

GAME DESIGN WORKSHOP

A Playcentric Approach to Creating Innovative Games

by Tracy Fullerton

with a foreword by Eric Zimmerman



**4TH
EDITION**



CRC Press

Taylor & Francis Group

AN A K PETERS BOOK

Chapter 7

Prototyping

Prototyping lies at the heart of good game design. Prototyping is the creation of a working model of your idea that allows you to test its feasibility and make improvements to it. Game prototypes, while playable, usually include only a rough approximation of the artwork, sound, and features. They are very much like sketches whose purpose is to allow you to focus in on a small set of the game's mechanics or features and see how they function.

Many first-time designers would rather jump in and start making the “real” game rather than starting

with a prototype. But if you invest the time, you will discover that there is nothing more valuable for improving gameplay than a well-thought-out prototyping process. When you are making a prototype, you do not need to be concerned with perfecting how it looks or whether the technology is optimized. All you need to worry about are the fundamental mechanics, and if these mechanics can sustain the interest of playtesters, then you know that your design is solid.

METHODS OF PROTOTYPING

There are many types of prototypes, including physical prototypes, visual prototypes, video prototypes, software prototypes, etc. A single project might require a number of different prototypes, each addressing a unique question or feature. The important thing to remember when prototyping is that you are not creating the final design, you are simply trying to formalize your ideas or isolate issues so that you can discover what works before going on to create the final design. This chapter will deal mainly with physical prototypes, those made using pen and paper, cards, dice, etc. to test the core game mechanics. Such paper designs are one of the most powerful tools a designer has to work with, but they are only one method of prototyping. [Chapter 8](#) will discuss digital prototyping and how to successfully use software prototypes in your design process.

Physical Prototypes

Physical prototypes are the easiest type of prototype for most game designers to construct on their own. These are typically created using slips of paper, cardboard, and household objects with hand-drawn markings. You are free to use anything you like, from lead figures to plastic army men to pieces borrowed from other games. Whatever you can cobble together is fine.

The benefits of physical prototyping are many. First, it allows you to focus on gameplay rather than technology. Over the years, in the many game design classes and workshops I have taught, I have found that when a team starts programming, they become very attached to their code. Making changes to the gameplay becomes a challenge right away. But if the

design is on paper, iterations do not seem as difficult. Don't like the way a turn structure works? Just change it and try it again. Games can go through more iterations in a shorter period of time and with little wasted effort. Another benefit of physical prototyping is that you can respond in real time to player feedback. If players come up with an issue or idea, you can incorporate it on the fly and see how it works.

Physical prototyping also allows for nontechnical team members to participate at a very high level in the design process. No one needs specialized knowledge or expertise in a programming language to give their input, which will allow for a wider variety of perspectives in the design process. And finally, physical prototyping allows for a broader and deeper experimentation process simply because it can be done without major cost or use of resources.

In early drafts of your physical prototype, I recommend that you pay very little attention to the quality of the artwork. Stick-figure drawings are the norm. The goal is to rough out system components so that you can see how the game operates on a mechanical level. Spending time on the artwork only slows down the process. Also, if you invest too much time crafting the look and feel of the prototype, you might become attached to your work and be reluctant to make changes. Because the prototyping process

is all about iteration and change, this becomes counterproductive.

Battleship Prototype

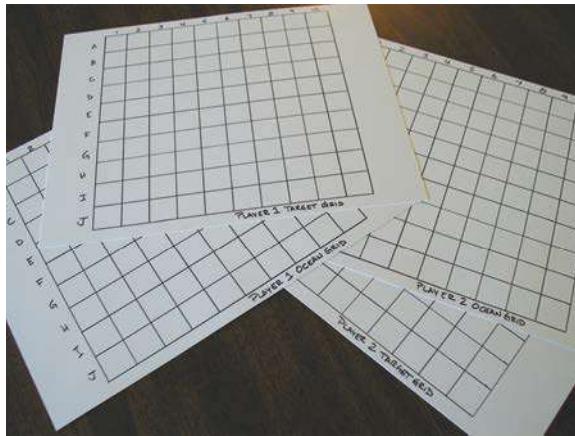
I am going to walk through the process of developing several physical prototypes so that you can get a sense of how they are made and used. I will start with a classic game with a simple system you have probably played before. If you are not familiar with Battleship, it is a popular two-player board game in which the object is to be the first player to sink your opponent's fleet.

Let's construct a physical prototype of this game. When starting with a prototype, it is best to identify the key elements of a game and then handcraft each element. In this case, take four sheets of paper and draw a 10 × 10 grid on each. Label the rows on each grid with the letters A through J. Label the columns on each grid 1 through 10. Put the following titles on the four grids: Player 1 Ocean Grid, Player 1 Target Grid, Player 2 Ocean Grid, and Player 2 Target Grid. The final set will look like [Figure 7.2](#).

Next find two players and give each an Ocean Grid, a Target Grid, and some colored markers, such as small beads. Players should shield their grids from their opponent's view. Each player distributes the following five ships by drawing on his Ocean Grid.



7.1 Prototyping materials



7.2 Battleship grids

The numbers in parentheses are the ships' sizes on the grid:

- Carrier (1×5 cells)
- Battleship (1×4 cells)
- Destroyer (1×3 cells)
- Submarine (1×3 cells)
- Patrol Boat (1×2 cells)

All segments of the ships should be drawn on the playing grid. Ships may not be placed diagonally. [Figure 7.3](#) shows an example of ships placed on the grid.

Now that you have the prototype assembled, it is time to play. On a player's turn, that player calls out grid coordinates, such as "B5." If the opponent has a ship on that cell, then he answers "hit." If not, he answers "miss." When all segments of a ship have been hit, the opponent says, "You sank my battleship!" Simple enough?

Players track hits and misses on their target grids. If B5 is a hit, the player marks a hit on his target grid. In my prototype, I've used different-color beads to mark the difference between hits and misses. Players take turns calling coordinates like this until one player sinks all five of the opposing ships. [Figure 7.4](#) shows an example of what grids will look like during play.

Play this game yourself. Think about it in terms of how it functions as a prototype. Does it accurately represent the game mechanics? Although the

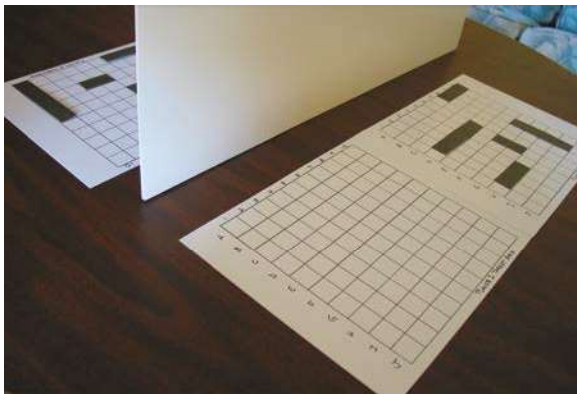
artwork is crude and the rules are rough, do they provide enough of an experience for someone to grasp the game and give feedback? If this is the case, then the prototype is a success.

As you can see, making a playable game prototype does not require programming skills or art skills. The experience generated by the paper version of Battleship is almost identical to the experience generated by the fully produced Milton Bradley version.

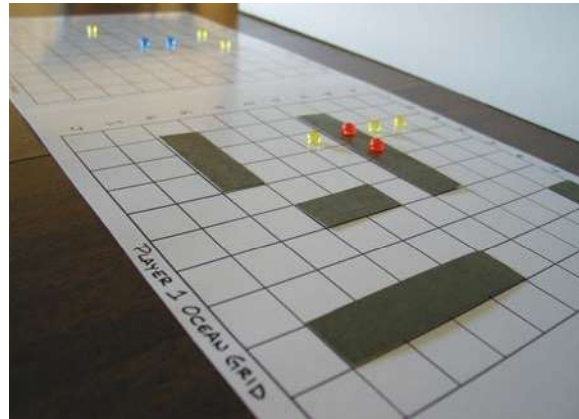
The advantage of prototyping is that as you assemble the game, you gain a tactile sense for how the mechanics fit together. Abstract rules suddenly become concrete. You can look at the grid and ask yourself, "What if I made that bigger? How would that affect the gameplay?" Enlarging the grid is simple; it is just a matter of getting out a piece of graph paper and drawing a bigger box. Then you can replay the game and see if the experience is better or worse.

Exercise 7.1: Modifying Your Prototype

Take your Battleship prototype and modify three aspects of the game. You can change the grids, the ships, the object of the game, the procedures for playing, etc. Get creative with the changes you make. After each change, play the game with a friend and describe how that particular change affects the gameplay.



7.3 Battleship grids with ships



7.4 Battleship grids during play

As you manipulate elements of the game structure, it will invariably spark more ideas, and it is not uncommon for entirely new systems to materialize during this process. You can then spin some of these systems off into their own games. After you become experienced at prototyping, you will find that this is probably the most effective way to create gameplay because it takes you right down into the mechanics and permits you to experiment in a way no other process can.

More Examples

Physical prototypes are critical for designing both board games and sophisticated electronic games. Many famous electronic games are based on paper games. The system for digital role-playing games such as *Diablo III*, *Baldur's Gate*, *EverQuest*, *Asheron's Call*, and *World of Warcraft* are derived from the paper-based system of *Dungeons & Dragons*. Likewise, the system for the famous computer game *Civilization* is based on a *Civilization* board game published by Avalon Hill.

The designers and programmers of these games used the paper-based originals to figure out what would work electronically. Also, many video game designers started out as board game designers, including Warren Spector, whose *Designer Perspective* you can find on page 27. Building and revising paper prototypes instills a deep understanding of gaming principles, and it does so in a setting that is not bogged down by the complexities of software development.

One good way to train yourself in the design of game mechanics is to challenge yourself with controlled design exercises in which you take an existing game system, set a new player experience goal, and make changes to the system to meet that goal. While not as difficult as designing a game from scratch, this is good practice in thinking through design problems and designing to meet a goal.

My next example will use another simple system, a children's game by the company Ravensberger called *Up the River*. You might not have played this game, but I will walk through the original rules and

the creation of an initial prototype in the same way I did with *Battleship*. I have done this particular exercise with hundreds of game design students all over the world, and the system, while simple, has lent itself to a wide range of resulting game concepts.

Up the River Prototype

Up the River has an unusual board design. The board is made up of 10 equally sized pieces, as can be seen in [Figure 7.5](#). These pieces are lined up to form the river. To make your own board pieces, just cut up regular white paper as shown in [Figure 7.6](#). When the game begins, the piece at the bottom of the river is the sand bar, and the fifth piece from the bottom is the high tide. These are special terrain that will be explained later; be sure to mark them on your prototype board pieces. At the top of the river sits the harbor or goal card, which you will need to create as well. This card has 12 numbered docks, or spaces, across its top. In addition to the board, you will need some player pieces and a six-sided die. You can use beads or buttons of four different colors for the player pieces, or boats, and you will need three of each color. To begin, players line all their pieces up on the fourth piece from the bottom of the board, as seen in [Figure 7.5](#).

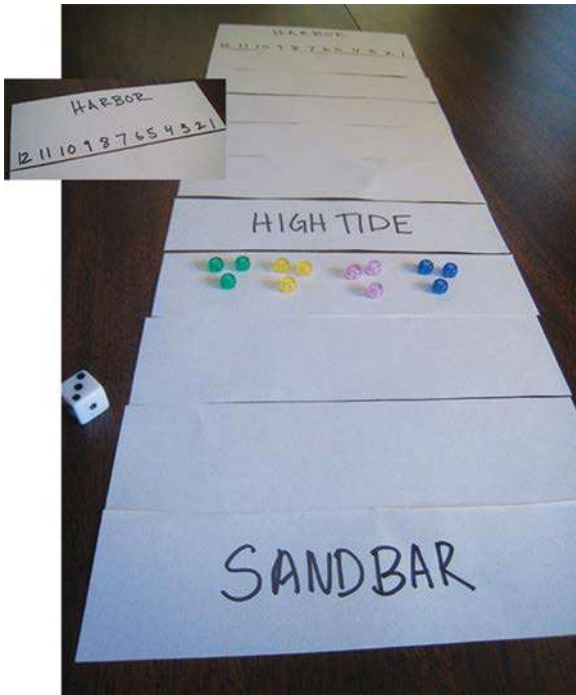
The objective of the game is to move all three of your boats to the harbor card and earn the most points. Your score is the total of all docks your boats are placed at, and the player with the most points wins.

The procedures of the game are simple: The youngest player goes first. On their turn, a player rolls the die and chooses one of their boats to move forward that many spaces. A player can only move one boat per turn. If a boat lands on the sandbar, it must stop there until the player's next turn, even if the roll was higher. If a boat lands on the high tide card with a direct throw, it advances three extra spaces, even if this means it advances into the harbor. A player does not need an exact throw to enter the harbor.

