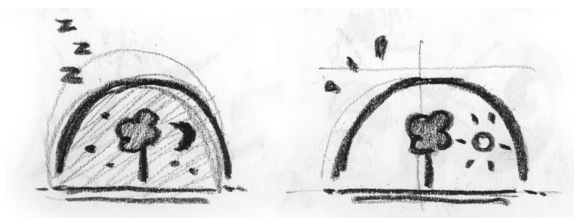

Figure 1: A Miniature Garden.

Shigeru Miyamoto, designer of *Super Mario Bros.*, often mentions his "miniature garden" aesthetic in interviews with journalists. Probably attributing this curious phrase to a mistranslation from Japanese, journalists never fail to not ask the question "what do you mean by that?" Miyamoto, without a doubt one of the greatest game designers, is telling us one of his fundamental design principles, and nobody bothers to ask him what he means. What follows is an attempt to interpret the phrase "miniature gardens" with respect to games using materials on Japanese gardens, literary microworlds, constructionist microworlds, play, and game analysis.

A garden has an inner life of its own; it is a world in flux which grows and changes. A garden's internal behaviors, and how we understand those rules, help us to wrap our heads and hands around the garden. The intricate spaces and living systems of a garden surprise, delight, and invite participation. Gardens, like games, are compact, self-sustained worlds we can immerse ourselves in. Japanese gardens often contain a multiplicity of environments and places, such as mountains, oceans, or forests that we can look at, walk around, or interact with. Gardens are a way to think about the aesthetic, cognitive, and representational aspects of game space.

A miniature garden, like a snow globe, model train set, or fish tank, is complete; nothing is missing, and nothing can be taken away. Clear boundaries (spatial and non-spatial), overviews, and a consistent level of abstraction work hand in hand to make the miniature world believable, complete, and tractable for both the author and player. Miniatureness makes a garden intelligible in the mind of a player, and emotionally safe in his heart. Miniature scale, clear boundaries, and inner life help players to wrap their heads, hands, and hearts around a world. [1]

2.1 Micro/Macro

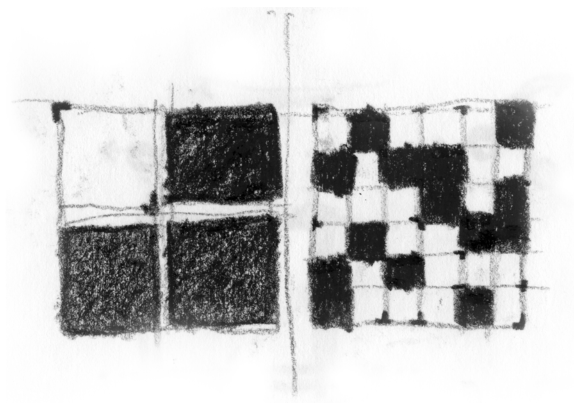

Figure 2: Micro & Macro

Miniature worlds offer simultaneous micro/macro readings. When looking at a model train set we perceive both an overview and ground level view. A model train set invites participation at the macroscopic scale, where the entire system is intelligible, plastic, and safe, as well as mental participation at the microscopic scale, where each train car can be walked into. This simultaneous play at micro/macro scales is a key pleasure of models. Using micro/macro views to engage people in unfamiliar worlds is a technique discussed later, in 2.4 Inviting Participation.

The aesthetic of micro/macro readings is evident elsewhere. A flower is beautiful to look at, and a time lapse video of a flower growing is fascinating in a wholly new way. The same aesthetic of multiple scales and their interplay is also present in *Go*. Tension and balance is evident in the arrangement of stones engaged in an intimate battle, as well as the visual tapestry and high level strategy of a game. Scale change in *Go* also marks a shift from a geometric to an organic aesthetic,

[1] "Miniature garden" most likely refers to *penjing,* miniature landscapes in containers. "The Chinese word *penjing* denotes a scenery in a container" [83, p38]. "Tree penjing or *shumu penjing* is called *bonsai* in Japan and the West" [83, p46]. My aim is to interpret the phrase with respect to game design.

and the transformation from one to the other is a source of continuous, surprising pleasure.

SimCity, *Go*, and *SimAnt* are games which encourage reading at multiple scales. *Go* and *SimCity* both have spatial structures at multiple scales that are intrinsically related and self-similar.

2.1.1 Overview

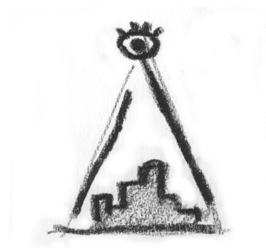

Figure 3: Overview

Miniature gardens are scale models of bigger phenomena. Fish tanks and gardens are scale representations of systems bigger than people. A fish tank is your own private ocean, and a garden is not just domesticated wilderness (a farm), but a scale model of wilderness as an aesthetic object to be designed and explored. If the garden is miniature, it follows that overviews of it should be accessible. Conversely, overviews help create the sense of miniatureness.

Miyamoto's *The Legend of Zelda* provides both spatial and temporal overviews at multiple scales. In addition to using a bird's eye view to describe the game's main action, *Zelda* incorporates maps to provide an overview of both a single dungeon and the entire world. The omniscient side view of *Super Mario Bros.* functions similarly:

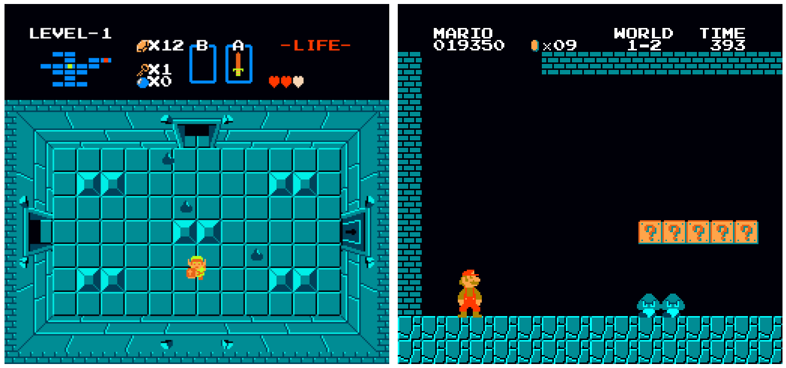

Figure 4: Overview via omniscient perspective in *The Legend of Zelda* and *Super Mario Bros.*.

Temporal overviews can be built by repeating activity landmarks. Rescuing princess Zelda requires collecting each piece of the triforce, and an overview of the collected triforce is a measure of one's progress in the game. To gather each piece of the triforce, the player, acting through Link, must: find the dungeon's entrance and enter it; discover a treasure; find keys, a map, and compass; defeat a dungeon monster; collect the triforce piece and exit the dungeon. This cycle of action repeats for each dungeon. Combined with the material side effects of collected triforce pieces, treasures, and dungeon keys, repeating action landmarks provide an overview of game scope, and locate the player within the game's progression.

The progress overview is a trope of other Miyamoto titles: *Super Mario Bros.* is divided into eight worlds with four levels per world, and the current level and world number is always visible. The fourth level of each world's cycle is always a castle. *Super Mario Bros. 3*, in addition to locating the player in a sequential world structure, spatializes each world into a series of locations on a map. *Mario 64* opens with the camera revolving around a castle, a space that encompasses all of the game's play spaces via an ingenious system of hyperlinked paintings. Player progression in *Mario 64* can be measured by castle traversal and visitation. Collecting stars is how one opens up parts of the castle, and the number of stars collected locate the player within the game at the local and global scales: How many more until I open the next section of the castle? How far through the game am I?

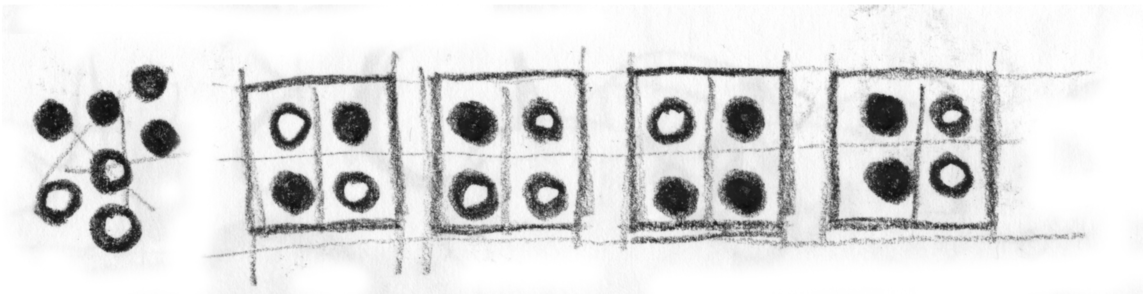

Figure 5: Overview via variations on a theme.

A key property of games is recombining familiar elements into novel configurations. Each *Zelda* dungeon repeats the same basic activity pattern, *Go's* tiny rule set and materials yield startling complexity, and *Super Mario Bros.* is a huge and engaging world built out of a small number of elements and rules. Variations on a theme can be used to provide players with an overview of a game. Introducing the rules and elements of a game gives an overview of a game's possibility space. It is for this reason that players can quickly get a sense for the size and shape of a game's possibility space.

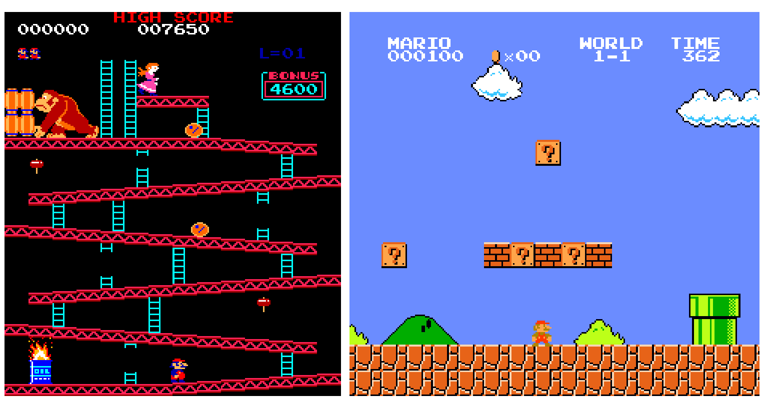

Figure 6: Overview via variations on a theme in *Donkey Kong* and *Super Mario Bros.*.

The first screen of *Super Mario Bros.*, like many of the single screen games that preceded it, contains almost all the dynamics the player will encounter throughout the game. Unlike single screen games such as *Donkey Kong*, *Mario's* world unfolds across a landscape of spaces, but an overview of game possibilities through variations on a theme is still provided. The first screen of *Mario* is a simple and comprehensible overview of the game's action, possibilities, and surprises: Mario jumps and moves; bricks can be stood upon; some bricks contain unpredictable presents; enemies harm Mario; enemies can be jumped upon and killed; objects ricochet; left to right traversal; Mario transforms by touching magical tokens; big Mario can break some bricks from below; there are coins to collect; some activities are elective; some obstacles must be jumped over. All of this is packed into *Mario's* first screen. [2]

Figure 7: Overview via tour.

Another means for conveying an overview is taking players on a tour, either by luring players along a path, or presenting a fly-through. Doug Church describes *Mario 64's* use of the first method:

> Often the first star (typically the easiest to get in each world) has been set up to encourage players to see most of the area. So even while getting that first star, players often see things they know they will need to use in a later trip. They notice inaccessible red coins, hat boxes, strange contraptions, and so on, while they work on the early goals in a world. When they return to that world for later goals, players already know their way around and have in their heads some idea about how their goals might be achieved, since they have already visited the world and seen many of its elements [13].

This type of overview, like variations on a theme, teases the player with a game's possibilities, and arouses the player's exploratory impulse with glimpses of potential rewards and intriguing contraptions. *Super Mario Sunshine* uses the fly-through technique to give players an overview of its sub-worlds. These worlds open with the camera flying through the space, from the traversal

[2] Technically, this is the second screen of *Super Mario Bros.*, but the first one is empty.

goal back to the starting location. An overview of the world's space, path, goal, and landmarks are established via overflight.

Compact scale can also be communicated without recourse to an overview. Many of the inhabitants and happenings of the Mushroom Kingdom, the setting of *Super Mario Bros.*, resonate a sense of the miniature through scale: Mario travels through pipes and plays with mushrooms, turtles, plants, and fish who are all his size. The Mushroom Kingdom's scale sets the stage for power reversals enacted by Mario's changing size.

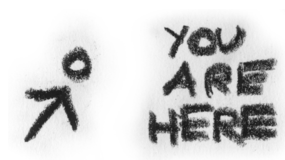

Figure 8: Overview via summary.

Another technique for conveying an overview is with a summary. The black screen that appears before each attempt at *Super Mario Bros.* locates the player by summarizing player lives, coins, score, level, and world.

2.1.2 Abstraction

Figure 9: Abstraction.

Miniature worlds are abstractions. A model train set is not a real rail system, but a model which captures the right details.

Will Wright points out that while playing games, people engage a game in their head, and what counts is this mental world. "So what we're trying to do as designers is build up these mental models in the player. The computer is just an incremental step, an intermediate model to the model in the player's head" [62]. His explanation of this concept works like this: somebody walks into a game store and looks at the cover of your game's box. Based on the front of the box, they start playing a game in their head, and if that game is interesting, they'll pick up the box and look at the back. They then play a new game in their head, closer to the one you've designed. If they like that game, then they'll buy the game and take it home.

The key insight here is that a game, despite the fact that it is a working system with moving parts, rules, and processes, is still a representation. Wright points out that a consistent level of abstraction is key to building believable worlds:

> One thing that we found in playing with *The Sims* is that it's pretty important that you have a consistent level of abstraction. It doesn't make sense to have everything highly detailed except one aspect and then have it abstracted. So in fact you want the

> entire world and the entire representation to be abstracted at almost the same level. At which point it holds together very nicely. It's kind of hard for you to go into a system and then be filling in the blanks of this one component, while everything else is highly detailed. So in *The Sims*, even the building is fairly abstracted. You can only put a wall within about a meter. The objects are somewhat abstracted in terms of selection: you don't have the full selection that you would really have in a furniture store. The granules of interaction in the game are kind of abstracted. So having that consistency, in your head, you fill in the blanks really well. And this is something that kids do quite well of course. You watch kids playing with toys. They're doing it all the time, very naturally. And even adults are doing that much of the time, with reading books, for example, where there are a lot of blanks to be filled in. [62]

Violations of abstraction consistency, such as carefully placed detail, can have powerful effects. While *SimCity's* primary level of abstraction and representation is at the scale of a city overview, representations of life at the microscopic human scale, such as cars, the newspaper, or the query tool (how many people passed through this train station?), provide carefully architected and intimate glimpses into a rich and detailed world that exists only in the player's head. An artist drawing a tree would never render every leaf; one or two leaves are drawn on the tree in detail, and that is enough to form a complete tree in the viewer's head. This is a kind of hide-and-reveal where something wanders in out of view through scale rather than space. The human life of *SimCity* wanders into and out of view with careful placed reveals of the city's microscopic life.

2.2 Boundaries

2.2.1 Frames

Figure 10: A World, Framed.

A miniature garden also has clearly defined boundaries. It fits in your backyard, a dresser drawer, or in the park. A fish tank has transparent walls that frame a miniature ocean. Boundaries can be ludic as well as spatial. Play is bounded in a separate ontology where the real world doesn't matter, and play doesn't matter to the real world. The boundaries of a world establish player expectations, which can then be subjected to play, teasing players into moving beyond a world's apparent boundaries. Playful violation of world boundaries and player expectations is described later in 2.5.1 Worlds in Flux and 2.5.3 Trespassing Across Worlds.

Will Wright uses a spatial metaphor to describe the state space of a game or system, and he describes a game with clearly defined boundaries as creating the effect of wandering around on an island with a visible shoreline. The boundaries of a game should be visible from the inside looking out: players should have a clear sense of what is and isn't possible. Doug Church describes how clear boundaries in *Mario 64* create the sense of a complete, internally consistent world:

> By offering a very limited set of actions, but supporting them completely, the world is made real for players. No one who plays *Mario* complains that they want to hollow out a cave and make a fire and cook fish, but cannot. The world is very simple and consistent. If something exists in the world, you can use it [13].

Player expectations can be called upon to make a world feel richer and more alive. *SimCity* establishes player expectations, and then makes good on them. Giving power to a neighborhood causes more people to move in. Miniature worlds live by establishing player expectations and then meeting them. A model train set establishes expectations through its appearance, and meets them when the cars motor around the tracks.

Clearly marked boundaries frame levels, worlds, and dungeons in *Super Mario Bros.* and *The Legend of Zelda*. *The Legend of Zelda* marks the spatial boundaries of its miniature garden world with a shoreline, mountains, and other particularized barriers. The start and end of a dungeon in *Zelda* is demarcated with discovering its location and conquering the monster. The spatial boundaries of *Super Mario Bros.* are clear: the world is one screen tall and scrolls from left to right until you get to the end of the level, which is always marked with a staircase and flagpole. *Super Mario Bros. 3* marks the end of each normal level with a kind of ragged black curtain and simple chance game:

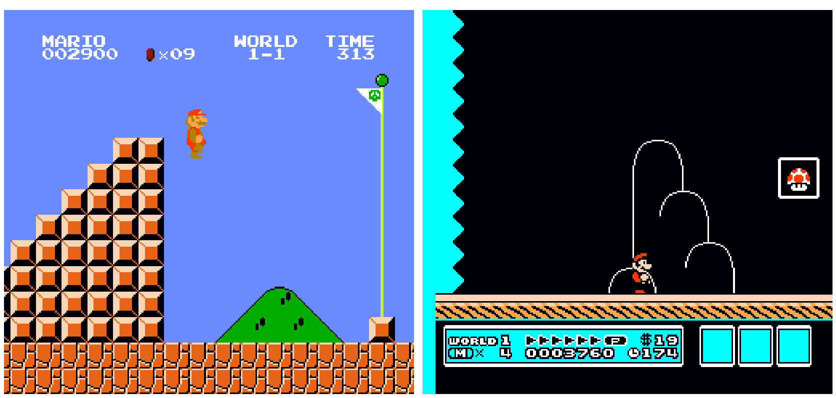

Figure 11: End of level markers in *Super Mario Bros.* and *Super Mario Bros. 3*.

Boundaries can be ludic as well as spatial: falling off the bottom of a level in *Super Mario Bros.* or touching another creature without landing on it will kill you. These boundaries are tidy and clear, and are visible from the inside of the game looking out, just as the shorelines of Wright's island are visible from the beach.

Clear boundaries reinforce the sense of an overview and compactness by helping us relate ourselves back to a garden's global structure. Although your view at any single moment is limited to one screen of a *Super Mario Bros.* level, the visible edges of a level's top, bottom, beginning, and end afford our cognitive mapping into a larger space. *Super Mario Bros.* and *The Legend of Zelda* support cognitive mapping with mechanisms such as edges, regions (worlds, overworld, underworld, dungeons, forest, mountains, graveyard, desert, etc...), nodes (warp zones), and landmarks. [3]

2.2.2 Hide-and-reveal

Figure 12: Hide-and-reveal.

The panels and gutters of comics provide incomplete views into a world reified in the viewer's mind. Panels and gutters in comics, implicit or explicit, are a manifestation of the more general principle of hide-and-reveal, and the mental closure it activates. This mental closure can be playfully manipulated, as Miyamoto does so well, to tease, goad, and lure the player into imagining and looking for alternate worlds.

Scott McCloud argues out that we piece together a world's action and life in the gutters between panels [48]. Hide-and-reveal is the spatial equivalent: it is a kind of teasing that places the conceivable and reachable states of a world in flux, puts them into play, and tickles the player into exploring the discrepancy and discovering what is hiding on the other side of the rabbit hole, up the beanstalk, below the manhole, through the pipe, behind the door, in the castle, or around the bend of the path. Overviews show us an entire world, and make it feel complete and safe;

[3]Kevin Lynch describes how edges, landmarks, nodes, regions, and paths afford cognitive mapping in *The Image Of The City* [40].

hide-and-reveal creates infinite worlds that invite exploration of their secrets.

David Slawson, in *Secret Teachings in the Art of Japanese Gardens* introduces *miegakure*, or "hide-and-reveal," as a playful technique for creating the illusion of a larger garden within a smaller space. Slawson quotes Tadahiko Higuchi, who describes the operation of *miegakure* in landscapes as relying

> heavily on the principle of overlapping perspective and involves making only a part of an object visible, rather than exposing the whole. The purpose is to make the viewer imagine the invisible part and thus create not only an illusion of depth but also the impression that there are hidden beauties beyond. *Miegakure* is, in short, a means of imparting a sense of vastness in a small space [74, p117 quoting Higuchi, p84].

Slawson goes on to discuss this careful manipulation of boundaries:

> One effective way of achieving this, as was mentioned earlier, is to design a meandering watercourse or path so that it now and then fades out of sight behind an object such as a hill or a tree or a rock to reappear at a greater distance from the viewer [74, p118].

The Legend of Zelda does in fact contain a river which wanders in and out of sight, as it meanders across the map and out of the garden. Games like *The Legend of Zelda*, *Super Mario Bros.*, and *Cosmic Osmo* all make use of portals such as doorways and pipes to interlink disparate spaces and pathways, and succeed in creating a sense of vastness and surprise within a limited space:

> In the case of a path, at each turn a new vista is revealed and a new one suggested by the point where the path fades out of sight beyond yet another bend. ... A device like this almost demands that the designer have a genuine sense of play, and revel in the way the garden can at turns entice and then surprise the observer [74, p118].

The effect of hide-and-reveal arouses an exploratory impulse in the viewer. Glimpses of a larger space tease you towards hidden pathways, seemingly unreachable spaces, and the potential pleasures of an infinite space. Comic panels are windows to look through, but hide-and-reveal makes passageways for traveling through.

If clear boundaries establish the perimeter of a game's island, hide-and-reveal plays with its boundaries and player expectations by providing glimpses onto secluded beaches, secret paths, and additional, hidden islands. Pushing aside statues, burning bushes, and bombing walls often reveals secret rooms, caves, dungeons in *The Legend of Zelda*. Boundaries in *Zelda* and *Super Mario Bros.* not only create complete worlds, but are barriers players can test, and hide-and-reveal encourages them to do so.

2.3 Life

Figure 13: Life.

> The dog that had terrorized him [Miyamoto] when he was a child attacks Mario. "I am especially proud of the dastardly, repulsive characters," he says. Miyamoto's dream was to make games that created worlds in which game characters could be more like players' companions, seemingly independent. "Perhaps they can even be ourselves at other times in our lives," he explains obliquely [72, p52].

Gardens are sites of endless recombination and amazement. A single creature or plant can manifest itself in an infinite number of ways as it grows, transforms, and interacts with other garden inhabitants. Familiar elements like grass, ants, trees, and flowers recombine in a kaleidoscope of immense sensual beauty, surprise, and wonder. Ant hills poke up in the most unlikely places; grass grows in the gaps between concrete slabs; vines climb up trees and doors of houses; squirrels and birds use whatever materials they can find, such as leaves and twigs, to build nests in trees, or maybe even the windowsill of your home. Ants and squirrels run off carrying accidentally discovered treasures. Delicate and surprising ecologies and interdependencies lurk everywhere as organisms of all sizes grow, transform, and adapt. Gardens are not only responsive, but demand human involvement: we must pull up weeds and plant seeds.

Gardens are dynamic living systems, full of secrets, autonomous agents, transformation, and emergent behaviors. A garden has an inner life all its own. It is a world which goes on without you.

Pre-digital games require human agency to animate them, but digital games are animated with the breath of computation, so garden is a tidy metaphor for self-animating systems. Not only are they dynamic, but gardens are reactive to human touch in a variety of ways, just like computers.

Both *The Sims* and *Lonely Time*, a lovely story-game by Takeshi Sakai, are self-contained, working miniature worlds teeming with life. The charm of simulation is that of a miniature train set: not only does the miniature world contain artificial mountains, trees, tracks, and trains, but it works; the trains move along the tracks and even emit smoke! My sims run around, get tired, go to sleep, and when they wake up, they have to go to the bathroom. In *Lonely Time*, I can enter the sliding rice paper doors of the bath house, lather up and rinse in the shower, and enter the communal tub. If I don't shower, then the man already in the tub will yell at me. I can order soup at the ramen restaurant, choose what extras I want, and consume each item in the soup individually. The restaurant even has other diners, and the waitress washes the counters! If I forget to pay, I end up in jail.

