Play as Research: The Iterative Design Process

Eric Zimmerman

Final Draft: July 8, 2003

Needs and Pleasures

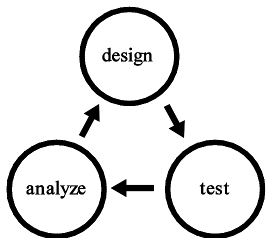
Design is a way to ask questions. Design *research*, when it occurs through the practice of design itself, is a way to ask larger questions beyond the limited scope of a particular design problem. When design research is integrated into the design process, new and unexpected questions emerge directly from the act of design. This chapter outlines one such research design methodology -- the iterative design process -- using three recent game projects with which I have been involved (SiSSYFiGHT 2000, LOOP, and LEGO Junkbot).

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

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

Iteration Iteration

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



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

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

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

Case Study 1: SiSSYFiGHT 2000

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

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

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

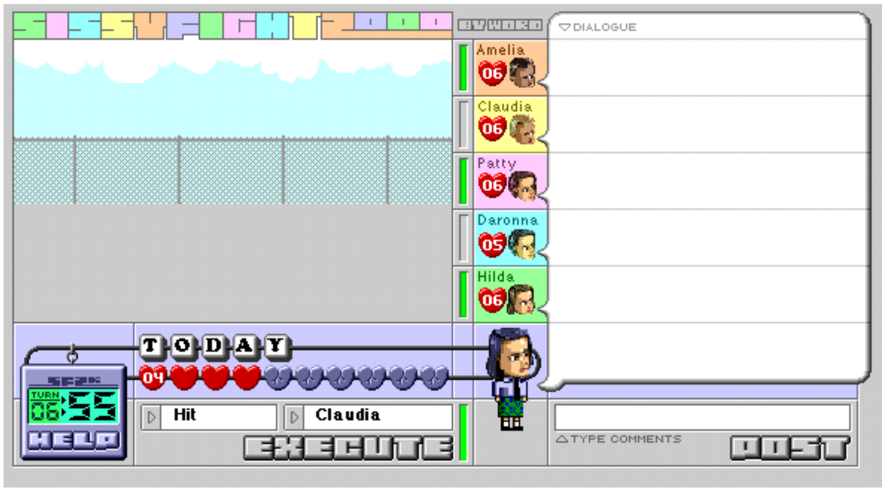
Once these basic formal and conceptual questions had begun to be mapped out, the shape of the initial prototype became clear. The very first version of SiSSYFiGHT was played with post-it-notes around a conference table. I designed a handful of basic actions each player could take, and acting as the program,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

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

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

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

While we tested gameplay via the text-only iteration, programming for the final version began in Director, and the core game logic we had developed for the IRC prototype was recycled into the Director code with little alteration. Parallel to the game design, the project's visual designers had begun to develop the graphic language of the game and chart out possible screen layouts. These early drafts of the visuals (revised many times over the course of the entire development) were dropped into the Director version of the game, and the first rough-hewn iteration of SiSSYFiGHT as a multiplayer online game took shape, inspired by Henry Darger's outsider art and retro game graphics.

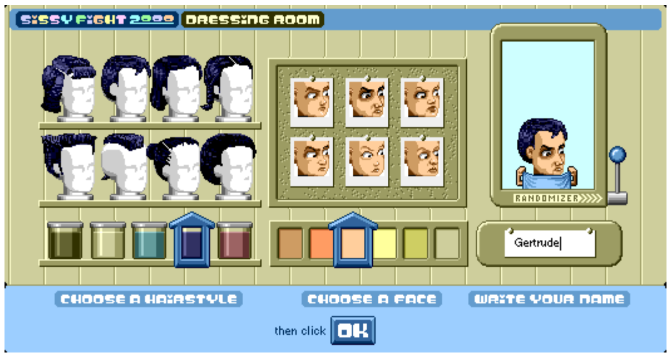


{zimmerman.chapter.02}

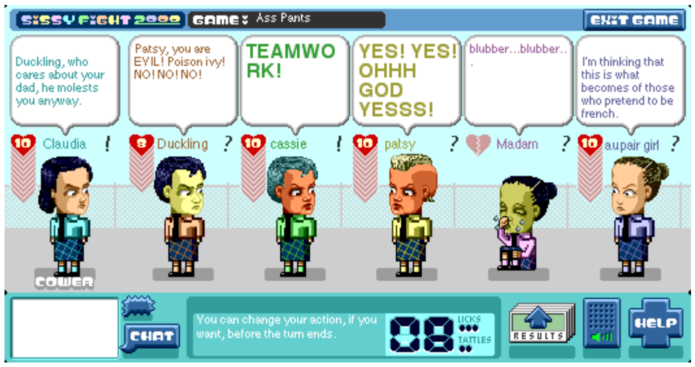
As soon as the web version was playable, the development team played it. And as our ugly duckling grew more refined, the rest of the Word.com staff were roped into testing as well. As the game grew more stable, we descended on our friends' dot-com companies after the workday had ended, sitting them down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

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

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



{zimmerman.chapter.03}



Duckling, who cares about your dad, he molests you anyway.

