Magic Lessons: Designing and Balancing Game Objects

K. Robert Gutschera
Director of Development
Wizards of the Coast
robert.gutschera@wizards.com

I. Introduction

Historically, games have typically had relatively few types of pieces in them. Checkers has only one piece and a simple board; chess has six different pieces. A standard deck of cards has four suits (more or less equivalent for most games) each of which has thirteen cards arranged in order. More important, the player typically cannot pick which pieces to use: I cannot decide to show up to a chess match with three queens.

Some modern computer games follow this pattern. A typical first person shooter might have a single character type and a dozen weapon types, all of which are available to the players (say by reaching a spawn point) — not too different from chess. But many computer games are far more complex. An RTS can have dozens of units or more (especially if you include upgrades and build paths), and a player can choose which units to use. An MMRPG may have hundreds of character abilities to choose from and thousands of items to equip. When I talk about “objects” I mean things of this sort: parts of a game that a player can select from when building a strategy.

Game designers want to create rich and interesting play environments. Individual players, however, want to win. So players will try to collapse the decision tree down to a single winning strategy by “breaking” the game: finding a small number of strategic choices that dominate other choices to the point of driving them out. Game designers need to prevent this by balancing the different choices so that enough of them (not necessarily all — in fact, it’s often desirable to have some “bad” choices) are viable, and the game remains strategically rich.

A great many computer games have wrestled with this problem (arguably, every game must deal with it to some extent), and games with large numbers of objects suffer the most. But there is also a category of non-electronic games that shares

The author would like to thank Richard Garfield and Cormac Russell for comments on an earlier draft of this paper.

the problems caused by player choice among many objects: collectable object games. By collectable object games, I mean trading card games like Magic: the Gathering, or collectable miniatures games like Warhammer 40K or Mage Knight. I'll sometimes refer to these non-electronic games as "paper games" even though in the case of miniatures they are metal or plastic. Similarly, when I talk about "computer games" I mean to include games on consoles and handhelds as well as PC games (although for practical reasons many of the electronic games that have many digital objects and much player choice are in fact PC games).

In this paper, I will discuss some of the viewpoints and techniques used by designers of (non-electronic) collectable object games with an eye towards things that might be interesting to designers of computer games. The next section will compare the worlds of computer game design and paper game design. Then I'll break game design up into different phases, followed by a discussion of some of the basic logic of costing of game objects. The last section will survey a number of practical tips for costing and balancing game objects.

Many of my examples are drawn from Magic, Starcraft, and World of Warcraft, because they are well-known, but the topics discussed should apply generally to games where players choose from among a large pool of objects: RTS games, MMRPGs, other competitive object games like Kart Rider, and online games based directly on the paper collectable games model like Pax Nora or ChronX.

II. Computer vs. Paper

If I'm going to try to apply lessons learned in the paper world to the digital one, the two had better be close enough that the lessons are relevant. I won't try to argue for that directly; instead, the individual points I make will, I hope, make that case. This section will briefly — in the interests of full disclosure — point out some of the ways that the two design problems are different. These differences are important enough to matter, but not enough to doom the enterprise.[1]

The most obvious difference with the paper design process is how much cheaper it is. The first playable prototype can be made very early on with a paper game. And if the designer wants to change a rule, he says to the person opposite him "let's try playing this way!" and they do that. If he wants to change a game object, he takes a pen and crosses out some bit of game data on a card or slip of paper and writes in the new value. This is enormously powerful, and allows a lot more time and effort to be put into the game design than into things like waiting around for a new build to show up. In other words, paper designs are much easier to prototype and test.[2]

[1] Of course, there are many other differences besides the ones I mention. For example, paper games tend to have more expansions than computer games. Also, the ease of creating a new virtual item that's just a stat tweak of an old one, and the difficulty of coding one that has an entirely new game mechanic, means that some computer games that have very large numbers of objects have proportionately fewer unique mechanic objects than a comparable paper game.
[2] Of course, some differences in ease of testing have to do with issues other than paper vs. electronic. For example, it's easier to test a one or two player game than a multiplayer one, and

Tied in with this question of prototyping is that of interface. Interface is a challenge for both paper and digital games, but a much bigger one for the latter. One can imagine a good paper game with a bad interface (e.g. a poorly designed cardface) — also, as a designer, one can work quite productively on the game while its interface is in bad shape. For many computer games, the interface and the enjoyability of the game are so tightly tied together that it can be quite hard to separate them.

Another difference between the paper and digital worlds that's tied in with the difficulty of coding is what percentage of design ideas are implementable at all. In a game like Magic, most new card ideas are at least possible to make (a few might not be due to technical problems with the rules system). With a computer game, a great many design ideas might not be practical to implement. So a large amount of Magic design work can be done before bringing in the technical experts. This would probably not be a good idea with a computer game.

Paper games have both the advantage and disadvantage that their rules are written largely (although not entirely, as anyone who has read the Magic rules knows) in English, rather than in computer code. On the plus side, the game can't really "crash"... if something is poorly written, players will make some reasonable decision about what to do and move forward. English being less than precise, though, means that paper game rules will contain ambiguities: a mild annoyance in casual play, but a more serious problem in highly competitive play. Efforts to be extremely precise with rules (and remember, if there are many different gameplay objects, there are many rules, and many rules interactions) can lead to unpleasant legalese. Computer games can hide complexities in code that players never see.[3]

Although paper games are on balance surely easier to "debug" than electronic ones — it's easier to catch a typo than a subtle code bug — they are harder to patch. If a card has a misprint, it has it forever. If a unit in Starcraft needs to cost 20 more crystal, the next patch can make it so, in both gameplay and tooltip; declaring that a card with a printed cost of 4 should be played as if it costs 6 (by whom? how do they know? what if they meet someone else who doesn't know?) is imposing such a burden on the user that it's arguably never worth it.