Scale and boundaries in both *The Sims* and *Lonely Time* set expectations for how detailed the world and its moving parts will be. Expectations for completeness are established, and nothing is missing. The neighbors come and visit your family in *The Sims*, for instance, and *Lonely Time's* world is filled with children at play, mothers and their babies, cats, and a romantic interest. These are miniature worlds teeming with an inner life of their own. Like a snow globe, model train set, or Japanese garden, nothing is missing, and nothing can be taken away.

Part of the simulation pleasure of *Lonely Time* is cross-cultural: I have a miniature Japan on my laptop. *The Sims* in Japan (*SimPeople*) probably has a cross-cultural appeal as well: care for and play with miniature Americans. Playing with miniature people is sure to entertain.

The moving parts in *Lonely Time* are largely situated in a dramatic universe. The game opens with an animated manga infused introduction, where a single salaryman comes home to find his cat missing. There's a lonely alleyway where you can sit, watch the sunset, and pine for things lost. A love interest works the cash register at a ramen restaurant. A playground serves as the stage for children at play, and a romantic encounter in the evening.

A system's capacity to surprise is a key criteria Will Wright uses to evaluate its interest and life. Wright uses mechanics such as cellular automata and system dynamics to create worlds full of life and surprise. *Super Mario Bros.* and *The Legend of Zelda* surprise players with worlds full of playful ambiguity and secrets.

World designers can draw upon a variety of techniques for creating illusions of life. *Lonely Time* and *The Sims* are unpredictable, living systems. *Lonely Time* makes use of story scripting and some simple dynamics to generate unpredictability and life. *The Sims* uses noise and threshold decision making mechanisms, like those described by Braitenberg [10], to give sims believable behavior and intentionality. System dynamics and cellular automata are used in *SimCity* to give its miniature cities life. Ken Perlin uses procedural noise to create the effect of life, a signature aspect of living and real things. The appearance of a drawn figure's balance, known by artists to be key in creating believability, is recreated in Perlin's three dimensional characters with procedures that dynamically balance body pose [63].

Games such as *SimCity* and *Go* call upon players to construct living things that will stand on their own against perturbations, feedback, or attack. Wright advises not making systems which are too sensitive to environmental fluctuations, as things players build in such systems are overly delicate, and will eventually die, much to the player's disappointment (personal conversation).

The aesthetic of life stems from a key pleasure of simulation and fictional worlds: making and playing autonomous, vibrant worlds.

2.4 *Inviting Participation*

2.4.1 Intelligibility & Plasticity

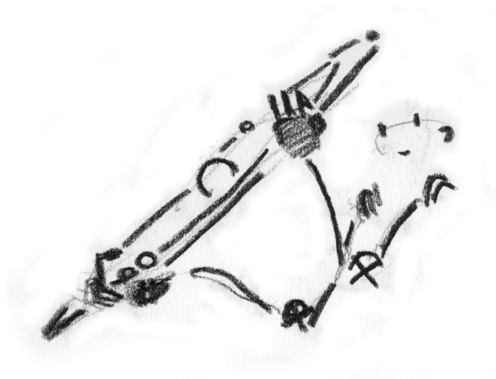

Figure 14: Intelligible and Plastic Miniature Solar System.

This person can get their head and hands around this solar system because it has clear boundaries, it's miniature, an overview is accessible, and the mechanics are abstracted.

Miniature gardens are scale models you can tinker with, and games are miniature worlds that fit in your hands. The world is big, scary, and uncontrollable, and games are systems which are so small that you can be the cause. *SimCity* transforms the inflexible and imposing built environment into a miniature city that you can build up, run like a model train set, and then smash to bits, like a child playing with wooden blocks.

Abstraction is the key to intelligibility and plasticity. Seymour Papert writes that "Newton 'understood' the universe by reducing whole planets to points that move according to a fixed set of laws of motion" [61, p117]. Miniature worlds have a plasticity unavailable in the "unintelligible limitations of matter and people" [61, p118], and the "assumptions and constraints" of a microworld, its abstractions, afford exploration "of a chosen microworld undisturbed by extraneous questions" [61, p117]. "[The] microworld is stripped of complexity, is simple, graspable. ... the child is allowed to play freely with its elements" [61, p162]. Abstraction enables malleable, comprehensible worlds. Consistent abstraction establishes expectations about what is possible in a world; meeting those expectations generates intelligibility. Abstraction imbues miniature worlds with intelligibility and plasticity, features that invite play.

Intelligibility and plasticity is how players get their heads and hands around a world.

2.4.2 Safety

Figure 15: Is This World Safe?

It must feel safe to experiment with a miniature garden. Caillois argues that play exists in a separate ontology, a domain in which what happens does not leak into real life [11]. The sense of a complete, closed world with clearly defined boundaries helps build this sense of safety, which Murray argues is the experiential function of the fourth wall in fiction and drama [51]. In the case of miniature gardens, overview, abstraction, and scale help to build this fourth wall. *The Sims* is a dollhouse world which feels safe to play with because of its scale and boundaries. The magic threshold of the fourth wall structures experience so that we obtain the pleasure of feeling terrified or elated, while cognizant that we are safely outside the fictive world. We participate richly in the miniature garden, but the fourth wall reminds us of the world's status as fiction.

There is an emotional facet to experimentation and the feeling of safety. Miniature gardens are safe in a way that a zoo or forest is not. A miniature garden fits in the palm of your hand, and doesn't contain sentient beings. It is doubtful players would feel comfortable torturing families in *The Sims* if the game was more realistic. *The Sims* as dollhouse world is miniature and abstract, and this abstraction invites manipulation and creates an emotionally safe space for experimentation. The stylized and exaggerated representation of *Grand Theft Auto 3* functions similarly for most players. Toru Iwatani wanted *Pac-Man* to be a play space that invited the participation of females, and settled on cute ghosts for the game's monsters [36, p141].

Safety and life help players get their hearts around a world.

2.4.3 Syntonicity

Figure 16: Syntonicity.

Players enter and exit microworlds through syntonicity. Syntonicity establishes strong bidirectional knowledge channels between ourselves and microworlds. Syntonicity helps us make sense of new microworlds, and import knowledge from a microworld back into the real world. Syntonicity not only works to make microworlds intelligible and plastic, but is a means by which we can learn from microworlds [61, p63].

Will Wright's games often use the interplay of micro/macro to map the player into the simulation and encourage participation. When playing *SimCity*, I mentally insert myself into my city's streets and look up and around at the surrounding buildings. This ground level scale, our position in the real world, helps me understand the simulated city and its needs: the citizens need somewhere to work; sims need somewhere to sleep; people must have a means of getting from place to place. *SimCity's* dynamic representation makes good on our expectations: we zone houses, people move in, and the activity, populace, and traffic of the city increases, magical transformations that are intelligible and consistent with our expectations. Simultaneously, through the game's overview, abstraction, boundaries, and scale, the city is transformed into a miniature garden. The ground level view creates intelligibility, and the macroscopic overview creates a plasticity unavailable to even the most authoritarian city planner. Insert some roads to help this section of the city develop; plant some residential seeds in the hope that homes will grow over here. The rigid and imposing built environment of the real world is magically transformed into a safe, intelligible cartoon miniature which fits in the palm of your hand and can be manipulated as such. The interplay of micro/macro is a key pleasure of making and playing with models and miniature systems such as gardens, model train sets, ant farms, model airplanes, and miniature cities.

A similar interplay of scales is at work in *The Sims*. *The Sims*, like a dollhouse, affords a mental mapping between the player's understanding of the real world and the microworld. It is straightforward to think about dolls or sims as people with needs and emotions like hunger, tiredness, or loneliness. *The Sims* and *SimCity*, when we first come to them, are intelligible at the micro scales, but their plasticity comes from their use of macroscopic scale; these worlds are miniature, malleable, and safe. Although we enter *SimCity* from the micro-scale, we exit at the macroscopic: playing and manipulating a simulated city leads to an understanding of the built environment as a dynamic entity.

Papert describes this kind of bidirectional transference of knowledge as syntonic learning, and he catalogues three syntonic modes. Something is *body syntonic* when it is "firmly related to children's sense and knowledge about their own bodies. Or it is *ego syntonic* in that it is coherent with children's sense of themselves as people with intentions, goals, desires, likes, and dislikes" [61, p63]. A microworld has the property of *cultural syntonicity* when it concretely connects with cultural practice. Angles in *Logo* are culturally syntonic with navigation: "The Turtle connects the idea of angle to navigation, activity firmly and positively rooted in the extraschool culture of many children" [61, p68].

The Sims taps into all three of these syntonic modes; the behavior of a sim is intelligible as a result of our understandings of our bodies (body syntonicity), material needs, and desires (ego syntonicity). The suburban setting and material concerns of *The Sims* affords a cultural syntonicity. These syntonic relationships make the game both intelligible and meaningful. Playing *The Sims* leads to a changed understanding of ourselves, human relationships, material things, and the role of architecture.

While Papert's primary interest in syntonicity is educational, broader aesthetic and cultural transferences are possible. Will Wright describes this operation in *The Sims*:

> I've seen a lot of people in testing start out by doing their own family, but very quickly

> they'll diverge, into either a fantasy of what they would like to see their life turn into or this voodoo doll thing, like "Let me see if I can kill myself" or "What happens if my sister gets sent off to military school?" We wanted people to read a lot into this by giving them the right level of ambiguity. We were trying to facilitate as much interpretation of the story as possible [73].

In addition to body, ego, and culture syntonicity, a microworld might have material syntonicity, and resonate with our knowledge of materials such as water, sand, or dust. Many game microworlds rely on game genre syntonicity: play requires leveraging game convention knowledge.

The syntonicity of Miyamoto's miniature gardens is quite whimsical. Besides the obvious body syntonicity Mario and Link build upon, Miyamoto appropriates and transforms a wide variety of cultural material for the construction of his worlds. Stories about hero journeys and rescuing princesses are drawn upon in both *Super Mario Bros.* and *The Legend of Zelda* to create narrative syntonicity. Link encounters magical helpers, descends into dungeons to recover treasures, and must vanquish an evil villain to rescue the princess Zelda.

Super Mario Bros. has more in common with how we process *Alice in Wonderland* than *SimCity*. *Super Mario Bros.* builds upon and then challenges our understanding of the stuff of our daily life: Mario collects coins, travels through pipes, can be eaten by flying fish, and is transformed by magic mushrooms. Exploring Miyamoto's miniature gardens teaches us little about system dynamics, but transforms our relationship to the world in a different way. Hyrule, the Mushroom Kingdom, and our own world are dynamic places filled with chance, danger, secrets, magical transformations, unstable boundaries, and invertible power relationships.

Syntonicity makes worlds that are intelligible and meaningful. Safety distances players and worlds, but syntonicity brings them together.

2.4.4 Scripting Play

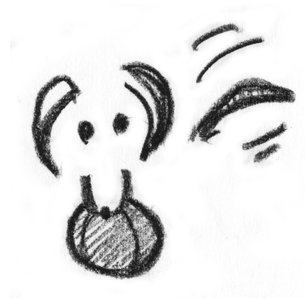

Figure 17: Dog With Ball Scripting Play.

Syntonicity is one tool to draw upon when scripting the interactor, the term Murray [51] uses to describe how a participatory world explains itself and encourages action by the player. *The Sims* scripts the interactor in a number of ways. Sims are people with needs (body and ego syntonicity). Objects in the environment can be manipulated to address these needs, and visual representations of objects script the interactor to their purpose and use: you call people on a telephone, fridges contain food, and showers clean people. *SimCity's* RCI (residential, commercial, industrial) demand meters script the interactor's activity by indicating what a city needs. The city's RCI needs are ego syntonic: the city can be conceptualized as an entity with needs, desires, likes, and dislikes. The need indicators of *SimCity* and *The Sims* script interaction by giving an overview of state and needs. Players adapt their play to this feedback: the city needs more residential zones, I better build more residential areas; my sim is hungry, I should feed him.

Player action in *The Legend of Zelda* is scripted, in part, through a narrative syntonicity that calls on our knowledge of myth and hero stories. *Pac-Man* scripts player action through representation. One generally runs from ghosts, and the chomping action of Pac-Man indicates that he eats things.

Games also script us to explore and create. Hide-and-reveal scripts exploration by teasing players into imagining hidden spaces. *SimCity* scripts the player into building things by presenting incomplete cities which need intervention.

2.4.5 Pleasures of Participation

Games provide us with a rich set of participatory conventions and anecdotal design knowledge that we can draw upon when building participatory artifacts. Thinking about games in terms of participatory pleasures is a productive way of thinking about why players do something, how they do it, and how games communicate and motivate this activity. Examining what players do allows us, as designers, to extract, replicate, and mutate the participatory conventions of games.

Below, I've inventoried some participatory pleasures of games. The list is not meant to be exhaustive, and the items are not mutually exclusive. Many complement one another. My intent is to demonstrate, by example, the usefulness of this analytic technique, and to highlight participatory pleasures that are drawn upon when reviewing examples and discussing *Comic Book Dollhouse*'s design.

Manipulation. Simple pleasure of agency, being a cause; putting your hand in the water and splashing about; something responds to me; pushing on something and having it move; knocking down a sandcastle to see it fall and the child who made it cry; knowing that I did it.

Problem Solving & Strategizing. Building a mental model of a world and thinking with it. The pleasure of thinking and working within a set of constraints to solve a problem. Imagining what the futures holds, and planning for it. Imagining many possible futures and making flexible plans.

Strategizing requires enough complexity and correspondence between mental and world model to afford short term and long term planning. *The Sims*, just as in Will Wright's original concept for the game, can be played tactically. How can you arrange your house to solve various optimization problems for your sims? The simulation is complex enough to surprise you, and the correlation between the player's mental model and the simulation mechanics is close enough to make long term planning possible. *The Sims* is interesting and intelligible enough to afford strategizing.

Lonely Time's simulation does not afford anywhere near the same level of strategizing. You can watch your stamina and money, and think ahead to when you need to trade cash for stamina at the restaurant or bath house, but this strategizing is shallow and not intrinsically related to the search for your cat, romance, or the lush microworld at the center of the game's interest.

Contest. Overcoming a challenge. Winning and losing. Playing against someone with a different goal.

Mastery. The pleasure of acquiring a new skill; shaping something; being good at something. Familiarity with difficult material. Knowing the rules.

Composition & Construction. The pleasure of building things, a common trope of Will Wright's work. Wright argues that giving players the opportunity to build things, and affording the creation of unique solutions, generates the satisfaction of creativity and a feeling of empathy in the player:

> So I guess what really draws me to interactive entertainment and the thing that I try to keep focused on is enabling the creativity of the player. Giving them a pretty large solution space to solve the problem within the game. So the game represents this problem landscape. Most games have small solution landscapes, so there's one possible solution and one way to solve it. Other games, the games that tend to be more creative, have a much larger solution space, so you can potentially solve this problem in a way that nobody else has. If you're building a solution, how large that solution space is gives the player a much stronger feeling of empathy. If they know that what they've done is unique to them, they tend to care for it a lot more. I think that's the direction I tend to come from [62].

SimCity and *The Sims* combine problem solving and composition. You build things to solve problems, sometimes you build things just to build things, and other times you solve problems in order to build the things you want.

A key participatory pleasure of *The Sims* and *SimCity* is compositional. Building a city or house feels like a creative act of composition, since the state space of possible designs is huge, and state accumulates over time. Also, the space of possible designs is dense with interesting results, unlike a paint program. Your household or city is probably unique as a drawing, and returning to the same composition is very compelling, especially since they feel like living, breathing entities. The compositional pleasure also has a high degree of agency associated with it, as the layout and design of your city or household has important effects on the way it behaves.

The Sims is also stimulating as a material to shape stories out of. *The Sims Exchange* storytellers compose stories out of their games by acquiring skins and objects, designing sets, organizing action, taking snapshots, and narrating the images with text. The album feature of *The Sims Exchange* seems to have not been intended for this purpose, yet its appropriation for this function is natural.

The degree of dramatic composition that takes place within *The Sims* game, rather than outside of it through snapshots and narration, is weak. While you do imagine rich stories in your head while playing the game, dramatically the components are not so suggestive.

Lonely Time also has a compositional pleasure. Rich building blocks are available for you to recombine into satisfying scenes, such as moody backgrounds, character actions (sit and think, slide, cry, laugh), and game world responses. This is composition through acting rather than design. The compositions have few consequences, but are satisfying in the same way I imagine authoring stories out of the materials of *The Sims* to be. Like *The Sims*, most compositions are interesting, but the space of possible compositions doesn't feel particularly large.

Exploration. Uncovering a world's surprises; the process of learning the rules of a world; gaining mastery.

> We started making the first *Zelda* game at the same time as *Super Mario Bros.* I wanted to create a game in which the player needs to think a lot to solve the mysteries; games before were quite simple and it was easy to figure out what would happen next. It was quite a reckless move on our part, quite a challenge, but it was widely welcomed by the players. – Miyamoto [14]

Reflecting & Watching. Sharing what you made; showing a friend; having your mom or dad hang your picture on the fridge. Watching the model train set go round and round; watching the movement of water.

Communicating. Sharing what you made. Feedback. The pleasure of a good conversation, listening to somebody, or having someone else understand you.

Transforming Identity. The pleasure of taking on a new identity, or putting oneself into flux. Becoming Mario or Link. In Miyamoto's words,

> One of the most important things with the Zelda franchise is that players really must feel that Link is really almost themselves in the game. In that sense, there has to be very natural and fluid interaction between the player and the character.[2]

The general pleasure of transformation, described by Murray as a principle aesthetic of digital media. Transforming identity; worlds in transformation; unstable worlds; potential transformations; play which leads to changed relationships with the actual world.

2.5 *Down the Rabbit-Hole*

> What if you walk along and everything that you see is more than what you see–the person in the T-shirt and slacks is a warrior, the space that appears empty is a secret door to an alternate world? What if, on a crowded street, you look up and see something appear that should not, given what we know, be there? You either shake your head and dismiss it or you accept that there is much more to the world than we think. Perhaps it really is a doorway to another place. If you choose to go inside you might find many expected things.
>
> – Shigeru Miyamoto [72, p37].

2.5.1 Worlds in Flux

Figure 18: A World, In Flux.

Once the boundaries and organization of a world are intelligible, they become potential targets for playful inversion, transgression, and transformation by both players and designers. Worlds with intelligible boundaries are constructed, made plastic, and placed into flux.

Testing boundaries is one way to find out what stuff a constructed world is made of. The clearly established boundaries of *Super Mario Bros.* become a target for playfulness and inversion by both Miyamoto and the game's players. Chuck Jones's classic *Duck Amuck* can invert and play with the conventions of cartoons because the boundaries of the cartoon and its system of representation were established at the time the film was made. In a similar move, Miyamoto plays with the conventions of a game from within it. Hidden passages take you underneath or on top of *Super Mario Bros.* levels, literally outside of the established spatial frame of the game's space. In Level

1-2 of *Super Mario Bros.* you can jump on top of the screen, and run behind the level's end into a warp zone that transports you to other sections of the game:

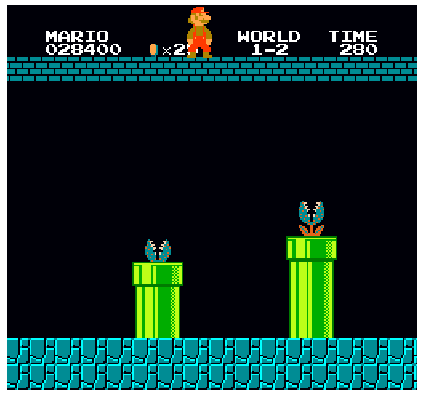

Figure 19: Sneaking outside the frame of *Super Mario Bros.*.

In Level 1-3 of *Super Mario Bros. 3* Mario can fall behind the level's scenery, and run behind the curtain which marks each level's end to recover a treasure. Objects sometimes dangle invisibly in the air, waiting for Mario to hit them before revealing themselves. Most pipes aren't passageways, but some pipes take Mario into alternate spaces. Both Miyamoto and players delight in playing with the boundaries and organizing principles of his miniature gardens, but this play is possible only because the boundaries are so clearly defined. Expectations are established against which surprises operate. Clearly marked boundaries form a backdrop against which instabilities can be constructed and made to feel safe, inviting, and charged with wonder.

Miyamoto describes the pleasure of testing boundaries and uncovering secrets:

> The players must be thinking, "Well, I don't see anything here, but it can be, it's possible." Then the player is curious enough to visit that place. When he finds something he never expected, he feels, "Ah, I did it. I made it." It's a great kind of satisfaction [72, p53].

Conceivable and reachable states are put into flux, inviting an exploratory play of world boundaries. The pleasure of finding reachable states in unexpected places teases the player into imagining additional states and testing their availability. The secrets, surprises, ambiguities, and alternate worlds of Miyamoto's miniature gardens lure players into toying with boundaries. Will Wright's constructionist toy worlds invite players to test the limits of the buildable, a basic pleasure of making things. *SimCity* goads players into conceiving of a variety of potential cities, and then exploring their reachability. Players of *SimCity*, in this sense, are exploring the boundaries of the buildable, just as Wright himself plays with the boundaries of what is digitally simulatable. Papert argues that testing the limits of a microworld is part of getting to know it:

> [In *Logo*] the primary learning experience is not one of memorizing facts or of practicing skills. Rather, it is getting to know the Turtle, *exploring what a turtle can and cannot do.* It is similar to the child's everyday activities, such as making mudpies and *testing the limits of parental authority*—all of which have a component of "getting to know" [61, p136] (Emphasis mine).

David Sheff relates an episode of boundary violation, spatial and parental (we can assume), from Miyamoto's youth:

> Miyamoto as a child had worked up the courage to go beyond the periphery of the forbidding cave he had discovered. "The spirit, the state of mind of a kid when he enters a cave alone must be realized in the game," he [Miyamoto] says. "Going in, he must feel the cold air around him. He must discover a branch off to one side and decide whether to explore it or not. Sometimes he loses his way. ... If you go to the cave now, as an adult, it might be silly, trivial, a small cave," Miyamoto says. "But as a child, in spite of being banned to go, you could not resist the temptation. It was not a small moment then" [72, p52].

Miyamoto's worlds *reward* players for testing limits rather than breaking under the duress of player subversion. The rules of a game world, its organizational structure, are a contract between the player and game that takes the form of boundaries and expectations. Games, however, revel in establishing boundaries and then putting them into states of instability, flux, and play. *Dominant world organization* is introduced and then subjected to a host of violations, reversals, suspensions, and transformations. [4]

Transformation of player power is a common world organization violation. The boundaries of player power are clearly demarcated and then reversed or expanded. *Pac-Man's* dominant organization is structured as a pursuit of the player character by ghosts, but Pac-Man can briefly invert the

[4]Conceptualizing a world's dominant organization allows us to articulate sanctioned cheating or rule violation within a formal system as rule bending or breaking. Stealing bases in baseball, sanctioned use of illegal words in a game of *Scrabble*, or special moves in *Street Fighter II* can all be described as interruptions of a dominant world order.

game's power structure by eating a power pellet. For a short interlude Pac-Man chases the ghosts. The invincibility star in *Super Mario Bros.* functions like the power pellets in Pac-Man; touching the invincibility star temporarily inverts the power relationship between Mario and hostile agents, and the musical accompaniment of the game switches to a faster tune.

In a miniature garden, relative size is power. Relative scale of creatures and objects in *Super Mario Bros.* is a world organizing principle against which transformations in size operate. Mario becomes larger and smaller through the transformative power of mushrooms and interactions with other creatures. The game's power relationships are not only mutable, but reward, at times, Mario's smallness with access to places unreachable when large. *Super Mario Bros. 3's* world four is populated with creatures and objects which are larger than their normal size; everything except Mario is four times its normal size. This world's organizing principle is a play on the other worlds' dominant use of relative scale. Normally Mario intrudes upon his world's order by growing larger, but in this world Mario is effectively miniaturized. Against this new organizing principle of a world which is larger than usual, Miyamoto continues to play tricks with scale. Passing through a doorway in level 4-6 causes the scale of the creatures to change back and forth between normal and large. Passing through a pipe in the *Mario 64* level *Tiny-Huge Island* causes Mario to shrink (or maybe the world enlarges), and the camera perspective and sound effects go along for the ride. Consistent relative scales are established and then intruded upon.