Patsy, you are EVIL! Poison ivy! NO! NO! NO!

TEAMWORK!

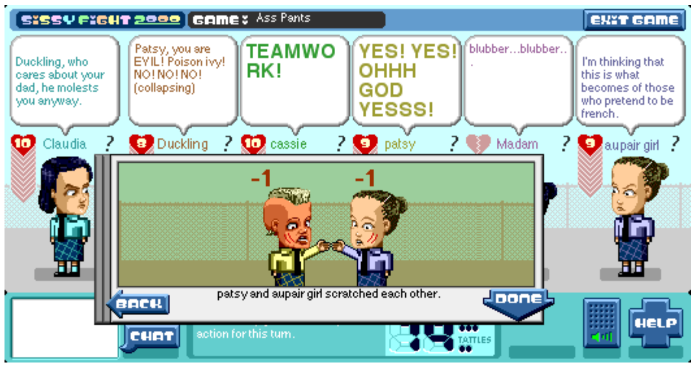
YES! YES! OHHH GOD YESSS!

blubber...blubber...

I'm thinking that this is what becomes of those who pretend to be french.

You can change your action, if you want, before the turn ends.

{zimmerman.chapter.04}



Duckling, who cares about your dad, he molests you anyway.

Patsy, you are EVIL! Poison ivy! NO! NO! NO! (collapsing)

TEAMWORK!

YES! YES! OHHH GOD YESSS!

blubber...blubber...

I'm thinking that this is what becomes of those who pretend to be french.

patsy and aupair girl scratched each other.

action for this turn.

{zimmerman.chapter.05}

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

Case Study 2: LOOP

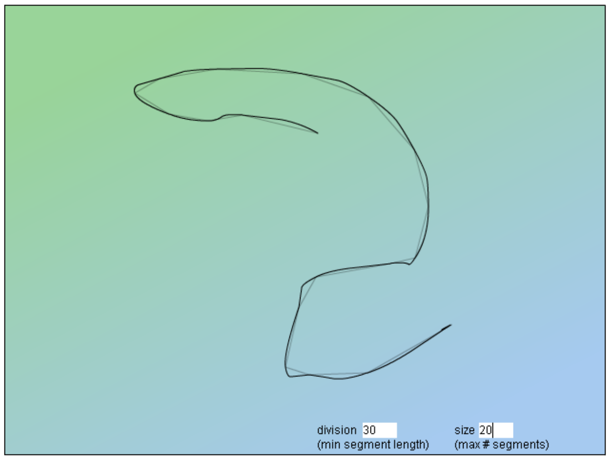
Summary: *LOOP is a singleplayer game in which the player uses the mouse to catch flittering, colored butterflies. The player draws loops around groups of* *butterflies of the same color, or of groups in which each butterfly is a different color (the more butterflies in a loop, the more points). To finish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

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

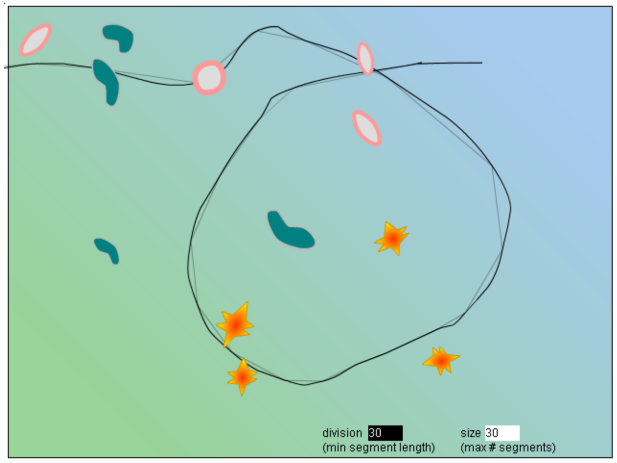
Another way of framing this problem is to ask, What is the *activity* of the game? Rather than asking what the game *is about*, ask what the player is *actually doing* from moment to moment as they play. Virtually all games have a **core mechanic**, an action or set of actions that players will repeat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