All of these factors influence an important large-scale decision: how much effort to put into game design versus coding (think for the moment of things like laying out the rulebook or designing the graphics for the card face as the paper

---

it's easier to test when a complete game takes twenty minutes than when it takes an hour (or several months!). And some things come down on the plus side of the ledger for a computer game—you can get thousands of testers in an open beta, and you might be able to code a good single player test mode even for a multiplayer game.

[3] As anyone who's played Monopoly with two different play groups knows, the problems of English come up even in relatively simple games. But those problems are orders of magnitude larger in games with many objects, each of which has its own associated rules.

equivalent of coding) that game design. It's not surprising that game design takes up a higher percentage of the effort in the paper world — the "coding" part is just plain easier — but given that a bit of time spent on game design can save a great deal of time coding, it's perhaps surprising how big that difference is. Wizards of the Coast is not a large company by the standards of the computer industry, but we have around 40 people devoted full-time to game design. They don't do graphic design, or layout, or printing... they just work on gameplay. Even a small trading card game will have 4 or 5 designers working on it, and Magic has many more. A typical computer game will have a much larger overall budget, but probably has fewer people devoted to gameplay, and many of them will have other duties. Designers of games that involve a large number of collectable objects, though, are discovering this may not always be the best mix.

So although the demands of coding might push one to spend almost all one's resources on it, the fact that coding is so expensive should make one look for ways to substitute cheaper (but not easier!) things for it... like careful planning. I don't mean this to be taken as an argument for long spec documents and no prototyping. On the contrary, I think there's no substitute for having a playable version of the game to see if it's fun or not. But once the basics of the gameplay are there, careful thought about balance and costing can save a great deal of time and effort over the more usual "keep trying stuff and adjust the broken bits".

III. Design vs. Development

At this point we run smack into a terminology problem. The game design process can be split into two parts: designing the initial game, and tuning that design. Either of those things would be called "game design" in the computer world, and they wouldn't necessarily be clearly distinguished. At Wizards, we distinguish them quite strongly, and we call the first one "design" and the second one "development". So when I use the word "development" I mean this process of tuning an existing design. I won't use the word "development" in the sense the computer world uses it: as a word meaning something like "coding" or "the general process of getting a game out, including everything from game design onwards". And a "developer" is someone doing this job of tuning a game, not someone who is writing code. If you think of a game as a house, the designers are the architects and the developers are the engineers. If the architects are no good, no one will want to live in the house; if the engineers are no good, the house will fall down.

Relatively early on Wizards realized these two roles were different, largely because we found very different people were good at one or the other. If we produce a new Magic set, one group of people (the designers) will produce a set of cards. Another group (the developers) will then take that set of cards and adjust the costs, the details of how the cards work, and so on. The designers create new cards from scratch in various odd individualistic ways; the developers more often work in groups and apply fairly well-developed techniques to get the card file in decent shape.[4] Designers playtest a fair amount to get a sense of how their design feels; developers playtest a lot, breaking the design, fixing it, and breaking it again.

Designers are rarer than developers, and it's hard to give useful formal training in design. Design is probably more art than science, while development is more science than art (each has elements of both, though). A game with many complex gameplay objects will need more developers than designers, so it's fortunate that developers are easier to train. We find 1 or 2 designers is often enough, whereas developers often work best in teams of 4 or 5.

This split between design and development applies pretty well to a new set of cards for an existing game. For a new game with lots of gameplay objects, it gets more complicated. There are four basic things going on:
1) design of the basic game ("game system design")
2) tuning and modifying that design ("game system development")
3) creation of the objects ("set design")
4) tuning and costing of that collection of objects ("set development")

Obviously all four of these things interact with each other, but we've found it very helpful conceptually to split them up. On large projects, we may even have four teams (with some overlapping personnel), one for each of these tasks. But even on a small project where it's just one team for all four tasks, splitting them up mentally can help clarify what needs to get done.

Let me try to explain a little more what the pieces entail, and then why it makes sense to split them up and how to get the pieces to talk to each other effectively.

The initial game system design is the trickiest part. There are no hard and fast rules, and I won't attempt to explain here how to do it (it's not clear it can be explained). I will say that at this stage, enough sample gameplay objects are needed to try out the design and see if it's fun, but designing a very large number of objects is probably a mistake, and tuning the costs in any concerted way certainly is.

Game system development is probably the hardest to explain. It includes basic tuning and filling in the holes in the rules. But probably the most subtle part of it is getting the costing system set up correctly. An otherwise solid design will often have serious problems in its costing system. Think of a costing system as a set of hooks and handles (or better yet, a collection of knobs and dials that you can adjust) around which you can build a large collection of balanced objects. The game design will want certain kinds of objects (weapons, or spells, or character classes). Do the right sorts of costs exist for those objects? By "right sorts" I mean something simple and elegant, so players can understand it, but still robust enough so that everything can be costed effectively — so that the variety of

[4] Well, those techniques are well-developed now — it took Wizards a while to get there, though.

gameplay won’t collapse as players use just a small fraction of the choices available to them. “Everyone gets to bring 16 pieces to the table” is a costing system for choose-your-own-army chess (each piece you bring costs you 1 of your 16 slots) but it’s not robust, since players will choose nothing but queens.