Mario undergoes a handful of transformations in *Super Mario Bros.* which reconfigure the world's power organization. Small Mario dies if harmed, becomes large Mario when a mushroom or fireflower is touched, and can fit into small spaces; large Mario becomes small when harmed, can destroy some blocks, and becomes fire Mario when a fireflower is touched; fire Mario can throw fire balls; touching a star renders Mario temporarily invincible.

Another play on power organization is establishing the boundaries of player power within the world, and then inviting the player to violate those limits. *The Legend of Zelda* begins with Link acquiring a sword, a first in a long series of steps in which player power is gradually incremented. Both Miyamoto and Papert describe violating or testing authority in relation to maturation or learning, an experience that can be exciting and awkward. Gaining experience, skill, and power is constructed in games like *Zelda* as testing and reconfiguring the organization of a world.

Another appeal of transformation is repeating the initial pleasure of immersion into a game world:

player character transformations are not unlike the experience of starting a game and acquiring a new form, and environmental transformations and transitions repeat the sense of novelty and dislocation associated with immersion into a new environment.

Super Mario Bros. 3 introduced additional mechanisms for placing the series' dominant world organizations into play. P-blocks, when activated, temporarily transform solid block into coin and coin to solid block:

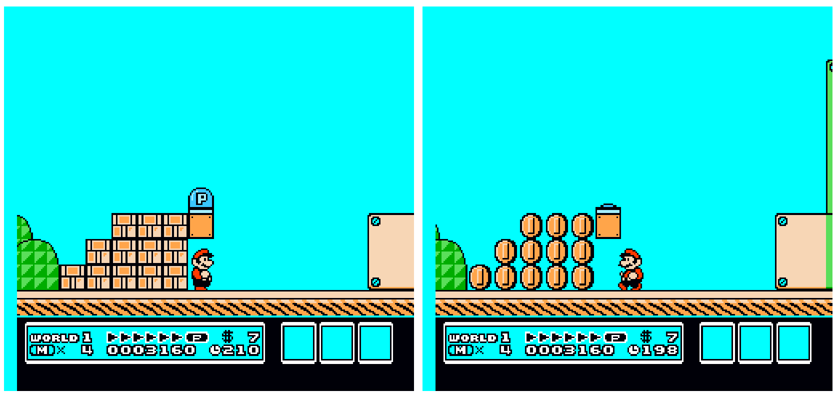

Figure 20: P-block in *Super Mario Bros. 3*.

This interlude is accompanied by a dreamy music box song which emphasizes the transient intrusion of an alternate world order. This delightful transformation allows Mario to reach spaces previously inaccessible and reap treasure from what was solid block. The possibility of this reversal teases the player into thinking about solid rock as passable, mountains of stone as hordes of treasure, and coins as solid platform that can be stood upon or destroyed.

This reversal is simply one more trick in a collection of transformations and disguises for which Miyamoto uses *Super Mario Bros.'s* ubiquitous blocks. Most blocks are breakable by Mario, but some contain hidden treasures and cannot be broken (one coin, multiple coins, extra lives, mushrooms, stars, fire flowers, magic bean stalks, etc...), and in *Super Mario Bros. 3* players can find such blocks jumping about with little critters living inside. *Mario* games articulate dominant world organizations that carefully build player expectation only to surprise with unexpected violations and transformations. Players are kept on their toes by Miyamoto's imaginative use of disguises, presents, surprises, transformations, and ambiguities.

Game worlds can also exist in states of vibrating instability. Rather than construct instability as playful violation of dominant world structure, worlds can be sites of oscillating order. *Ikaruga* is a vertically scrolling shooter with an intriguing instability: the game can be played with your choice of two sets of rules, and the rules can be switched during play as often as desired. The majority of agents and all the projectiles in the game are either black or white. Black agents shoot black projectiles, and white agents shoot white projectiles. In one mode, your vehicle is black, and black projectiles, when touched, give your agent power, and white projectiles destroy it. In the opposite mode, your vehicle is white, and black projectiles harm and white ones aid. Switching modes inverts danger into safety and safety into danger. An unlimited number of reversals are available to the player, and the game demands rapid switching, as the play field is often filled with both black and white projectiles. *Ikaruga's* structure is an oscillating bipolar order. The dominant organization vibrates between two modes, yielding an exhilarating feeling of ongoing instability and dizzying reversal.

The spatial structure and power distribution of a *Go* game is in continuous flux until its end. A large part of the pleasure of playing *Go* is the fluctuating relationship between stones and power boundaries. Group boundaries and player power is very fluid in *Go*: a single stone can reverse who has which space, or turn a live group into a dead one. The boundaries between black and white power, and the identity of groups are continually transforming, vibrating, and reversing. As the game approaches its end, these instabilities approach zero, tension is resolved, and play becomes less and less interesting until additional moves are meaningless, which is marked by both players passing.

How do worlds in flux remain intelligible to players? What prevents a world in flux from disintegrating into nonsense? A world's rules might be in play to such a degree that it dissolves into unintelligibility, but this would be an extreme case. Violations are readable when they operate against a dominant world order. *Super Mario Bros.* is intelligible because its boundaries are clearly established and then subverted.

Dominant world order might change in such a way that we find ourselves in an altogether different world. The dominant organization of a game's multi-world universe might then be a particular sequence of worlds, as in *Super Mario Bros.*. The sequence of worlds can then be placed into flux, and violated with diversions into alternate worlds via warp zones or other magic portals. But before we can trespass across multiple worlds, a compound world must be constructed.

2.5.2 Compound Worlds

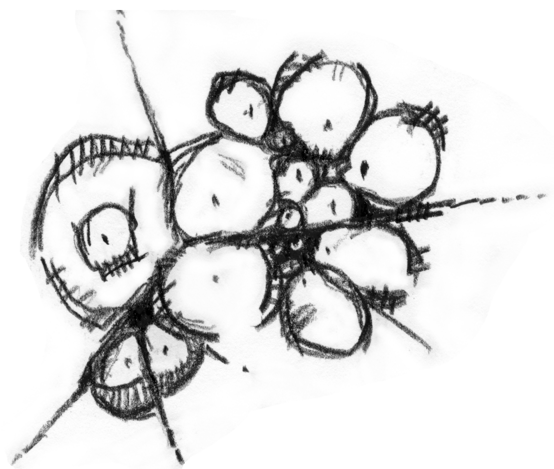

Figure 21: A Compound World.

In *Super Mario Bros.*, players travel above ground, underwater, underground, through castles, the clouds, dark worlds, and winter worlds. Players delight in the exploration of these multiple worlds, and the movement between them. The phenomena of compound worlds is a trope of both fiction and art, and happens to be a particularly postmodernist preoccupation. How does one go about building a compound world, and once you have one, what can be done with it?

Compound worlds are collections of microworlds, each microworld framed and set off from its neighbors through boundaries and changes in world organization. A single boundary can play many roles in a game world: it can be put into playful flux, used to frame a world and make it feel complete, and also delineate the boundaries between multiple worlds. The framing conventions of a *Super Mario Bros.* level, such as its end of level markers and spatial boundaries, are simultaneously used by Miyamoto to make each level complete, mark the boundary between levels and worlds, and are limits that players can violate, at times, by sneaking outside of.

The last chapter of Boris Uspensky's *A Poetics of Composition*, describes how miniature worlds are created in both literary fiction and visual art. Such miniature worlds are constructed through a variety of techniques, and are often composed into compound worlds. Literary and pictorial works of art, Uspensky writes, both "possess the features of a closed system—each work presents a unique microworld, organized according to its own laws and characterized by its own spatial and temporal structure" [81, p167]. [5]

> In many instances, it seems to be psychologically necessary to mark out the boundaries between the world of everyday experience and a world which has special semantic significance. Thus in the theater the frame is expressed through such stage devices as footlights, curtains, and so forth. ... Sometimes the borders of the conventional artistic space may fluctuate, without, however, being entirely destroyed, as in the case of carnival or mystery plays where theatrical conventions expand into life [81, p138].

Uspensky writes that changes in descriptive system, such as visual perspective or narrational point of view (ideological, psychological, spatio-temporal, phraseological), or beginning and ending markers, are enrolled to signify the frames of a microworld. Furthermore, this "framing may apply not only to the whole narrative but also to parts of the work. These parts (microdescriptions) have their own organization, based on the same principles as the organization of a whole work—that is, each has its own internal composition and its own frame" [81, p151].

The frames, internal laws, and structural principles which define the boundaries of a microworld can recursively repeat, producing compound worlds. Worlds within worlds are created by repeating the elements that frame worlds. Microworlds are combined into compound worlds through adjacency (spatial, temporal, or otherwise), embedding (forming hierarchies), and layering. The boundaries between microworlds can be clearly demarcated, or organically merged and "detected only by discovering the implicit framing devices ... which, so to speak, constitute the internal seams of the work" [81, p155].

[5] Two definitions for *microworld* are now in play, one from Papert, and one from Uspensky. For our purposes, the definitions are complementary. Papert uses *microworld* to describe a miniature world that embodies a "set of assumptions and constraints" [61, p117]. *Logo* is a microworld for playing with geometry. Uspensky uses *microworld* to describe an entire work, or the units of a compound work, where each unit has its own organizational structure, or is set off in some way. *Microworld* conveniently absorbs both meanings. Papert and Uspensky's definitions reinforce one another.

Compound game worlds can be analyzed in terms of variation in descriptive mode or changes in world organization with the tools put forth by Doležel and Uspensky. Each room in *The Legend of Zelda*, for instance, has its own independent point of view. [6]

Multiple worlds in *Super Mario Bros.* are marked by changes in description and organization. Music, environment, and rules change as Mario moves between worlds. *Super Mario Bros.* uses spatial boundaries, different ludic orders, and various markers to demarcate the boundaries of its microworlds. Dominant world organization, discussed earlier with respect to worlds in flux, can be used construct a neighborhood of microworlds. Multiple world organizations means multiple microworlds.

Game interface elements are often layered directly on top of the primary representation. This layering doesn't create confusion because each layer belongs to its own microworld. The microworld of interface elements, such as the player's score and lives, frames the game microworld, and is clearly separated by differences in descriptive style and function. The layering of an interface microworld on top of the game microworld, furthermore, helps construct a fourth wall between the player and game world. Games often use embedding to form a hierarchy of microworlds: *Final Fantasy 7* embeds games within games; in *Mario 64*, Mario jumps into the represented space of paintings in a castle, which forms a hierarchy of microworlds.

Super Mario Bros. 3 is organized into a sequence of eight microworlds, and are literally referred to as worlds within the game. Each world, in turn, contains a set of levels, which are microworlds embedded within the space of each world's map:

[6] Doležel argues that narrative microworlds are constructed by selection, which "determines which constituent categories will be admitted into the world under construction," and formative modalities [22, p113]. Doležel puts forth a typology of such modalities: alethic constraints govern what is possible, impossible, and necessary in a world, "especially causality, time-space parameters, and the action capacity of persons" [22, p115]. Deontic constraints govern what is permitted, prohibited, and obligatory. Axiological constraints govern what is good or bad, and epistemic constraints describe what is known and unknown [22, p113-128]. Doležel describes compound worlds as dyadic. This modal system gives us traction for describing ludic microworlds through their rules. The rules of a game articulate what can and cannot be done, what is known and unknown, and what actions get rewarded or punished. Also, the difference between alethic and deontic constraints provide a language for describing sanctioned rule violation, such as stealing bases, within the formal constraints of a game world.

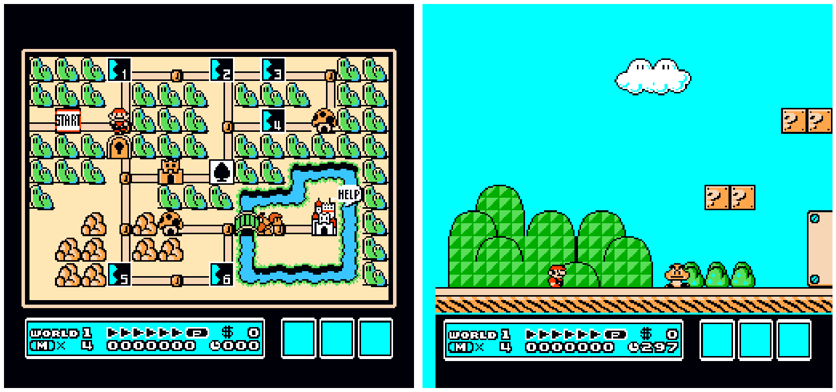

Figure 22: Embedded microworlds in *Super Mario Bros. 3*.

Thumbnails of each level are placed on the world's map, which spatially embeds the level microworld within a world. Also, the point of view changes from a 3/4 overview perspective on the map to a side view for each level.

Super Mario Bros., rather than use a map to represent the embedding of each level within a world, frames levels by other means. The start of each level and play attempt summarizes the player's location within a particular level and world. The spatial limits, end of level staircase, and flag pole are frames that demarcate the boundaries between level microworlds:

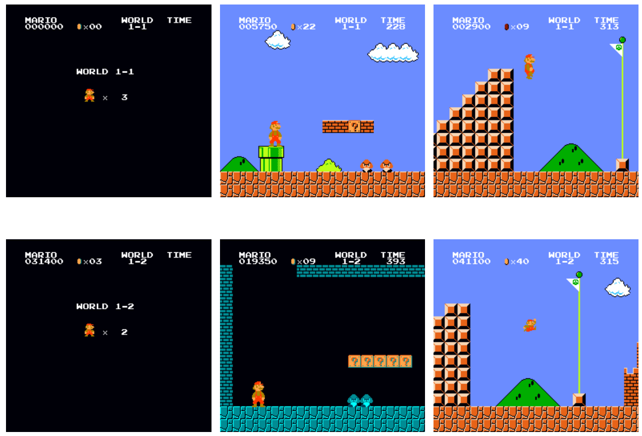

Figure 23: Level demarcations in *Super Mario Bros.*.

Super Mario Bros. uses heterogeneous environments to construct multiple, internally consistent microworlds. The game's environments include, but are not limited to, above ground, below ground, underwater, and castle interiors:

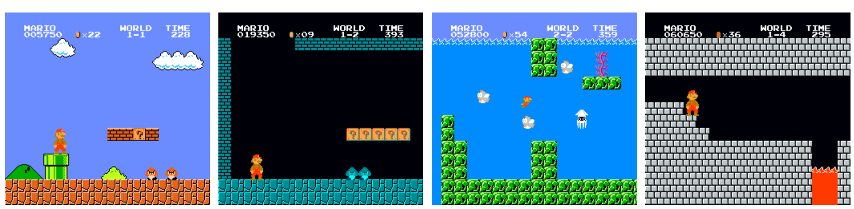

Figure 24: Compound worlds via varied environments in *Super Mario Bros.*.

Japanese gardens and miniature gardens also construct compound worlds by representing a multiplicity of environments, such as mountains, oceans, and forests:

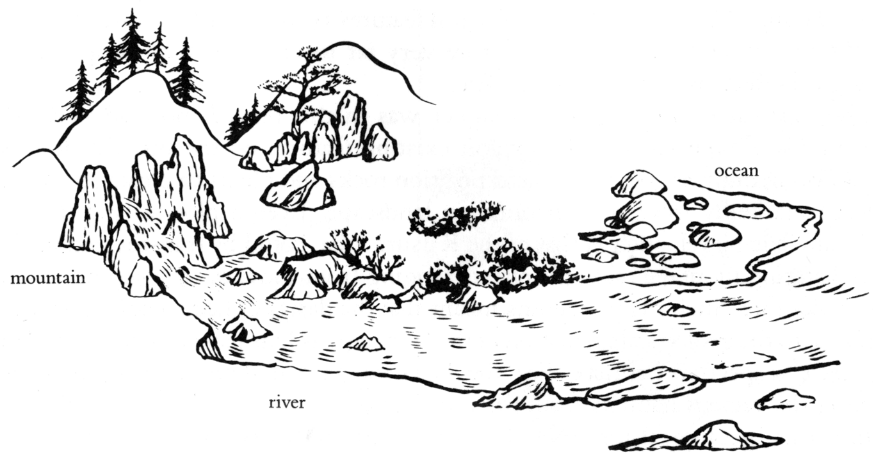

Figure 25: Japanese garden with multiple zones [74, p68].

Slawson writes that the "emphasis on natural features and scenic places reveals a 'feature-oriented' approach to landscape design. Chosen features of natural scenery were re-created in the garden so that they were convincingly present and evoked some of the same feelings one had when actually viewing them in nature" [74, p58]. Japanese gardens build landscapes containing multiple, embedded zones, through techniques for maintaining the internal consistency of each microworld, such as natural habitat and geologic zones, and clearly marked zone borders. Slawson quotes *Illustrations for Designing Mountain, Water, and Hillside Field Landscapes*, a pre-medieval Japanese gardening text:

> In the planting of trees and herbs, you make their natural habitats your model. You will not go astray so long as you bear in mind the principle of planting trees from deep mountains in the deep mountains of the garden, trees from hills and fields in the hills and fields, herbs and trees from freshwater shores on the freshwater shores, and herbs from the seashore on the seashore. For the landscape garden mirrors nature. And thus it is said that in each and all we must return to the two words, natural habitat [74, p62].

Rocks taken from mountain, river, and ocean environments re-create those environments, as the natural forces operating on the rocks have left their mark. Such rocks and plants are abstracted elements that stand for particular habitats and geological zones. Each garden zone is an internally consistent microworld with boundaries marking the frontier between microworlds. Slawson writes that

> it should be borne in mind that in an art seeking to convey the experiential quality of natural landscape, the designer, like the painter, may compress a great distance into a relatively short interval of time and space. For example, in the left rear corner of a site no more than thirty feet square, the designer might plant several cypress on an artificial hill to create the effect of a deep mountain, with a retaining wall of boulders to simulate a rocky cliff. The mandarin orange could then be planted in the right front corner near the residence, in a sun-filled meadowy space that extends out to the front bank of a stream bed (wet or dry) running along the base of the retaining wall. The stream bed thus serves as the primary demarcation between the two habitats and their corresponding geological zones [74, p64].

Internal consistency and clearly marked boundaries create compound landscape microworlds. Landscape elements can suggest different zones and moods, a design principle Miyamoto seems to be well aware of:

> **Nintendo Power:** The first two stages of the game [*Pikmin*] convey a strong, "backyard" feeling, but the next two stages have a grander atmosphere and don't feel like an adventure in a small world.
>
> **Miyamoto:** Perhaps, because those two areas don't have objects like empty boxes or cans. [1]

The castle levels in *Super Mario Bros.* are foreboding environments because of more than a change in the music or scenery. The materials the castles are built out of are unbreakable silver bricks, and the castles contain fewer secrets and hidden treasures than elsewhere. Change in environmental design shifts the emotional tone to one of less playfulness, which makes the castle seem more dangerous, and less subject to Mario's power reversals and whimsical transformations. If there's a knob of miniature garden-ness, it has been turned down for the castle levels.

Brian McHale argues that a trope of postmodernist fiction is constructing complex, compound worlds for the purpose of putting the interfaces between these worlds into flux and exploring the world building process itself [49]. Compound game microworlds afford travel and trespassing between worlds, as well as play with the world building process itself.

2.5.3 Trespassing Across Worlds

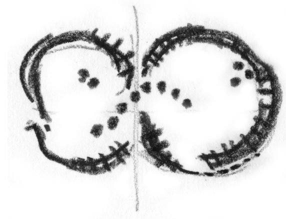

Figure 26: Traffic Across Worlds.

All of the techniques for framing and composing microworlds into compound worlds come together in *Super Mario Bros.* to produce the effect of travel between worlds. A hierarchy of microworlds (worlds and levels) is established through variations in internally consistent environments (above ground, below ground, underwater) and clearly marked microworld boundaries (staircases, status screens, flag poles). The game's microworlds share ludic properties: Mario must evade or beat enemies and travel to the rightmost edge of each level. The compound world of *Super Mario Bros.* carries with it a dominant world organization, that of linear progression from world one through world eight. Traveling between worlds creates the pleasure of immersion into a new environment, and trespassing across worlds adds the pleasure of sneaking across worlds in violation of a dominant organization. Trespassing across worlds is made possible by repeating the technique of putting a world in flux not at the scale of a single world, but at the scale of a compound world as a whole. Trespassing occurs when a compound world's dominant organization is in flux and subject to playful transgression. Warping from world one into world four (skipping two worlds) is a violation of *Super Mario Bros.'s* compound world organization.

Miyamoto, like postmodernist authors, moves us between multiple worlds and across unstable world boundaries. *Super Mario Bros.* communicates Mario across worlds with warp zones, pipes, and magic bean stalks. *Super Mario Bros. 3* adds magic whistles, doors, and magic note blocks to this repertoire.

Trespassing begins with articulated multiple worlds, clear boundaries between those worlds, and a clear dominant organization for the compound world as a whole. The player is then invited to play with the compound world's organization, and the interface between separate microworlds. The progression and size of each level and world in *Super Mario Bros.* is clearly established and then compromised: a hidden bean stalk carries you out of the current level and into the clouds, an alternate microworld; a pipe in level 1-1 brings you briefly into the underground world, a rupture in the normal composition and progression of microworlds; a room behind the end of level 1-2, a space which can only be reached by traveling outside the boundaries of the level and over the interface, lets you warp into worlds two through four, a microworld sequence violation. Some of these ontological diversions are also marked by variations on the dominant game rules. The underground microworld detour in level 1-1 has a different ludic organization than that of each level: the screen doesn't scroll, no enemies threaten Mario, and the room is filled with an unusual number of coins. Many games contain reward microworlds where threats are absent and treasure is abundant, and *Super Mario Bros.* combines this trope with world trespassing, which makes it into a kind of bonus level with additional, intrinsic rewards: the microworld is discovered by testing a microworld's limits, trespassing across them, and discovering a new microworld which other players may not have discovered.

Playing with the boundary between the microworlds of above ground and underground is a trope common to Miyamoto games. The compound world of *The Legend of Zelda* is organized into an overworld, and a system of underground dungeons and caves whose entrances must be found. These underground microworlds have different ludic, musical, and environmental organizations than the overworld. *Super Mario Bros.*, like *Zelda*, is organized into above ground and underground microworlds with both obvious and secret interconnects:

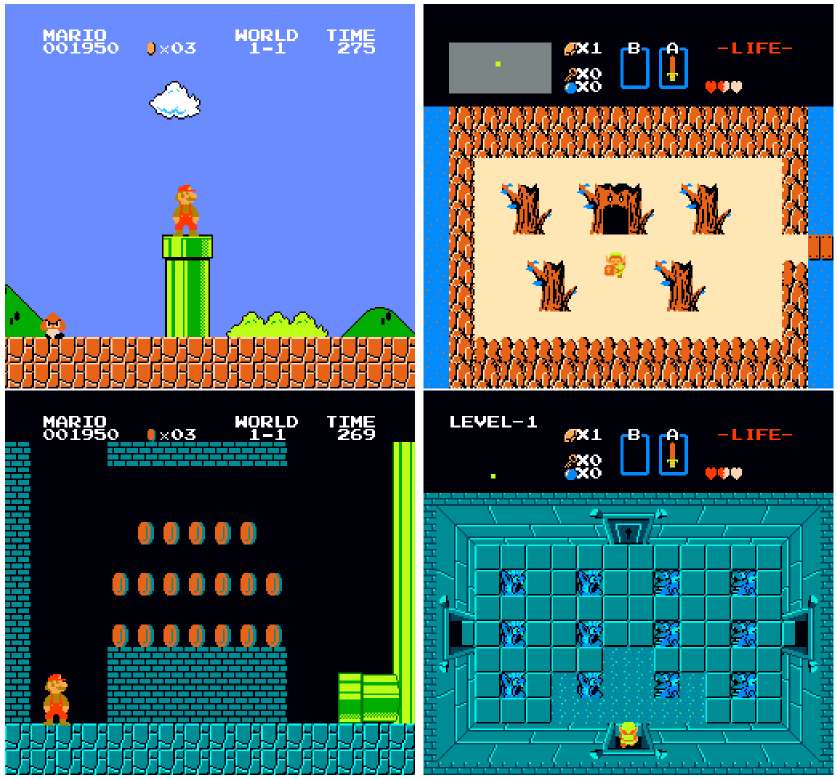

Figure 27: Interlinked overworlds and underworlds in *The Legend of Zelda* and *Super Mario Bros.*.