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

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


division 30
(min segment length)
size 20
(max # segments)

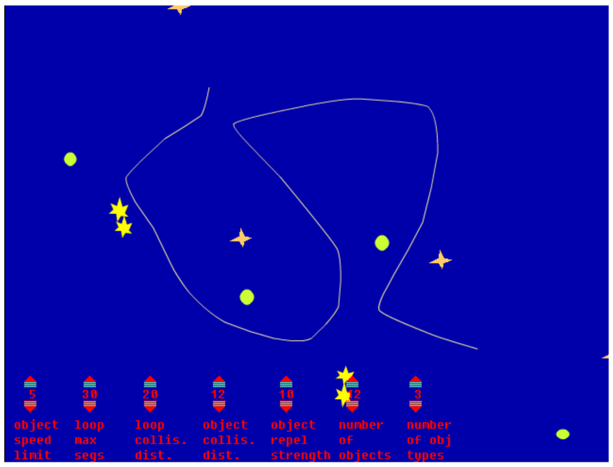
{zimmerman.chapter.06}



division 30
(min segment length)

size 30
(max # segments)

{zimmerman.chapter.07}



5
object
speed
limit

30
loop
max
segs

20
loop
collis.
dist.

12
object
collis.
dist.

10
object
repel
strength

12
number
of
objects

3
number
of obj
types

{zimmerman.chapter.08}

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

```
-- LOOP SCORES
score_same=0,5,10,20,40,80,150,250,350,500,700,1000,1400,1900,250
0,3100,380
0,4600,5500,7000
score_different=0,0,30,75,200,500
score_badloop=-20

-- # of caught butterflies for each level of loop sound effect
loop_sound_num=1,4,6,8,10
```

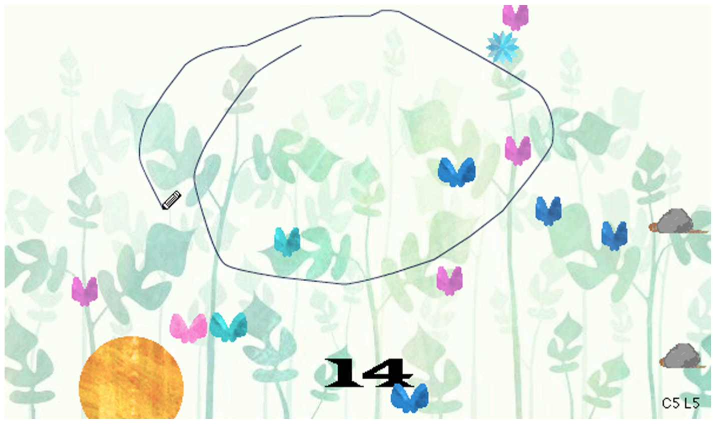
```
-- BONUSES
-- butterfly-borne bonus (x2):
bonus_lifetime=60
-- leaf-blown bonus (longer, moretime, freeze, flock):
freebonus_speedlimit=15
bonus_freeze_duration=4
bonus_flock_duration=12

-- HAZARDS
snail_speedlimit=1.2
killerbee_speedlimit=12,
killerbee_attackrate=3,killerbee_stingduration=6
beetle_speedlimit=3, beetle_fighttime=4, beetle_aborttime=10,
beetle_effectradius=300
stinkbug_speedlimit=2, stinkbug_tag_radius=40,
stinkbug_effect_duration=10, stinkbug_effect_radius=300
spider_speedlimit=9,
spider_climblimit=22,spider_stingduration=6,spider_loop_length=5
```

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

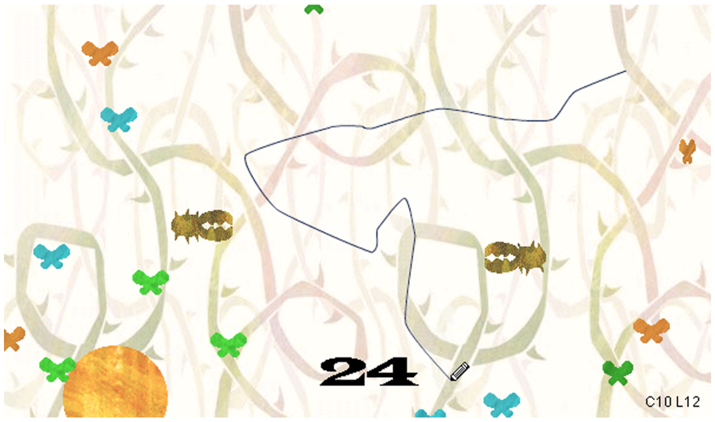
As the game code solidified, the many adjustable parameters of the game were placed in a text file that was read into the application when it ran. These parameters controlled everything from the behavior of game creatures to points scored for different numbers of butterflies in a loop to the progression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