Set design is where all the objects for the game are created. The focus should be on fun and not balance at this stage. Objects may have costs, but they are just guesses at this point.

In set development, the objects are given more accurate costs, and tested repeatedly for balance. Mistakes from earlier stages will become clearer here. If the game wasn’t fun before, set development won’t make it fun. If game system development didn’t create the right knobs, the set developers won’t be able to balance the set. Probably some objects will need to be tossed out or drastically altered, but the set developers should be making a good faith effort to include as many as possible, and a robust costing system will help.

Of course, these four phases can’t be completely separated, nor should they be. Step 1 pretty clearly comes first, and step 4 last, but steps 2 and 3 are problematic: each wants the other to come first. Designing the detailed rules for a digital object system without knowing what objects it needs to support is pointless, and designing objects without a system is impossible. Running them in parallel and having them talk to each other a lot is probably the best bet.

For smaller games, there may be a lot of overlap on the various teams. A typical setup might be a couple of people doing the system design, which they then hand off to a team of three or four developers. Those developers do system development while one of the designers designs the set, at which point the same developers develop the set.

Even if there are four separate teams, it’s important to have overlap among the team members for continuity. In particular, if developers need to change something in the design, it’s very important to have one of the designers right there who can say “well, this is the reason we designed it the way we did”. Then the developers can look for a change that preserves that original design intent, or at very least can override that intent knowingly rather than accidentally. In general, it’s best to have the overlapping team members *not* be the team leads. A team lead is more likely to be wedded to a particular choice and might well feel called upon to defend it rather than simply explain it.

In general, the designers are responsible for putting the fun in the game, and the developers for taking the problems out. But sometimes in development the game just doesn’t seem like fun: maybe the game simply wasn’t that fun to begin with, or maybe development had to make so many corrections to problems that the game suffocated under the weight of all the changes. At this point it’s key to resist the temptation of having a bunch of developers redesign a game. An experienced lead developer will spot this happening and go to his boss and ask that the game be kicked back to design (perhaps that same design team, perhaps a new one).

Looking back on what I've written, I realize this all may seem like an attempt to automate the production of games and thereby suck all the life out of it. I don't think that's true — let me give an example to try to explain why breaking things into pieces like this can actually help free up people's thinking.

Imagine a new set is being created for an existing trading card game. Relatively early on (in set design) someone proposes an interesting and rather unusual card. It often happens that a somewhat junior R&D member says "we can't make that, that card's broken!" Now often the card *is* "broken" (grossly overpowered) at whatever cost it currently has in the design file. A more senior developer may respond "well, it costs 3 now... would it be broken if it costs 8?" to which the answer will usually be "well, nobody would play it at 8, it would be terrible." At this point hopefully it's clear that in fact this card can go in the design file... it's an issue for development as to whether it should cost 3, or 8, or something in between. That's not to say any card is OK if you find the right cost; some game mechanics are just plain not fun. But if the design team worries about what's fun and what isn't, and leaves the costing for later, they are more likely to take the design risks that make for a fun card set. This applies even if the design team and the development team are one and the same. Knowing you will think carefully about costing later keeps you from worrying about it before you should.

IV. Why We Cost

The basic goal of costing is preserving the fun that a game with lots of gameplay objects can have. If those objects are truly varied, some will be more powerful than others, in the sense that some will help you win the game more than others will (I include things like defeating AI monsters more efficiently under "winning"... basically any goal that a large fraction of your player base wants to achieve can count as "winning"). Starcraft would be a boring game if Drones, Marines, and Carriers were all equally powerful in combat. But if we want players to use objects other than the most powerful ones, we need to assign high costs to the powerful ones and low costs to the weak ones. (This leads to an unfortunate problem of terminology where "powerful" can also mean strong for its cost, rather than simply strong.)

Accurate costing and good game balance aren't things you do just for your most sophisticated players. They're also (and perhaps more importantly) things you do for your weaker players, to protect their gameplay experience. If Carriers are overcosted, so that it's never right to build them, or if Rogues are not as good as other character classes, players who make these choices (due to ignorance, or due to being flavor-driven rather than mechanics-driven) will be "wrong". Other players will let them know, and they may well give up on the game. So there are some basic gameplay choices that you want to be sure are reasonable, not in the sense that they are always right, but in the sense that they aren't always wrong.

How many of the choices in the game are reasonable depends very much on the game. In an RTS, you may well want every unit to be a reasonable choice, at least in some situation. Surely you want every character class to be viable in an MMRPG. But when the number of objects is very large — cards in a Magic set, or items in an MMRPG — you probably don't want them all viable, and you probably couldn't make them all viable even if you wanted them to be.

The reason not to want them all viable is that sorting through the objects is part of the fun of the game. Upgrading your deck, your army, or your character is a big part of why people play. If there are choices that are clear to your beginning player (and there should be), like dropping your non-magical newbie sword for a shiny new one that gives +2 Strength, then some of those choices will be flamingly obvious to your more advanced player. That's OK, as long as there are plenty of harder choices for the advanced players to puzzle over.