So far the game seems fairly mundane—a dice race. But there are two special rules that give the



7.5 Up the River



7.6 Up the River prototype

system just enough of a twist to make it interesting. The first is called the waterfall. After each player has had a turn, the bottom piece of the board is moved to the top. This simulates the current of the river pushing the boats back downstream. Any boats that are on that bottom piece are lost and taken out of the game. Suddenly our simple dice race has a more dramatic twist! Each time you choose a boat to move, you must consider the placement of your other boats as well. Are they at risk? Will they be at risk the next time the waterfall occurs? As I have discussed, this simple dilemma adds conflict to the system.

The next special rule is called good wind/ill wind. This occurs when a player rolls a six. Instead of moving six spaces, the player must now make a choice: whether to move any one of their boats *up* to join their next boat farther up the river—good wind—or whether to move one of their opponent's boats *down* to join the nearest boat of the same color—ill wind. When choosing good wind, if a boat moves past the sandbank, it must stop there. When choosing ill wind, if a boat moves past the sandbank, it need not stop. If the player who rolled the six has only one boat, or if all their boats are on the same card, the good wind option is not available. If their opponents have only one boat or all their boats are on the same card, the ill wind option is not available. If neither option is available, the player who rolled loses their turn, and the turn moves to the next player. The good wind option may not be used to move into the harbor.

The good wind/ill wind option adds an interesting choice to this simple system. Players can, in effect, choose to act for themselves or against their opponents. This moment of choice is an example of player-to-player interaction that creates an interesting moment of gameplay. When players move their boats into the harbor, they place them on the next available dock and score the number of points on that dock. The game ends when all the boats have either gone over the waterfall or entered the harbor. All the points are added up, and the player with the highest score wins.

Play your Up the River prototype and analyze how each element in this simple system adds to the

game. Ask yourself these questions about the formal system:

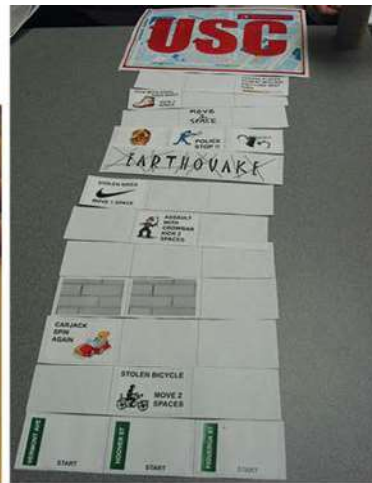
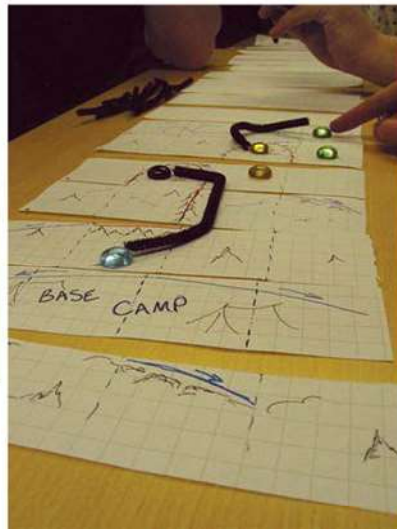
- What is the relationship between the size of the board and the number of points on the die? What happens if you change the size of the board?
- What is the relationship between the number of boats each player has and the starting position? What happens if you change the starting location?
- Why is the starting position of the sandbar important? What about the high tide card?
- What skills are necessary to play this game? Is the game ultimately decided more by skill or by chance?
- What does the good wind/ill wind option add to the game?
- Why does play begin with the youngest player?
- Who is the market for this game?

Thinking about these and other questions should lead you to see some potential changes you might

make in the game system, but change for the sake of change is not the goal here. Before you begin modifying this system, brainstorm several possible player experience goals for your new version of the game. Here are some examples:

- The game is resolved primarily by strategy rather than by chance.
- The game has teams on which each player has a special role to play.
- The game has more player-to-player interaction, including negotiation.

In addition to your player experience goal, you will want to come up with a dramatic metaphor for your new game that reflects your player experience goal. Figure 7.7 shows a number of variations on Up the River, including a mountain climbing game that required teamwork, a sea rescue game where players collaboratively worked to save swimmers from drowning, and a parking game that enhanced the formal structure by breaking each row into three columns.



7.7 Variations on Up the River

Exercise 7.2: Up the River Variation

Create your own variation of Up the River. Set a player experience goal first and brainstorm ideas to change the system to meet that goal. Then modify your Up the River prototype, or build a new one, to reflect your changes to the system. Play your variation with friends and see if you have met your experience goal.

You can create your own controlled design exercises and continue practicing your design process. Just start with an existing game system and analyze it to clearly understand its formal, dramatic, and dynamic elements. Then come up with a new player experience goal and make changes to the system to meet your design goal. I advise starting with very simple games. And remember, even small changes to a tightly balanced system can have a great impact on the gameplay. By practicing this process, you will become a stronger designer and gain a deeper understanding of many different types of mechanics.

Prototyping a First-Person Shooter

It's one thing to prototype a simple board game, but you are probably wondering if it is possible to create a physical prototype of an action-packed video game. The answer is yes. While a paper prototype of a digital system has limitations, it is still quite valuable to the design process. For example, you can create a paper prototype of a game in the first-person shooter (FPS) genre. Classic examples of first-person shooters include Quake, Castle Wolfenstein, Battlefield 1942, Half-Life, Unreal Tournament, and Call of Duty. The core game mechanics of these games involve player units running around shooting other units. That is simple to understand, but how do you model one of these games on paper, and what can that teach us?

A physical prototype of a first-person shooter can help you understand the larger tactical and strategic issues of weapon balance, territorial control,

etc., but it won't help you to understand the fluid process of running, aiming, and shooting in a 3D environment. In this way, it is possible, in fact probable, that an accurate paper prototype of a first-person shooter will fail to capture the essence of the game's player experience while still providing a valuable design process. As I will discuss in the following chapter on digital prototyping, one game can have many different prototypes, each addressing different questions about the design. A paper prototype is well suited to some questions about the design of a first-person shooter, for example, those regarding level design and weapon balance, while not being suited to others. The distinction should become clear to you as we construct our physical prototype for a first-person shooter.

Arena Map

Take a large sheet of hexagonal graph paper. Hexagons are nice for prototypes because they allow units to move diagonally. You can purchase this graph paper at most board game stores or print it out using one of several freeware and shareware programs available online, such as Hexographer. The grid will serve as the arena for your game.

Cut out a small paper chit and color it red to mark spawning points. A spawning point is the cell on the grid where units materialize after they are killed.

Put lines on the grid to represent walls. Units cannot move or shoot through walls. It is helpful to make walls out of objects that can be repositioned on the grid. Matchsticks are perfect for this. Having moveable walls makes it easier to tweak the system.

You probably already have questions like: How many hexes should be on the grid? How big should each hex be? How many spawning points do I need? and Do I need lots of walls or only a few? The answer to all of these questions is: Take your best guess. There is no way to know what will work until you play the game. No matter what you decide, you will probably wind up changing it later on. Pick whatever parameters you deem reasonable and proceed with the process.

CATASTROPHIC PROTOTYPING AND OTHER STORIES

by Chaim Gingold

Chaim Gingold is a game designer and theorist with 20 years of research and design experience in both industry and academia. He is most well known for designing the Spore Creature Creator and creating Earth: A Primer, and his design expertise and research interests include authoring tools, simulation, and play. Currently, he is a researcher at Y Combinator Research (YCR) / Human Advancement Research Community (HARC).

My hard drive was full of failures. Twelve years after learning to program, I looked back on all my software: Almost none of it was finished, and what was wasn't ambitious enough. The projects that started out ambitiously always seemed to fall back to earth, like failed rockets lacking the power to propel their own weight into orbit. Sure, there were interesting ideas in there, lots of wacky toys, and I had even attempted a few large projects, but none of them ever came together like the cool games and software I had always admired.

Sure, I had become a pretty good programmer and learned to make cool stuff, but clearly none of it would ever amount to anything. I just didn't have what it took.

I went to graduate school at Georgia Tech and read some Chris Crawford. I learned that he had the same problem. But he didn't think of it as failure. For him, this was an organic part of the development process. The failures filling his head were actually prototypes that helped him decide which ideas were worth pursuing. For each good idea, there were a large number of stupid ones that didn't work out. Failing, for this successful designer, was a way to find the good ideas. The revelation hit me like a ton of bricks. Maybe I had a chance after all.

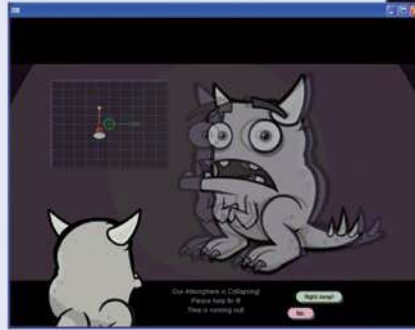
Ken Perlin came to Georgia Tech and gave a talk on his work with emotional software actors. His work blew my mind. He had an infinite series of cool little toys, which he considered to be sketches or studies. Master artists like Escher or van Gogh don't just sit down and crank out a finished piece. Artists create numerous sketches and studies before they undertake finished paintings, let alone masterpieces. Ken's larger demos clearly built on top of what he had learned in previous ones. It all formed one long line of inquiry and research. In Ken's world, my failures, which I was now calling *prototypes*, are like an artist's studies, a necessary part of any major undertaking.

All of my software failures, which I was now thinking about as prototypes, sketches, and studies, had taught me a thing or two about design and programming. If you want to learn to draw, you have to make a ton of bad drawings first. The difference between practice and failure is simply a matter of attitude. One thing led to another, and my experience, plus being in the right place at the right time, led to an internship with Will Wright. He was working on a new game at the time, code named Spore, and had a small team working on prototypes.

I went to Walnut Creek for the summer, joined the impossibly small Spore team, and it all finally started to click into place. Maxis was in the thick of The Sims Online, and the other intern and I were placed in the hallway outside of Will's office, next to the Elvis shrine, on folding tables. Under my "desk" was one of Will's old Macs, a fancy machine from the mid-1990s, which, to my delight, we hooked up. Spending the summer at Maxis was like going to Santa's workshop at the North Pole and finding out how the elves made the toys.

The old Mac was like a treasure cave, a historical archive of blueprints, prototypes, projects, and concepts. I could study the source code to some of my favorite games, like SimAnt, SimCity, and SimCity 2000. But that wasn't even the best part. I found an ambitious Maxis project about tribal civilization from the early 1990s that was never completed. It was like a murder mystery. Why had this project died? The hard

drive was full of prototypes for a secret project, which turned out to be The Sims. Apparently, Maxis had been working on the game for a long time, and many aspects of it had been prototyped in isolation, including a 2.5d character animation system and editor, the motive- and decision-making AI, and a house editor. The code to the last prototype was clearly a hacked version of the SimCity 2000 engine. I found a program that used genetic algorithms to procedurally generate SimCity-



Prototypes from Spore

style buildings with a blind watchmaker-style interface. That program had clearly been written as efficiently as possible, not from a runtime point of view, but from an implementation standpoint. It was using the SimCity 2000 code base as a host organism for some rapid experimentation. Will's imagination had clearly been running faster than proper software engineering practice allowed. All of this, plus the awesome array of prototypes the Spore concept team had been cranking out, made a big impression on me. I joined in and contributed some of my own wacky prototypes to Spore.

What was going on here? What did all of this mean? Thinking back, I realized I had made two classic mistakes. First, my eyes were bigger than my stomach. The ambitious projects I had undertaken in the past "failed" because I made the mistake of not proving out the core ideas in prototypes. You can't send a rocket to the moon if you haven't first experimented with launching simple toy rockets. My sense is that the tribal civilization game died for similar reasons: Its author had launched into an ambitious finished project without doing the proper research, sketches, and prototypes.

Secondly, my success/failure evaluation function had been wrong. While the code to the The Sims prototypes wasn't in the final game, they had clearly informed the final product. All of my small "failures" were actually a series of small successes that had improved my design skills, and they were in fact studies I could incorporate into larger projects. I had it backward the whole time. My "successful," but incomplete, large projects were the real failures. I had invested too much energy into large projects that would fail because I hadn't done my homework. It's a hard lesson to take and one that most people probably have to learn the hard way.

So I dusted off my ACM programming competition skills, which taught me to make tightly focused programs in minimal time and with minimal frills. I became a better designer, really fast. I gained tons of design experience points by slaying so many gremlins.