LOOP
"My pleasures are the most intense known to man: writing and butterfly hunting"
- Vladimir Nabokov
Loop me to Play

{zimmerman.chapter.09}



{zimmerman.chapter.10}



{zimmerman.chapter.11}

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

Case Study 3: LEGO Junkbot

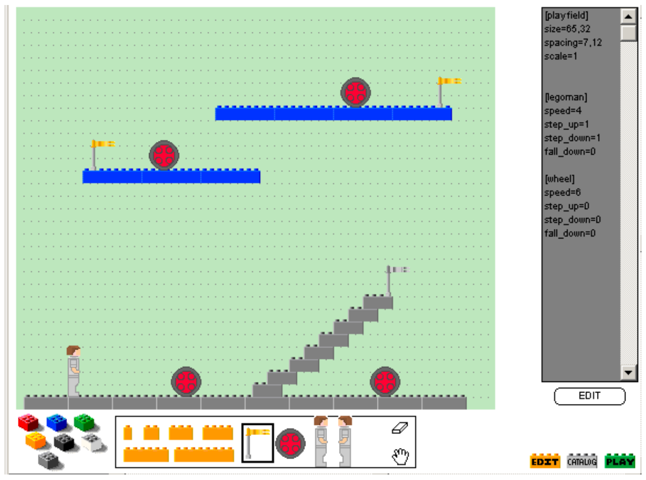
Summary: *LEGO Junkbot is a singleplayer game in which the player helps the robot character Junkbot empty trash cans throughout a factory. The player doesn't control Junkbot directly but instead uses the mouse to move LEGO bricks around the screen, deconstructing and reconstructing his environment brick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

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

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

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

{zimmerman.chapter.12}



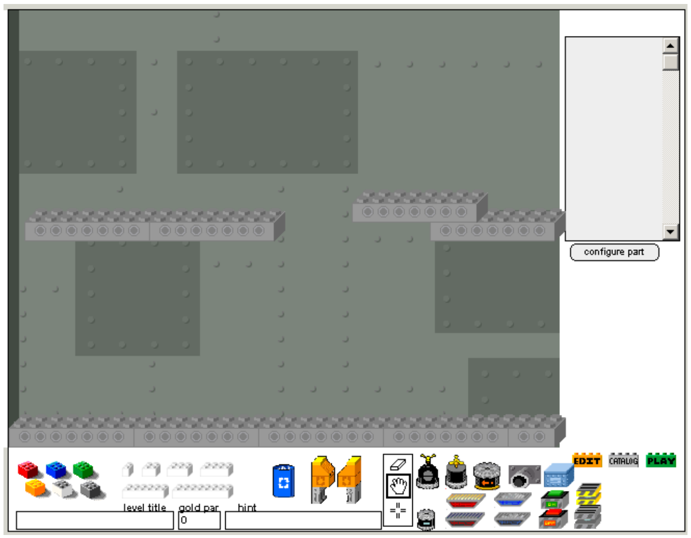
```
[playfield]
size=65,32
spacing=7,12
scale=1


[legoman]
speed=4
step_up=1
step_down=1
fall_down=0

[wheel]
speed=6
step_up=0
step_down=0
fall_down=0
```

{zimmerman.chapter.13}

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

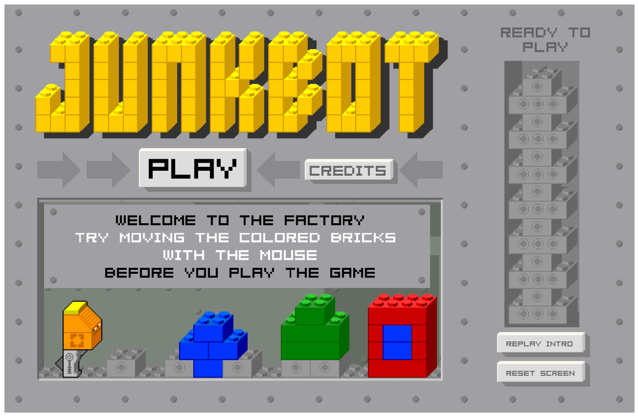
Each new element addressed something that was lacking in the experience of the previous prototype: it was monotonous to move bricks one by one, so we implemented code that let players stack bricks and move them as a group. We needed a way to move the main character vertically on the screen, so we added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

