

Good Design Makes Happy Customers

A Microtransaction Design Primer

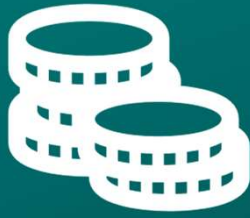
Crystin Cox

Game Director
ArenaNet, LLC



10 years of microtransaction design and strategy.

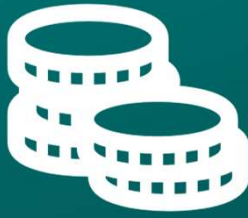
Microtransaction Design



Today I am going to talk about microtransaction design. Specifically, I am going to focus on the role of *good* design in microtransactions.

~~Microtransaction Design~~

Good Microtransaction Design



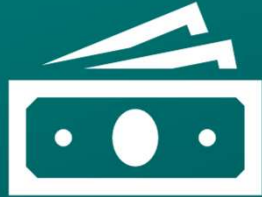
Today I am going to talk about microtransaction design. Specifically, I am going to focus on the role of *good* design in microtransactions.

What do you mean *good* design?



What do I mean by good design? In this case I am talking about designs of systems and content that please players, inspire players to engage with them repeatedly, and increase the quality of the game as a whole.

Microtransaction == Game Design



This should be the standard for all areas of our games. Microtransactions are no exception but they do come with some special considerations, some of which are simple perception shifts on classic design standards while others are highly unintuitive.

Understanding Value Player Motivations Pricing Using Feedback and Data Designers' Role in Microtransactions

I am going to walk you through the considerations I feel are the most important when it comes to microtransaction design organized around a few big areas of interest:

Understanding Value

We have to ask, why?



One of the most basic questions we answer in design is why. Why does this system exist? Why does the player want to go into that cave? All the way up to Why are we making this game?

Good whys make players happy



Bad whys solve developer problems



Good design comes from having good answers to why, answers that are focused on the player and their needs rather than the designer's needs. When design decisions are motivated by solving a developer problem ('we need the game to be longer so we don't get dinged in reviews') we inevitably end up with weak design and weak player engagement.

Why are we making this
microtransaction?
~~Because we need to make money!~~



This is why the answer to the question 'Why are we making this microtransaction?' should never be 'Because we need to make money.'

But I need money to live!



‘But wait’, you’re saying, ‘we really do need to make money!’ Of course you do! Game developers have many needs and you likely work for a company with even more needs and all those needs require money.

Good design comes from a focus on
the player's needs



But you are a designer and if you want to do great design you're going to need to set aside your needs, and your company's needs, and focus on what your players' need.

People only buy things that have
value



And this isn't some anti-capitalist, art above money appeal, your microtransaction items will be more successful if you focus on player benefit more than your need for cash, because the first rule of commerce is that people only buy things that have value.

Players don't care about your needs



Your players will be applying a brutal value assessment to everything you release and your need to generate revenue will not factor into that assessment. A great example of this at a high level is the decline and failure of the monthly game subscription business model.

Why do monthly game subscriptions exist?

- To fund continued development
- To pay for server costs
- To keep players invested in the long term
- To make ongoing payments convenient

Why do monthly game subscriptions exist?

To fund continued development

To pay for server costs

To keep players invested in the long term

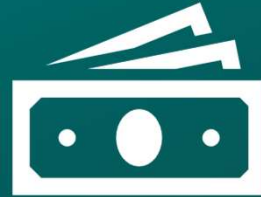
To make ongoing payments convenient.

Developer value is high
Customer value is low

Monthly subscriptions are weak
design

This list is strongly skewed toward value for the developers/publishers, there is very little value for the customer. As audiences for online games became larger and more sophisticated, the willingness for customers to see funding continued development as a community effort diminished and subscription fees became an even worse value proposition. Of course, we all know that the industry shift from subscription fees to free-to-play was more complex than just this, but from a design perspective subscription fees were coming from a place of weak customer value which contributed to them underperforming in the face of newer models.

What about game access? Inherent vs. Derived Value



Many of you have probably noticed that I left a significant customer value out of that last example; access to the game. I did this because it makes for a good opportunity to discuss the difference between inherent and derived value.

Hats are cool no matter what!



Items or systems with inherent value provide some benefit to the purchaser that was not created by the seller. An example of this would be cosmetic items like hats; the player wants to express themselves and the hat provides that all on its own.