I finished at Georgia Tech and joined Spore. My vast collection of tiny student projects, bite-sized personal projects, and work prototypes I had made added up to a huge amount of experience and intuition. Compared to my peers, I had a tremendous amount of design experience, simply from writing, evaluating, and throwing away so many ideas. I witnessed good prototypes move mountains. I like to think of good

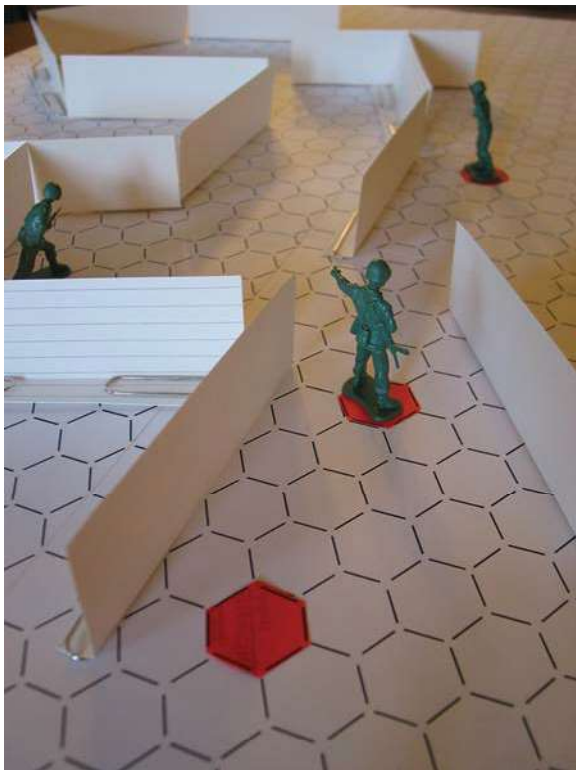
prototypers as powerful ninjas who can drop into hard design challenges, or tedious design debates, and cut them to shreds with one swift movement of their prototyping blade.

Here are a few prototyping rules of thumb. Even with years of experience, I often find a prototype going nowhere and can usually trace the problem to not following one of these rules:

- *Always Ask a Question.* Always ask a question, which will give you purpose, and have a hypothesis, which is a specific idea you are testing out. For example, you might be thinking about mouse-based control schemes for a school of fish. Your question is: How do I control these fish with a mouse? A hypothesis might be: Flocking will make the fish move together, and every mouse click will drop an invisible “bomb” that will act as a repulser upon every fish’s steering AI, and it will take a few seconds to complete exploding. A good way to make sure you aren’t going to waste time implementing ideas you don’t actually have, which happens to me more often than I’d like, is to diagram the idea on paper first and work out as many details with a pen as possible. This also speeds up writing the prototype.
- *Stay Falsifiable.* Just like good science, you must validate the results of your experiment. Did your hypothesis work? Does your fish flock control scheme feel good to you? Do your friends find that it feels good? Does it work in the context of your game idea? You can never user test and playtest an idea too early. I have seen many cool ideas go down in flames because their owners were overprotective, didn’t think it was ready, didn’t believe the feedback they were getting, explained away people’s responses, or thought that only their opinion mattered. Eventually users will play with your work, and by then it will be much harder to fix the design. Incorporate the user into the design process as early as possible. Be honest with yourself and your players, and you will be richly rewarded. This one is easy for me because, as a designer, my main intent is to entertain and transform other people, so I’m always interested in what effect my work has on others. Watching people use what you make will also make you a smarter designer.
- *Persuade and Inspire.* We’re making entertainment and art—your prototype should be cool and fun and should excite people. If you and your peers are compelled, your players will be, too. On the flip side, if something isn’t resonating with other people, perhaps your idea or approach should be reconsidered. Prototypes can be powerful persuasive devices. Keita Takahashi, the designer of *Katamari Damacy*, couldn’t convince anyone that rolling around a giant sticky ball would be fun. Until they played the prototype.
- *Work Fast.* Try to minimize time to your first “failure” (rejecting a hypothesis), and don’t be afraid to push the eject button. A classic error is to spend months working on an engine, architecture, or something else that has nothing to do with proving out your core design idea. Prototypes don’t need engines. Prototypes are slipshod machines held together by bubble gum and leftover bits of wire that test and prove simple ideas as quickly as possible. If you find yourself weeks or months into a project with only an engine, you’ve failed. Perhaps you need to articulate a specific gameplay idea to validate. For me, the ideal window of time to start and finish a prototype (including design, implementation, testing, and iteration) is two days to two weeks. Anything longer than that sets off alarm bells.
- *Work Economically.* You’re making something small and beautiful, so invest development effort wisely. To work fast, you must stay small: Don’t do too much at once or you’ll never make progress. Be realistic. Here are some questions to ask yourself when you are considering how much effort to spend on proper engineering, art, interface design, or any aspect of your prototype. What’s the

purpose of this prototype? Who will use it? What's important? Look? Kinesthetics? Load time? Run time? Usability? Persuading your peers? Be a cheap, lazy, slothful programming bum. Just make it work so you can test your idea. Don't go above and beyond the call of duty in programming, art, or any other aspect of your prototype.

- *Carefully decompose problems.* Don't bite off more than you have to at once. If you prototype all systems simultaneously, you will fail because you can't work fast or reach any kind of conclusion. To build it all at once is to build the actual game, which is hard. The prototype designer's job, like a good Go player, is to cut and separate the enemy stones (your design problem) into small, weak groups that can be killed or manipulated at will. Wisely divide your problem into manageable pieces. You must be careful because problems are sometimes connected in nonobvious ways and bite you later. Through practice, your designer's intuition and experience will help you see the connected nature of the problem you are trying to subdivide and make the most judicious cuts.



7.8 FPS prototype example

Units

Units are your characters in this game. You can represent them with coins or plastic army men or other household objects. Whatever you use should fit within one cell on the grid. In addition, a unit should clearly show which direction it is aiming. For example, if you use coins as units, draw an arrow on them to indicate their direction.

This prototype is designed so multiple units can play at the same time. To determine starting cells for the different unit on the grid, roll a die. The player with the lowest number places their unit on the grid first. Go in clockwise order from there and have each player choose a starting cell. An example of what your prototype might look like is presented in [Figure 7.8](#).

Exercise 7.3: Movement and Shooting

If you want a challenge, stop reading now and come up with your own movement and shooting rules. Explain your reasoning behind this set of rules.

Movement and Shooting Rules

Here is one possible solution for movement and shooting. There are endless other creative possibilities, and I encourage you to experiment with them.

Each player gets the following nine cards:

- Move 1 space (1)
- Move 2 spaces (1)
- Move 3 spaces (1)
- Move 4 spaces (1)
- Turn any direction (2)
- Shoot (3)

Play is executed in rounds.

1. *Build stack*: Each player chooses three cards and places them face down on the table in a stack.
2. *Reveal*: Each player turns over his top card.
3. *Resolve shoot cards*: Players with a shoot card fire in the direction their unit is pointed. They follow an imaginary line across the grid. If this line intersects with a cell containing another unit, the shot hits. If this line comes to a wall or otherwise does not intersect with a unit, it misses. Shots happen simultaneously so that two or more players can hit at the same time.
4. *Resolve turn cards*: Players with turn cards turn their unit to whatever direction they please. If two or more players have turn cards, roll a die to determine who turns first.
5. *Resolve move cards*: Players with move cards move their units the number of spaces specified on the card. If two or more players have move cards, roll a die to determine who moves first. Players cannot occupy the same cell.
6. Repeat steps 2-5 for the second card in the stack.
7. Repeat steps 2-5 for the third card in the stack.

If a unit is shot, it is removed from the grid, and the player chooses one of the spawning points on the grid and reappears there at the beginning of the next round.

This process of prototyping an action-based game might at first seem to be complex, but if you think about what I have described, it is pretty amazing. In just a few pages, I have completely detailed how to build a prototype of a first-person shooter using only pen and paper. When you play with this model, you will see that it is both flexible and simple to use.

Some suggested additions to your first-person shooter prototype are as follows:

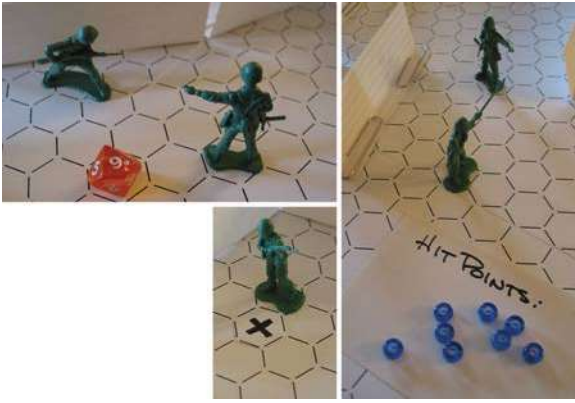
- *Add a scoring system*: Make players track the number of kills they get. The first player to get 10 kills wins the game.
- *Include a hit percentage*: Suppose the chance that a shot hits is 100% when two units are standing on adjacent hexes on the grid. This percentage decreases by 10% for each hex of distance added. Calculate hits and misses using a 10-sided die.
- *Provide hit points*: Have each unit start with five hit points. One shot suffered removes one hit point.
- *Drop in first aid*: If a unit stands on a first-aid hex on the board for a full round, then his hit points return to their original amount.
- *Add in ammo*: Units start with 10 rounds each. Every time they shoot, one bullet is removed. If a unit stands on an ammo hex for a full round, he will reload his clip.
- *Introduce other weapons*: New weapons can be placed on the grid. If a unit stands on the weapon, he can use it in the next round. Enhancements to weapons include more damage per shot, higher accuracy, more bullets, etc.
- *Introduce alternative objectives*: What if this game were not focused entirely on direct combat but had alternative objectives for the players, such as rescuing nonplayer characters or finding narrative objects within a constrained time limit?

Exercise 7.4: Build It Yourself

Build the physical prototype described just previously and test it out. Describe any problems that you encounter. Also, list out any questions you have while building it.

Exercise 7.5: Features

Add some or all of the features mentioned previously plus a few that you dream up yourself and incorporate them into the physical prototype. Write down how these features affect the gameplay.



7.9 FPS prototype example with additions; clockwise from top left: hit percentage, hit points, and first aid

New rules and features can continue to be added, altered, and removed. You can use the system to create capture-the-flag games, cooperative play missions, and other interesting twists. You can continue adding, testing, and tweaking until you come up with the right combination. Each time you add a rule or feature, it might spark new ideas and lead you down a path you did not expect to go. This is the heart of the creative process, and you should encourage yourself to try things that might seem ridiculous or absurd and just see what happens when you play the game. Completing these exercises will give you insight into how many first-person shooters and 3D adventure games are designed.

Exercise 7.6: Working Backward

Now let's apply what you have learned to a different type of game.

1. Take two different real-time strategy (RTS) games, such as StarCraft II and Age of Empires IV, and work backward. Strip away the external feature set and show what both games have in common. These are the core game mechanics.
2. Translate the core game mechanics for one of the RTS games to paper in a playable format.

Remember, all we care about are the rules that correlate between the two games. These rules represent the core gaming system and will form the basis for your RTS physical prototype.

Perspective on Physical Prototyping

People who are not used to physical prototyping might argue that this method does not accurately represent the player experience on a computer. They might think a pen and paper prototype might work for a turn-based game, but not for an action-based shooter because gameplay is integrally tied to the 3D environment and the ability of the players to act in real time. I am not arguing that physical prototyping replaces those things. What I am saying is the overall gaming system can benefit tremendously in its early stages by building a physical prototype.

Physically prototyping allows you to build a structure for the game, think through how the various elements interact, and formulate a systemic approach to how the game will function. The sensory experience created by a digital game—that is, the feeling of moving through a 3D space—is only one component of an engaging game experience. Although it is a critical component, it can be isolated and focused on later in the process. At a minimum, physical prototyping forces you to think through the design elements and define them. You can always change them down the road, but this gives you a framework to build upon, and that in itself can provide better focus when it comes to working and communicating with a production team.

Imagine starting a project with a new group of programmers who know nothing about the concept, and imagine trying to describe to them the game you have in your head. It is not easy. If you want to create gameplay that people have never seen before, it might be impossible. A physical prototype that they can sit down and play ensures that they will be able to grasp your vision of the game. It gives you all a starting point for concrete discussions about how

the game will function. A written treatment or design specification is part of the process, of course, but when it comes to communicating a complex system,

these documents are best supported by a prototype, or set of prototypes, that the team can actually play and discuss.

PROTOTYPING YOUR ORIGINAL GAME IDEA

Now that you have some experience creating and modifying prototypes, it is time to take one of your game concepts and create your own original prototype. The first step is to pick one of the ideas that you brainstormed in [Chapter 6](#), “Conceptualization.” When your idea and concept treatment are done, you will be ready to make your first prototype. But before I dive into the mechanics of constructing the prototype, make sure you have clearly articulated the core gameplay that will be created.

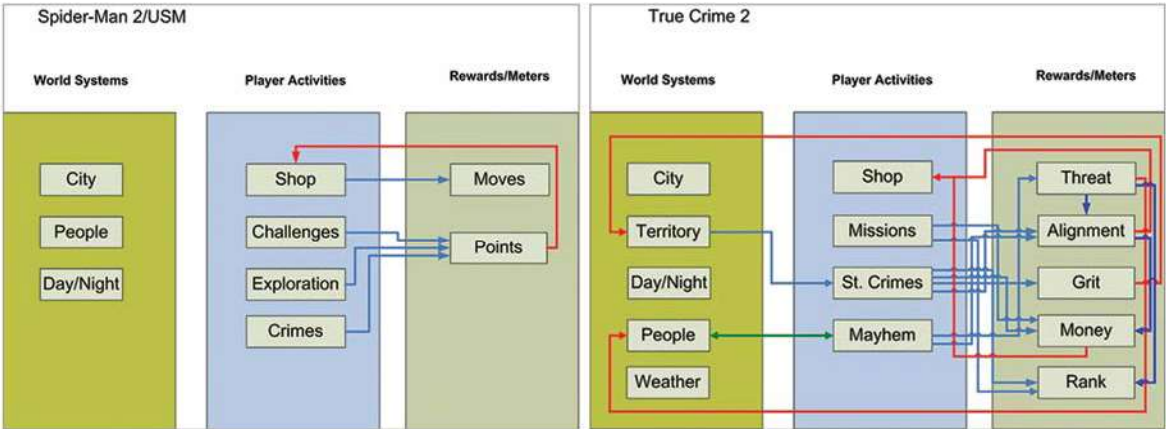
Visualizing Core Gameplay

If you try to design the entire game at once, you might become confused and overwhelmed. There are so many elements in a typical game that it is difficult to know where and how to start. What I recommend is that you isolate the core gameplay mechanisms and build out from there.