If you could somehow create a game where thousands of gameplay objects were all equally powerful, it's not clear that would be a good thing. If a player's strategy is a subset of, say, 20 of those objects (cards in a deck, or items owned by a character), and all of your thousands of objects need to be considered, the number of equally viable player strategies would be so vast no player (or developer, for that matter) would be able to make sense of it. A game with three or four fundamentally different ways to build a Rogue is cool and interesting. A game with three or four thousand is just baffling and annoying.

The reason you can't make all the choices viable even if you wanted to is because you can't balance them all exactly — if your game has thousands of functionally different objects, how can you make them all exactly equal in utility? But there are only so many slots on a level 70 World of Warcraft character, only so many slots in a Magic deck. Only the top 100 or so objects (of a given type) can effectively compete for those slots. If you make some weak object more powerful by adjusting its cost, you may push it into the top 100 by bumping some other object out of the top 100. But most likely you won't really have changed the number of viable objects. So a plan to make every object viable, aside from being misguided, is also unlikely to succeed.[5]

In any case, the danger of having all your objects perfectly balanced is not something to worry about (no game has this problem). The important thing is to

[5] If the number of objects is small—units in an RTS or classes in an RPG—you can hope to balance them all. And even in Magic you could try to have most if not all of the cards viable for sealed. But with very many objects and complete player freedom to choose among them, only some objects will be viable. Also note strictly dominated objects are never viable with complete player freedom, although they can be viable in limited environments like Magic sealed-deck play.

figure out, for your type of game, what percentage of your objects you want to be viable, balance those, and accept that the other objects are worse. From now on I'll act as if you've identified a set of objects to balance and just focus on that.

So to balance some set of objects, you'll need to assign costs. Note that costs may be what players think of as costs, but they may not be. The cost of a unit in Starcraft is primarily its build cost in gas and crystal, but the tech tree you need to get through to be able to build it and its build time factor in as well. The cost of a card in Magic is primarily its mana cost, but it's often helpful to think of a card as costing its mana cost plus 1 card (i.e. the fact that you used up one of the cards in your hand matters too — in games where you replenish your hand at the end of every turn, this cost doesn't exist). This is why a card with a 0 mana cost isn't really free. The gold value of an item in World of Warcraft has very little to do with its cost. In some sense its cost is the time and trouble it takes to get it (which monster drops it and how often), but that's pretty hard to work with, so instead it's easier to work with something like item level (more on that below).

V. How We Cost

This section is, in some sense, the meat of the paper. It's a listing of various tricks for costing and balancing objects. I've tried to arrange them very roughly in order of importance.

Some caveats: these are very much rules of thumb, not absolute laws. There are plenty of exceptional cases. Also, depending on the type of game you're working on, some issues will be more relevant than others, although I've tried to pick issues that have fairly wide applicability.

1) Adjust costs, not effects
If something is too powerful for its cost, people will notice its effect and want to tone it down. Resist this impulse. It's better to adjust the cost first. That way you can hope to preserve the coolness of the object and the original design intent. Even if an object just seems unfun, ask yourself: is it unfun for the user? Maybe it should cost less, and then the user would have fun. Is it unfun for the victim? Maybe it should cost more so the victim feels less abused.

In addition to preserving design intent, adjusting costs first has other benefits. It's easier to understand what you're doing over several iterations if you adjust costs up and down, whereas a big effect change often means your testing must start from scratch because now you have what amounts to a new object. Also, if you've started with an object that's too good for its cost and you adjust the cost upward, you can hope to wind up with a big expensive object that is worth its high price. Such objects tend to be more fun than cheap weak objects.

Sometimes effects do need to be changed. They might be unfun at any cost (in games for young kids, we often avoid effects that make your opponent discard). Sometimes an effect really can't be costed effectively without doing damage elsewhere to your system (see "The dangers of non-scalable effects" below).[6] But changing costs should be your first thought, and usually your second and third as well.

2) The single knob theory
In some games, the objects have many numbers associated with them. You generally want a single number that represents your cost in the sense that it's the number you will tune. Otherwise costing problems can just become too hard to think about. That doesn't mean you won't adjust the other numbers as well; it's just that you will think about everything in terms of one number. For instance, if your RTS units cost gold and wood, come up with some reasonable conversion rate (it won't be perfect, and you'll have to adjust it over time) and write all your costs in, say, gold.[7] That way you will know if unit A costs more or less than unit B, and you'll know when you are raising or lowering a cost. If there's no natural number to use, you might need to make one up that otherwise wouldn't exist — this is what item level is in World of Warcraft. You're creating what economists would call a numeraire (a single, perhaps arbitrary, unit in which all your costs can be measured), and for essentially the same reasons.

If you think about it a bit, rule 1 pretty much implies rule 2.

3) Color wheels are everywhere
In Magic, cards come in one of five colors, and the mana that you use to pay for them comes in five colors as well. This looks like a specific phenomenon, but I'd like to argue it is in fact a very general one.