Just as manholes and gutters in the real world suggest the existence of vast underground networks through the principle of hide-and-reveal, the pipes, caves, and rabbit holes of *Super Mario Bros.* and *Zelda* suggest vast underground microworlds. These underground microworlds lure players into pushing on world boundaries to find hidden seams and enter secret worlds that are always nearby, just below our feet, and tantalizingly out of reach. The seam between worlds, even if it is impassable, is always present, reminding us of an alternate world with an inchoate scope. Hide-and-reveal creates the sense of a vast shadow microworld, and players are teased into imagining palpably close other worlds, and searching for passable interfaces:

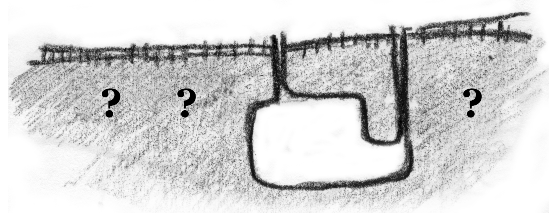

Figure 28: Hide-and-reveal across a seam between an overworld and underworld.

Every game boundary can be exploited as a seam (shared boundary) between parallel worlds. The spatial boundaries which frame a microworld can be used as world seams, as can the rules of a microworld. Many adjacent microworlds can be simultaneously juxtaposed via such seams; *Super Mario Bros.* simultaneously juxtaposes an underworld, a world in the clouds, and alternate world orders in this way. The p-blocks of *Super Mario Bros. 3* activate radical world transformations, and create a shadow world of inverted block and coin. Passageways communicate between spatially adjacent worlds, but transformations superimpose shadow worlds. Transformation tokens such as the p-block, mushroom, star, and fireflower are glimpses into alternate worlds, but rather than transmit us to a nearby space, these tokens are interconnects where shadow worlds intrude on our own.

Leakage between microworlds can occur when materials move from one microworld to another, as in *The Sims* and *Animal Crossing*. Sims can travel from one house to another, between saved games, which reinforces the sense of each household as a distinct, self-sustaining world with its own life. Neighborly visits suggest that each household has an independent life that perpetuates even when you aren't around. Each *Animal Crossing* saved game is a unique environment, a microworld unto itself. Just as two *SimCity* saved games are independent microworlds, each *Animal Crossing* saved game is its own unique village. In one *Animal Crossing* village, peaches might be a common fruit, while in another village peaches might be rare and valuable. Players, however, can communicate objects and characters across these microworlds. It is possible to take a fruit from one village, mail it to a friend playing an altogether different *Animal Crossing* game, and have the object materialize in someone else's village. If two saved games are inserted into the same console, it is possible for your character to ride a train and travel from its home village into the village on the other saved game.

Besides adding ludic interest to a game, the juxtaposition and leakage between microworlds brings into focus the boundaries of each, and heightens the sense of independent, complete worlds. Juxtaposing worlds throws their shared interface into sharp relief, which reinforces the boundaries and completeness of each world, just as the ritual of crossing national borders foregrounds the sovereign and unitary existence of each country. The train and mail link between *Animal Crossing* microworlds foregrounds the self-contained nature of each village, and transforms each saved game into a complete world. The same holds true in the compound world of *Super Mario Bros.*; boundaries between microworlds frame individual microworlds as independent, unitary entities, and leakage between worlds draws attention to this fact.

Why build compound worlds? Compound worlds allow both designers and players of game worlds to repeat the pleasures of world construction and immersion. Compound worlds also contain interfaces between microworlds, boundaries which can be placed into playful flux. Worlds in flux and compound worlds foreground the world making process itself, and the excitement of a new medium for making worlds.

2.5.4 Ontological Foregrounding

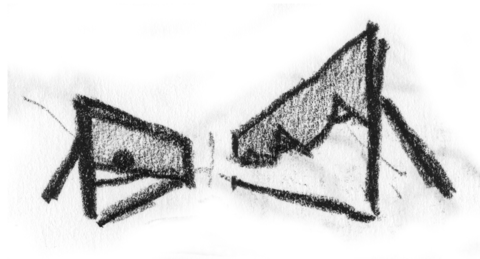

Figure 29: Ontological Foregrounding.

Foregrounding ontology, or the constructed nature of a world, is a preoccupation of postmodernist writers. Foregrounding the world making process is another way to frame a world and call attention to its status as fiction [49, p27]. Two operations are accomplished by this move. Ontological foregrounding is a technique for framing, abstracting, and make a world miniature, intelligible, and safe. Secondly, ontological foregrounding is a ludic operation. The fictional world is simultaneously real and unreal; it's a world, but it isn't.

Calling attention to the world making process itself foregrounds the fourth wall, the interface which demarcates the boundary between our own world and that of the game. The interface between any pair of microworlds calls attention to the separateness, unity, and constructed nature of each. Foregrounding the world making process is another way of establishing the fourth wall that divides the real world from a fictional one.

In some sense, much of the pleasure of playing with worlds in flux and traveling across the interconnected microworlds of *Super Mario Bros.* is sharing with Miyamoto the designer's delight in making a constructed world. It is not hard to imagine the warp zones of *Super Mario Bros.* being used by the game's creators as a debugging tool for quickly navigating a large world, or at least being inspired by such a mechanism. Likewise, each playful change of rules remind us of the designer's ability to do whatever he wishes with a game world. Miyamoto invites us to share his own pleasure in seeing what can and cannot be done, like a baby playing with its fingers, discovering a new medium and its world making possibilities. All of Will Wright's games directly share with players the experience and exhilarating pleasure of constructing, and then bringing whimsical destruction upon miniature worlds. *SimCity* reminds us of its fictional status through our own construction of cities, secret codes for free money, and disasters that can be inflicted upon a city at whim. Cyan titles like *Cosmic Osmo* and *The Manhole* share with players the designer's delight in exploring the ability to fabricate and play with multiple worlds.

Ultimately, what is being played with is not only the boundaries between the embedded microworlds of *Super Mario Bros.* or *Cosmic Osmo*, but the fourth wall and ludic frame itself. The world of *Super Mario Bros.* flickers between a carefully framed play space and an immersive world. Playful reminders of *Super Mario Bros.'s* constructed nature make exploration, experimentation, and play safe, comfortable, and inviting for both players and designers. Playing with the dominant world order of a game repeats the first act of game making and playing: establishing a new world order counter to the existing one.

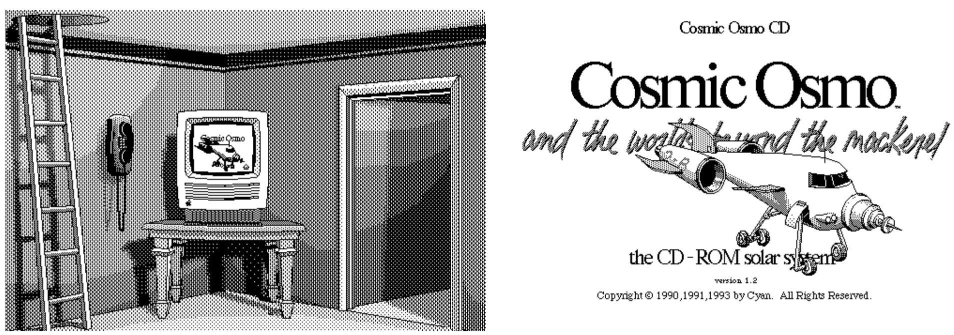

Figure 30: *Cosmic Osmo* foregrounding the ludic frame.

CHAPTER III

MAGIC CRAYONS

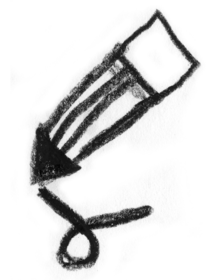

Figure 31: A Crayon, Possibly Magic.

3.1 Building Worlds

Having demonstrated the power of thinking of games as worlds, we turn to the problem of making worlds. How does one represent and author a world? How can we leverage the interpretive processes of players, and representational techniques of authors?

Abstraction is the foundation of representation. Abstraction, Papert observes, creates worlds which are intelligible and plastic. Consistency of abstraction builds complete worlds in the minds of players, as Will Wright's work and design reflections suggest. Scott McCloud argues that abstraction invites audience participation both cognitively and emotionally. Abstract cartoons of faces and people invite our identification, and the gutters between panels encourage the conceptualization of denser worlds [48]. This same principle is at work in the operation of hide-and-reveal, evident in both landscapes and Shigeru Miyamoto's game worlds. Abstraction, also, cuts out the boring bits: films, novels, and video games don't, as a rule of thumb, contain all the details of life (even *The Sims*), but are dramatically compressed worlds, carefully foreshortened to maximize interest.

Besides making worlds intelligible, plastic, complete, and engaging for players, abstraction also helps make world construction tractable for authors. A corollary consideration to a world's intelligibility and plasticity to players, is the world's intelligibility and plasticity to its author. How easy is it to build this world? Artists don't need to render each leaf on a tree; one or two leaves are enough to suggest the complete tree. The problem of representing a world can be reframed in terms of load balancing: what is the weight of realizing a world, who carries it, and what are the implications?

Computers are machines that can represent all of our existing representational modes as well as behavior. Conventions from many genres and media are leveraged in digital media creation, which raises some hard design questions: how do we get the load bearing elements of representation (players, authors, conventions, bits, computers) to support and reinforce one another, rather than antagonize and establish false expectations?

One can pose the design question of abstraction and scale like this: if you have some fixed quantity of representational resources, how will you allocate them to build the richest model? What will the boundaries of the model be, how detailed will you make it, and what are the moving parts? You can't draw every leaf on the tree, so which ones will you draw, and which ones will simply be suggested? Different media forms and genres use different conventions for breaking up the task of world representation. How a representation is load balanced affects a world's believability, representational flexibility, ease of authorship, participatory plasticity, and architectural feasibility.

Representational resources must be properly balanced to create both intelligibility and plasticity. Improper balances create snarled or inert worlds. Worlds become snarled when their computationally complexity resists human intelligibility and manipulation; inert worlds result from representations which rely on dead bits that establish false player expectations and are opaque to computational dynamics.

How a world is constructed also affects the ease of importing, appropriating, subverting, and inventing new representational modes. Maxis was happily surprised by player's appropriation of *The Sims* and the family album area of *The Sims Exchange* as story telling tools [30].

The quality of a world representation scheme exists relative to authorial intent. My orientation is to minimize the amount of authorial time spent building things and remove computational complexity that is irrelevant to the user's experience. I'm interested in making things that are highly plastic in the user's hands, to the point where users can participate in the process of authorship themselves. Sometimes computational complexity, or the lack thereof, is the point, as is the case with both *Eliza* and *Terminal Time* [23]. *Eliza's* computational model is very simple, and yet it produces believable conversation. Appreciation of *Terminal Time* is predicated on audience awareness of how it dynamically assembles documentaries tailored to audience preference.

3.2 Exploiting the Player

The interpretive and participatory processes of players can be leveraged for world building in a number of ways. When an artist constructs a representation of dimensional space on a picture plane, she relies on the audience's complicity in reconstructing the scene. One or two leaves are drawn on a tree, and fuller leaf growth is suggested. People can be teased into imagining more complete worlds, as Wright, Miyamoto, and McCloud demonstrate. Wright describes how characters are represented in *The Sims*:

> Especially right now with current technology, there are a lot of limitations in terms of what we can do with character simulation. So, to me that seemed like a really good use of the abstraction because there are certain things we just cannot simulate on a computer, but on the other hand that people are very good at simulating in their heads. So we just take that part of the simulation and offload it from the computer into the player's head. ... So you know, it's parallel processing of a sort [62].

Player world reconstruction can also be used to achieve other effects. Miyamoto masterfully manipulates expectation and teases players into exploring his worlds by modulating the gap between the world imagined by players, and the one simulated in the game. Analogously, *SimCity* lures players into imagining possible cities and testing their potential existence by constructing them. Players, in this sense, are being called upon to play the role of authors and help build actual digital worlds. McCloud observes that abstraction engages reader identification. Player complicity in world construction is invited through mental completion, exploration, construction, and identification.

3.3 *Exploiting the Author*

Leveraging authorial knowledge from other media is one way to exploit authors in the representation of worlds. Visual artists, for instance, have a huge range of tools and techniques for manipulating the picture plane to create illusions of dimensional space. A bitmap lets painters bring to bear known illusionistic techniques to the problem of world building, while a computational ontology of three dimensional space shifts representational work to the machine, and demands a different set of compositional tools from authors.

Author versus computer representation is not a binary distinction. *Emotisketch* is a paint program I wrote for making dynamic faces [25]. Authors draw free form lines, and wire those lines to deformation primitives:

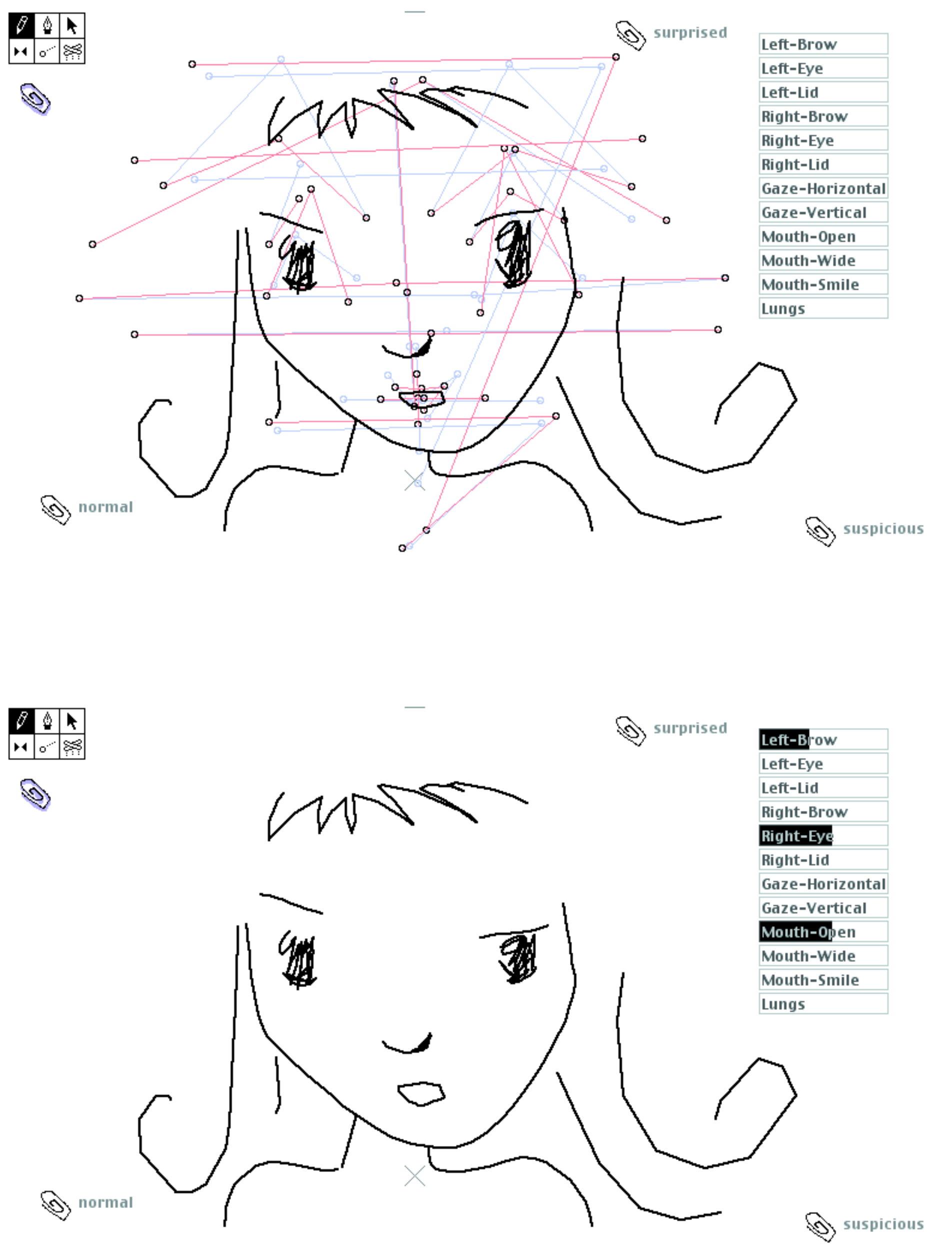

Figure 32: *Emotisketch.*

Artists can leverage conventions for creating dimensional space within *Emotisketch*. All the standard tricks for creating dimensional space are valid, and they can even be extended with deformation parallax. *Emotisketch's* computational representation lends itself to how artists traditionally manipulate the picture plane, and adds the magic of life through animation. The system also holds out the possibility of discovering new representational possibilities with the line and deform primitives.

The difference between an *Emotisketch* face and a bitmap of a face is the difference between live bits and dead ones. The bits that represent an *Emotisketch* composition are structured as lines and muscles that my procedural descriptions of animation and behavior can manipulate. A photograph of a face is dead bits, and is very difficult to procedurally manipulate, and thus cannot be particularly dynamic or interactive. Computation can be used to operationalize representations, and add the magic of procedurality and interactivity; dead bits offer authorial plasticity to authors trained in non-computational disciplines. *Emotisketch* strikes a balance between live and dead bits. Ken Perlin's *Facial Nuance*, which inspired *Emotisketch*, is a thicker computational representation of an expressive face [64]. The entire construction is live bits; even the three dimensional model is constructed procedurally. Building faces in Perlin's program would take more work than in *Emotisketch*. In fact, no such author interface exists, unless you want to write the code yourself. As an artist, I can create an expressive, animated face in *Emotisketch* in under five minutes, and even animate things that aren't faces at all, like cartoon frogs and letters. The medium of *Emotisketch* is very malleable, and can easily be used to author things which were never considered when I wrote the program. By shifting the representational load from the machine (not to mention the programmer) to the author, artists can combine their tricks with *Emotisketch's* computational primitives. Perlin's *Facial Nuance*, on the other hand, operationalizes three dimensional space, so the face can be drawn from any angle. The three dimensional space an author might create in *Emotisketch*, on the other hand, is purely an authorial illusion, totally opaque to the computer.

An extreme example of shifting work from the author to the programmer and machine is *Minstrel,* Scott Turner's story generation system [80]. *Minstrel* is an artificial intelligence program which tells stories using a deep computational model of authorial practice. What an author does is simulated by computationally modeling problem solving, authorial goals, character goals, and meaning. Despite the incredible computational depth of *Minstrel*, the stories it tells are interesting only to theorists and practitioners of artificial intelligence. How is an author going to leverage artistic practice into such a system? Is it necessary or even productive to take something that humans are good at, e.g. storytelling, and simulate the process in a machine? Writing within a computationally thick framework is hard work, even for experienced programmers. *Façade*, an artificial intelligence based interactive drama, has taken two expert programmers two years to author story content for fifteen minutes of real time interactive drama. This does not include the incredible time invested in building *Façade's* architecture.

Minstrel and *Emotisketch* embody quite different design strategies. *Minstrel* attempts to get the effects of what authors do by automating what authors do. *Emotisketch* provides computational primitives for authors to build new types of things. *Minstrel* is a robot that tells stories. *Emotisketch* is a magic crayon.

Just as *The Sims* simulates human emotion in the player's head rather than the computer, simulation can be offloaded from computer to author. Authors, with the proper languages, can leverage their skills in creating artifacts with depth and meaning by tapping into both computational primitives and artistic practice. Artists and authors are masters at exploiting the physical limitations and material properties of a medium for representational gain: pages, panels, and gutters are used in comics to engage reader imagination in world reification; gaps between theatrical scenes are telescoped to represent the passage of variable time; playwrights use off-stage dialogue and sound effects to efficiently convey action. The online storytelling practice that grew out of *The Sims* and *The Sims Exchange* is a case in point, and was encouraged by the offloading of human emotion and intent modeling onto players. Overloading the computer with all the details of a represented world abbreviates a critical role of artists and authors: reinterpreting existing materials for new representational purposes.

3.4 Snarled Worlds

Figure 33: A World, Snarled.

Generally, the thicker and more complex the ontology in the machine, the more viscous, implastic, and unintelligible the material becomes in the hands of both authors and players. If computational complexity is required, and most digital games are computationally complex, dynamics and syntonicity are used to quicken and illuminate.

SimEarth, a landmark interactive simulation with numerous visualization, interface, and computational innovations that designers will no doubt return to, is in many ways a snarled world. In Will Wright's words: "... I have still never seen anyone do an integrated model with an integrated lithosphere, hydrosphere, and atmosphere together like that. And we were getting some effects in the model that were real effects, that really show up, that even some of the more elaborate models that NCAR [National Center for Atmospheric Research] makes weren't capturing" [67, p444]. *SimEarth's* behavior, however, is hard to make sense of. "When *SimEarth* came out I realized at the end that, God, this is like sitting in the cockpit of a 747 in a nose dive. That's what it feels like to most players" [67, p447]. The computational complexity and representation is out of balance with what most people can understand, manipulate, and form goals and hypothesis about. The metaphor of a 747 in a nose dive evokes the additional problem of feedback loop length, the distance between player action and perceivable effects: "The biological systems tend to be very soft, squishy things that you can do something to, and then it kind of reacts and adapts. It's not really clear what you did to it, because it'll then evolve around you. ... people would be playing and all of sudden their planet would freeze up and they'd have no clue why it happened. And I, as the simulation engineer, couldn't tell them either!" [sic] [67, p444]. While *SimEarth* used plenty of dynamics to make the world's complex state plastic, the dynamics lacked intelligibility and syntonicity, a hallmark of Wright's more popular simulation games: "I think people can reason through their failures and assign credit to the failures more easily with the economic models. Plus the idea that you have money and you make money this way and you spend money on that all seems very natural to people, whereas when you get into the complex things like diversity, food webs, and things like that, people just don't have an instinct for it."

Monetary resource management is culturally syntonic. Syntonicity, as Papert argues, lets people relate to and make sense of new systems. Syntonicity helps make systems intelligible, a precondition for player constructed goals, hypothesis, plans, and action evaluation; properties necessary for a system to be used as a game or toy.

Dynamics also help make complex systems intelligible and plastic. How is a player to navigate a system with a huge state space? Dynamics carve our reachable states, transitions, and autonomously change state. Legos, for instance, have connection and stacking dynamics that make construction easy. The dynamics of Lego bricks constrain the kinds of things that can be made (reachable states), and suggest a limited set of interconnections (state transitions) for a player to choose from. It is certainly easier to build a miniature castle out of legos than cardboard. *SimCity* began life as a kind of paint program, and *SimCity* the simulation, we can easily imagine, makes building a city a lot easier than painting one. Without dynamics, a player would have to trudge through each state, carefully placing each image tile. Dynamics, also, give life to a world and are fun to play with. *SimCity's* dynamics create a world with an inner life and vibrancy that a paint program lacks. Syntonic dynamics, such as a city that grows, offer players cognitive hooks for getting their heads and hearts into a miniature world.

A design rule of thumb for adding computational complexity to a system is whether the added work and complexity, on the part of both the programmer and computer, has a corresponding experiential payoff for the player.

3.5 Inert Worlds

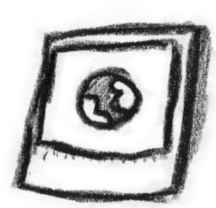

Figure 34: A World, Inert.