The core gameplay mechanism, or “core mechanic,” can be defined as the actions that a

player repeats most often while striving to achieve the game’s overall goal. Games are repetitive by nature. While the meaning and consequences of what a player does can change over the course of game, the core actions tend to remain in place and are built on as gameplay progresses. See Dan Cook’s discussion of loops and arcs on page 153 for more about understanding these core game mechanics.

[Figure 7.10](#) is a visualized analysis comparing the core actions of Spider-Man 2 with True Crime that was done by Jeff Chen, game analyst, and Carl Schnurr, senior director of game design, both at Activision. As you can see, player actions in these diagrams are interrelated with meters and rewards. In the case of Spider-Man 2, challenges, exploration, and rewards all translate into points, which can be spent in the Spidey Store to buy upgrades, combos, health, etc. This is a very simple reward system that will motivate players to look out for opportunities to gain points. If you look at the diagram for True Crime, you can see it is a much more complex design. The



Diagrams courtesy of Activision Central Design (Jeff Chen and Carl Schnurr)

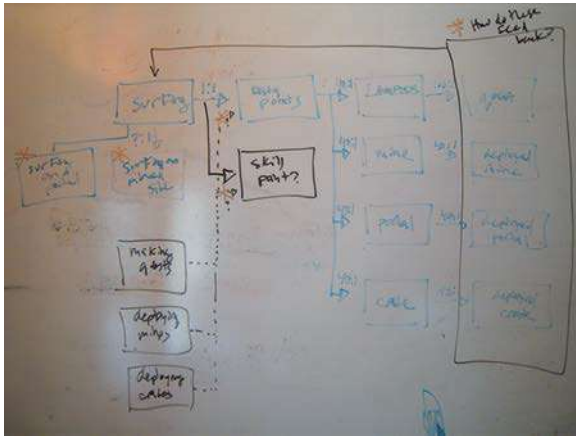
7.10 Visualization of gameplay mechanics for Spider-Man 2 and True Crime 2

player activities pay off in multiple forms of rewards, and these rewards and meters in turn have an effect on the overall world system. It should be noted that more complex designs do not always make for a better player experience.

As I discussed in [Chapter 5](#), sometimes you will find your mechanics are creating a positive or negative feedback loop that throws the play out of balance. By diagramming your core game actions, you are likely to spot such a problem early on. Your visualization does not have to be done in a formal presentation style, like the examples from Activision. You can just sketch it on a piece of paper or on a whiteboard as in [Figure 7.11](#), which is a very rough visualization of the core actions in a student game prototype. Even a rough sketch such as this can expose features that are not integrated into the main mechanics and allow you to go back and redesign to better integrate these features.

Here are some examples of popular games and brief descriptions of their core gameplay mechanisms:

- *WarCraft III*: Players build and move units on a map in real time with the intent of engaging opposing units in combat and destroying them.
- *Monopoly*: Players buy and improve properties with the goal of charging rent to other players who land on them in the course of play.



7.11 Rough visualization of core gameplay

- *Diablo III*: Players battle monsters, seek treasure, and explore dungeons in an attempt to amass wealth and become more powerful.
- *Super Mario Bros.*: A player controls Mario (or Luigi), making him walk, run, and jump, while avoiding traps, overcoming obstacles, and gathering treasure.
- *Atomic Bomberman*: Players move their Bomberman around a maze and drop bombs next to their opponents in an attempt to blow them up.

Exercise 7.7: Diagramming Core Gameplay

If you are familiar with these games, you can probably sketch out a visualization of their core gameplay mechanisms fairly quickly. If you are not familiar with them, choose two or three games that you know, write a short description of their core gameplay, and then sketch a visualization of it like the ones previously shown.

Exercise 7.8: Diagramming Core Gameplay 2

Now try diagramming the core gameplay of your own game idea. Your treatment from Exercise 6.8 on page 190 should give you a head start with this. If you find you do not know how some of the activities should interrelate, just take your best guess. The answers are going to evolve as you prototype and revise your game, so do not let them slow you down here at the beginning.

Building the Physical Prototype

Now that you have practiced by making and changing prototypes of existing games, you are ready to begin prototyping your original game concept. Here are four steps that will help you build a physical prototype efficiently.

1. Foundation

Build a representation of your core gameplay. Get some arts and crafts materials, such as cardboard, construction paper, glue, pens, and scissors. Draw

a board layout or rough map if you want to, and cut pieces out of the cardboard and paper.

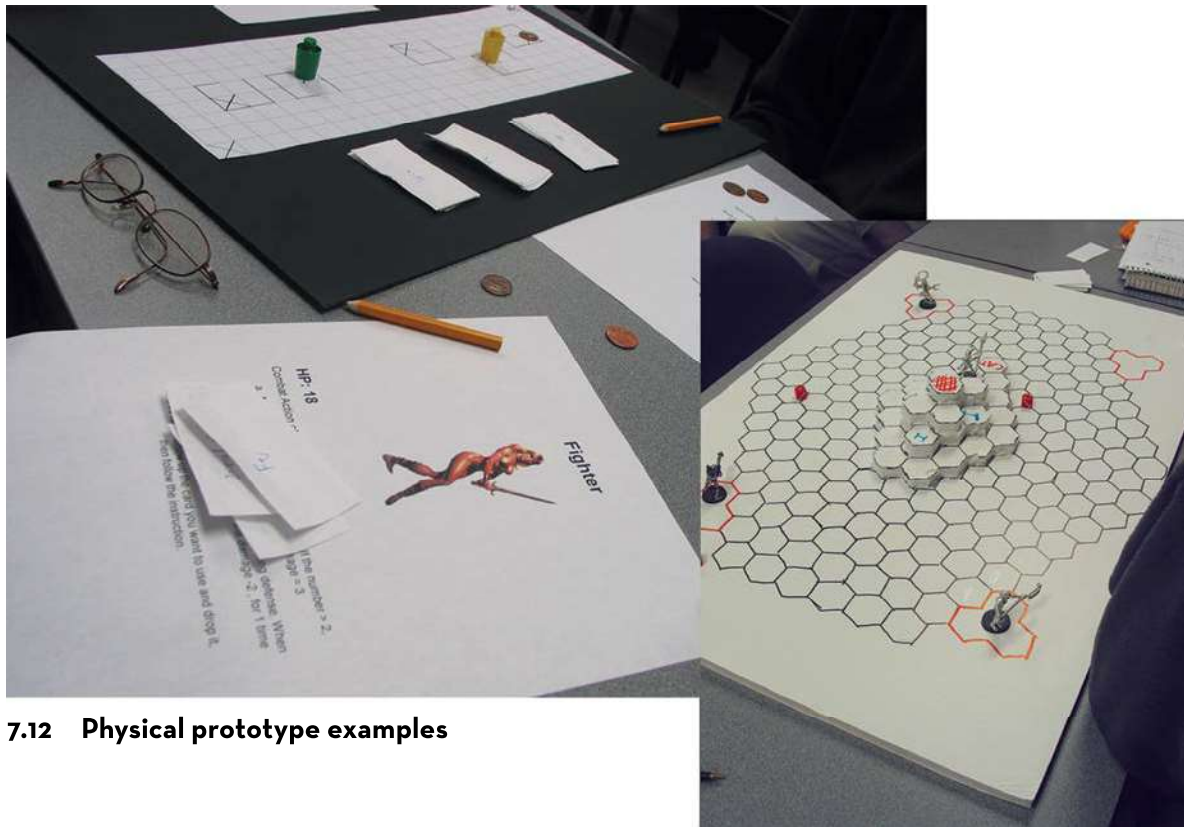
As you do this, questions will come to your mind. How many squares should a player be allowed to move? How will the players interact with one another? How is the conflict resolved? Do not try to answer all these questions at once. In fact, place the questions on the back burner and focus on the core gameplay.

Designing the basic game objects (physical setting, units, resources, etc.) and the key procedures for the game (those repetitive action cycles that keep the game in motion) are the heart of the foundation stage.

Try playing your core gameplay on your own—it might not be much of a game, but you will be able to see if the basic concept is worth pursuing. After you have your foundation in place, questions that you will want to answer become evident. But watch out. Try to test the game without expanding the rules at

this point. If you have to add a rule to make the prototype playable, then add it, but only do this if it is absolutely necessary. Your goal should be to keep the core gameplay mechanism down to as few rules as possible.

In the FPS prototype, the first element that I fleshed out was simultaneous movement because this is the core mechanic of the game. The idea that all players should reveal an action card at the same time to simulate real-time movement was conceived. This was a foothold to build upon. From there, the next logical question was: What are the options on the action cards? The answer was: move, turn, or shoot. Other ideas for action cards popped up as well, such as stand, crouch, go prone, etc. However, I decided to keep the options as simple as possible at first. These options lead us to the next stage of our prototype: structure.



7.12 Physical prototype examples

THE DESIGN EVOLUTION OF MAGIC: THE GATHERING

Magic: The Gathering is one of the most important and influential games of our time. It was an instant hit when it first appeared at the Gen Con game convention in 1993 and has grown steadily in popularity since. This is a special two-part look at the creation and development of the game as written by the designer, Richard Garfield. Richard wrote the first part, “The Creation of Magic: The Gathering,” over 25 years ago when the game was first released. In it, he muses about the design challenges of a collectable trading card game, and he recounts the game’s fascinating playtest history.

The second part, “Magic Design: A Decade Later,” is a retrospective on the original design notes. In it, Richard provides insight about how and why the game evolved the way it did, including thoughts on the Magic Pro Tour, Magic Online, and the following 10 years for the game.

THE CREATION OF MAGIC: THE GATHERING

by Richard Garfield (written in 1993)

The Ancestry of Magic

Games evolve. New ones take the most loved features of earlier games and add original characteristics. The creation of Magic: The Gathering is a case in point.

Though there are about a dozen games that have directly influenced Magic in one way or another, the game’s most influential ancestor is a game for which I have no end of respect: Cosmic Encounter, originally published by Eon Products and rereleased by Mayfair Games. In this game, participants play alien races striving to conquer a piece of the universe. Players can attempt their conquest alone or forge alliances with other aliens. There are nearly 50 alien races that can be played, each of which has a unique ability: The Amoeba, for example, has the power to Ooze, giving it unlimited token movement; the Sniveler has the power to Whine, allowing it to automatically catch up when behind. The best thing about Cosmic Encounter is precisely this limitless variety. I have played hundreds of times and still can be surprised at the interactions different combinations of aliens produce. Cosmic Encounter remains enjoyable because it is constantly new.

Cosmic Encounter proved to be an interesting complement to my own design ideas. I had been mulling over a longtime idea of mine: a game that used a deck of cards whose composition changed between rounds. During the course of the game, the players would add cards to and remove cards from the deck so that when you played a new game it would have an entirely different card mix. I remembered playing marbles in elementary school, where each player had his own collection from which he would trade and compete. I was also curious about Strat-o-matic Baseball, in which participants draft, field, and compete their own teams of baseball players whose abilities are based on real players’ previous year statistics. Intrigued by the structure of the game, I was irritated that the subject was one for which I had no patience.

These thoughts were the essence of what eventually became Magic. My experiences with Cosmic Encounter and other games inspired me to create a card game in 1982 called Five Magics. Five Magics was

an attempt to distill the modularity of Cosmic Encounter down to just a card game. The nature of Cosmic Encounter seemed entirely appropriate for a magical card game—wild and not entirely predictable, but not completely unknown, like a set of forces you almost, but don't quite, understand. Over the next few years, Five Magics went on to inspire entirely new magical card games among my friends.

Ten years later, I was still designing games, and Mike Davis and I had come up with a board game called RoboRally. Mike was acting as our agent, and among the companies he approached was a brand new gaming company called Wizards of the Coast. Things seemed to be going well, so that August, Mike and I made our way to Portland, Oregon, to meet over a pizza with Peter Adkison and James Hays of Wizards of the Coast.

Both Peter and James were very receptive to RoboRally, but they informed me that they weren't really in a position to come out with a board game right away. This wasn't what I had come out to hear, of course, but I didn't want the trip to be a total waste. I asked Peter what he would be interested in. Peter replied that he really saw a need for a game that could be played quickly with minimal equipment, a game that would go over well at conventions. Could I do it?