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



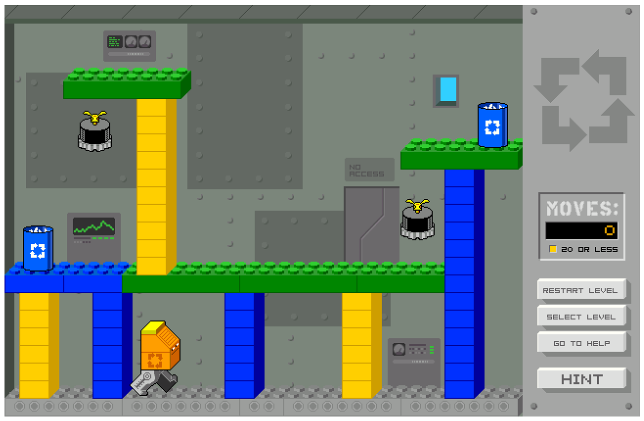
{zimmerman.chapter.14}

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

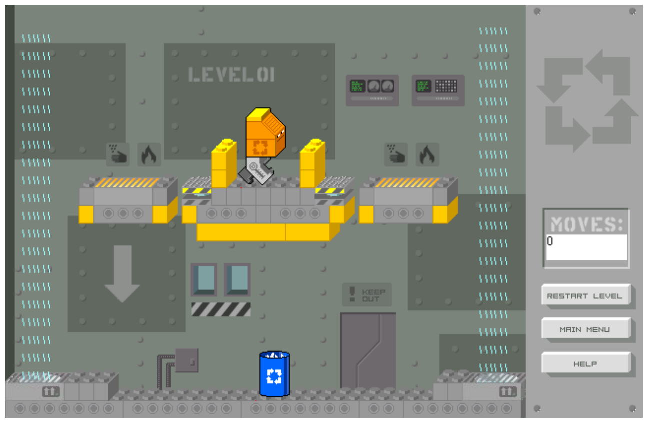


WELCOME TO THE FACTORY
TRY MOVING THE COLORED BRICKS
WITH THE MOUSE
BEFORE YOU PLAY THE GAME

{zimmerman.chapter.15}



{zimmerman.chapter.16}



{zimmerman.chapter.17}

A good rule of thumb for iterative testing is to err on the side of observation rather than guidance. While it may be difficult to keep your hands off the tester's mouse, instead sit back and see what your audience actually does, rather than telling them how it is supposed to work. What you observe can sometimes be painful to watch, but it will help you design more successful play. Part of iterative design is simply learning how to listen.

Conclusions

Iterative design is a process-based design methodology, but it is also a form of design research. In each of these three case studies, new questions emerged out of the very process of design, questions that were not part of the initial investigation but were nevertheless addressed through iterative play and design.

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

The principles of the iterative process are clearly applicable beyond the limited domain of games. *Rules* and *play* are just game design terms for *structure* and *experience*: a designer creates some kind of *structure* (a typeface, a building, a car), and a reader, visitor, or car passenger *experiences* it: encountering, exploring, dwelling in, and manipulating the system -- using it, playing with it, delighting in it. Games provide particularly clear examples of iterative design, but any design field can benefit from such an approach.

In iterative design, there is a blending of designer and user, of creator and player. It is a process of design through the reinvention of play. Through iterative design, designers create systems and play with them. They become participants, but do so in order to critique their creations, to bend them, break them, and re-fashion them into something new. And in these procedures of investigation and experimentation, a special form of research takes place. The process of iteration, of design through play, is a way of discovering the answers to questions you didn't even know were there. And that makes it a powerful and important form of design research.

Project Teams

SiSSYFiGHT 2000
Word.com, 2000
Marisa Bowe, Ranjit Bhatnagar, Tomas Clarke, Michelle Golden, Lucas Gonze, Lem Jay Ignacio, Jason Mohr, Daron Murphy, Yoshi Sodeka, Wade Tinney, Eric Zimmerman

LOOP
gameLab, 2001, published by Shockwave.com
Ranjit Bhatnagar, Peter Lee, Frank Lantz, Eric Zimmerman, & Michael Sweet / Audiobrain

LEGO Junkbot
gameLab, 2001, published by LEGO.com
Ranjit Bhatnagar, Nick Fortugno, Peter Lee, Frank Lantz, Eric Zimmerman, & Michael Sweet / Audiobrain