Increasing the resolution of a world carries with it the risk of building worlds which are inert, devoid of computational life. Following's Wright's principle of consistent abstraction, increasing the visual detail of a world requires a corresponding increase in behavioral detail. As Perlin pointed out in his 2002 *Game Developers Conference* talk, we have digital characters that look more realistic, but don't move or act with comparable realism [65]. False expectations are visually created which a system's behavior does not live up to. *Final Fantasy 8*, for instance, uses very detailed bitmaps as backgrounds, but the world's dynamic behavior doesn't match its visual realism. High resolution representations that involve detailed bitmap images, complex geometries, or large collections of linear animations not only establish false expectations for the player, but pose serious challenges to world authors. Giving a highly detailed but static world procedural life is hard from the point of view of both computation and design, as large, inert representations are hard to breathe procedural life into. Too many dead bits produce highly detailed but frozen worlds, worlds inert to both player and author manipulation.

Super Mario Sunshine breaks with previous *Mario* games in this regard. In previous *Mario* games, if something was visually represented, such as a dinosaur, cave, or water, it generally behaved in a manner consistent with the visually established expectations. Players could expect to dig into the behavior of a dinosaur through the procedural power of the computer. Even elements in the background, such as clouds and mountains, were operationalized when players romped on clouds, or travelled across mountainous terrain. *Super Mario Sunshine*, on the other hand, builds representations which establish behavioral expectations, but fail to deliver. What appears to be a submarine is in fact only a platform for Mario to stand on. Expectations are aroused and then thwarted, creating a partially frozen world.

3.6 What is a Magic Crayon?

The danger of moving too much representational work to the programmer/machine is a snarled world; a medium whose complexity is unintelligible and, as a result, implastic. Keeping representation locked up entirely in an artist's static two dimensional images, video, and models, on the other hand, is also untenable, as inert worlds result. Too many dead bits require Frankensteinien acts of computation to gain procedural life.

New representational materials which build upon the authorial practices of both computation and art will enable artists to leverage artistic techniques with computation. Inventing magic crayons, or new languages that combine existing authorial practices with the computer, is coterminous with both the practice of art and computational language design. Designing new languages of representation is at the heart of both computer science and the arts. Magic crayons are a class of representational languages that combine art and computation.

As a society's computational literacy level changes, the standards for what is and isn't a magic crayon will shift. TeX is a computational language that integrates the practice of page layout, typesetting, and printing. TeX, however, is not a generally accessible language. If everyone could think more like a computer, it might be a magic crayon.

Director and *HyperCard* are examples of magic crayons. These authoring tools combine computation and representation into systems which are accessible to ten year olds. The magic crayon-ness of *Director* and *HyperCard*, however, bottom out when we leave the domain of the stage, score, cast, cards, and buttons. Leaving that domain takes us back to the land of imperative programming.

The challenge of inventing magic crayons is combining the power of computation with expressive possibility in a novel form.

What are the properties of a magic crayon?

ACCESSIBLE. Magic crayons, like normal crayons, must be readily available, cheap, and robust. Accessibility also means that children can obtain satisfying results from picking up and playing with a magic crayon.

SKETCHABLE. Magic crayons enable authors to obtain satisfactory results with a small amount of effort. Things can be sketched, or partially specified, and they work. Sketching also means that creation is cheap: what one makes is playable in an incomplete form, and doesn't require a large time commitment. Although fully functioning, sketched things remain malleable, both mentally and physically, because a sketch isn't fully reified. Sketching means that the state space authors navigate is dense with interesting results, and that these interesting results are easily accessible.

The notion of sketching and low creation overhead is applicable to tools which aren't related to drawing at all. Apple's *Stickies* program is more of a magic crayon than Microsoft *Word*, since *Stickies* doesn't require users to think very hard about creating, saving, and destroying notes. Sticky notes are low cost and friction-free.

COMPUTATIONAL. Magic crayons are magic because they are imbued with the power of computation. Magic crayons allow non-programmers to engage the procedural qualities of the digital medium and build dynamic things.

Expressive. Magic crayons are artistically expressive, and rely upon authors to create meaningful worlds. One can draw many things with a crayon. Magic crayons allow authors to leverage and improvise a wide variety of artistic techniques.

3.7 Authorial Modes

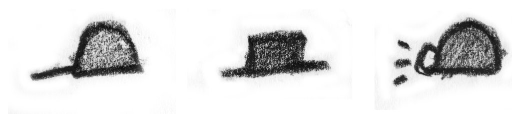

Figure 35: Participatory Hats: Player, Author, and Programmer.

Authorship is a participatory mode that deserves special consideration: how does a representational system afford authorship? What are the primitives an author builds with, and how do these primitives affect what can be built, and the process of building itself? What is the relationship between authorial and non-authorial modes of participation?

Making systems (i.e. Legos) for constructing new things (space ships, castles, abstract structures) is a fundamental practice of digital design: language design, a topic close to the heart of computer science, is making new systems (grammars, compilers, interpreters) of representation for building new types of things; authoring tools like *Director* are systems of representation for making other systems; *Photoshop* is a system that lets one build and manipulate images, but unlike *Director* and most computer languages, what one makes in *Photoshop* is not dynamic. Systems that can be used to make systems gives rise to nested authorial levels. Espen Aarseth identifies as levels of usership the phenomena of nested participatory modes which result from systems for making systems [3, p172].

The participatory and authorial modes afforded by a medium impact who gets to do the making and playing. If we put aside the vertigo of recursive systems, and orient ourselves at a particular usership level, we find that participatory modes break down into some configuration of programmer, author, and player roles. In the hands of the public, *Super Mario Bros.* can only be played, but at Nintendo, someone programmed the architecture of *Super Mario Bros.*, and someone authored the world's content. Programmer and authorial modes in *Super Mario Bros.* are private.

Chris Crawford's *Erasmatron* distinguishes all three of these participatory modes: Chris Crawford is *Erasmatron's* programmer, other people build storyworlds with *Erasmatron*, and someone else can play the storyworlds. *Façade* consists of numerous architectures that bring its storyworld to life, and Mateas and Stern have also authored the content of *Façade's* world, keeping the system's programming and authorial roles to themselves.

In reality, who gets to play what roles is controlled by availability and accessibility: Nintendo doesn't share *Super Mario Bros.'s* world authoring tools with the public, and authoring content for *Façade* is a non-trivial task.

3.8 Encouraging Authorship

The problem of making a public authorial mode accessible can be framed as facilitating mobility from player to author roles. Authorial accessibility is governed by the intelligibility and plasticity of a system's primitives to an author. Bill Atkinson's *HyperCard* provides graded user levels: browsing, typing, painting, authoring, and scripting. The elegance of *HyperCard's* graded design is that the machinery which one makes use of transparently at one level becomes exposed at the next level down. When browsing, for instance, one sees lots of images and text, and the next two authorial levels, typing and painting, allow users to fabricate text and images. *HyperCard's* authoring mode allows users to link cards, and this interface automatically generates HyperTalk code at the scripting level, which facilitates mobility between authorship and scripting levels. Multiple paths for solving the same problem allow users to compare knowledge across paths, deepening understanding of *HyperCard's* workings. Interrelated and graded participatory levels encourage users to wade deeper and deeper into authorial waters, and leverage familiarity in understanding the unknown.

HTML's widespread use resulted from features intrinsic to the language's design. HTML is robust, open, fluid, easy, incremental, and modular. HTML is a reliable system, and tools for writing and viewing HTML are publicly available. Authorial descriptions of web pages, the HTML source, are available for other users to learn and steal from. Switching between the role of HTML user and author is fluid, because the relationship between a page's code and appearance is clear. HTML is also an easy language. The relationship between code and web page is intelligible, and the language substrate (ASCII) and syntax is a plastic and forgiving material. Authorship in HTML can be incremental; one doesn't have to build an entire web site or even a complete web page to successfully make something. HTML's incremental authorship affords sketching. The structure of HTML is modular: it is easy to steal or reuse the code that describes an entire site, single page, chunk of layout, table, or hyperlink.

Modularity also means encapsulation, which makes reuse and authorship easier. Encapsulation produces modular entities with clearly defined boundaries and overviews. Getting your head around a cleanly encapsulated unit is straightforward. Rather than have a snarled spaghetti world where the authorial description of every object is interconnected to a hundred other objects, each entity is neat and self-contained. Non-interdependence means that reusing and exchanging entities is straightforward.

3.9 Primitives

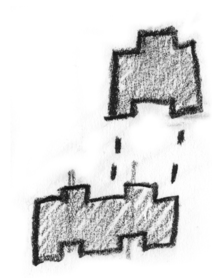

Figure 36: Interlocking Primitives.

One way of conceptualizing a system for constructing things is as a set of primitives, or self-contained building blocks. The primitives of Legos are individual blocks. Rather than consider the representational system all at once, we break it down into pieces which may operate differently, but nonetheless operate in the same kinds of ways. There are different kinds of lego pieces, but we can compare and think about how each piece connects to others, has a color, weight, size, or may be particularized into flags, people, or wheels.

Primitives stand for the fundamental set an author builds out of, but the primitives given by a system are not necessarily the same as those conceptualized and used by an author. Papert points out that microworlds are both material and conceptual: the sock pairing microworld exists in a child's head [61, p162]. One can build with Legos and add the constraint that we are only building with blue pieces. This constructs a mental microworld with a different set of primitives than those articulated by the Legos themselves. The primitives one uses to build something are both material and conceptual entities.

Chapter 7 of Michael Mateas's dissertation, *Interactive Drama, Art and Artificial Intelligence*, treats this topic in depth [41]. His discussion takes place in the domain of semiotics, which he proves useful for grounding out fundamental issues of computer language design. The semiotic language, however, isn't light weight enough for talking efficiently about authoring tool and language design. I propose using a higher level language oriented around the concept of primitives which can be grounded out in Mateas's semiotic work, but is more malleable for designers.

Primitives exist at the nexus of programmer, author, and player roles. Primitives are built by programmers for authors who, in turn, build things for players. Designing or modifying primitives is the practice of changing the fundamental set, the primitive elements in play. To consider the design of primitives, we must have a language that grasps their existence in three worlds simultaneously. It is useful to think of primitives as having valences that bind them into the domain of the programmer (physical valence), author (logical valence as building block), and player (experiential valence).

A card in *HyperCard* exists as a computational construct made of data structures and code, as a unit of segmentation authors can use to make stacks, and as a place that users can be at when browsing stacks. The two dimensional space of a card has a representational valence in the worlds of the programmer (the data structures and code that manages its bitmap and ordered list of objects, for instance), the author (a place to put things), and player (a place to be). A card's two dimensional space has code, authorial, and player valences.

Some primitives give physical and logical order to other primitives. *HyperCard* cards are two dimensional spaces which lend form and intelligibility to buttons, fields, and bitmaps. *Director* makes use of two dominant organizing primitives: the score and stage. Other primitives, such as cast members, operate within the physical and logical contexts carved out by the score and stage.

3.10 Domain

The material properties of a medium impact what can be represented within it. Primitives are intimately related to what one can build, and different media forms have non-uniform representational powers. In the language of comics, for instance, time is very elastic. Simulations of cartesian space and time lend themselves to games whose primitive actions are jumping, moving, and shooting. *Myst* uses an entirely different model of space than *Super Mario Bros.*. Building a game whose primitive actions are jumping and shooting within *Myst's* spatial ontology is absurd, and *Super Mario Bros.* lacks the spatial elasticity which makes *Myst's* worlds navigable. Films often contain car chases, but stage plays do not.

Certain primitives lend themselves to particular domains of representation: it's quite easy to build cities with *SimCity*, animations in *Director*, and spaces in *HyperCard*. Building an animation in *HyperCard* requires scripting or sequencing and rapidly playing back cards to create the illusion of motion. The most common way to build space in *Director* is manipulating the flow of time to loop or stop, creating a solidity which suggests space.

There is no single representational material which we can use to build every type of world. A plurality of magic crayons are needed to make a plurality of worlds.

3.11 Origami Possibility Spaces

A system's primitive elements combine, interact, and fold into a multiplicity of worlds and possible worlds:

Figure 37: Primitives folding into an origami possibility space.

Primitives fold into possibility spaces with particular sizes, domains, and densities. The primitives that authors work with influence the kinds of worlds built, and whether construction is easy or hard. The primitives that players interact with govern the shape of a game's possibility space.

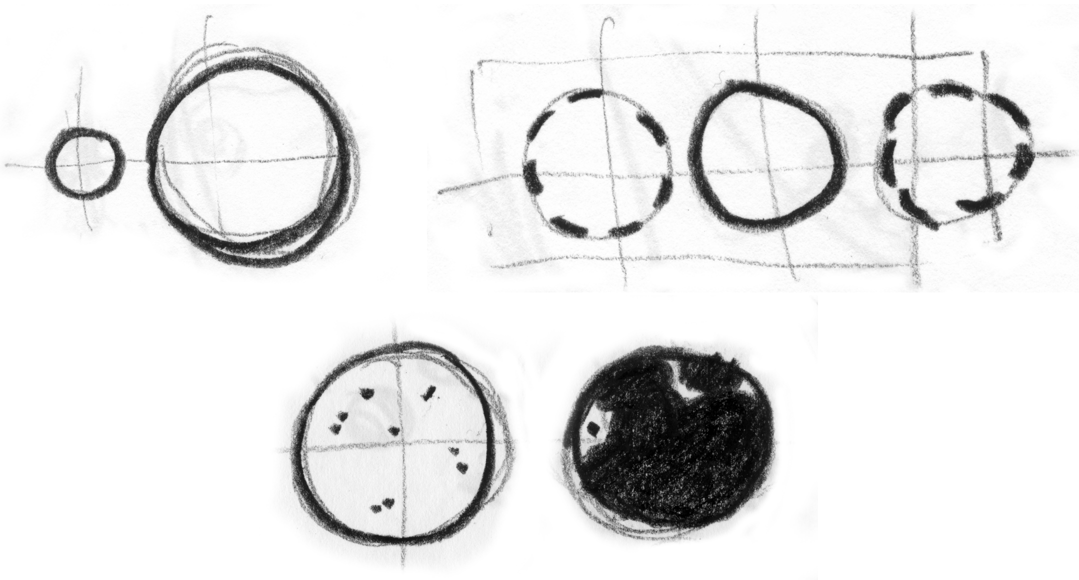

Figure 38: Primitives fold into possibility spaces with particular sizes (how many states are in here?), domains (what kinds of things can I do?), and densities (how many states are interesting?).

A programmer working in a language like C navigates a huge space of possible programs that is only sparsely populated with interesting results. Programmers make primitives for authors, and the space which authors work within is smaller, denser with interesting results, and more tuned towards a particular domain. It's easier to make things of a certain type in *HyperCard* than C. The possibility spaces that players navigate are fabulously dense with interesting results, but are far smaller than the spaces authors and programmers navigate:

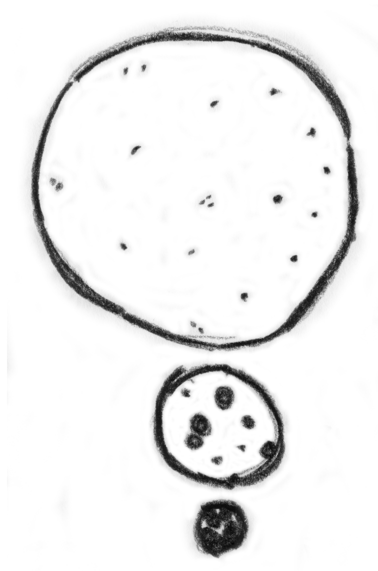

Figure 39: Programmer, author, and player possibility spaces.

Programmers work in a huge possibility space with few interesting programs. Authors work in a smaller, but denser space. Players engage smaller possibility spaces that are densely populated with interesting results.

Authors define the primitives engaged by players, and try to optimize the ratio between game primitives and game possibilities. Variations on a theme, which players use to "sniff" and obtain an overview of a game's possibility space, is viable because of how game primitives fold and recombine into a possible worlds landscape.

Programmers, by defining authorial primitives, influence the shape of the possibility space authors navigate. Programmers try to optimize the ratio between authorial effort and player experience. *Comic Book Dollhouse* offers authorial primitives that are accessible to many people, which means that the total output of author produced work is potentially high. *Façade* provides authorial primitives which require great skill to use, but can be used to fabricate player experiences that would otherwise be unreachable. *A Pattern Language* is an authorial microworld, a set of primitive elements and rules that establish a possibility space dense with interesting results.

Magic crayons fold into particular types of origami worlds. The possibility spaces authors navigate with magic crayons are dense with interesting results (accessible and sketchable), big (expressive), and occupy flexible and procedural domains (computational and expressive). As one might imagine, making authorial primitives requires tradeoffs along the dimensions of possibility space size, domain, and density. A language like TEX affords exploration of a huge space in the domain of typesetting (navigation is aided by the procedural power of the computer), but this space is not dense with interesting results.

CHAPTER IV

PULLULATION & LUDIC PLAYABILITY

Figure 40: Possible Worlds, Bifurcating.

4.1 Possible Worlds Landscapes

Digital worlds are procedural, which means that they can exist in a variety of states. The procedural description of a digital world defines a landscape of possible worlds: multiple world states and their relationships to one another. A digital world's dynamics, defined by its makers, gives rise to a possible worlds landscape that is traversed by players.

One of the experiential pleasures of both narrative and games is pullulation, the feeling of negotiating multiple potential worlds, the tension of multiple possibilities, and the resolution and celebration of this indeterminacy. The term *pullulation* comes from a Borges story, "The Garden of Forking Paths" [9]. This story makes explicit what all stories have in common, which is the negotiation of potential worlds in the fictional world, the minds of fictional characters, and the mind of the reader.

Narrative action is predicated on conflict, and conflict is generated by incompatible possible world states: incompatibilities between the actual state of the fictional world and the dreams, plans, and beliefs of characters within that fictional world [68, p156]. Ludic interest is also predicated on diversification of possible worlds. Will Wright describes the ludic interest of divergent possible worlds in *Go*:

> In *Go*, both players have a model of what's happening on the board, and over time those models get closer and closer and closer together until the final score. At that point you have a total shared model of, you know, "you beat me." (Laughter.) Up until that point, though, there's quite a large divergence in the mental models that players have. Especially if you ask them what the score is, or "How are you doing?" They'll frequently say, "I'm doing pretty well, here," or "He's whipping me." Or that backwards thing, "Oh, he's whipping me," when really you're the one winning. And it really comes down to how each person is mentally overlaying their territories onto this board. In each player's mind, there's this idea that "Oh, I control this and they control that, and we're fighting over this." They each have a map in their head of what's going on, and those maps are in disagreement. And it's those areas of maximum disagreement where the battles are all fought. You play a piece there, and I think "Oh, that's in my territory, I'm going to attack it cause you're in my territory." Whereas you're thinking, "Oh, that's my territory, you're invading me." And finally, the battle resolves that in our heads, and then it's pretty clear that, "Okay, that's your territory and that's mine." So the game is in fact this process of us bringing our different mental models into agreement. Through battle. [62]

Marie-Laure Ryan's *principle of narrative tellability* seeks "the diversification of possible worlds in the narrative universe" [68, p156; emphasis removed]. A corollary *principle of ludic playability* is responsible for ludic interest: diversification of possible worlds in games gives substance to phenomena such as contests, challenge, agency, winning, and losing. Placing a stone in *Go* closes off certain potential game states and opens up others. The game's combinatorics generate a wild bifurcation of possible worlds from the placement of a single stone. Indeterminacy and interest approach zero as a *Go* approaches its end. The possible worlds which orbit the actual game board state converge towards equilibrium. No further moves are necessary. Both players pass, the score is counted, and the game is over.

When the possible worlds of a game converge, it ends. If convergence takes place before the game's formal ending, participants end play on their own, as games with undiversified possible worlds are dull. Players often resign when it is clear they will lose, and *Go* has a handicap technique for making matches between players of disparate skill interesting.

The kernels of possible worlds diversification, such as chance, agency, and transformation, are operationalized differently in narrative and games. While narrative pullulation toys with the expectations and plans of characters and the expectations of the reader, ludic pullulation engages the expectations, plans, and actions of the player directly, since the reader is no longer a reader, but a player who participates in the action and directs travel across a landscape of possible worlds. Agency and chance are narratable ("Jane buys a lottery ticket and wins."), but agency and chance in games can exist at the same level of reality as us, the physical reality of the narrational machinery. Players act within the nexus of pullulating worlds.

The topographical features of a possible worlds landscape generates interesting narrative and ludic experiences. The structural relations that bind bifurcating possible mental and material worlds give rise to narrative and ludic experiences like planning, risk, strategy, threats, forced moves, power, contests, disguise, surprise, and optimism.

Game designers define possible worlds landscapes that players traverse. The dynamics and rules of a game shape its possible worlds landscape. Rules and dynamics constrain player input, limit reachable game states, limit transitions between possible game states, and automate game state transitions. A game's rules are a web of constraints which hold its moving parts and experiences together, enable and prompt action on the part of the player, give intelligibility and plasticity to the system, imbue a miniature world with dynamics and life, generate the experience of responsibility, enable problem solving and contests, and give value to particular game states.

SimCity's evolution from paint program to game is paradigmatic for thinking about the role of constraints and dynamics. *SimCity* began life as a level editor, a paint program for making "islands with little roads and cities on them."

> **I read that the level editing tool for *Bungeling Bay* was your inspiration for *SimCity*.**
> It was a character set that actually described a bunch of islands with little roads and cities on them. And so there was such a big area that I developed my own little character editing program. I found that I was having so much more fun with the paint program than I was with the game that after I finished the game I kept playing with the paint program. And it eventually evolved into *SimCity* [67, p436].

Apart from illustrating Wright's propensity for making things rather than blowing them up, a feature that sets apart Maxis titles, the evolution of *SimCity* from paint program to game illustrates the role of constraints and dynamics. Comparing *SimCity* the paint program (*Bungeling Bay* level editor) and game (*SimCity*) is a productive thought experiment for evaluating the structure, function, and experiential effects of constraints.

Constraints can be instantiated in digital worlds as rules and dynamics, but they can also be mentally deployed by designers. One of Will Wright's design practices is to build open simulation worlds, and then mentally impose a system of constraints, which makes the system more game-like (personal conversation). Possible constraints are play-tested alongside an existing system before they are fixed into a game's code.

Rules are the abstractions that define a microworld, and give it intelligibility and plasticity. Constraints, dynamics, and rules give shape to a landscape of possible worlds, and carve out the topographical features that give rise to experiences such as agency, chance, exploration, planning, strategy, risk, regret, threats, forced moves, power, contests, winning, losing, challenge, inevitability, optimism, responsibility, empathy, disguise, and surprise.

4.2 Constraints & Ludic Playability

Experiences such as responsibility, risk, regret, and strategy are structurally dependent on constraints that interrelate possible worlds. If I choose to cross a rickety wooden bridge over a canyon which drops hundreds of meters into rapids swarming with sharks, I'm taking a risk. This risk is predicated upon the distance between the possible worlds of death below and living above:

Figure 41: Risk on a possible worlds landscape.

If after landing in the water I can somehow teleport myself back up to the top of the bridge, falling in the water has no consequences, and the risk involved in crossing the bridge evaporates. Threats are also predicated on a particular relationship between possible worlds:

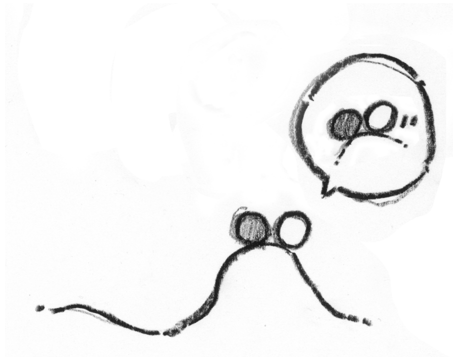

Figure 42: Possible worlds landscape in which a threat is being made.

Loftus & Loftus articulate the experience of regret in terms of possible worlds:

> "If only I had put on the radiation suit," you say to yourself, "I wouldn't have died that horrible death in the radiation chamber." And since the alternative world in which you put on the radiation suit is very close to the "actual" world in which you didn't, regret is very high. But since you saved the game, you can go back and create the alternative world, thereby eliminating the regret. So you do. Computer games provide the ultimate chance to eliminate regret; all alternative worlds are available [39, p32].