Within a few days, the initial concept for a trading card game was born, based on another card game I had developed in 1985 called Safecracker. It hadn't been one of my best games. But then I remembered Five Magics.

The First Designs

I went back to graduate school at the University of Pennsylvania and worked on the card game in whatever spare time I had. It wasn't easy; there were three months of false starts on the project, there are so many aspects of card game design that have to be reconsidered when designing trading card games. First of all, you can't have any bad cards—people wouldn't play with them. In fact, you want to prevent too much range in the utility of cards because players will only play with the best—why make cards people won't play with? Besides, homogeneity of card power is the only way to combat the “rich kid syndrome” that threatened the game concept from the start. What was to keep someone from going out and getting ten decks and becoming unbeatable?

It was a major design concern. I had numerous theories on how to prevent purchasing power from unbalancing the game, none of which were entirely valid but all of which had a grain of truth. The most compelling counter to this “buy out the store” strategy was the ante. If we were playing for ante, the argument ran, and your deck was the distilled fruit of ten decks, when I did win, I would win a more valuable card. Also, if the game had enough skill, then the player purchasing their power would surely be easy prey for the players dueling and trading their way to a good deck. And of course there was the sentiment that buying a lot of poker chips doesn't make you a winner. In the end, however, the “rich kid syndrome” became less of a concern. Magic is a fun game, and it doesn't really matter how you get your deck. Playtesting showed that a deck that is too powerful defeats itself. On the one hand, people stopped playing against it for ante unless a handicap was invoked; on the other, it inspired them to assemble more effective decks in response.

The first Magic release was affectionately named Alpha. It consisted of 120 cards split randomly between two players. The two players would ante a card, fight a duel over the ante, and repeat until they got bored. They often took a long time to get bored; even then, Magic was a surprisingly addictive game. About 10 o'clock one evening, Barry “Bit” Reich and I started a game in the University of Pennsylvania Astronomy lounge, a windowless, air-conditioned room. We played continuously until about 3:00 A.M.—at least that's what we thought until we left the building and found that the sun had risen.

I knew then that I had a game structure that could support the concept of individually owned and tailored decks. The game was quick, and while it had bluffing and strategy, it didn't seem to get bogged down with too much calculation. The various combinations that came up were enjoyable and often surprising. At the same time, the variety of card combinations didn't unbalance the game: When a person started to win, it didn't turn into a landslide.

From Alpha to Gamma

Except for the card mix, little has changed about Magic since alpha. In alpha, walls could attack, and losing all your lands of a particular color destroyed the associated spells in play, but otherwise, the rules are much the same now as they were in the early stages of playtesting.

Moving from alpha to the beta version was like releasing a wild animal. The enjoyable game that was alpha now burst the confines of the duel to invade the lives of the participants. Players were free to trade cards between games and hunt down weaker players to challenge them to duels while gamely facing or cravenly avoiding those who were more powerful. Reputations were forged—reputations built on anything from consistently strong play to a few lucky wins to good bluffing. The players didn't know the card mix, so they learned to stay on their toes during duels. Even the most alert players would occasionally meet with nasty surprises. This constant discovery of unknown realms in an uncharted world gave the game a feeling of infinite size and possibility.

For the gamma version, new cards were added and many of the creature costs were increased. We also doubled the pool of playtesters, adding in a group with Strat-o-matic Baseball experience. We were particularly anxious to find out if Magic could be adapted for league play. Gamma was also the first version that was fully illustrated. Skaff Elias was my art director: He and others spent days poring over old graphic magazines, comic books, and game books searching for art for the cards. These playtest decks were pretty attractive for crummy black-and-white cardstock photocopies. For the most part, the cards were illustrated with serious pictures, but there were a lot of humorous ones as well. Heal was illustrated by Skaff's foot. Power Sink showed Calvin (of *Calvin and Hobbes*) in a toilet; after all, what is a toilet but a power sink? Berserk was John Travolta dancing in *Saturday Night Fever*. Righteousness pictured Captain Kirk, and Blessing showed Spock doing his "live long and prosper" gesture. An old comic book provided a Charles Atlas picture for Holy Strength, and a 98-pound weakling getting sand kicked in his face for Weakness. Instill Energy was Richard Simmons. The infamous Glasses of Urza were some X-ray glasses we found in a catalog. Ruthy Kantorovitz constructed a darling flame-belching baby for Firebreathing. I myself had the honor of being the Goblins. The pictures and additional players greatly added to the game atmosphere. It became clear that while the duels were for two players, the more players playing, the better the game was. In some sense, the individual duels were a part of a single, larger game.

Striking the Balance

Each playtest set saw the expulsion of certain cards. One type of card that was common in alpha and beta was rare in gamma and is now nonexistent: the type that made one of your rival's cards yours. Yes, Control Magic used to permanently steal a creature from your opponent. Similarly, Steal Artifact really took an artifact. Copper Tablet no longer even remotely resembles its original purpose, which was to swap two

creatures in play. (“Yes, I’ll swap my Merfolk for your Dragon. On second thought, make that my Goblins—they’re uglier.”) There was a spell, Planeshift, that stole a land, and Ecoshift, which collected all the lands, shuffled them and redealt them—really nice for the user of four or five colors of magic. Pixies used to be a real pain—if they hit you, you swapped a random card from your hand with your opponent. These cards added something to the game, often in the form of players trying to destroy their own creatures before their opponents took them for good or even trying to take their own lives to preserve the last shreds of their decks. However, in the end it was pretty clear that the nastiness this added to the game environment wasn’t worth the trouble, and no card should ever be at risk unless players choose to play for ante.

It was around this time that I began to realize that almost any decision made about the game would be opposed, often vehemently, by some players. The huge amount of dissent about what should and should not be part of the card mix has led players to make their own versions for playtesting—a significant task that involves designing, constructing, shuffling, and distributing about 4000 cards. Each of these games had its merits, and the playtesters enjoyed discovering the quirks and secrets of each new environment. The results of these efforts will form the basis of future Deckmaster games that use the structure of The Gathering while containing mostly new cards.

To Build a Better Deck

Playtesting a Deckmaster game is difficult. Probably the only games harder to playtest are elaborate, multiplayer computer games. After developing a basic framework for Magic that seemed fairly robust, we had to decide which of the huge selection of cards to include, and with what relative frequencies. Common cards had to be simple, but not necessarily less powerful, than rare cards—if only rare cards were powerful, players would either have to be rich or lucky to get a decent deck. Sometimes a card was made rare because it was too powerful or imbalancing in large quantities, but more often, rare cards were cards that were intricate or specialized—spells you wouldn’t want many of anyway. But these design guidelines only got us so far. The whole game’s flavor could change if a handful of seemingly innocent cards were eliminated or even made less or more common. When it came down to actually deciding what to include and what to do without, I began to feel like a chef obliged to cook a dish for 10,000 people using 300 ingredients.

One thing I knew I wanted to see in the game was players using multicolor decks. It was clear that a player could avoid a lot of problems by stripping down to a single color. For this reason, many spells were included that paralyzed entire colors, like Karma, Elemental Blast, and the Circles of Protection. The original plan was to include cards that thwarted every obvious simple strategy, and, in time, to add new cards that would defeat the most current ploys and keep the strategic environment dynamic. For example, it was obvious that relying on too many big creatures made a player particularly vulnerable to the Meekstone, and a deck laden with Fireballs and requiring lots of mana could be brought down with Manabarbs. Unfortunately, this strategy and counter-strategy design led to players developing narrow decks and refusing to play people who used cards that could defeat them flat out. If players weren’t compelled to play a variety of players and could choose their opponent every time, a narrow deck was pretty powerful.

Therefore, another, less heavy-handed way to encourage variety was developed. We made it more difficult to get all the features a player needs in a deck by playing a single color. Gamma, for example, suffered from the fact that blue magic could stand alone. It was easily the most powerful magic, having two extremely

insidious common spells (Ancestral Memory and Time Walk), both of which have been made rare. It had awesome counterspell capabilities. It had amazing creatures, two of the best of which are now uncommon.

Blue magic now retains its counterspell capability, but it is very creature poor and lacks a good way to do direct damage. Red magic has little defense, particularly in the air, but it has amazing direct damage and destruction capability. Green magic has an abundance of creatures and mana but not much more. Black is the master of anticreature magic and has some flexibility, but it is poorly suited to stopping noncreature threats. White magic is the magic of protection, and it is the only magic with common banding, but it has little damage-dealing capability.

Sometimes seemingly innocuous cards would combine into something truly frightening. A good part of playtest effort was devoted to routing out the cards that contributed to so-called “degenerate” decks—the narrow, powerful decks that are difficult to beat and often boring to play with or against. Without a doubt, the most striking was Tom Fontaine’s “Deck of Sooner-Than-Instant Death,” which was renowned for being able to field upward of eight large creatures on the second or third turn. In the first Magic tournament, Dave “Hurricane” Pettey walked to victory with his “Land Destruction Deck.” (Dave also designed a deck of Spectres, Mindtwists, and Disrupting Sceptres that was so gruesome I don’t think anyone was ever really willing to play it.) Skaff’s deck, “The Great White Death,” could outlive just about anything put up against it. Charlie Catin’s “Weenie Madness” was fairly effective at swamping the opponent with little creatures. Though this deck was probably not in the high-win bracket of the previous decks, it was recognized that, playing for ante, Charlie could hardly lose. Even winning only one in four of his games—and he could usually do better than that—the card he won could be traded back for the island and the two Merfolk he lost, with something extra thrown in.

In the end I decided that the degenerate decks were actually part of the fun. People would assemble them, play with them until they got bored or their regular opponents refused to play against them, and then retire the deck or trade off its components for something new—a Magic version of putting the champion out to stud. Most players ended up treating their degenerate decks much like role players treat their most successful characters: They were relegated to the background to be occasionally dusted off for a new encounter.

After the pursuit of sheer power died down, another type of deck developed: the Weird Theme deck. These decks were usually made to be as formidable as possible within the constraints of their theme. When Bit grew bored of his “Serpent Deck” (he had a predilection for flopping a rubber snake on the playing surface and going “SsssSsss” whenever he summoned a Serpent), he developed his “Artifact Deck,” which consisted of artifacts only—no land. It was fun to see the “Artifact Deck” go up against someone who used Nevinyrral’s Disk. But the king of weird decks was, without a doubt, Charlie Catin. In one league, he put together a deck that I call “The Infinite Recursion Deck.” The idea was to set up a situation where his opponent couldn’t attack him until Charlie could play Swords to Plowshares on a creature. Then he would play Timetwister, causing the cards in play to be shuffled with the graveyard, hand, and library to form a fresh library. Swords to Plowshares actually removes a creature from the game, so his rival has one less creature. Repeat. After enough iterations, his rival was bloated with life given by the Swords to Plowshares, having maybe 60 life points, but there were no creatures left in his deck. So Charlie’s Elves started in—59 life, 58 life, 57 life—and the curtain closes on this sad game. I still can’t think about this deck without moist emotional snorts. The coup de grace is that this league required players to compete their decks 10 times. And, because his games often lasted over an hour and a half, he received at least one concession.

Words, Words, Words

It was not just determining the right card mix that players and designers found challenging. This becomes increasingly clear to me as I participate in the never-ending process of editing the rules and the cards. As my earliest playtesters have pointed out (in their more malicious moods), the original concept for Magic was the simplest game in the world because you had all the rules on the cards. That notion is long gone.

To those who didn't have to endure it, our struggle for precision was actually rather amusing. My own rules discussions about card wordings were mostly with Jim Lin, who is the closest thing you will ever encounter to a combination rules lawyer and fire hose. A typical rule-problem session would go:

Jim: Hmm—there seems to be a problem with this card. Here is my seven-page rules addition to solve the problem.

Richard: I would sooner recall all the cards than use that. Let's try this solution instead.

Jim: Hmm—we have another problem.

[Repeat until...]

Richard: This is silly—only incredibly stupid and terminally anal people could possibly misinterpret this card.

Jim: Yes, maybe we have been thinking about this too long. If you're playing with that kind of person, you should find some new friends.

A specific example of something we actually worried about is whether Consecrate Land would really protect your land from Stone Rain. After all, the first says it prevents land from being destroyed and the second says it destroys the land. Isn't that a contradiction? It still hurts my head getting into a frame of mind where that is confusing. It is perhaps a little like wondering why anyone would give you anything for money, which is, after all, just paper.

But, then again, I could never tell what was going to confuse people. One of the playtesters, Mikhail Chkhenkeli, approached me and said, "I like my deck. I have the most powerful card in the game. When I play it, I win on the next turn." I tried to figure out what this could be; I couldn't think of anything that would win the game with any assurance the turn after casting. I asked him about it, and he showed me a card that would make his opponent skip a turn. I was confused until I read exactly what was written: "Opponent loses next turn." It was my first real lesson in how difficult it was going to be to word the cards so that no two people would interpret the same card in a different way.