Derived value is once removed
Customer → Sub Fee → Game Access



Items or systems with derived value are once removed from the benefit. The monthly subscription fee is a simple example. The value is actually the game, not the subscription, the only reason the subscription has value is because the seller gated the game behind it. This creates a situation in which the opportunity cost evaluation is made removed from the actual value which puts a lot more pressure on that value to be extremely high.

Wow is so good I will jump through
hoops!



In the example of the monthly subscription fee, a few games did extremely well with this model, the value of access to their games was such an overwhelmingly valuable commodity it outweighed the rest of the value proposition. I would assume that every game that released under a monthly subscription fee model hoped that this would be true for their game.

Without the hoops these games are
valuable too



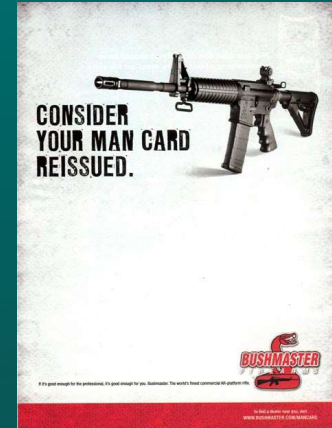
And in a way they may have been right; monthly ARPPU (average revenue per paying user) for free-to-play MMORPGs is generally much higher than \$15, it could be that people really do value these games at a healthy, steady rate per month, but they don't value subscriptions that much.

Will I get what I want?
Did I get what I wanted?



In general, players actually make two distinctive value evaluations. One before purchase and one after. Before purchase players rely on you to communicate the value of microtransaction items and systems. After the purchase players evaluate how satisfied they are by the purchase. Microtransactions are depended on the second evaluation because we need repeat purchasing (these transactions are micro afterall), but you can doom that second evaluation by misrepresenting the items value upfront. This usually comes in two forms:

Value implied by marketing



Attempting to manufacture value through marketing. Everyone is familiar with the adage "Sex sells." Unfortunately, many people take this to mean that if you simply imply that your product will lead to sex you will sell things. Of course, this is absurd, although many people try.

This erodes player trust



Similarly, implying that your microtransaction will result in some vague benefit that is not based on hard value will erode player's trust. It can be subtle like showing a pro player using an item to suggest that it will help an average player win more often or it can be downright manipulative like promoting that you've doubled the drop rate of a rare item in a loot box while the resulting drop rate still well outside the achievable range for a player. Some of these are classic marketing techniques but so what? A spender isn't going to come back after feeling burned just because you explain that other industries use the same tactics.

Obfuscating Value



This second form is unfortunately very prevalent; obfuscating value. The most egregious form of this is intentionally making items or systems hard to understand because if they were clearly understood by the player they wouldn't seem valuable. This can happen unintendedly too because games are complicated. Either way, obfuscating value is bad for player trust, people don't like to feel tricked.

Random Boxes



So, let's talk about random boxes. Remember, good design comes from satisfying player needs and random boxes do fill some very specific player needs, but they are also incredibly easy to design in such a way that their value is murky, leading to confusion and even resentment. Once a player opens a random box and feels ripped off they are much less likely to buy another one.

How do we design a better random box?