Part of the allure of games, Loftus & Loftus argue, is risk, regret, and repair through replay. Replay does not eliminate risk, as *all* alternative worlds are *not* available in games. Limited chances and incomplete alternate world recall means that replay doesn't violate the constraints that create risk and regret. *Pac-Man* has multiple lives, but a fixed quantity of them, so getting caught by ghosts is a threat. *The Legend of Zelda* has no lives, but when Link is killed, game state components are rolled back. Link is returned to the dungeon entrance with additional costs incurred: Link only has three hearts, and resources such as money or bombs spent during the attempt are not refunded. Undoing what the player has done is an intrinsic penalty that gives weight to actions and can be used to generate regret, risk, danger, failure, and winning. This intrinsic cost is similar to what happens when some other kid comes over and knocks down the wooden blocks you've been carefully assembling into a castle. Bullies are one way to add risk, strategy, failure, and winning to a playground.

Eliminating alternative possible worlds from a game's state space can yield powerful experiential effects. While getting killed in *Pac-Man* feels like getting caught and starting over rather than death, Aerith's death in *Final Fantasy 7* generates substantial empathy and loss, rather than a simple feeling of regret and a desire to retry. Pac-Man's demise is preceded by a high degree of perceived agency and replay potential, but Aerith's death is marked by a loss of control and a sense of inevitability, a feature shared by the death of Floyd in *Planetfall* (many players cried), and also the death of the homeless man in *Groundhog Day*, who dies despite the efforts of Bill Murray's character in numerous parallel worlds. Agency in a multiform world, it seems, can be used to heighten a sense of impotence and loss, by affording the traversal of multiple possible worlds only to reveal that a particular outcome is rigidly immutable. While thwarting agency is generally not a designer's goal, *Final Fantasy 7*, *Planetfall*, and *Groundhog Day* illustrate the expressive potential of such a strategy.

SimCity deploys limited funds as a ludic constraint on player activity. A fixed quantity of funds demands allocation and strategizing about how to obtain more. If a limitless quantity of money were available, we would have something closer to a paint program; there would be no limit to how one could modify the city, and agency allocation would not be a problem to solve. Problem solving only exists within a context of constraints.

Constraints carve out a space of reachable states, and the pathways of the state space add intrinsic meaning to particular states. Winning and losing states of a game are not interesting if they are adjacent to every other game state. If at any point during a game of chess a winning move is available to both me and my opponent, we no longer have a contest, but a collaborative exercise in arranging chess pieces; such free accessibility robs winning and losing of meaning. A degree of inaccessibility is a necessary condition for winning and losing states to exist. This principle can be generalized to goal oriented activity in systems which aren't contests at all, such as Will Wright's constructionist toys. If any sequence of moves yields a player goal state, the sequence of moves and goal states lose their value. Winning exists in the face of challenge:

Figure 43: Challenge and winning on a possible worlds landscape.

Goals and solutions have meaning when alternate event sequences are conceivable where the goal's attainment was thwarted. This principle of ludic playability is isomorphic to Ryan's principle of narrative tellability; interesting conflict is constructed by divergent event sequences, imagined or real. Alternate event sequences give agency, conflict, problem solving, failure, and goal states meaning. The participatory pleasures of contests and problem solving are predicated on constraints.

If a system allows players random access to possible world states, then we end up with a system more like a paint program than a game. Paint programs, as Wright points out, provide access to huge state spaces, but only a tiny fraction of accessible states are actually interesting. Games, on the other hand, constrain accessible states to those which are interesting. Interesting failure states are an important design rule of thumb proposed by Wright. If a player fails to direct a sim in *The Sims* to the bathroom, the results are still entertaining. When players push on the boundaries of *Super Mario Bros.*, the game world rewards players, rather than breaking. An extreme case of interesting failure states is *Bridge Builder*, a game by Alex Austin. Watching a failed bridge's spectacular collapse is far more interesting than watching a bridge work properly, because a bridge that is working properly doesn't do much. All of these games reward subversion.

What about games in which players always lose, such as *Missile Command*? Is failure meaningless because all play results in loss? At the global scale failure is meaningless, since winning the game isn't possible or conceivable, but at the scale of shooting down incoming missiles, protecting a city, and completing a level, failure and success have meaning, since alternate event sequences and outcomes are possible. The principle of ludic playability, conceivable alternate event sequences where a goal's attainment is thwarted or successful, is adhered to in *Missile Command*.

Wright describes game players as climbing solution landscapes of possible world states to reach goals. These landscapes are formed by the geological processes of game dynamics. Game dynamics constrain reachable states and establish transitions among them. *SimCity's* dynamics offer resistance along certain event pathways, and help along others; some states are hard to reach, and others are not. A possible worlds landscape with no constraints is the equivalent of mountain climbing in Kansas.

4.3 Input Constraints

Lack of input constraints poses a huge interface design hurdle: if millions of possible worlds are accessible from this one, how is the player to select, let alone conceive of, an adjacent world? Trained designers and sophisticated players use mental microworlds to navigate through vast landscapes of possible worlds. Christopher Alexander's *A Pattern Language*, for instance, is an architecture microworld whose constraints help people navigate the vast space of possible structures. Games constrain the types of things players can do for reasons of ludic interest as well as intelligibility.

The graphical user interface championed by Apple structures user input so that only valid inputs are allowed. A command line interface will happily accept any sequence of characters, but only a miniscule fraction of possible inputs actually do something. Crawford conceptualizes the design advance of the GUI as a reconfiguration of the ratio between conceivable and reachable states. GUIs are better than command line interfaces in establishing expectations consistent with what the system actually does, resulting in a tighter mapping between conceivable and reachable states [16, p62].

Elegant input structures use a small vocabulary of user commands for a wide variety of purposes. *Super Mario Bros.* is a masterful example of how elegant dynamics can give intelligible and multiform meaning (activating blocks, breaking blocks, leaping over things, squashing creatures, variable jumping height) to a simple input, such as pushing the jump button. Mario's water pack in *Super Mario Sunshine* allows Mario to float, fly, clean things, water plants, push switches, and swim more quickly through water.

4.4 Motivating Rules

Motivating the rules of a digital world is a topic that is too large to be treated here in depth. In general, rules can be motivated and explained through *syntonicity* and *scripting play*, topics discussed earlier. Limited resources, for example, are often represented as money or minerals, and the removal of entities from a game world is often represented as death. *The Legend of Zelda* uses myth syntonicity to organize the world into a sensical whole, and motivate player action. *Go's* dynamics are syntonic with space, power, and contests. As discussed earlier, syntonicity lends microworlds intelligibility, and allows games to have cultural currency beyond their magic circles.

CHAPTER V

WHO AM I? POINT OF VIEW IN GAMES

Digital worlds invite player participation. Participation is the difference between screen savers and games. Both digital games and screen savers are procedural worlds, but participation sets them apart. One can participate in a game world, but only watch a screensaver.

Participation raises some complex questions: who am I in this world? If I, as a player, can act upon a world, what is the relationship between me and the world?

The participatory qualities of digital worlds complicate point of view: what is the player able to manipulate? How does manipulation impact identification? How can modes of participation, such as degree of control, be accounted for?

A common game trope is the player character, something in the game world that stands for the player. Players manipulate and identify with Mario in *Super Mario Bros.* What do we make of that? Although some games present the world through the eyes of a player character, others do not. Many games, such as *Tetris*, have no player character at all. Sometimes player manipulation is channeled through a single entity, frequently the player character, but divergent strategies are employed in games such as *Tetris*, *Lemmings*, and *The Sims*, where the player manipulates a sequence of falling blocks, pokes and prods at a herd of simple-minded critters, or directs one of a set of people to take action.

Although point of view is an ancient focus of storytelling craft, and a significant topic in literary and film criticism, we do not have an adequate language for cataloging the structure and species of participatory point of view witnessed in digital worlds. To date, the best conceptual tool is Marie-Laure Ryan's internal/external and exploratory/ontological typology [69]. This typology, however, isn't sharp enough for the task at hand.

SimCity and *SimLife* are both placed by Ryan into the category of external/ontological: "In these games, the user rules over a complex system, such as a city, an ant colony, or an empire, and his decisions affect the evolution of the system." But what about *SimAnt*? Players not only oversee the development of an entire ant colony, but are also placed directly into the shoes of a particular yellow ant, much like player control of Mario in *Super Mario Bros.*. Internal/external, also, isn't sharp enough to distinguish between narration and manipulation. Player actions are often channeled through a particular character, such as Mario, a kind of first person manipulation, but the world is described from the third person. The most basic typology of literary point of view is considerably more elaborate than the internal/external modes proposed by Ryan: first person, second person, third person limited, and third person omniscient; unreliable/reliable; stream of consciousness.

The structure of point of view in participatory media can best be described with four variables:

Epistemic Access. *What can I see? How is the world described to me?*

Locus of Manipulation. *What do I control, and how?*

Player Character. *Who am I?*

Identification. *What do I care for?*

Figure 44: Epistemic access, locus of manipulation, player character, and identification.

Epistemic access and identification operate in non-participatory narrative media such as the novel and film. Narratological tools for understanding epistemic access and audience identification can be brought to bear on these aspects of point of view in games, but accounting for film and literary point of view theory falls outside the scope of this thesis. Participation demands we account for both agency and the player character. The player character, a representation of the player within the realm of a game's projected world, is a trope of participatory media in general, and games in particular. The player character variable interacts with epistemic access and locus of manipulation to generate a host of point of view architectures. In addition to its descriptive power, the peculiarity of the player character convention to participatory media merit its modeling as an independent variable.

In *Eliza*, the player is instantiated as a patient in a fictional world that also contains a psychotherapist. The world is narrated through the discourse of the character Eliza, and we have no reason to believe that she is an unreliable narrator or that the fictional world she inhabits does not correspond to her description of it. Eliza (the character) speaks as if we, the player, are her patient, which constructs the player character as a patient. Player discourse also narrates the world, and this input is interpreted by the program as the patient character's speech, which links player manipulation to the patient. In *Eliza*, manipulation and epistemic access are both located in a patient player character with whom the player identifies. Narrative and manipulation are both in the first person. Sherry Turkle observes that when you play a video game, "you have to do more than identify with a character on the screen. You must act for it" [79, p83]. Turkle argues that this control sets into motion a new mode of identification: *identification through action.* Players act for the patient in *Eliza*, which motivates the player to identify with the patient and engage the dramatic situation. The patient character is inchoate, and this invites *identification through abstraction,* a mechanism Scott McCloud identifies at work in comics [48].

Player control in *Super Mario Bros.* is located in a single player character, just like *Eliza*. Epistemic access in *Super Mario Bros.*, however, is in the third person. Although Mario is our player character, we see him in the world from a vantage point different than that of Mario. In this regard, *Super Mario Bros.* is different from *Eliza*, which narrates its fictional world from the point of view of the patient player. We can make sense of Mario as a player character through body syntonicity; his body, like *Logo's* turtle, behaves in a manner consistent with our body knowledge. Pushing left makes Mario walk left; pushing right makes Mario walk right. Player input is directly mapped to Mario. In addition to identification through action, players also identify with Mario through body syntonicity.

Puzzle Bobble is a game where bubbles must be shot at an advancing array of bubbles. When groups of same colored bubbles are formed, they are removed from play, much like the elimination of completed lines in *Tetris*. Manipulation is located in a bubble gun at the bottom of the screen: pushing left and right causes the gun to rotate, and pushing the button causes it to fire a bubble.

Epistemic access is in the third person: players have an omniscient view of the game's action. What makes *Puzzle Bobble* peculiar is that it has a player character, and the player character is not the bubble gun. In both *Super Mario Bros.* and *Eliza*, manipulation is located in a player character, but *Puzzle Bobble* locates manipulation elsewhere. A little dragon serves as the player character in *Puzzle Bobble*: when the player wins or loses the dragon cries or rejoices, and the bubble gun is represented as being steered by the dragon. *Puzzle Bobble* encourages identification with the dragon player character, narration is omniscient, and manipulation is located not in the player character, but a bubble gun. *Puzzle Bobble* also encourages identification with the bubble gun through Turkle's principle of identification through action.

Quake's three dimensional world is narrated through the eyes of the player character. *Puzzle Bobble* unglues epistemic access and manipulation from the player character, but *Quake*, like *Eliza*, firmly situates both epistemic access and manipulation in the player character. Partial representations combine to construct a player character with whom we identify through linear perspective, which suggests a body we are looking out of, and visibility of the player character's hand or weapon. Manipulations available to the player also suggest the presence of a player character. Body syntonicity in *Quake* is stronger than in *Super Mario Bros.* as a result of situating epistemic perspective in the player character. Linear perspective is independent of epistemic point of view: *Mario 64* uses linear perspective to narrate the world from the third person, and *Eliza's* world is narrated from the first person without the aid of three dimensional graphics and linear perspective.

The standard for three dimensional games had been locking narrative perspective into the first person, as in *Quake*, but *Mario 64* constructed a three dimensional world where epistemic point of view was externalized from the player character into a third person. This uncoupling was one of *Mario 64's* greatest innovations, and Miyamoto has commented that placing the camera in the third person was harder on programmers and designers, but easier on the player. The relationship between player character, epistemic access, and locus of manipulation in *Mario 64* is the same as *Super Mario Bros.*.

Tetris has no player character, an omniscient epistemic point of view, and locates manipulation in one of a sequence of falling blocks. The entire game board is visible and one can even peer one step into the sequence's future. Manipulation is located in one block just as player manipulation in *Super Mario Bros.* is located in Mario; pushing left and right causes Mario or the block to move left and right, which is a body syntonic operation. Block rotation is also consistent with our body knowledge. Player manipulation isn't located in a single block for too long, as after each block comes to rest, the locus of manipulation is moved to the next falling block. Located manipulation is embedded within sequential manipulation.

Point of view in *Go* shares many features with *Tetris*: an omniscient view of the game world, sequential manipulation, and lack of a game constructed player character. While players might identify with a *Tetris* block via identification through action, manipulation isn't situated in a single *Go* stone long enough for this to take place. *Go* players do, however, identify with their stones, territory, and power on the game board.

Epistemic point of view in *StarCraft* is similar to third person limited. Only one screen at a time of the game world is visible to the player, and fog of war limits epistemic access to what the units of a player's army can see. Identification through action encourages player identification with units, as does tying epistemic access to units. Linking manipulation and narrative access to game world characters encourages character identification, even if the character is fragmented as a collection of agents. Player units in *StarCraft* have some autonomy, and resist and assist player control. This negotiation of control is part of the pleasure of games such as *StarCraft* and *Lemmings*, which fragment the player character into collections of resistant and assistant semi-autonomous agents.

Miyamoto describes *Pikmin's* point of view, which combines a player character (similar to *Super Mario Bros.*) with control of a collection of semi-autonomous agents (similar to *Lemmings*):

> When you play this game, you control only Olimar, but you get the impression you're actually moving the Pikmin, don't you think? When you play *Super Mario Bros.*, you control just Mario, but this game will leave you feeling like you're actually Olimar and you're controlling the Pikmin. I don't know any other game that can achieve such a feeling [1].

Epistemic access in *SimAnt* and *SimCity* is omniscient. One screen of the world is visible at a time, but the entire world is always available. Although players in *SimAnt* identify with their entire ant colony, players do have a single player character, a yellow ant, in which manipulation is located. Players identify with both the yellow ant and the entire colony. *SimCity* also encourages identification with an entire city, an entity more diffuse than an individual or fragmented player character. Wright's observation that construction leads to empathy is congruent with Turkle's principle of identification through action. *Identification through construction* is evident in *The Sims*, *SimAnt*, *SimCity*, *Go*, *StarCraft*, and role playing games that call upon players to take part in character design. In these games, players guide the growth of an ant colony, city, or simulated person by placing stones, manufacturing units, or equipping and particularizing characters.

Games put the self in flux. What is perceived as *me* or *my body* is firmly located in a particular character (*Super Mario Bros.*), grows and multiplies (*Go*, *StarCraft*), is fragmented into a plurality of tokens or agents (*Go*, *StarCraft*), resists or amplifies player intent (*Lemmings*), undergoes amputation and dissolution (*Go*), transforms into alternate versions (*Super Mario Bros.*), migrates from body to body (*Avenging Spirit*), absorbs the form of others (*Kirby's Adventure*), or switches among multiple selves (*The Sims*).

CHAPTER VI

COMIC BOOK DOLLHOUSE

6.1 Goals

The goal of *Comic Book Dollhouse* is to create a magic crayon which encourages a practice of storyworld authorship and play. *Comic Book Dollhouse* is intended to be a complete, robust, and publicly available system that will be iterated based on user feedback. *CBD* isn't as computationally sophisticated as other interactive storyworld systems, such as *Façade*, but as an interactive storyworld construction toy, the system is unique, complete, robust, and usable. *CBD* is the simplest system that works. Design began from an analysis of relevant artifacts and a subsequent articulation of desired player experiences. Participatory conventions and pleasures from games are drawn upon, since games are experientially the richest and most sophisticated participatory media form. The visual language of comics is used because of its plasticity and intelligibility along the axes of narrative, authorship, and computation. Comics are representationally powerful, easy to write in (with the proper support), and lend themselves to procedural manipulation.

6.1.1 Encouraging a Storyworld Practice

What is meant by a practice of storyworld authorship and play? Users making, sharing, and playing storyworlds. Users should be able to engage storyworlds in a variety of ways, including reading stories made by other users, playing with authored storyworlds, modifying existing storyworlds, or creating storyworld materials for others. Participatory modes should be graded in sophistication and make use of overlapping interface conventions to facilitate mobility between participatory modes. The design intent is to lure someone who comes across a completed story on a web page to download the software and play with an existing storyworld, and also tempt players into becoming authors and making storyworlds. The interactive fiction community and the storytelling community that gravitates around *The Sims Exchange* are examples of the desired social structures: people specialize in reading/playing, tool development, and making particular types of world content. On *The Sims Exchange*, more sophisticated users help new ones, exchange of work is facilitated, player stories are foregrounded, and demands are made upon authors and tool makers [30]. *CBD's* simplicity is designed to encourage usership and feedback on avenues for elaboration. *Comic Book Dollhouse* is unlike interactive fiction software such as *Inform,* because *Inform* is not a magic crayon, and *CBD* storyworlds are visual rather than textual.

6.1.2 Experiential Goals

6.1.2.1 Immersion into a Miniature Dramatic World

CBD affords creation of miniature dramatic worlds. The primitives authors and players work with, such as actors, props, actions, and comics lend themselves to the domain of story.

Comic Book Dollhouse storyworlds will have an internal life of their own. A conscious decision was made to target the participatory pleasures of composition and construction first, and then add constraints and dynamics that add life after iterating the system based on feedback by real authors and users. Dynamics and constraints that create life will also add strategizing and problem solving to the experience of playing a dollhouse world. What kinds of dynamics and constraints do authors want to add to their miniature comic book dollhouse worlds?

6.1.2.2 Pleasures of Participation

CBD engages the participatory pleasures of composing and making things. Authors and players should be able to look at what others have made, export their stories to other forms, like web pages and printouts, and save and exchange storyworld materials. *CBD* offers primitives that authors can use to make a variety of things, and authors, in turn, can make primitives for other authors and players to assemble into storyworlds or stories. *The Sims Exchange* is a medium that storytellers use to share stories, and functions as a catalyst for the practice of using *The Sims* to tell stories [30]. I plan on making a web site where authors can share storyworlds, which will add a social impetus to the practice of storyworld construction.

Eventually, *CBD* will engage players, and not just authors, in strategizing and problem solving, and player decisions will evoke a stronger sense of responsibility. The constraints and dynamics that generate problem solving possible worlds landscapes will be the same mechanisms that imbue *CBD's* miniature worlds with more life and reactivity. Rather than implement dynamics and constraints that may or may not be useful for authors, I'll wait for authors to demand certain kinds of constraints and dynamics.

What is the player's point of view within a *CBD* storyworld? The player will not manipulate the world from the point of view of a player character, as there is a problem with having a character you manipulate in the first person also have its own internal simulation life, as Maxis discovered when transmogrifying *The Sims* into *The Sims Online*: "I'm having fun, why isn't my sim?" Also, the system gives players broad control over a dollhouse, which is inconsistent with linking the player manipulation to a specific character. Players will be temporally located at the end of a story. Although players can go back and retroactively edit any part of a story's visual layout, the player interface is structured around manipulation at a story's leading temporal edge.

6.2 Strategy

6.2.1 Representation & Comics

The Sims and *Lonely Time* suggest representational mechanisms for efficiently conveying dramatically compressed action on the computer. Both *Lonely Time* and *The Sims* use some very clever shorthands (abstractions) for explaining the life of their worlds. In *The Sims*, one cannot zoom in and see the expressions on character faces, so body language is how one assesses mood. Also, conversation, desires, and dreaming take place through evocative icons. Sims will produce mops and garbage bags out of nowhere, the fridge contains an infinite supply of foods that can be purchased at will, and all of this feels completely natural. Although *Lonely Time* is animated, it borrows conventions from manga. When the bathhouse is entered, it occupies the entire display, but superimposed in the corner is a small man in a circular frame thanking me for paying. Both displays are simultaneously animating disjunct moments in time, and yet the narrative compression and flow is intelligible. Also, the waitress in the restaurant doesn't walk between the cash register and counter, but jumps back and forth, which feels natural despite the combination of animation and cutting between separate moments in time on a background of continuous time. Representational shorthands and abstractions are used to make the systems' construction tractable, cognitively easier on the user and author, and to maintain dramatic interest.

CBD uses conventions from comics to represent the unfolding action of a storyworld. Video is combinatorially inflexible, computationally unwieldily, and very labor intensive to fabricate. Another problem with video is that it looks very realistic, and so false expectations about the domain and realism of potential storyworld events are established. The vast majority of video games use animated cartesian spaces for representing their worlds. Representing a cartesian space, however, is at odds with the needs of representing compressed dramatic action. Fabricating animated graphics, while not as labor intensive as video, is still a great burden to place on an author. Text, while exceptionally cheap to author, hardly takes advantage of the representational strengths of the computer.

Representation of time and space in *The Sims* is too literalistic, but *Lonely Time* points the way towards appropriating conventions from comics into an interactive system. *Lonely Time's* economy of representational affords the construction of a dramatically oriented world. Comics are a strong language for dramatic compression and abstraction, and have been used to cover every conceivable type of narrative ground.

While comics don't have any established conventions for participation, they are an exceptionally powerful and flexible representational medium. Comics make use of a light weight and highly malleable language for representing time, space, action, and emotion. Conventions such as the panel and motion lines are well understood. *The Sims*, for instance, taps into comic iconography. Comics are flexible along the axis of visual abstraction, and with the proper tools, writing in the visual language of comics is very easy.

Comics have a natural discreteness that lends itself well to computational recombination. Visual elements, such as frames, faces, speech balloons, and character bodies, can be parametric. Not only can the computer parametrically pull the strings that describe the appearance of such visual elements, but users can directly manipulate them. Layouts can also be parametric: rather than specify exactly which visual assets go where, layouts can be templates that afford substitution. Also, the computer can add animation to the visual language of comics.

Interactive and procedural generation of comics has been done in *Comic Chat* and *Comic Diary*. *Comic Chat* is chat room software where the conversation is displayed as a comic strip, and participants have parametric control of their characters' expressions and speech balloons [38]. *Comic*

Diary uses data collection and templates to generate comic strip diaries of users' conference experience [71]. Neither program is an authoring tool for making comics, which is the kind of interface *CBD* provides.

6.2.2 Authorship

HTML's widespread use resulted from features intrinsic to the language's design. HTML is robust, open, fluid, easy, incremental, and modular, features *CBD* aspires to.

CBD is a robust, stable system, and will be freely available for both Macintosh and Windows. The authorial description of *Comic Book Dollhouse* storyworlds are available for players to examine, and interface conventions are reused across player and author modes, which gives fluidity to the participatory roles. Storyworlds are represented to players and authors in relatively similar ways, and the cross-role consistency of representation and manipulation should encourage migration across roles. Authorship in *CBD* is incremental. One doesn't have to author a huge world to have something which works.