What does the Magic color wheel do? It's a way of encouraging gameplay variety. If there were no colors of mana, I'd just choose the best cards (at each given cost) and put them in my deck. But with the color wheel, if I choose red cards, I need red mana to pay for them, which makes further red cards easier to play. Now (assuming those pesky developers got the balance right) I might build a red deck, or a green deck, or a blue deck... I have five times as many deck types. In fact, I have more than that, because I can trade consistency for variety by using two colors... if I have a red-green deck, sometimes I will be sad because I am sitting on red mana and I want to play a green card, but I will be choosing from a larger pool of cards, which should help me. Playing all five colors is probably too much — I'll too often have the wrong color for the card I want to play — but depending on the details of how mana works (which of course is dependent on the cards as well as the game system, since cards generate mana as well as use it) maybe I can play 1 to 3 colors. (If I can always play 5 colors

[6] A more subtle situation where effect changes are often needed occurs when an object is submitted by design at a low cost with the intention of it being generally useful. That low cost might be right for the object's general use, but a specific use (perhaps in combination with some other unusual objects) might lead to the object needing to cost a great deal more. To preserve the original idea of the object, one needs to tweak the object's effect so that the general use still works but the specific use does not. (For example, in Magic, cards like Darksteel Colossus or Serra Avatar have clauses to prevent their reanimation.)
[7] Write everything in gold for internal costing work, that is—that's not to say the game itself can't use both wood and gold.

easily, the whole color wheel is doing nothing for me — I'm back in the world where I just play the best cards at any given mana cost.)

Why is this a general phenomenon? Because the basic principle is simply one of forcing correlations in play between various gameplay objects, to increase variety. Most often those correlations are 100%: if you are a Protoss, you build Protoss units. If you are a Priest, you cast Priest spells. But sometimes they are not: if you build an archery upgrade in an RTS, you will presumably be led to make more archers, but you may still have some melee units. If you mine a lot of gas, you might be pushed towards several different units that use gas, but you can still use crystal-heavy units also.

Identifying the "color wheels" in your game and balancing the tension between breadth (easy to use many "colors") and narrowness (colors force you to commit to them) is a big part of good system development. Two factors come into play:

a) How hard is it to use multiple "colors"? Forcing you to build a gas mine or pay for an archer upgrade nudges you in the direction of gas units or archers. Saying you must choose a character class says your Priest is not in any way a Rogue.

b) How much unique stuff does each "color" have? If only red spells directly damage your opponent, you are more willing to add red to your deck (even if it's a bit difficult to use multiple colors). If every color can do everything, you'll just pick one and use it — why pay even a small price in difficulty in that case?

Systems like Magic with a good balance between a) and b), so that you can mix and match colors or play with just a single color, are very appealing when done right. They are also very difficult to get right. Balancing a class-based RPG (where you are forced to play a single "color") is hard enough; balancing a pure point-buy RPG (where you can put any card you like in your "deck") is very difficult, and has arguably never been done successfully. These systems appear to offer great variety, but in practice tend to degenerate to a single viable character build.

Note that if it's trivial to use multiple colors, or if different colors have nothing unique about them at all, you don't have a color wheel (except maybe artistically). Also note that your players and your more junior designers will tend to push in the direction of making it easier to use multiple colors, and letting each color have the cool things the other ones have. Don't succumb.

4) Rock-paper-scissors

Rock-paper-scissors is an incredibly powerful design tool. If your game has three gameplay units A, B, and C, and you try to balance them so that each one wins against the other 50% of the time, you will surely slip up and find that B, say, wins 53% of the time against A and 57% of the time against C.[8] Everyone will stop playing A and C and just play B, and your game will be less interesting. But if you try to balance so that A beats B beats C beats A, each 60% of the time, then as long as you're off by less than 10% in each case, all three units remain

[8] Of course, all statements about A beating B are meant to be interpreted as applying to equal costs of A units and B units, not on a unit-by-unit basis.

viable (not in exactly equal proportions, but that's OK). Of course, you're even safer if you go for 100% instead of 60%, but that may not be a very fun game.

Good games have rock-paper-scissors buried all over the place (note there can be more than 3 objects, but I'll stick with 3 object examples for simplicity).[9] Sometimes those places are quite overt: cavalry gets +2 versus archers, archers get +2 versus infantry, infantry gets +2 versus cavalry (which never made much sense to me, frankly). Sometimes they are more elegant: many RTS games have melee ground beats ranged ground beats air, with air beating melee ground simply because melee ground can't shoot back.[10]

A number of examples fall under the relationship of special ability units losing to units that counter that ability which in turn lose to regular units. So if a unit uses a foo attack (which does 25% more damage), it loses to a unit with foo defense (half damage from foo attacks, but with slightly lower stats, say 5% fewer hitpoints), which loses to a regular unit. A similar example on the defensive side might include an armored unit, an armor-piercing unit, and a vanilla unit.

One subtle rock-paper-scissors common to many games, including most trading card games and RTS games, comes from the ability to create more powerful units later in the game (something you may want anyway for reasons of pacing). If units that come late in the game are not only more powerful, but more powerful in relation to their cost (in gold/mana/crystal — not including the cost of coming late in the game itself), you get a dynamic like:

$1 < 2 < 3 < ... < n-1 < n < 1$

Arriving a little late but with a much better army will win you the game. Planning to arrive much, much later with an infinitely better army is no good... you lose the game before your army is ready. (The numbers above might represent a research tier in an RTS, or the mana cost of a vanilla creature in a trading card game.)

Note you have to be careful about applying heavy-handed rock-paper-scissors in cases where players feel wedded to their choices. If blue decks always beat green decks, or Rogues always beat Priests, your overall play environment will still be varied (if you get the rest of the rock-paper-scissors right), but individual players will not have much fun when they find themselves on the losing end in those matchups.