The Magic Marketplace

Another thing I realized in the second year of playtesting really surprised me. Magic turned out to be one of the best economic simulations I had ever seen. We had a free-market economy and all of the ingredients for interesting dynamics. People valued different cards in different ways—sometimes because they simply weren't evaluating accurately but much more often because the cards really have different value to different players. For example, the value of a powerful green spell was lower for a person who specializes in black and red magic than for one who was building a deck that was primarily green. This gives a lot of opportunity for arbitrage. I would frequently find cards that one group of players weren't using but another group were treating like chunks of gold. If I was fast enough, I could altruistically benefit both parties and only have to suffer a little profit in the process.

Sometimes the value of a card would fluctuate based on a new use (or even a suspected new use). For example, when Charlie was collecting all the available spells that produced black mana, we began to get concerned—those cards were demanding higher and higher prices, and people began to fear what he could need all that black mana for. And, prior to Dave’s “Land Destruction Deck,” land destruction spells like Stone Rain and Ice Storm were not high-demand spells. This of course allowed him to assemble the deck cheaply, and after winning the first Magic tournament, sell off the pieces for a mint.

Trade embargoes appeared. At one point a powerful faction of players would not trade with Skaff, or anyone who traded with Skaff. I actually heard conversations such as:

Player 1 to Player 2: I’ll trade you card A for card B.

Skaff, watching: That’s a moronic trade. I’ll give you card B and cards C, D, E, and F for card A.

Players 1 and 2 together: We are not trading with you, Skaff.

Needless to say, Skaff was perhaps a bit too successful in his early duels and trades.

Another interesting economic event would occur when people would snatch up cards they had no intention of using. They would take them to remove them from the card pool, either because the card annoyed them (Chaos Orb, e.g.,) or because it was too deadly against their particular decks.

I think my favorite profit was turned during an encounter with Ethan Lewis and Bit. Ethan had just received a pack of cards and Bit was interested in trading with Ethan. Bit noticed that Ethan had the Jayemdae Tome, began to drool, and made an offer for it. I looked at the offer and thought it was far too low, so I put the same thing on the table.

Bit looked at me and said, “You can’t offer that! If you want the Tome you have to bid higher than my bid.”

I said, “This isn’t an offer for the Tome. This is a gift for Ethan deigning to even discuss trading the Tome with me.”

Bit looked at me in disbelief and then took me aside. He whispered, “Look, I’ll give you this wad of cards if you just leave the room for 10 minutes.” I took his bribe, and he bought the Tome. It was just as well—he had a lot more buying power than I did. In retrospect, it was probably a dangerous ploy to use against Bit—after all, he was the person who was responsible for gluing poor Charlie’s deck together once, washing a different deck of Charlie’s in soap and water, and putting more cards of Charlie’s in the blender and hitting frappé.

Probably the most constant card-evaluation difference I had with anyone was over Lord of the Pit. I received it in just about every playtest release we had, and it was certainly hard to use. I didn’t agree with Skaff, though, that the only value of the card was that you might get your opponent to play with it. He maintained that blank cards would be better to play with because blank cards probably wouldn’t hurt you. I argued that if you knew what you were doing, you could profit from it.

Skaff asked me to cite a single case where it had saved me. I thought a bit and recalled the most flamboyant victory I had with it. My opponent knew he had me where he wanted me—he had something doing damage to me, and a Clone in hand, so even if I cast something to turn the tide, he would be able to match me. Well, of course, the next cast spell was a Lord of the Pit; he could Clone it or die from it, so he Cloned it. Then each time he attacked, I would heal both of the Lords, or cast Fog and nullify the assault, and refuse to attack. Eventually, he ran out of creatures to keep his Lord of the Pit sated and died a horrible death.

Skaff was highly amused by this story. He said, “So, when asked about a time the Lord of the Pit saved you, you can only think of a case where you were playing somebody stupid enough to clone it!”

Dominia and the Role of Role Playing

Selecting a card mix that accommodated different evaluations of the cards wasn't enough; we also had to develop an environment in which the cards could reasonably interact. Establishing the right setting for Magic proved to be a central design challenge. In fact, many of our design problems stemmed from an attempt to define the physics of a magical world in which duels take place and from building the cards around that, rather than letting the game define the physics. I was worried about the cards' relationship to each other—I wanted them to seem part of a unified setting, but I didn't want to restrict the creativity of the designers or to create all the cards myself. Everyone trying to jointly build a single fantasy world seemed difficult because it would inevitably lack cohesion. I preferred the idea of a multiverse, a system of worlds that was incredibly large and permitted strange interactions between the universes in it. In this way, we could capture the otherworldly aspects of fantasy that add such flavor to the game while preserving a coherent, playable game structure. Almost any card or concept would fit into a multiverse. Also, it would not be difficult to accommodate an ever-growing and diverse card pool—expansion sets with very different flavors could be used in the same game, for they could be seen as a creative mingling of elements from different universes. So I developed the idea of Dominia, an infinite system of planes through which wizards travel in search of resources to fuel their magic.

In its structured flexibility, this game environment is much like a role-playing world. I don't mean to suggest that this setting makes Magic a role-playing game—far from it—but Magic is closer to role playing than any other card or board game I know of. I have always been singularly unimpressed by games that presumed to call themselves a cross between the two because role playing has too many characteristics that can't be captured in a different format. In fact, in its restricted forms—as a tournament game or league game, for example—Magic has little in common with role playing. In those cases, it is a game in the traditional sense, with each player striving to achieve victory according to some finite set of rules. However, the more free-form game dueling with friends using decks constructed at whim embodies some interesting elements of role playing.

Each player's deck is like a character. It has its own personality and quirks. These decks often even get their own names: “The Bruise,” “The Reanimator,” “Weenie Madness,” “Sooner-Than-Instant Death,” “Walk Into This Deck,” “The Great White Leftovers,” “Backyard Barbeque,” and “Gilligan's Island,” to name a few. In one deck I maintained, each of the creatures had a name—one small advantage to crummy photocopied cardstock is the ease of writing on cards. The deck was called “Snow White and the Seven Dwarves,” containing a Wurm named Snow White and seven Mammoths: Doc, Grumpy, Sneezy, Dopey, Happy, Bashful, and Sleepy. After a while I got a few additional Mammoths, which I named Cheesy and Hungry. There was even a Prince Charming: my Veteran Bodyguard.

As in role playing, the object of the game in the unstructured mode of play is determined largely by the players. The object of the duel is usually to win, but the means to that end can vary tremendously. Most players find that the duel itself quickly becomes a fairly minor part of the game compared to trading and assembling decks.

Another characteristic of Magic that is reminiscent of role playing is the way players are exploring a world rather than knowing all the details to start. I view Magic as a vast game played among all the people who buy decks, rather than just a series of little duels. It is a game for tens of thousands in which the designer acts as a gamemaster. The gamemaster decides what the environment will be, and the players explore that environment. This is why there are no marketed lists of cards when the cards are first sold: Discovering the cards and what they do is an integral part of the game.

And, like a role-playing game, the players contribute as much to an exciting adventure as the gamemaster. To all the supporters of Magic, and especially to my playtesters, I am extraordinarily grateful. Without them, if this product existed at all, it would certainly be inferior. Every one of them left a mark, if not on the game itself, then in the game's lore. Any players today who have even a tenth of the fun I had playing the test versions with them will be amply pleased with Magic.

MAGIC DESIGN: A DECADE LATER

by Richard Garfield (written in 2003)

Magic and the trading card game industry have undergone a lot of changes since the time I wrote those design notes. In the meantime, Magic has grown stronger with each successive year—as the game itself is improved and as more people are brought into trading card games from products such as Pokémon and Yu-Gi-Oh!

It is difficult for people these days to appreciate how little we knew about the game design space we were entering in the early nineties. My design notes failed to mention what, in my mind, is the strongest sign of that—after describing the concept of a trading card game to Peter Adkison, I concluded with the cautious statement, “Of course, such a game may not be possible to design.” It is hard for me to imagine that state of mind today, in a world where trading card games have reached every corner and are a part of almost every major entertainment property. This is a world where trading card games have left their mark on all areas of game design, from computer games to board games, and where trading card games have directly inspired games ranging from trading miniature games to trading tops games. This is a world where Jason Fox, from the comic strip *Foxtrot*, complained that a deck of cards coming with only four aces was some sort of ploy to get people to buy expansion kits.

That could be left as the end of the story; Magic was designed—as the design notes of a decade ago portray—and 10 years later, it was still going strong. But this leaves out a large part of the story because Magic has been anything but a static game since then. The changes and improvements to Magic warrant design notes of their own.

First and Foremost—A Game

One thing that might look arcane in my notes to people who know something about the game market is my reference to the form of game that Magic launched as a “trading card game,” rather than a “collectable card game.” I still use TCG rather than CCG, which became the industry standard despite my efforts from its earliest days. I prefer “trading” rather than “collectable” because I feel it emphasizes the playing aspect rather than the speculation aspect of the game. The mindset of making collectables runs against that of making games—if you succeed in the collectable department then there is a tendency to keep new players out and to drive old ones away because of escalating prices. One of the major battles that Magic fought was to make it perceived principally as a game and secondarily as a collectable. Good games last forever—collectables come and go.

This was not merely theoretical speculation—Magic's immense success as a collectable was severely threatening the entire game. Booster packs intended to be sold at a few bucks were marked up to 20 dollars

in some places as soon as they hit the shelves. While many people view this time as the golden age of Magic, the designers knew that it was the death of the game in the long run. Who is going to get into the game when it was immediately inflated in price so much? How many people would play the game if doing so was wearing holes in some of their most valuable assets? We might be able to keep a speculation bubble going for a while, but the only way Magic was going to be a long-term success—a classic game—was for it to stand on its game play merits, not on its worthiness as an investment.

During “Fallen Empires,” the fifth Magic expansion, we finally produced enough cards that the speculative market collapsed. The long-term value of Magic could perhaps thrive, but it wouldn’t immediately price itself out of the reach of new players before they got a chance to try it. There was an inevitable negative patina that Magic got for a while, and Fallen Empires still has, but from this point on, Magic was sinking or swimming on its game merits. Fortunately, Magic turned out to be a strong swimmer.

Binding the Unbounded

The part of my notes that, I believe, reveals my biggest change in thinking over the last decade is the statement that in the future we would publish other games with mechanics similar to Magic. What I was referring to is what became Ice Age and Mirage, two expansions for Magic. Why did I think these would be entirely new games, rather than what they ended up being—expansions for the main game?

We all realized from the start that we couldn’t just keep adding cards to Magic and expect it to stay popular. One reason for that is that each successive set of cards were a smaller and smaller percentage of the entire pool of cards, and so they would necessarily have less and less impact on the whole of the game. This was illustrated vividly by players of Ice Age talking about how the entire set introduced two relevant cards to the game. One can imagine how the designers felt—working for years to make Ice Age a compelling game to have it boil down to a mere two cards. Another, perhaps more important reason, is that new players wouldn’t want to enter a game where they were thousands of cards behind, so our audience would inevitably erode.

Initially we saw two solutions to this problem:

1. Make cards ever more powerful. This is a route many trading card game makers followed, and one I greatly dislike. It feels like strong-arming the players to buy more and more rather than really providing them more game value. But it would bring new players in because they wouldn’t need the obsolete old cards.
2. Eventually conclude Magic: The Gathering and start a new game—Magic: Ice Age, for example. I advocated this approach because I believed we could make exciting new game environments indefinitely. When one set was finished, players wouldn’t be forced to buy into the new game to keep competitive, they could move on if they wanted a change, and new players could begin on equal footing.

When it actually came time to do Ice Age, it was absolutely clear that players would not stand for a new version of Magic, so we had to think of something else. Additionally, we were also worried that fragmenting the player audience was a bad idea; if we made a lot of different games, people would have a harder and harder time finding players.

The solution we found was to promote different formats of game play, many of which involved only more recent sets of cards. Today there are popular formats of play that involve only the most recently published cards, cards published in the last two years, and cards published in the last five years, in addition to many

others. While this does fragment the player base—because you might not be able to find players who play your format—it is less draconian than different games because you can apply your cards to many different formats over time. This was a far more flexible approach than the first because it didn't command players to start fresh; it allowed them to, and it allowed new players to join the game without being overwhelmed.

Trading Card Games Are Not Board Games

I used to believe that trading card games were far more like board games than they are. This is not surprising because I had no trading card games before Magic to draw examples from, and so I was forced to use the existing world of games to guide my thinking on TCGs. A lot of my design attitudes grew from this misconception. For example, my second trading card game was designed to be best with four or more people, and it took several hours to play. These are not bad parameters for a board game, but trading card games really want to be much shorter because so much of the game is about replaying with a modified, or entirely new, deck.

In a similar vein, I used what I saw board game standards to be when it came to rules clarifications. It was common in board games to find that a different group played a slightly different way or had house rules to suit their tastes. With board games, different interpretations of the rules and ways of play were not a major problem because players tended to play with fairly isolated groups. This led me to be quite antiauthoritarian when it came to the “correct” way to play. It turned out that a universal standard for a trading card game was far more necessary than a board game because the nature of the game form made the interconnectivity of the game audience far greater.