CBD storyworlds are modular to make authorship easy, and to encourage circulation of storyworld materials between authors and dollhouses. Storyworlds are modular as dollhouses (they contain everything else), objects (props and actors are self-contained units), and verbs (each is self-contained and lives within an object). This modularity will make it possible for users to exchange actors, props, and other storyworld materials. Modularity also means encapsulation, which makes authorship easier for authors. Each entity has clearly defined boundaries and an overview, which makes getting your head around each primitive straightforward. Each *CBD* object and verb is a neat, self-contained entity.

Modularity is key to the *The Sims* and the storytelling which takes place on *The Sims Exchange*. At the heart of the *The Sims's* architecture is an object oriented design which allows new objects to be incrementally added to a world. Sims know how to use and integrate new objects into their decision making and behavioral patterns because objects contain the properties, scripts, and animations that define their desirability, behavior, and representation. Stories on *The Sims Exchange* make heavy use of distributed authorship. Sophisticated storytellers use props and objects created by other users.

CBD, unlike other story generation or interactive storyworld software, emphasizes authorship and story construction as the primary mode of participation for both players and authors.

6.2.3 Participation and Authorship, then Generation

CBD is intentionally designed to solve the problem of storyworld participation and authorship, and leave the question of generation for later.My hypothesis is that the problem of generation will not be so hard once we have a grasp on the design problem of authorship and player participation. Making and playing storyworlds should encourage users to ask for specific generative features whose solutions might not be so hard. Current work in generative systems, such as *Universe*, *Tale Spin*, *Minstrel*, *Erasmatron*, and *Façade*, privilege generation, authorship, and participation differently. All these approaches take the stance that the solution to participation and/or authorship will follow logically from generation. None treat generation as a problem to be solved after participation and authorship is understood. It seems to be quite easy to build generative systems that are hard to author in or build satisfying participatory conventions for. *CBD* addresses the problems of participation and authorship first, with the hope that simple generative models will follow from the demands and usage patterns of players and authors.

6.2.4 Shallow Architecture, Deep Surface

Pencil and paper is a medium whose material architecture is simple and intelligible, but yields a deep surface that authors can put to a variety of uses. HTML, also, is a shallow architecture with a deep surface. HTML's simplicity makes it easy to understand, and its surface can be manipulated by authors to fabricate all kinds of things not anticipated by the language's designer.

CBD strives for computational simplicity to maximize intelligibility and plasticity for both authors and players. Simultaneously, its representational primitives are designed to maximize expressiveness. Panels, line drawings, actions, actors, and props are primitives that can be put to a wide variety of representational purposes, and appropriated for new uses. Computational complexity is deployed only when it results in a corresponding experiential payoff for authors and players.

6.3 Design

6.3.1 Overflight via Screenshots & Descriptions

Conveying a computer program through screen shots and narrative text is a poor excuse for playing with the real thing. Nonetheless, here are some screen shots and descriptions to communicate a taste of *CBD*.

The first screenshot is what the player sees after loading the storyworld "A Tale of Two Robots." The column on the left gives an overview of all the objects in this storyworld: stories, actors, props, and sticker sheets. A story object is open on the right. In this story object, which is a comics layout, players engage the objects of the storyworld to play/make a story:

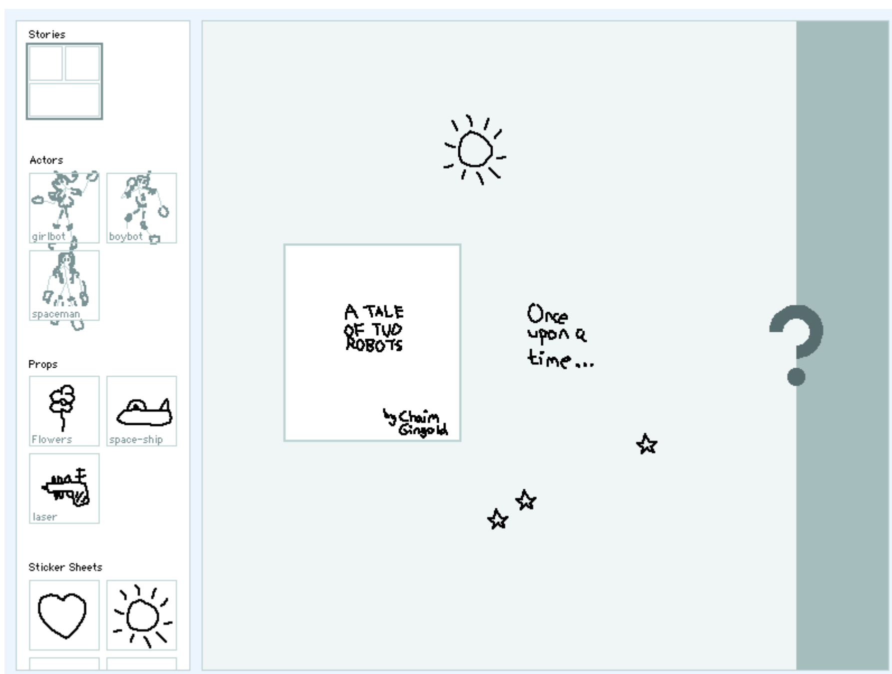

Figure 45: A story object in *Comic Book Dollhouse*.

Players drop objects into the story, and pop-up menus attached to on-stage objects prompt the player with possible sentences involving the on-stage objects:

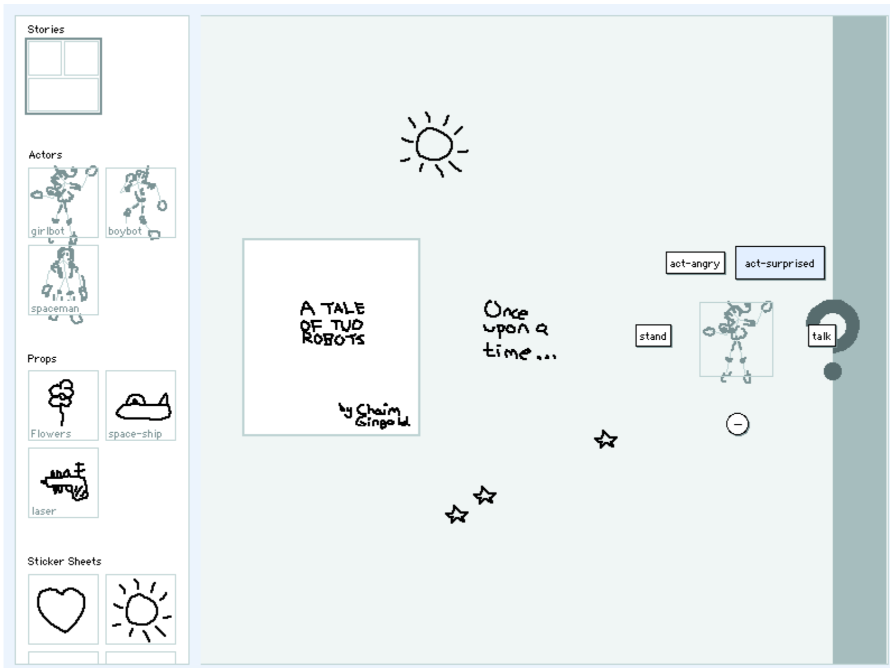

Figure 46: *Comic Book Dollhouse* verb pop-up menu.

The curtain previews options for what comes next in the story. These options are generated from verbs the player selects in the pop-up menu, or the plot generation mechanism:

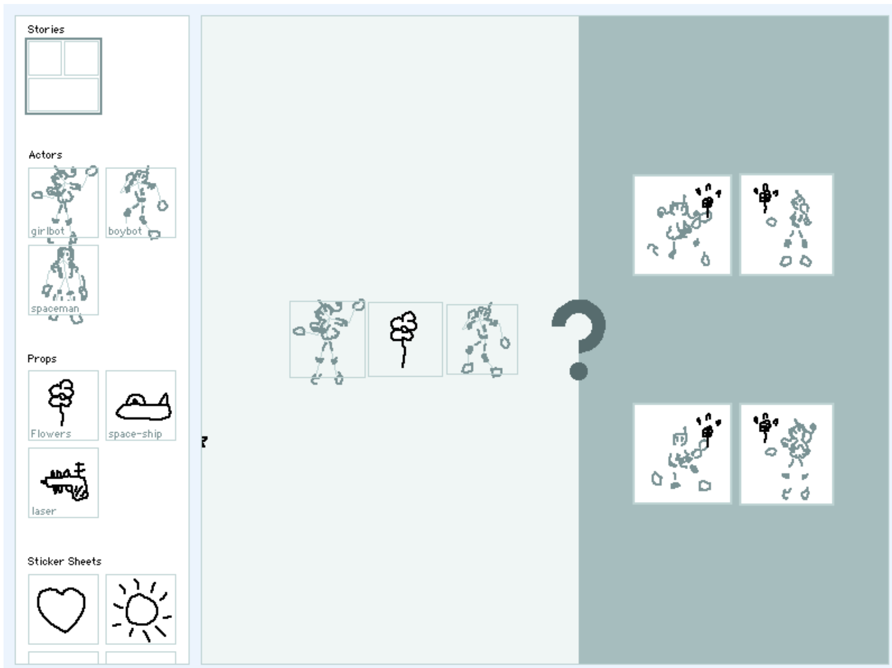

Figure 47: Question mark curtain previewing story possibilities in *Comic Book Dollhouse*.

Users can drop below the curtain, and edit the story's layout directly:

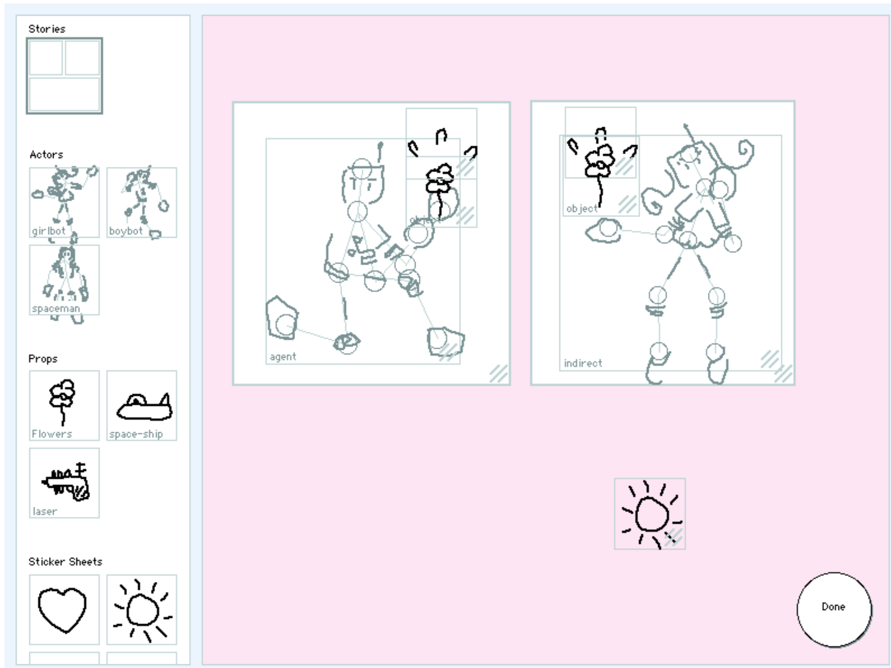

Figure 48: Editing a story's layout in *Comic Book Dollhouse*.

Editing the graphics for "boybot." The actor graphics component is based upon *Soude,* a program I wrote for sketching posable figures [27]. You can directly manipulate the pose of the figure, and draw upon it:

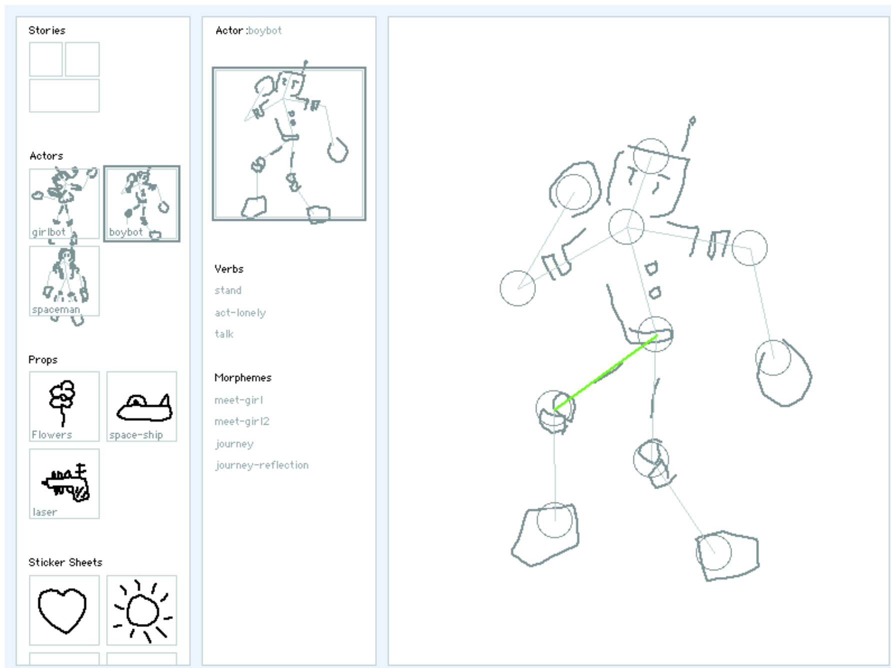

Figure 49: Editing an actor's graphics in *Comic Book Dollhouse*.

Editing the verb "hug," which belongs to the actor "girlbot:"

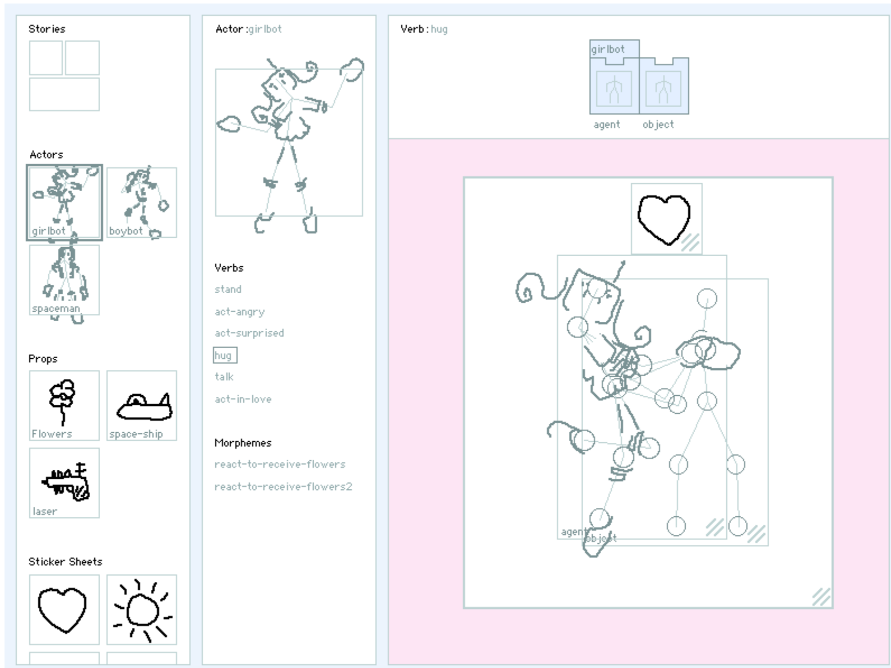

Figure 50: Editing a verb in *Comic Book Dollhouse*.

Editing the morpheme "meet-girl," which belongs to the actor "boybot." First, "boybot" acts lonely, he finds some flowers, and then gives them to "girlbot:"

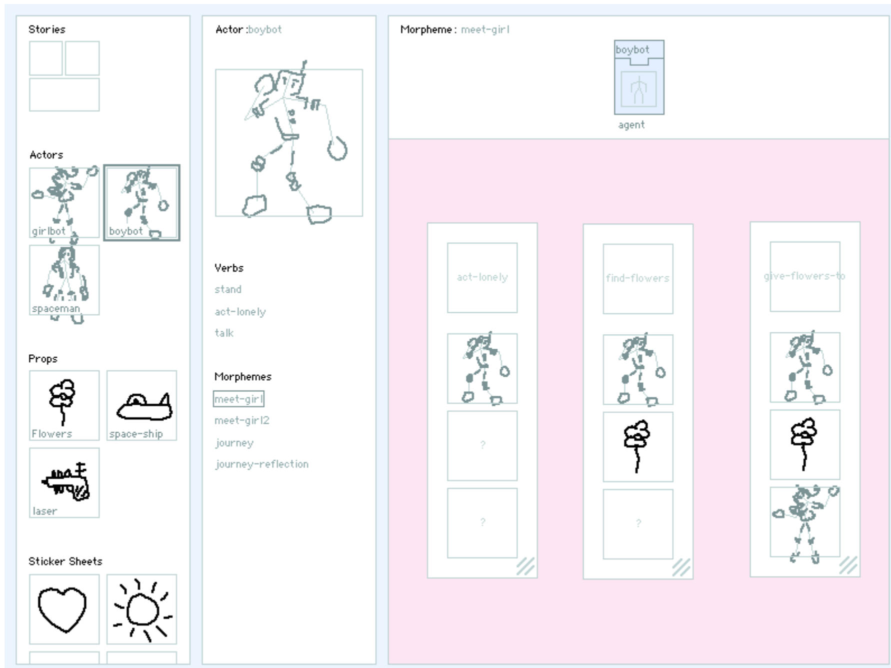

Figure 51: Editing a morpheme in *Comic Book Dollhouse*.

6.3.2 Storyworld Primitives

6.3.2.1 Objects

The basic architecture of *CBD* descends from *The Sims*. The world of *The Sims* contains objects which provide actions for sims to perform. The sink in *The Sims* provides the action "wash hands," which becomes available when the player clicks on the sink. The action "wash hands" is also advertised by the sink to sims, which is how sims make autonomous decisions. Sims can also be the providers of actions to other sims, such as "kiss." This object oriented architecture is intelligible and modular to players and authors.

CBD prop and actor objects provide verbs for actors to perform. Dollhouses also contain objects that don't provide verbs, like sticker sheets and stories. Sticker sheets are reusable collections of icons that users can make, use, and trade. Stories are where players interact with dollhouses, making stories from dollhouse components.

6.3.2.2 Verbs

Actors and props provide verbs for actors to perform. Verbs describe storyworld actions, and are the atomic units of activity. Each verb has one to three parameters. Verbs always have an agent, which is the actor performing the action. The object that contains a verb is always a parameter of the verb, which enforces the object oriented structure of *CBD*; verbs cannot float free without existing in an object. Parameters can also require certain types of objects. The structural constraints of a verb are designed to give intelligibility to a world's actions: all verbs are always performed by a actor on another object. A need for more than three parameters has not been encountered, and adding a totally variable number of parameters would add significant complexity without a corresponding experiential payoff. Each verb is a comics layout, and the parameters can be matched to elements in the layout, which is how layout templates interlock with substitutable verb parameters.

An example verb is "hug," which belongs to the actor object "girlbot." "Hug" has two parameters, one of which is always "girlbot," since she owns the verb. The other parameter is the actor whom she will hug. The verb describes the linkage between layout and parameters: which parameter should be drawn where?

Why verbs and not a spatial simulation, goal seeking artificial intelligence, or generative grammar? Verbs with parameters and comic layouts capture *CBD's* core representational needs: computational recombination through substitution, player participation, and compressed dramatic action. Structuring verbs as sentences whose parameters needs to be filled in accounts for player participation at the lowest level of storyworld description.

The descriptive language of verbs and layouts is intelligible and plastic to both authors and players. Many possible substitutions result from a small number of objects and verbs, yielding interesting player choices, and big payoffs for authorship. The structure of verbs is generic enough to provide great representational flexibility: a verb can capture a small action, like a actor looking up at the sky, or a big action, like the betrayal of one lover by another. A single verb could even represent someone joining a cult and living out the rest of their life alone in Montana.

The object-verb system successfully gives rise to a combinatoric explosion of object interactions and possible sentences, which makes composition and construction possible and interesting. The problem to solve, however, is making this possible worlds landscape topographically interesting. Dynamics that guide and push the player into these diversified possible worlds are needed, along with constraints that hold possible worlds into configurations that correspond to experiences such as responsibility, problem solving, and risk.

6.3.2.3 Plot Fragments

CBD uses a very simple plot generation mechanism to suggest possible worlds pathways to the player. Objects can contain morpheme declarations, which are layouts that describe linear sequences of actions, such as somebody finds a flower, and then somebody smells a flower. This declarative language describes story event sequences, and once the user interface is improved, writing in it should be very easy for authors. *CBD* is continually reading over the history of a story and looking for partially completed morphemes. If part of a morpheme's event sequence is found, the missing parts are suggested to the player. Morphemes can be recursively embedded within one another, have variables, and unspecified variable values. *CBD* unrolls unspecified variable values into possible parameters, which generates more bifurcation. If a morpheme suggests that one actor should give flowers to an unspecified actor, then *CBD* will suggest recipients. Future work in this area will include improving the interface, allowing more complex expressions, and prioritization of morphemes based on embedding relationships. The plot generator is run each time the story advances, which suggests zero or more sentences for the player to choose from. Only three sentences are previewed, so suggestions must be prioritized. Prioritization will be opportunistic: the system will favor sentences that fulfill more than one morpheme, and those that help construct overarching stories.

The plot mechanism was implemented to give dollhouses a little bit of internal, reactive life. The original design called for an event handling mechanism: verbs would generate events that event handlers could listen for and respond to. The advantage of declarative action sequences is that they can do everything an event response mechanism could have done, but without adding an entire event signature, generation, and recognition apparatus. Also, plot morphemes can respond to sequences of actions, not just single actions. Another advantage is representational: a sequence of verbs can be represented as a sequence of comic panels.

Plot morphemes do in fact give worlds a bit of internal life and surprise, but are still missing constraints and state that place the player into a situation where choices generate player responsibility. Choosing one action over another should have consequences, but all possible worlds remain available, which decreases player responsibility.The plot mechanism solves the problem of reactive life, but as a dynamic, it doesn't yield responsibility and problem solving. State modification and precondition syntax are needed to constrain the possible worlds landscape of a *CBD* dollhouse.

It is desirable for a single system of rules to be responsible for giving a world internal life, surprise, intelligibility, and player problem solving. More rules will also provide input constraints that make navigating a richly populated dollhouse feasible. While creative players can add constraints in their head to make new microworlds with these properties (I shot girlbot, so don't use her again), I want to computationally instantiate these constraints.

6.3.2.4 Visual Primitives

CBD provides visual primitives for making comics. Speech and thought balloons are primitive visual elements that can be directly manipulated into different shapes. A user can create speech a speech balloon, for example, and interactively drag on the spoke to stretch it out and point it towards an actor's mouth.

Actor figures behave like rag dolls, and can be easily posed through direct manipulation. The rag doll behavior of actors is achieved through an inverse kinematics system based upon springs. This system is a subset of *Soude,* a program I wrote for making hand drawn actors [27]. *Soude* allows users to create arbitrary skeletons, but *CBD* uses a fixed skeleton to enable visual substitution of actors.

Panels are another primitive *CBD* provides to facilitate making comics. *CBD* panels can be used to prompt mental closure. This is a form of hide-and-reveal that operates on the space of a page, rather than the represented space of a world.

Stickers are icons users can make and reuse in layouts. Examples of stickers are hearts, exclamation marks, question marks, motion lines, and sighs. The idea behind stickers is that there exists a vocabulary of icons users will want to author, reuse, and trade.

Visual primitives interact to make compositing easier. Legos snap together in certain ways to ease construction, and *CBD's* visual primitives adhere to one another in ways that afford constructing comics. Placing a visual primitive inside of another one causes *Comic Book Dollhouse* to structure them into a hierarchy. Dropping a sticker inside of a panel, for instance, causes the sticker to attach to the panel, so moving the panel will move the sticker as well. The spoke of a speech balloon directs its attachment: if a balloon's spoke lands inside of another element, such as a panel, the balloon will attach to it. *CBD* doesn't place logical constraints on hierarchy. Speech balloons can contain actors, additional speech balloons, panels, or anything else. An entire story could be written with everything contained in thought balloons.