The trick in using rock-paper-scissors in your game is to identify early where the RPS structures are and which units are playing which roles, and make sure each unit beats someone (so it has a use) and is beaten by someone (so it doesn't obsolete everyone else). It's easy to remember armor-piercing units need to beat armored units, but harder to remember that vanilla units need to beat the armor-piercing ones.

[9] Also note what we're really talking about here is Von Neumann/Nash game theory. A "viable object" is what game theory would call a "non-dominated strategy".

[10] A particularly elegant rock-paper-scissors is found in the RTS Impossible Creatures, where ranged AoE damage hits friendly units, with the result that ranged AoE beats regular ranged (each attack hits several enemies yielding high total dps), regular ranged beats melee (ranged has more dps and better focus firing), and melee beats AoE ranged (once the melee closes, the AoE ranged is hitting itself as well).

5) Don’t neglect the vanilla curve
A lot of time and attention will be lavished on gameplay objects that do unique things. Costing them can be a real challenge. But especially at the system development stage, it pays to focus on the vanilla curve: the costs for those objects that have just basic stats and no special powers. Although it may seem boring or trivial, getting right how the power of the objects needs to scale as the costs rise, or how much of one stat is worth how much of another, is very important, both for itself (most games have lots of vanilla objects) and as a base to build on. You may find that the developers who are best at “breaking” unique objects (typically by finding clever combos between them and other unique objects) are not the ones who are good at getting the vanilla objects properly balanced.

When you do get your vanilla curve (mostly) right, you’ll be able to look at objects and say “well, the stats are worth this much, and we’ve made other objects with this ability, so we know the ability is worth that much”. And if you know how to add the ability cost to the vanilla cost correctly (which you probably will if you’ve thought hard about the vanilla curve, or maybe you’ve lucked out and it’s all linear) you can cost the object. If you can’t do it this way, then each object becomes a new thing to playtest, and you will run out of testing resources long before you are done. As much as possible, as you create new objects, you want to focus testing on the ones with new abilities, not the ones with new combinations of old abilities. A solid vanilla curve helps you do that.

6) Processes
I’ve discussed above different ways to organize designers and developers into teams, how to get the teams to work with each other, and so on. I won’t repeat that here. But how you organize the work is important.

7) Watchlists
Simply cutting people loose and having them play a lot is valuable, and as things get more formal make sure you still have people doing this. At an early stage, they should be looking more for fun and less for balance. Later on, when you are focusing on balance issues, at least some of your best people (and maybe everyone) should be spending some time mucking about looking for whatever broken things occur to them.

That said, it still makes sense to divide up the work in an organized fashion as well. I find a good technique is to get all the developers in a room and list all the things that people think *might* be broken. Then assign each one to somebody to test. When it comes time to evaluate, the person testing it may well have become fond of it, so make sure to listen especially hard to the people she was testing against. If they think it’s OK, it probably is. If she thinks it’s fair but they all hate it, that’s a danger sign.

8) Simple databases to help designers
We like to have all our card (or miniature) gameplay data in a simple database that designers and developers can easily use. For a new game, it’s important that

at least some of the designers and developers can actually modify the database itself (e.g. by adding new fields). There's lots of space for comments as well as the data itself.

For some complex games, where the single knob isn't apparent, it can be useful to put some formulas into the database that convert everything into that single cost. In the ideal situation, each different gameplay stat is converted into a cost, any special abilities are converted to costs (unique abilities will need to be estimated and entered by hand), and some reasonable math will combine them all into a number that represents (with as much accuracy as possible) the utility of the item. Then the cost of that item in the game is also represented. Note these numbers should *not* necessarily be equal. A game, to be interesting, needs good bargains and bad bargains. But a good game developer should know which are which, and by how much.

Good prospects for testing are items where the cost is a lot less than the value. Good prospects for re-evaluating your database formulas or unique ability values are items where the listed value is less than the cost, but people are playing them a lot anyway.

The amount of work needed to get this kind of partially automated costing right may not be worth it for all games, but it may pay off for a game that's an ongoing franchise (an MMRPG, or an RTS that's likely to have sequels). Having a database where everyone involved can make comments (rather than, say, a scripting file that only one person ever reads) is always valuable, though.

9) Playtesting vs. theorizing

Much of what I've written here might give the impression that I think theorizing is better than playtesting, or that fancy formulas will tell you more than actually playing the game. In fact, I believe quite the opposite. Theorizing is very often wrong, and playing is by far the best way to be sure. But playing can be very expensive (if you are testing a possibility and you have to write special code for it). And there just isn't time to test everything, or even an appreciable fraction of everything (once you throw interactions into the mix). So even though a lot of theorizing turns into hot air in the light of later playtests, it's worth it. I find that a good two hour discussion consists of a hundred minutes of pointless theorizing and twenty minutes of good stuff that saved a week's worth of work.

Do remember, though, that if people are arguing back and forth on theory for a while and making no progress, it's often best to table the discussion and just try it and see. (Sometimes it's worth coming back to the argument later, though, after everyone has some real data to draw on.)

Good theories are especially valuable for identifying danger areas that need special testing. If you know that fast mana or card drawing tends to cause problems, you can test those things more carefully.