This meant that we had to take more and more responsibility for defining the rules and standards of play. In some ways, this is analogous to being forced to construct the tournament rules for a game. The rules to bridge are not that complex, but when you write out the official tournament rules—really try to cross the Ts and dot the Is—you have a compendium.

I had also hoped that players could moderate their own deck restrictions. We knew that certain card combinations were fun to discover and surprise someone with, but they were not fun to play with on an ongoing basis. So we figured players would make house rules to cover those decks and the responsible cards. The highly interconnected nature of Magic made it unreasonable to expect that, however, because every playgroup came up with a vast number of restrictions and rules, and they all played with each other. This meant we had to take more responsibility in designing the cards and, when necessary, banning cards that were making the game worse.

The Pro Tour

All this precision invested in the design of the rules and cards made Magic a surprisingly good game to play seriously. We began to entertain ideas of really supporting a tournament structure with big money behind it—big enough that players could, if good enough, make a living off of playing Magic. This was a controversial subject at Wizards of the Coast for a while, the worry being that making the game too serious would make it less fun. I subscribed fully to the concept of a Pro Tour, thinking of how the NBA helped make basketball popular and didn't keep the game from being played casually as well.

The Pro Tour had an almost immediate effect. Our players rapidly became much better as the top-level ones devoted time to really analyzing the game and as that game tech filtered down through the ranks. Before the Pro Tour, I am confident that I was one of the best players in the world, now I am mediocre at best.

Now there are thousands of tournaments each week, and many players have earned a lot of money playing Magic, some in the hundreds of thousands of dollars. At the last World Championship, there were 56 countries competing. There is a never-ending buzz of Magic analysis and play as players attempt to master the ever-changing strategic ground of Magic. I believe this is a major part of Magic's ongoing popularity—if even a small group of people take a good game very seriously, there can be far-reaching effects.

Magic Online

Online Magic didn't come into its own until last year. For a long time, I have wanted to see an online version of Magic that duplicated real-life Magic as closely as possible. That is, the online game would connect people, run the games and the tournaments, and adjudicate rules, but little else. At first, we tried to form partnerships with computer game companies to do this, but our partners always had other ideas about how to do computer Magic. Eventually we hired a programming studio to do it our way, and now we have Magic Online.

One of the striking things about Magic Online is that we use the same revenue model as in real life. Despite exhortations to use a subscription model, we chose to sell virtual cards, which you could trade with other players online. This allows players to buy some cards and then play them indefinitely with no further fee—as in real life.

It was important to us that we not make it a better deal playing online than off—we wanted it to be the same. That is because we feel the paper game contributes a lot to Magic's ongoing popularity, and it could be threatened if many of its players go to the online game.

For this reason, one of the prime targets for the online game was going to be lapsed players. Many studies had been done on how long people play Magic and why they leave the game, and for the most part, they didn't leave because they were bored with the game; they left because they had life changes that made it more difficult to play, for example, getting jobs or having kids. These players would potentially rejoin the game if they could play from their own home on their own hours.

Magic Online is still a bit too young to be sure about, but it appears to have acquired a dedicated sizeable audience of players without hurting the paper game. Many of the players are formerly lapsed players, as we had hoped.

The Next 10 Years

Who knows what the next decade will bring? Ten years ago, I had no clue at all; it was an exciting time, and we were riding a roller coaster. Now I am more confident. I believe that Magic is fairly stable and that there is every reason to believe that it will be around and as strong in another 10 years. At this point, it is clear that Magic is not a fad, and as many new players are coming in each year as are leaving the game.

Certainly Magic has stayed fresh for me. I get into the game every few months—joining a league, constructing a deck, or perhaps preparing for and participating in a tournament. Every time I return I find the game fresh and exciting, with enough different from the previous time to keep me on my toes but enough the same that I can still exploit my modest skills at the game. I look forward to my next 10 years of the game.

2. Structure

After the foundation is in place and seems to function, it is time to move on to structure. The best technique for doing this is to prioritize what is most essential to the game. In my FPS prototype, some structural elements I added were the three action options: (1) number of spaces a unit could move, (2) procedures for turning, and (3) hit and miss rules for shooting. Our army men were moved and turned on the table as mock units using the rules.

These experiments solidified some ideas about moving and shooting and caused other ideas to be dismissed, which resulted in a very crude system for simultaneous movement and the basics of shooting. I also considered adding rules about movement and starting points, as well as assigning a turn order to the players.

Think of it this way: You have built the foundation, and now you need to build the framework for your game. It is not a matter of what you think is coolest or most saleable; it is about constructing a skeletal structure that can support the rich and varied feature set that will be your finished game. What you need to do first is decide which rules are essential and which are features that those structural elements have to support. Your gameplay visualization should help you make these decisions.

At this point in the construction of the FPS prototype, the movement and shooting foundation begged for the structure of a scoring system and unit hit points. As I added these elements, my crude movement and shooting system was retested with them in place. The tests illuminated problems that could only be seen with the system in motion. The whole system was revised to address the problems. At this point, the system was still messy and ill defined. Nothing had been written down. There were open questions everywhere. However, the system was basically functional.

When working through this, keep in mind the distinction between what are features and what are rules. Features are attributes that make a game richer, like adding more weapons or new vehicles or a nifty way to navigate the space. Rules are

modifications to the game mechanics that change how the game functions, such as winning conditions, conflict resolution, turn order, etc.

You can add rules without adding features, but you can never add a feature without changing or adding rules. For example, if you added a new type of laser gun to your game, the rules would dictate how this gun could be used, what damage it would do, and how it would relate to all aspects of the game. One new feature might introduce 10 or more new rules to support it. As you modify your game, you will be constantly tweaking the rules to enhance gameplay and accommodate a growing feature set.

Your best strategy for adding structure is to focus on rules first and features later. Rules, by their very nature, tend to be inextricably linked to the core gameplay, while features tend to be peripheral. That is a generalization, but if you keep it in mind, it will help you to structure the development of your game.

3. Formal details

The next step is to add the necessary rules and procedures to the system to make it into a fully functional game. Focus on what you know about formal elements to decide what your game needs. Is the objective interesting and achievable? Is the player interaction structure the best choice? Are there rules or procedures that you wanted to add, but they were not part of the core mechanic? The trick is to find an appropriate level of detail to add. Beginning game designers typically add too much. The art of game design often involves paring a bunch of feature ideas down to a small, important set of features that all contribute to meeting your experience goal for the game.

At this point in the development of the FPS prototype, I added the hit percentage, health, and scoring. Many other ideas were considered, including mines, shields, vehicles, mechanisms for hiding, and more. However, I scrapped all of them and focused on rules affecting the central gameplay, rather than a set of new features that I believed would create the most interesting game. How did I decide on some

elements and not others? It was a creative judgment backed by input from my playtesters.

One way to add formal details efficiently is to isolate each new rule and test it individually. If you feel the game cannot function without this rule, then leave it in the game and add another rule. But do not overuse this privilege. Not every rule is critical, and the less you add, the cleaner your skeleton. A lot of what you consider rules are probably features. Try to draw a clear distinction, and keep your core rule set as clean as possible.

Test each rule, then remove it, and add another rule, and test it. It will be clear that some of the rules are optional and others must be included in the game if you are to continue to expand the gameplay. This is a litmus test. If you can continue to build out the game without a specific rule, no matter how amazing that rule seems, you should leave it out. You can always add it later, but it should not be included at this early stage.

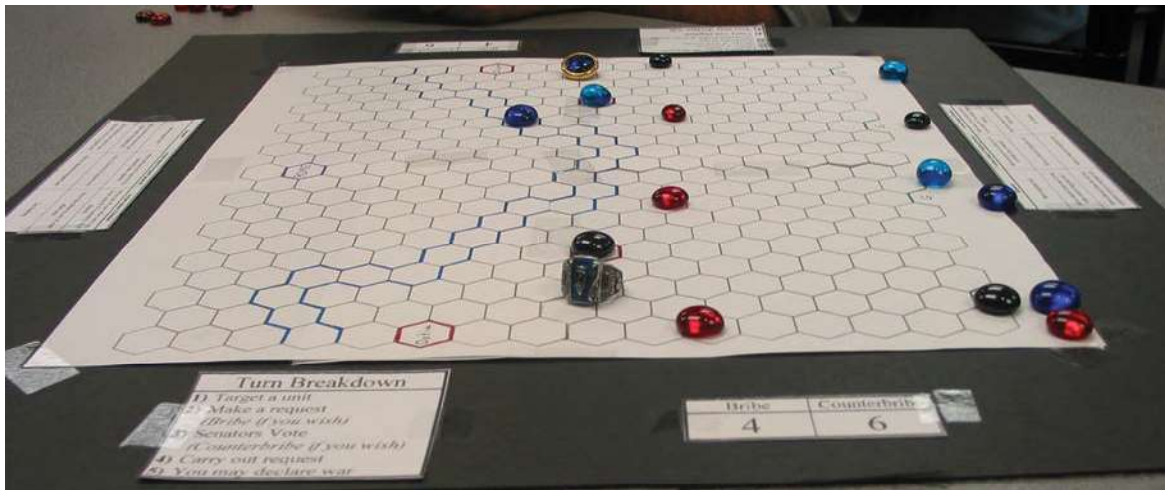
4. Refinement

At this point in the process, the prototype is a playable system, although it might still be somewhat rough. By experimenting and tweaking, the play system will become more refined. The play experience

created by the game will flow. Instead of questioning the fundamentals of the game (and possibly thinking it will never work), you will switch to questioning the smaller details, and, of course, the big question: Is your game compelling? If not, what will make it so? This process of refinement can continue for a number of iterations.

During refinement is also the time to add all those great ideas for features that have come up while testing but were not really essential. I went back to those ideas about mines and teleport pads for my FPS at this point. Again, be careful not to get ahead of yourself. It is tempting to add five new features, create a bunch of rules to support those features, and then start playing, but this blurs your view of the game. It becomes difficult to tell which features are making the game more fun to play and which are causing problems.

To avoid this, rank your features in terms of necessity. Then introduce and test each one. Test how it affects overall gameplay, and then remove it. This might seem cumbersome, but it will keep your game structure from getting convoluted. If you add too many features too early, you will find yourself losing your grasp on what the game is about. I have seen this happen over and over again with beginning



7.13 Physical prototype with procedures outlined

designers, and this is why I caution you to postpone the pleasure of creating the ideal game from the outset, and instead recommend that you focus on what is needed step by step.

As you do this, you will discover that some rules and features that seemed like great ideas actually diminish the playability, while others that appeared dull add a whole new dimension to the player experience. You can only know this by testing each one in a controlled environment without the interference of other features. After testing each new twist, write down an analysis. Be sure to use your playtesters, and incorporate their feedback into your analysis. They are your eyes and ears. You might love a rule or feature so much that you are blind to its flaws. Trust your testers.

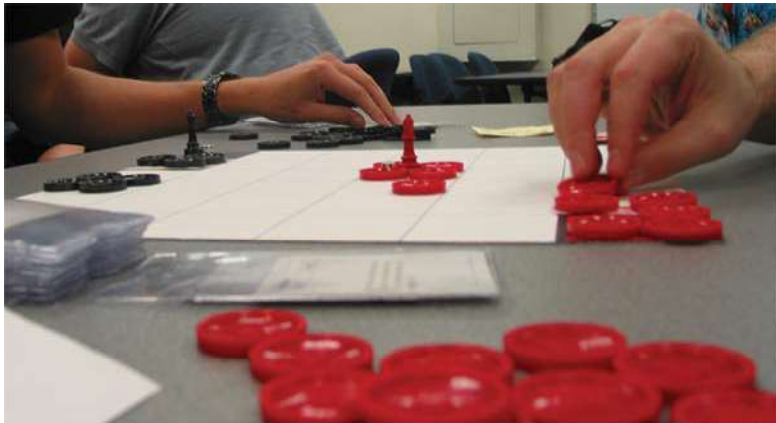
Exercise 7.9: Prototype Your Own Game

Use what you have learned to create a paper prototype of the game idea you described in Exercise 6.8

on page 190. This is a hard task. Break it down into the iterative steps described on pages 217–218 and 231–233 (i.e., foundation, structure, formal details, refinement). If you get stuck on a step, just take your best guess and move on. With prototyping, you always have room to iterate.

Refining Your Visualization

As you prototype, you will probably wind up changing the relationships of the various activities in your game. I recommend refining your gameplay visualization as you go along so that you can see how your changes affect the overall flow of the system. As you analyze and refine the structure, you will be able to see if there are activities that have little or no payoff for the player or other activities that are overvalued. You will want to make sure that the core actions have significant impact for the player and that each is there for an appropriate reason. I will talk more about balance and tuning of your game in [Chapters 10](#) and [11](#).



7.14 More physical prototype examples



MAKING THE PHYSICAL PROTOTYPE BETTER

The prototype you have created may or may not be very playable. Parts might be out of balance, and rules might conflict. Your game might also feel slow or disjointed. Some beginning designers become discouraged at this point and walk away. They feel that their game is hopeless, and the only solution is to start from scratch with a new game idea.