6.3.3 Playing & Authoring

6.3.3.1 Time

Story time advances in one of three ways: the inner life of a storyworld suggests options for what should happen next, players choose a specific action, or players dig into a story's layout and manually arrange panels, make speech balloons, and pose actors.

Players often ask about arbitrary insert into a story's past. The ability to go back and change a story's past is probably suggested to users by *CBD's* interactive spatialization of time. There is no architectural reason why this type of play shouldn't be possible. Random insert as well as parallel, branching stories are avenues for further exploration. *Squares,* an interactive comics piece I made, allows intervention in a story's past to see how changes cascade into the future [28]:

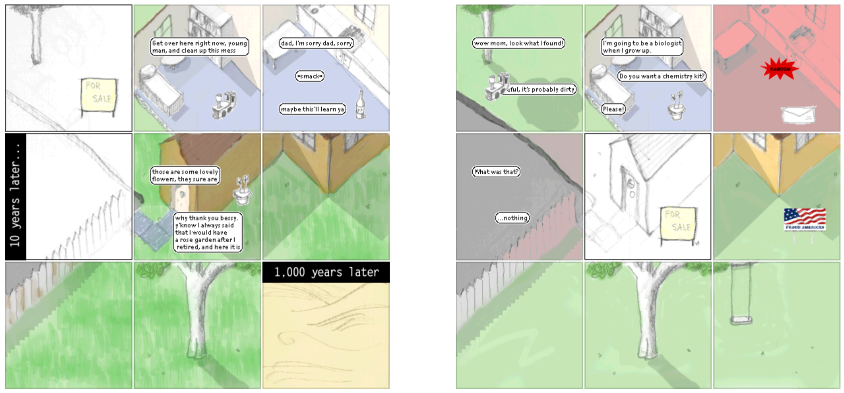

Figure 52: *Squares,* an interactive polyptych.

Architecturally, this will require capturing the state of a dollhouse in each time step of a story, which should be tractable with an object prototype architecture. An object instance in one time step will inherit properties from the global definition of that object. What we'll have is an architecture which can represent multiple, interdependent ontologies. Such a system could be stretched for more bizarre uses. A story could contain a dream sub-ontology which descends from a particular world state. Stories could branch into parallel paths and recurse into dreams and embedded narratives. These multiple ontologies could interpenetrate in interesting ways. A character could meet a version of himself from the past or an alternate instance of himself from a parallel world. Dream and actual worlds could collide, letting people enter each other's dreams, or disjunct dreams could leak into one another. Scott McCloud achieves similar effects in his multiform comic "Choose Your Own Carl" [47].

6.3.3.2 Question Curtain

The question mark curtain separates a story's past from its possible futures, and visually prompts the player to take action. The curtain also functions as a boundary that generates player safety and invites participation. The boundary divides a story's completed past from its inchoate future, so that the empty space to the right of a story doesn't make players feel uncomfortable. Having a totally blank piece of paper in front of you can be exciting and inspiring, but can also be overwhelming and intimidating. This boundary between the actual and possible is fluid, and the curtain moves around as players add to the story and preview possible futures. The curtain completely disappears when players edit a story's layout, which gives the sense of going behind the curtain. When players are done editing, the curtain descends again. The question mark curtain also functions as a seam between the microworlds of editing story layout, and playing a story.

Earlier in *CBD's* design, an interface style more closely related to *The Sims* was used, where players selected an agent to control, and then clicked on objects to see a menu of things the selected agent could do with the object. This approach, however, requires players to formulate ahead of time who will perform the action. *CBD* adopted an interface where players throw multiple objects into the story, and then click on objects to get a menu of verb sentences which result from the interaction of on stage objects. The chosen action is previewed in the story scratch space, and the user can choose to commit one of these previews. If a chosen action can be interpreted by the system in multiple ways (should the girl give the boy flowers, or should the boy give the girl flowers?), both possibilities are previewed in the story scratch space. The strength of this approach is that players don't have to know ahead of time what objects will be what parts of speech. Players simply choose what they want to work with, and compare possible sentences. The verb grammar is hidden from the player, but this makes learning how to author with the verb grammar harder.

If users don't place a sufficient number of objects into the story scratch space, then no sentences can be made. Users often place a prop into the story scratch space and are then surprised that no verbs are available, but this is because an actor is needed to do something to the prop. The solution is to have *Comic Book Dollhouse* automatically place previews of possible sentences into the story scratch space as soon as just one prop is added. Results from this search could be pruned against the sentences suggested by the plot generation mechanism.

6.3.3.3 Overviews

The dollhouse view provides an overview of the entire dollhouse as a collection of thumbnails. The intent is to give the user an overview of the world, and make the world feel miniature and safe. *CBD's* original design called for scenes, which were also overviews. A story would be located in a particular scene, and a scene would contain the objects in play. The purpose was to scope the objects in play for both the player and computer. Scenes, however, made the interface too complicated.

The dollhouse view isn't sufficient for dealing with a large number of objects. If a dollhouse contains more than a dozen objects, navigation will be hard. The solution to this problem is having a dollhouse reveal and hide its objects during play. Such dynamics will give us input constraints that make using a richly populated dollhouse feasible. A dollhouse might only start with one or two visible objects, but reveal additional objects as a story unfolds. Using a dollhouse could then become an exploratory activity in which players look for new props and characters.

6.3.3.4 Exchange

Building an exchange where authors and players can exchange storyworld materials is an important step in encouraging a practice of authorship and play to develop. One of the pleasures of building something is sharing it with others. Part of *CBD's* web site will be a virtual fridge door, like the Maxis Picks section of *The Sims Exchange*, where authors have their work foregrounded. Research into *The Sims Exchange* indicates that authors post things for others to read, and also enjoy helping others make things. Social dynamics such as recognition inspire making and playing [30].

An exchange will encourage authorship, as users will be able to download and leverage other people's storyworld objects into their own storyworld, leading, hopefully, to what Will Wright describes as a big bang of content. Will Wright has suggested transparent mechanisms for extracting actions and plot sequences from player activity, and cross-pollinating material across *CBD* storyworlds.

Users often ask about exporting completed stories and printing them. This follows naturally from the impulse to look at and share what one has made. *CBD* will need to support export of finished stories to web pages and printable files. It would be nice to privilege web page export, as this kind of sharing provides another entry point for users to see what can be done with *CBD*.

The biggest obstacle to the exchange of materials between storyworlds is a fixed taxonomy. *CBD's* architecture is ready for user defined abstract data types, but proliferating taxonomies will work against object portability. User defined data types are also key to specifying more interesting verb parameter constraints. One solution is to package the abstract data types of a dollhouse into a genre package that can be used across multiple dollhouses as a lingua franca. A genre package for a domestic situation comedy might define prop types and character stereotypes that would allow authors to exchange particular characters and household objects. Exposing abstract data types will require authors to agree on object taxonomies before exchanging storyworld materials.

6.3.3.5 *Authorial Plasticity*

Watching a creative user play with *CBD* is very rewarding, as users will reinterpret the basic materials provided by *CBD* for a variety of representational purposes. Space, time, drawings, and actors are highly plastic in the hands of a creative author, and the given materials of a storyworld prompt creation. Ideas are planted in a player's head by what is available, and if the imagined action isn't provided, players will often construct it on their own. Players are successfully called upon to participate in the imagination and construction of a world. The activity of making, arranging, and adding things is fun. [1]

An interesting problem is encouraging users to abstract and build reusable storyworld components. Users will often make new designs in the story layout editor without adding new verbs to objects. It is my hope that that when users become more proficient, they will add new verbs and objects to a dollhouse's objects for later reuse, rather than simply make things in the story for one time use.

Adding a "learn new verb" command might bridge this gap. Players could build something concrete in the story view, select part of a layout, and tell *CBD* they want to make it into a new verb. *CBD* could then ask the user which object the verb should be added to, and guide the user through the task of abstracting a layout with all parameters specified into one with substitutable parameters.

The lack of drawing tools in the layout editor forces people to make reusable stickers, which has good and bad side effects. The fact that players make reusable images is good, but drawing becomes a less improvisational activity. This could be solved, similarly, with a "learn new sticker" feature. Guiding players through the act of migrating things made in the story view into reusable verbs and stickers will make the transition between playing and authoring easier.

CBD doesn't intelligently layout panels appended to a story. While this is, in one sense, a problem, it is also an opportunity for player participation. Players will often choose verbs and then go in

[1] Informal user testing has been performed throughout *Comic Book Dollhouse's* development.

and fix any layout issues, a satisfying design activity. On the other hand, this does get tedious, and I believe that a dynamic layout system that allows for user control would make tinkering with the layout possible, but not always necessary. My *Paper* prototype demonstrates one approach to the problem of dynamic layout. A simulation is used to position panels and adhere to constraints such as reading order [26].

CBD would benefit from a richer graphics model. Allowing the user to draw anywhere will add a tremendous amount of authorial plasticity to the system. The design problem is figuring out how to segment user drawn graphics. Users will probably be expected to create canvas objects to draw on. Also, *Emotisketch* was originally going to be integrated into the system, but time was short [25]. At present, one character body skeleton is hard coded into all dollhouses. In the future, dollhouses will be able to define their own character body skeletons, and afford multiple skeletons within the same dollhouse. Having one skeleton makes substitution easy, but diversifying skeletons within a dollhouse requires more sophisticated substitution and mapping techniques.

Rather than specify the appearance of a visual asset in a top down manner, by telling a character to draw itself in a particular pose, the state of a character could inform how it is drawn. A character whose state specifies it is sad or wearing a hat would know to inflect its appearance with these emotional and visual modifiers. This isn't a hard problem, and games do it all the time. The difficult part will be integrating bottom up constraints specified by a character's state with top down constraints specified by the user or layout template.

6.3.4 Interface

6.3.4.1 Navigating Multiple Editors

CBD is organized into multiple views. On the left is the dollhouse view, which provides an overview of everything in the dollhouse. Double clicking on an object in the dollhouse opens it in a new view to the right of the dollhouse. Keeping everything in one window makes navigation easier on the user, but sometimes users have problems getting back to the story they were editing. For example, users will edit a story, decide they need to make a new sticker, make the sticker, and then be unsure of how to get back to their story. One possible solution is to add a "back" or "done" button that takes users back to the story view. A "done" button was added to the story layout editor mode to make switching back to the story play mode easy. *HyperCard's* recent window shows a collection of thumbnails of the last several dozen cards visited by the user, and clicking on a thumbnail returned the user to that location. Another possible solution is tinting the last object edited with red in the dollhouse overview, so users can easily identify where they were. Multiple saturation levels could be used to indicate the last few objects visited, and in what order.

Another problem is visually connecting a view, such as an actor, with its parent view, the dollhouse overview. When an actor is opened in the dollhouse, a new view appears to the right, but the connection isn't always clear. The Macintosh *Finder* solves this problem with a zooming rectangle animation that connects a folder window to its icon. I couldn't get zooming rectangles to look nice, but it seems some sort of animation would help. Repeating visual elements across views, such as labels, colors, or thumbnails helps to visually connect related views.

Lateral navigation elements would help move the user across different editors and modes. The "done" button that hovers over the story layout editor is an example of lateral navigation. Users can use a menu to turn off the layout edit mode, but users generally want to return to the story view after editing the layout, so a button appears at the right moment to take users immediately back. Other buttons could hover at appropriate times to prompt users for the next logical action or mode change. More user observation will be required to streamline navigating multiple editors.

6.3.4.2 Selection

CBD integrates multiple editors together into the same window. It is general interface design practice to give each pane of a multi-paned window its own selection, which means that only one pane has the user focus. Focus modes, which is understood to be bad design practice, is used to resolve user actions such as copy and paste against a particular pane's selection. *CBD* solves this problem not by introducing additional modes, but by allowing only one active selection across multiple editors. If something is selected in the verb editor pane, and the user clicks on an object in the dollhouse view, only the object in the dollhouse view is selected. The experience is of working with a single editor that supports editing many different types of things, rather than multiple editors placed into a moded window.

6.3.4.3 Sculpt vs. Specify

If a user wants to make a speech balloon, should they draw the outline of what they want, or get a speech balloon and shape it into what they want? *CBD* follows *HyperCard's* approach. Rather than ask users to describe an object before creating it, *CBD* creates an object for users to sculpt. Users ask for a new balloon from the menu, and a new balloon is created in the center of the layout, which can then be shaped. Sculpting is better than specification, because specification adds a redundant interface for describing objects, as well as an additional mode. Users would have to choose the speech balloon drawing tool, or specify the balloon's dimensions through some other mechanism. Also, sculpting lets users see what they are getting as they describe it. Prompting users with default content is a strategy *CBD* tries to follow elsewhere. New dollhouses always contain an actor, prop, and story; new actors always contain one verb; verbs always contain default layouts that users can manipulate.

Starting from zero is harder than sculpting something given, and *CBD* should extend this idea. New dollhouses should automatically inherit more default content from a template. Giving authors partially completed material that demands completion is a design strategy for encouraging authorship. The character body editor in *CBD* presents the user with a humanoid stick figure, and users rarely have trouble understanding what it is and building on it.

6.3.4.4 Fluid Authorship

CBD leverages code and interface conventions between authorial and player modes to encourage mobility between playing and authoring. The same layout editor is used in the story view and verb editor. The verb editor can preview a verb as it will appear in the story. Authors can also preview what the verb will look like with particular objects plugged into the parameters.

CBD needs to follow *HyperCard's* lead in constructing multiple user levels that hide parts of the interface that aren't relevant to a user's proficiency. *CBD* doesn't make its use clear to new users. It should always open into a storyworld, and prompt players to participate. Graded participatory levels would focus the interface, and not distract less experienced users with options. It is better to pleasantly surprise the user with more authorial possibility than distract with irrelevant widgets and options.

6.3.4.5 Drag and Drop

Much of *CBD's* interface was designed around drag and drop. Direct manipulation is great, but if everything can be dragged to something else to issue commands, then the possible things you can do with a system becomes opaque. Making everything drag and drop seems to approximate the experience of being at a command line, except one uses direct manipulation rather than typing free text to make sentences.

6.3.4.6 Undo

CBD affords infinite undo, which encourages users to experiment. This raises a problem with integrating multiple editors into a single window. One undo history exists for the entire dollhouse, which isn't natural from the user's point of view. Each document in a word processor, for instance, has its own undo history. Undoing in a word processor affects the frontmost window. Undoing in *CBD* will simply undo the last thing the user did, even if it isn't the object being edited. *CBD's* undo history needs to be scoped, but what should the units of segmentation be?

6.3.4.7 Direct Manipulation Scrolling

Scrollbars are banished from *CBD*. Originally this was done to ease porting from Macintosh to Windows, but Scott McCloud argues that direct manipulation is preferable. He observes that scrollbars were preferable to direct manipulation only because our computers couldn't keep up with real time redrawing of smoothly scrolling content. Now they can, but a new issue is raised: with a scrollbar, one can keep scrolling in one direction by holding the button down on a scrollbar arrow; direct manipulation requires a back and forth motion to keep grabbing and moving the data. McCloud's direct manipulation idea includes physics, which solves the back and forth problem. Story scrolling in *CBD* has physics, and the content can be grabbed and thrown around.

6.4 Future Work

The next phase of *Comic Book Dollhouse's* development is encouraging people to use it. If a small group of people begin using it to make and play storyworlds, then *Comic Book Dollhouse* will be considered a success. I am also eager to see if people find alternate uses for *Comic Book Dollhouse's* primitives.

Future work will include development of an exchange web site where authors and players can upload and download storyworlds. Once this exists, and it contains a handful of storyworlds, it will be time to invite more people to download *Comic Book Dollhouse*, play these storyworlds, and make their own. Feedback from users will direct *Comic Book Dollhouse's* development.

Ultimately, I want players of storyworlds to feel they are taking risks, making strategic choices, and solving problems while engaging miniature dramatic worlds, and authors should feel they have a magic crayon for making these worlds.

REFERENCES

[1] "Pikmin Promotional Brochure," 2001.

[2] "Nintendo's Shigeru Miyamoto and Eiji Aonuma Interview at D.I.C.E. Summit 2003," *GameCubicle.com*, 2003.

[3] Aarseth, E. J., *Cybertext: Perspectives on Ergodic Literature.* Baltimore, Maryland: John Hopkins University, 1997.

[4] Alexander, C., Ishikawa, S., and Silverstein, M., *A Pattern Language: Towns, Buildings, Construction.* Oxford University, 1977.

[5] Atari, *Missile Command.* 1980. Arcade game.

[6] Atkinson, B., *HyperCard.* Apple, 1987. Macintosh software.

[7] Austin, A., *Bridge Builder.* 2000. PC game.

[8] Blizzard, *StarCraft.* 1990. PC game.

[9] Borges, J. L., *Labyrinths: Selected Stories and Other Writings.* W.W. Norton & Company, Feb. 1988.

[10] Braitenberg, V., *Vehicles: Experiments in Synthetic Psychology.* MIT, 1996.

[11] Caillois, R., *Man, Play and Games.* Urbana, Illinois: University of Illinois, 2001.

[12] Capcom, *Street Fighter II.* 1991. Arcade game.

[13] Church, D., "Formal Abstract Design Tools," *Gamasutra*, July 1999.

[14] Colton, S., "Exclusive Miyamoto Interview!," *Gamer.tv*, Feb. 2003.

[15] Crawford, C., *Erasmatron.* Macintosh software.

[16] Crawford, C., *Understanding Interactivity.*

[17] Cyan, *The Manhole.* 1987. Macintosh game.

[18] Cyan, *Cosmic Osmo and the Worlds Beyond the Mackerel.* 1989. Macintosh game.

[19] Cyan, *MYST.* Brøderbund, Sept. 1993. Macintosh game.

[20] DMA Design, *Lemmings.* Psygnosis, 1990. PC game.

[21] DMA Design, *Grand Theft Auto III.* Rockstar Games, Oct. 2001. PlayStation 2 game.

[22] Doležel, L., *Heterocosmica: Fiction and Possible Worlds.* Baltimore, Maryland: John Hopkins University, 1998.

[23] Domike, S., Mateas, M., and Vanouse, P., "The Recombinant History Apparatus Presents: Terminal Time," 2002.

[24] Eskelinen, M., "The Gaming Situation," *Game Studies*, vol. 1, July 2001.

[25] Gingold, C., *Emotisketch.* 2002. Software prototype.

[26] Gingold, C., *Paper*. 2002. Software prototype.

[27] Gingold, C., *Soude*. 2002. Software prototype.

[28] Gingold, C., *Squares*. 2002. Software prototype.

[29] Gingold, C., *Comic Book Dollhouse*. 2003. Software application.

[30] Gingold, C. and Lichti, N., "The Storytelling Community of The Sims Exchange." Unpublished manuscript, 2002.

[31] Goodman, D., *The Complete HyperCard Handbook*. Toronto: Bantam, Sept. 1987.

[32] HAL Laboratory, Inc., *Kirby's Adventure*. Nintendo, 1993. NES game.

[33] Higuchi, T., *The Visual and Spatial Structure of Landscapes*. Cambridge, Massachusetts: MIT, 1983.

[34] id Software, *Quake*. 1996. PC software.

[35] Jaleco, *Avenging Spirit*. 1991. Arcade game.

[36] Kent, S. L., *The Ultimate History of Video Games*. Roseville, CA: Prima, 2001.

[37] Knoll, T., *Photoshop*. Adobe, Feb. 1990. Macintosh software.

[38] Kurlander, D. and Skelly, T., "Comic Chat," SIGGRAPH, 1996.

[39] Loftus, G. R. and Loftus, E. F., *Mind At Play: The Psychology of Video Games*. New York: Basic Books, 1983.

[40] Lynch, K., *The Image of the City*. Cambridge, Massachusetts: MIT, 1960.

[41] Mateas, M., *Interactive Drama, Art and Artificial Intelligence*. PhD thesis, Carnegie Mellon University, Dec. 2002.

[42] Maxis, *SimCity*. Brøderbund, 1989.

[43] Maxis, *SimEarth: The Living Planet*. 1990.

[44] Maxis, *SimAnt: The Electronic Ant Colony*. 1991.

[45] Maxis, *The Sims*. Electronic Arts, 2000. PC game.

[46] Maxis, *The Sims Online*. Electronic Arts, Dec. 2002. PC game.

[47] McCloud, S., *Understanding Comics*. New York: HarperPerennial, 1994.

[48] McCloud, S., "Choose Your Own Carl," 2001.

[49] McHale, B., *Postmodernist Fiction*. London: Routledge, 1996.

[50] Meretzky, S., *Planetfall*. Infocom, 1983. PC software.

[51] Murray, J. H., *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York: The Free Press, 1997.

[52] Namco, *Pac-Man*. 1981. Arcade game.

[53] Nintendo, *Donkey Kong*. 1981. Arcade game.

[54] Nintendo, *Super Mario Bros*. Sept. 1985. NES game.

[55] Nintendo, *The Legend Of Zelda*. 1987. NES game.

[56] Nintendo, *Super Mario Bros. 3.* Feb. 1990. NES game.

[57] Nintendo, *Super Mario 64.* June 1996. Nintendo 64 game.

[58] Nintendo, *Animal Crossing.* Sept. 2002. GameCube game.

[59] Nintendo, *Super Mario Sunshine.* July 2002. GameCube game.

[60] Pajitnov, A., *Tetris.* 1985. PC game.

[61] Papert, S., *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, 1980.

[62] Pearce, C., "Sims, BattleBots, Cellular Automata, God and *Go*: A Conversation with Will Wright," *Game Studies*, vol. 2, July 2002.

[63] Perlin, K., *Emotive Actors.* Java applet.

[64] Perlin, K., *Facial Nuance.* Java applet.

[65] Perlin, K., "Implementing Procedural Personality Effects Over a 3D Model," Game Developers Conference, 2002.

[66] Ramis, H., *Groundhog Day.* 1993.

[67] Rouse III, R., *Game Design: Theory & Practice.* Plano, Texas: Wordware, 2001.

[68] Ryan, M.-L., *Possible Worlds, Artificial Intelligence, and Narrative Theory.* Bloomington, Indiana: Indiana University, 1991.

[69] Ryan, M.-L., "Beyond Myth and Metaphor: The Case of Narrative in Digital Media," *Game Studies*, vol. 1, July 2001.

[70] Sakai, T., *Lonely Time.* 2001. Macintosh game.

[71] Sakamoto, R., Nakao, K., Sumi, Y., and Mase, K., "Representing Individual Experiences through Comic," The 15th Annual Conference of Japanese Society for Artificial Intelligence, 2001.

[72] Sheff, D., *Game Over: Press Start to Continue.* Wilton, CT: GamePress, 1999.

[73] Sieberg, D., "The World According to Will," *Salon*, Feb. 2000.

[74] Slawson, D. A., *Secret Teachings in the Art of Japanese Gardens.* Tokyo: Kodansha, 1987.

[75] SquareSoft, *Final Fantasy VII.* Sony Computer Entertainment America, 1997. PlayStation game.

[76] SquareSoft, *Final Fantasy VIII.* Square/Electronic Arts, 1999. PlayStation game.

[77] Taito, *Puzzle Bobble.* 1994. Arcade game.

[78] Treasure, *Ikaruga.* 2001. Arcade game.

[79] Turkle, S., *The Second Self: Computers and the Human Spirit.* New York: Touchstone, 1985.

[80] Turner, S. T., *The Creative Process: A Computer Model of Storytelling and Creativity.* Lawrence Erlbaum Assoc., Aug. 1994.

[81] Uspensky, B., *A Poetics of Composition: The Structure of the Artistic Text and Typology of a Compositional Form.* Berkeley, CA: University of California, 1973.

[82] WEIZENBAUM, J., "ELIZA: A Computer Program For the Study of Natural Language Communication Between Man and Machine," *Communications of the ACM [Association for Computing Machinery]*, vol. 9, pp. 36–45, Jan. 1966.

[83] ZHAO, Q., *Penjing: Worlds of Wonderment*. Athens, GA: Venus Communications, 1998.