10) Developing for multiple environments

Often your game will be played in very different ways, but the same gameplay pieces will get used each time. In Magic, the same cards might get used by casual players and by tournament players, or perhaps the same tournament players will use cards to draft and also to do constructed. In World of Warcraft, there are casual players and more intense ones; the more intense ones might be in a raiding guild, or they might be doing PvP. What's balanced in one environment might not be in another. Thus there's no Platonic perfectly balanced set of items, not even in theory.

If you accept that not all objects have to be viable in all environments, it's not a big problem in theory: balance the object to the environment it's best in, and accept that it's of low quality in some other environments. In practice, you won't have the resources to test all objects in all environments. Your best bet is probably to pick the environment that puts the most pressure on object balance (this is probably your most competitive environment) and test there. If an object looks like it will be unusually good in some other environment, you can make a special exception and test it there.

Beware of the trap of testing in your most important or most popular environment. If you think limited play in Magic or PvE in World of Warcraft is the most important, that doesn't mean you should focus your balance testing there (of course, you should play them to see that they're fun). You might decide constructed Magic play or WoW PvP puts the most pressure on your game object balance, and you would want to focus your testing there. If it's balanced in the high-pressure environment, it should generally be fine in the lower-pressure environments, and you should be able to spot the exceptions.

11) The Black Lotus effect

What does it do to your game to have an object that's not balanced? Perhaps not completely broken (to the extent of dominating everyone's experience of the game, whether they are playing with it or against it), but so powerful that almost everyone wants one?[11] If you become enveloped in the developer/balancer mindset, you might feel it's your job to stamp out such things, and in general that's right. I'm going to argue here, though, that it's not entirely right. If it's not too overwhelming, the Black Lotus (or, say, in World of Warcraft, the Hand of Justice) can have good effects as well. Everyone wants it. Everyone can be excited about it. If it's not too hard to get, everyone has a chance to own an item that's top of the line, and they can feel good that they have something that's absolutely a good thing, not just a relatively good thing, or a thing that's only as good as they can afford.

Too many such things (enough to make a deck, or enough to fill every equipment slot in your character) and all players' object collections (which is what a deck or a character is) look the same. But maybe just a few, and your game has gained more than it's lost. Note it's still lost something: if your game has the Boots of Superlativeness, and nothing is better, then once a character puts them on, it's like deleting a slot from that player's character. All other boots in the game are now irrelevant. But maybe that cost is worth bearing, once in a while. It certainly gives people something to talk about.

[11] If you know about the power of Black Lotus in Magic, you might argue it's too close to completely broken to be giving its name to this topic, but humor me — it's too good a name not to use.

If the item is unfun to play against, or if having it versus not having it becomes *the* key factor in the game's outcome, it's probably too much.[12]

12) Dangers of non-scalable effects

Most objects are scalable: they add to your stats, or they do a certain amount of damage, or they are creatures with numbers that show how powerful they are. But some objects are not: a spell that destroys an enemy creature (no matter how large), a spell that heals a friend for all damage (no matter how much),[13] or a kick that cancels an enemy's spell. These objects can be dangerous. They typically can be costed (the question of "well, if it's too good at 2 mana would you play it for 8?" still applies), but their lack of scalability can lead to pernicious effects.

In games like Magic or an RTS, which have systems where costs are directly paid by the player, these items can do damage to other items. If your spell that destroys any enemy creature costs (say) 4, it will be very hard for you to make good creatures that cost much more than 4. Maybe 5, maybe 6, but probably not 8. Your one spell that kills a creature is cool, but is it worth all the expensive creatures that you can no longer make? You can make objects like this, but they affect your other objects very strongly, and you need to think about them early. Although game system development normally focuses just on vanilla curves, you need to decide if you'll have all-or-nothing spells like this early on, and balance your vanilla curves with their existence in mind. In terms of balance, a game about armies of creatures that has a "kill a creature" effect is quite different from a game without one; a game about damaging monsters and healing your friends that has a "heal all your damage" spell is quite different from a game without one.

In MMRPGs this problem tends to arise in a slightly different form. There, items are meant to be obsoleted over time as the player rises in level (or advances through a series of ever more difficult raids). But non-scalable items don't obsolete easily: healing someone for all their hit points is always good, and it's hard to make a healing spell that's better. It's always good to get a free attack. It's always good to stop an enemy from casting a spell. Once you give a player a good ability of this form, if it doesn't scale, you may be hard pressed to give that player something better later on.

It's hard to avoid non-scalable items, because getting rid of them often means taking something simple and cool ("gain an extra attack") and turning it into something less elegant ("gain an extra attack against an opponent of level 40 or below"). But be aware of the potential problems they cause, and decide if they are worth it to you. Perhaps they can be tweaked, perhaps they can be lived with — maybe as one of your "black lotuses".

[12] So the Boots of Superlativeness might be OK, but the Black Lotus is too much, at least if players know how to use it. How much is too much depends on how good the players are at getting the maximum possible benefit out of an item, which in turn depends on how sophisticated the audience is and how quickly information propagates through that audience.

[13] Everquest players might recall the problems around Complete Heal.