This might be true, but before you take such drastic measures, it is good to go back to your core game mechanics. Strip away all the additional rules and then reintroduce them one by one in an attempt to isolate the problem. In doing this, you will come to understand how each rule and feature actually fits into the system. Some features and rules might seem innocuous at first, but as you add and remove them, it will become apparent how they can throw the whole system out of balance.

Your game is a complex system, and specific elements might interact with others to produce a result that is unexpected. Your job is to systematically determine the problems and experiment with solutions until you solve them. Sometimes this can be a painstaking process, as you rip apart rules and rebuild them over and over again, but it is the only way to truly figure out what part of your game is actually broken.

When you get to the point where you are absolutely certain that your prototype is both playable and fun, then you are ready to start all over again. Yes, that's right. Just because your game is good does not mean it is brilliant. Before you move on to the next stage, though, you will want a great prototype. And even if it is great, there might be a way to make it better.

BEYOND THE PHYSICAL PROTOTYPE

Now that you have experimented with physical prototyping and iterated several designs, you are probably beginning to get a good sense of what it means to be a game designer. The physical prototype of your original game concept is working, though perhaps not perfectly. At this point you will want to do some playtesting of your prototype, as discussed in [Chapter 9](#).

But physical prototyping is only the first in a long set of steps to completing a functional digital game. You and your team can use the physical prototype as the blueprint for a software prototype. Because you have spent a lot of time thinking through the core mechanics and most important features of your game by building a physical prototype, articulating those mechanics will be much simpler.

Obviously, taking your physical prototype from a physical to a digital design will change the nature of how players access the game, but the core

mechanics of the system are still valid. For example, in the FPS prototype, I could lay out the arena, spawning points, ammo, first aid, etc., in the software prototype exactly as I had them in the physical prototype. The programmers would implement a real-time system for movement and shooting, making my card system obsolete, but the basic gameplay would remain intact, and the map I created would provide a good design guide.

Some core challenges you will find in translating your physical prototype to a digital design are in the controls and interface for the target system. Rather than players moving their army men on the grid, now you have to provide a control map for a keyboard and mouse, a proprietary controller, or whatever other input device for which you are designing. Also, you have to design a visual display of the game environment that is compatible with the platform you are targeting. [Chapter 8](#) goes into more detail on this process.

CONCLUSION

Creating a physical prototype is a critical step in the design of your original game concept. It will save your team tremendous amounts of time because everyone will have a clear understanding of the game you are making. In addition, a physical prototype will enable you to focus your creative energy on

the game mechanics, without becoming distracted by the production and programming process. And most importantly, making a prototype gives you the freedom to experiment—and through experimentation comes innovation.

DESIGNER PERSPECTIVE:

JAMES ERNEST

President, Cheapass Games

James Ernest is a prolific designer of tabletop games, including Kill Doctor Lucky, Lords of Vegas, Button Men, Diceland, Give Me the Brain, Lord of the Fries, Falling, Brawl, and Fightball. He has also done freelance design work for several major game companies, including Hasbro and Microsoft.



On getting into the game industry:

I met some of the people at Wizards of the Coast in 1993, shortly before they released *Magic: The Gathering*. I worked with Wizards on support material for that game, and I designed new games to submit to them for publication, with limited success. Eventually, with a backlog of unpublished designs, I started my own game company in 1996.

On favorite games:

I like simple games, whether they have deep strategy or not. For example, I'm a sucker for casino games of all types. Judging by the time I spend playing them, my five favorite games would have to be poker (by a wide margin), *Diceland*, blackjack, *Dungeons & Dragons*, and my family's unique version of cutthroat pitch. I also play plenty of computer games, mostly of the casual, puzzle, and arcade variety. Poker tops the list because of several factors: I can make money at it, so I'm pretty fascinated by that. The rules are incredibly simple, but the strategy is deep. Hands are only a couple of minutes long, so in a sense you're always getting a fresh start. And it has strong components of both mathematical and psychological strategy, so I can focus on whichever one I'm most interested in at the time.

On game influences:

Magic: The Gathering inspired me both to imitate the things it did right and to learn from its mistakes. Before *Magic* came out, I'd given very little thought to formal game design (though I'd written a few games, including a chess variant). Being closely associated with *Magic* in its early stages, I became aware that games were something you could design for a living. One of the most inspiring things about *Magic* was its original format: To imitate that success, I continue to experiment with new formats. I haven't made a hit yet, but at least I deserve a little credit for trying. I'm also a fan of popular European games like *Settlers of Catan* and *Puerto Rico*, which I like for their structure and balance, and traditional and casino games, which prove that it doesn't take many rules to create a compelling game.

On design process:

Many designers seem to invent game mechanics first, then theme, or at least they give preferential treatment to mechanics. In my experience, if a game is going to have a theme or a story, you need to settle on

that part first because it's so much harder to do it last. I have been in too many design meetings (for my own games and others) where we have a perfectly functional game but now need to come up with a name or theme. Those sessions are awful, and it's often impossible to come up with the right answer. Conversely, if I know the theme of a game, I have absolutely no trouble coming up with mechanics that deliver on it. In fact, a good theme usually suggests new mechanics that I would otherwise never have considered.

On prototypes:

Even when designing computer games, I try to build a paper prototype if it's at all possible. I need to put the game in front of real players for several rounds of quick, iterative testing, and paper prototypes are much quicker to modify. On the paper side, I try to prototype every meaningful element of the game as soon as I can. For example, when creating *Pirates of the Spanish Main*, a miniatures game for Wizkids, I built modular, miniature pirate ships exactly the size of the final models, using Lego bricks. The result gave us a very good idea of what would and wouldn't work. Having real models, rather than generic pieces, made testing and refining the game much easier.

On balancing *Diceland*:

Diceland is one of the most challenging games I've ever designed, and it took about six years between the original concept and the final product. It's basically a miniatures game that uses paper dice, and it deals with damage, range, and distance in very abstract ways. Each character is an eight-sided die, and each face of the die represents the character in a different state, such as wounded, healthy, blinded, in command, etc. A core design challenge was to balance light, nimble fighters against large, bulky ones in a way that made sense. The solution involved mapping the surface of the die, controlling the damage path, and understanding all the relationships between sides. When a character moves, he tips from one side to an adjacent side. A similar thing happens when he takes damage, though not always in the same direction. Understanding and mapping the "recovery" moves, that is, those that move a character from a weaker to a stronger face, gave me the control necessary to give smaller fighters a real sense of agility.

Advice to designers:

When you're new to a discipline, everything will feel like a lot of work, and something that was a lot of work can be hard to let go. Don't get so attached to your work that you can't be honest about it. Change everything if it needs changing, even down to the roots. Also, become addicted to simplicity. It's always tempting to fix a game by adding rules, but it's better (and much harder) to take bad rules away.

A lot of game designers will tell you to borrow liberally from existing games, and you can, but only when you understand what you're doing. Don't decide that your game uses seven cards just because your favorite game also uses seven cards. Decide because seven cards is the right number for your game.

It's easy to copy what you see in the market, but it's more productive to look for what isn't there. You can try to write a game for a player you don't know, but your best target will always be yourself. That's why the sign in my office says, "Write the game you want to play."

DESIGNER PERSPECTIVE: KATIE SALEN

Co-founder and chief designer, Connected Camps

Katie Salen is a game designer, writer, and educator whose games include Squidball (2003), Big Urban Game (2004), Drome Racing Challenge, The Last Fax (2006), Forget Me (2006), Skew (2006), Cross Currents (2006), and Gamestar Mechanic (2011). She is the co-author of Rules of Play from MIT Press and the designer of Quest to Learn, an innovative public school in Manhattan focused on game-based learning in grades 6–12. Currently she is co-founder and chief designer of Connected Camps, a benefit corporation providing connected learning experiences that foster creativity, problem solving, collaboration, and interest-driven learning.



On getting into the game industry:

I fell into working with games via a project I ran with some students on the Texas Lottery. At the time, I was interested in thinking about how lottery tickets effectively functioned as formal, social, and cultural interfaces and quickly realized that games were an incredible platform for creating compelling, interactive experiences. I started making games as part of this work, met a bunch of interesting game designers in Austin and New York, and starting digging into a myriad of gaming subcultures, including machinima. My career as a game designer and writer on games grew from there.

On favorite games:

This is always a challenging question to answer because there are so many games I love. But if I were to name the games that have most influenced my thinking and work, Rez, Mafia, Guitar Hero, four square, and DDR would top that list. Each of these games totally transformed the way I think about designing play to transform how a player relates to their social and physical surroundings, and each has a particular interactive aesthetic to which I am drawn. I also am deeply affected by the cultures of production that have emerged around some of the games on the list and the way each of these games create performative spaces as part of their play.

On inspiration:

I tend to find design inspiration in particular moments of a game, rather than in a game as a whole. Sometimes the inspiration comes from a particular moment of play of a game that was unexpected but traceable to the game's design or from an especially elegant core mechanic. The exquisite feeling of the handholding

mechanic in Ico, for example, or the egalitarian structure of a race game like the New York Marathon gets me thinking about the kinds of experiences games can provide. Katamari Damacy inspired me on the level of a core mechanic that led to the invention of strange stories and worlds. SuperMario for the DS inspired me on the value of game balancing and the pure pleasure of failure. Because I was a high-level athlete into my postcollege years (volleyball), I find that I also look for inspiration in that experience, which was intensely competitive and wholly collaborative. As a player, I learned to respect what the game demanded from me; as a designer, I work to translate this feeling of respect into a kind of social contract binding player to player and player to game. I really feel like it is this state of mutual respect that makes game design so interesting.

On the design process:

I see game design as requiring a balance between a systematic analysis of constraints, understanding of precedent, and sheer imagination. Most often I begin by trying to define exactly what it is I want a player to experience—how I want them to feel, what physical movements or actions I want them to enact, in what ways they might interact with other players or contexts. I also think a lot about what kinds of meanings I want the game to express and where and by whom the game will be played. I explore core mechanics that fit with answers to these questions, doing physical or paper prototyping to gauge the effects of the mechanics, and then work with a team to embed those mechanics into a larger design system. Game ideas also sometimes start with an image—the Big Urban Game, for example, started with the image of giant bowling pins wandering through a city; the slow game Skew with an image of players running through stores scanning game pieces. I also rely heavily on brainstorming ideas with other designers and making lists and lists of kinds of experiences I'd like to create.

On prototypes:

Prototyping is a key part of my design process because it is the very best way to understand what your player will experience and to begin to see the kind of possibility space the game provides. I use many different prototyping methods: paper prototyping, physical prototyping, scenario writing, interactive prototyping, etc. Prototypes become the basis for playtesting, which I use throughout the entire design process. Sometimes the prototypes are incredibly simple—a few chits and cards used to model a core mechanic or balancing scheme; later in the process, especially with digital games, the prototypes can become quite complex and require a number of people to produce. In all cases, I use prototyping and playtesting to help me see what is and isn't working about the game, to explore unexpected outcomes, and to constantly assess the quality of experience the player is having.

On solving a difficult design problem:

I remember with the Big Urban Game, Frank Lantz and Nick Fortugno and I were struggling with the overall balance of the game. As a race game that took place between three teams over the course of five days, we needed to figure out how to make sure that the race felt dramatic throughout and that each team, each day, would have a chance to win, even if one of the teams took an early lead. We chose to add a “power-up” feature where anyone could show up to one of the race checkpoints and roll a pair of giant dice to help power

up their team's game piece. Formally, this feature did what we needed it to do—a team could come from behind if they received enough dice rolls, but the feature also did something else: It invited a whole other kind of player into the game. These were the super-casual players whose entire engagement with the game was a single role of the dice. Surprisingly, because the mechanic of the dice roll was embedded in a social context (the checkpoint where lots of players were hanging out), those players felt as much a part of the game as the hardcore players did. Designing a game where a single move was felt to be equally valid as days of play by players was pretty incredible, and it was an experience that continues to inform my work. If we were to redesign the game, we would have built on this feature even more.

Advice to designers:

Be open to the possibilities of history, diversity, and ideas that you feel matter. Prototype ideas rather than talking about them. Practice. Practice. Practice.

FURTHER READING

Buxton, Bill. *Sketching User Experiences: Getting the Design Right and the Right Design*. San Francisco: Morgan Kaufmann, 2007.

Henderson, John. "The Paper Chase: Saving Money via Paper Prototyping," [Gamasutra.com](https://www.gamasutra.com/view/feature/131099/the_paper_chase_saving_money_via_.php), May 8, 2006. https://www.gamasutra.com/view/feature/131099/the_paper_chase_saving_money_via_.php

NielsenNormanGroup. *Paper Prototyping: A How-to Training Video*. DVD available at (<https://www.nngroup.com/reports/paper-prototyping-training-video/>)

Sigman, Tyler. "The Siren Song of the Paper Cutter: Tips and Tricks from the Trenches of Paper Prototyping," [Gamasutra.com](https://www.gamasutra.com/view/feature/130814/the_siren_song_of_the_paper_.php), September 13, 2005. https://www.gamasutra.com/view/feature/130814/the_siren_song_of_the_paper_.php

Snyder, Carolyn. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Francisco: Morgan Kaufmann, 2003.