13) Balancing powerful effects that occur late or rarely
A unit that appears very late in a card game or an RTS (typically due to a high mana cost or a long tech tree) can be tricky to balance. It's only human to judge these objects by the games they appear in. But the point is that, because they appear late or rarely, they often don't appear at all. If most cards cost 4 mana and Mighty Dragon costs 8, many games will end with it still sitting in your hand. To make up for that, it needs to be pretty overwhelming when it does hit the table. If you want Mighty Dragon to be a viable choice for a good player to put in her deck, or you want the Battlecruiser to be a viable choice for a good Terran player to work towards in Starcraft, it probably needs to be better than at first sight you'd think it should be. Pushed too far, you have a unit that needs to be so powerful you automatically win if you ever manage to play it, which may not be fun (and even if it's fun, you certainly can't make any units that are even more expensive than that!). Thus most costing systems have upper limits, at least among their viable units.

14) Flexibility is sometimes worth less than you think
In general, focus is best. Objects that help you do one thing well are, all else being equal, better than objects that help you do several different things moderately well. So a ring that adds +3 to four different stats should cost less than a ring that adds +12 to one stat. A card that lets you pick one of two different effects is not twice as good as a card that lets you do just one of those effects (in fact, our experience is that to first order the two are equally good, that is, their cost difference tends to be less than 1 mana). Flexibility isn't worthless, but it's usually worth less than you'd think.

15) Miss on the other side of the target
If a particular object is proving tough to balance, you often want to overcorrect once rather than undercorrect again and again. If Rogues keep coming up weak, tune them to be too strong. If the Sword of Doom is too strong, tune it to be terrible. Once you've missed on both sides of the target, it should be much easier to hit it on the next iteration.

These and other methods for balancing game objects won't turn a boring game into a fun one. But they can save a game that would have been fun from collapsing under the pressure of thousands or even millions of players looking to exploit flaws in the game system. Now that more and more players are sharing strategies online (either through strategy websites or simply by playing together in-game), the pressure on games to get balance issues right is enormous. Game designers and developers will need to use every trick they can to keep up.

Appendix: Object Lifecycles

In many games with a large number of objects, you expect to add more objects to the game over a period of time (collectable paper games and MMRPGs tend to do this more than RTSs). If so, managing the object lifecycle is very important. As you add new objects, you want them to be interesting choices for the players. This is a big topic that I'll barely touch on, but I'd feel remiss not mentioning it at all.

You have three basic options as you introduce new objects:
a) they are as powerful, on average, as your existing objects
b) they are more and more powerful ("power creep" or "inflation")
c) they are as powerful on average, but you somehow get rid of the old ones ("set rotation")

In the short term, a) or b) are quite reasonable choices. Eventually, though, (a) will lead to your new objects being all but useless as your set of 100 new objects competes with thousands of old ones. Most decks (or armies or characters) will use the old ones — there's just more stuff to choose from, so more of the good stuff will live in the big pool — and your new objects will seem worthless to people even though on average they are no worse. But for small object pools (one or two expansions to an RTS) this can be a fine strategy.

With (b), your new sets will seem exciting, but as the objects get more and more powerful, your game system is highly likely to break down under the strain. If in your game more powerful objects mean quicker wins, you will reach a state where games are over almost instantly. If powerful defenses play a big part in your game, perhaps certain strategies will become unstoppable to all but a few overwhelming assaults, and most games will bog down. In theory, a game can scale indefinitely (a game of pure numbers might be essentially identical with every number doubled), but in practice highly complex games always break down at some scale. You might feel your game will never reach that point, though, so (b) might be a good strategy for you. Computer games tend to come with built-in lifespans for technological reasons, so very long-term thinking isn't always the right approach.[14]

As an aside, note that even without adding new objects, you can see a sort of power creep in the process of RPG leveling: you throw away your level 20 items and replace them with level 30 ones, thus ensuring you have a desire for new items. But I'm mainly talking here about the addition of new content as a game goes on.

Strategy (c) is necessary if you are thinking very long term. If you want your game to last for ten years, and you want to keep introducing new and relevant items, you have to get rid of the old ones somehow, and power creeping won't work. But the players will be angry at you for taking away their old items. You can mitigate that anger in various ways, although you can't completely cure it. One

[14] Although computer games have been treated as having an inherently short lifespan, it's imaginable that some of them could reach classic status. Tetris and Starcraft, for example, are still played today. So long-term thinking can still be relevant, at least for the ambitious designer.

approach is to have a special place where they can play with all their old items... that place won't be affected much by the new items you create, and new players won't be able to compete effectively in it, but players who want to enjoy their old stuff will be able to do so. Another approach is to try to have a sort of "Escher staircase" of power, where each set seems better than the last (so players feel that the objects entering are better than the ones leaving), but this is a subtle and difficult task to do well (to see it's possible in principle, imagine two sets are considered "in play" at any one time, and sets are published in order rock-paper-scissors-rock-etc, with a paper set object getting +2 against a rock set object, and so on).

Whichever strategy you adopt, make sure that as you test your new objects you keep in mind how they compare to the old ones. The natural tendency when testing is to play with the new toys. But it's easy for all your new objects to look reasonable compared to each other, and yet not have the right balance with your old objects.

Note also there are a host of issues surrounding how you react when some particular object is damaging the gameplay experience, and you have to decide whether to eliminate that object, restrict its use, or modify its functionality. The focus of this paper is how to get the balance right before launch, though, so I'm omitting that topic. But be aware that no matter how carefully you balance ahead of time some problem objects will sneak through. Dealing with them is painful enough when it does happen (for both your customers and for you) that working to minimize their occurrence is well worth it. And of course too many of them and your game is simply irreparable.