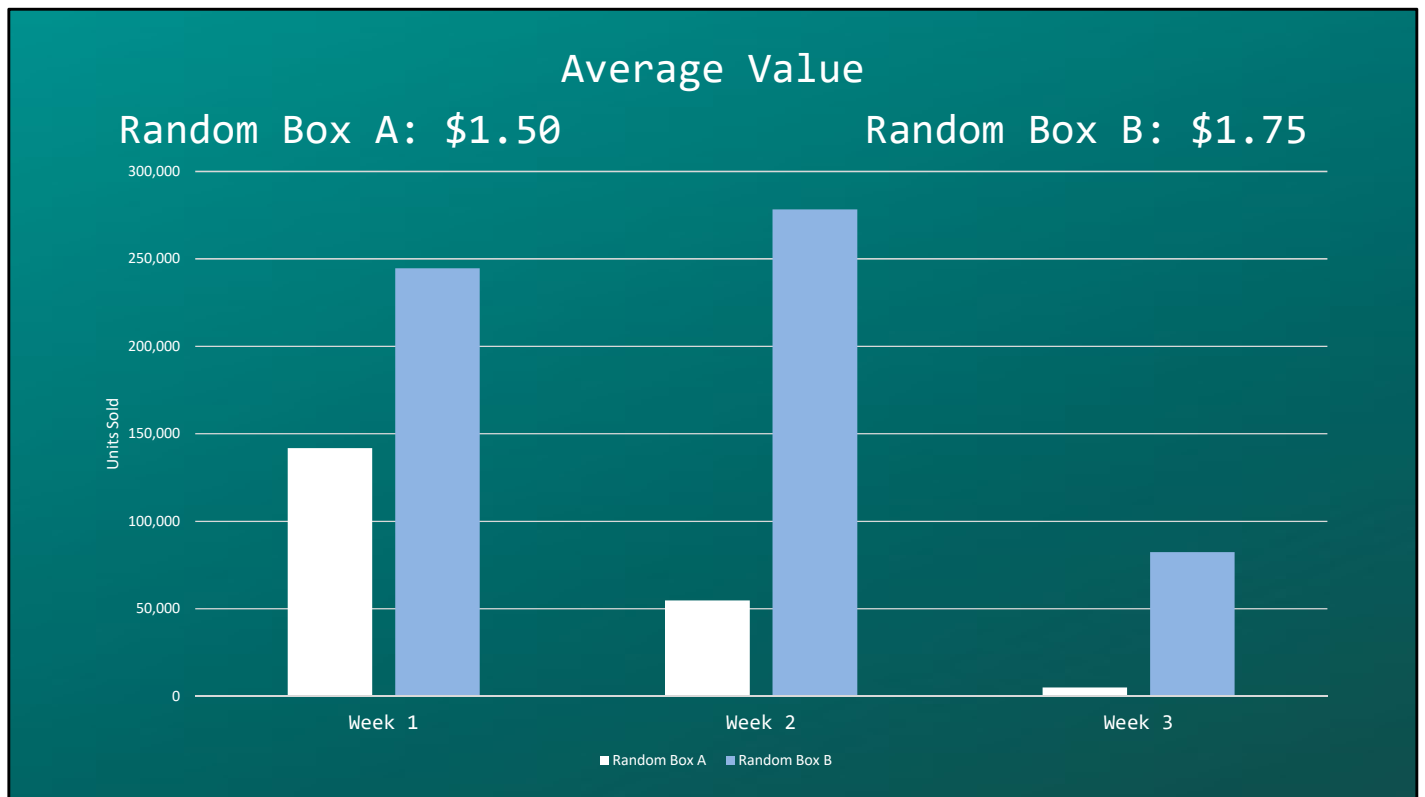


How can we avoid this? How do we design a better random box? Let's go back to our whys. Why do players buy random boxes?

~~Because people love to gamble!~~



'Because they like to gamble' is not a strong answer. Sure, people like gambling but they're still not going to buy something unless it has a real value to them. And adding a random element doesn't prevent people from evaluating their satisfaction with an item after purchase. Players are very sensitive to the value output of systems.



Here are the sales over time of two random boxes released into the same game. Both random boxes cost the same amount but random box B had a slightly higher average value per box. Even though the players did not have access to the drop tables of these boxes and both boxes have a positive ratio of value, players were still able to evaluate and pass judgement on the value of both boxes very quickly. Satisfaction after the purchase is the strongest indicator of future purchasing and we all rely on repeat purchasing. I've seen this play out in random box data time and again; second day revenue from random boxes are a better indicator of total revenue generation than first day sales. Just like always, value is what really drives repeat purchasing of random boxes.

Random Box Types



Okay, players are sensitive to value, but where does that value come from? Random Boxes satisfy a couple of specific demands that usually fall into two general types of random boxes.

Jackpot Style



Jackpots: In these random boxes most pulls are small, incremental, or consumable rewards with a small number of clear jackpot items at a lower chance. These boxes provide players a chance to get something that by necessity must be rare or to get something expensive for a cheaper price by getting lucky. Balancing the price of an individual box versus the rarity of the jackpot items is key for these. Remember the jackpot items are what will attract players to these boxes and will likely be how they determine the box's value. If a player looks at your box and determines that the jackpot item is worth \$50 to them, if they do not get a jackpot after spending \$50 they are going to feel ripped off. Don't get caught up in rarity rates viewed at a population level, the experiences of individual players is what matters most for value perception.

Grab Bag Style



Grab Bags: For these random boxes, all items are close to equally valuable and there is very little variation on rarity of drops. Grab Bags allow players to get something for a cheaper price when they don't have a strong preference for exactly which thing they get. Players judge value for these random boxes based on the entire contents of the box. The key to a good value assessment with these is to make sure that even if the lowest value items is drawn the player still feels like they got a good deal. Depending on what type of items are in your grab bag, you may use the same drop table every time the player opens one, or you may remove items the player has already received from the drop table.

Clear Audience = Clear Value



Of course random boxes might be used to satisfy other needs specific to your game but the key is to have a clear audience that sees clear value in your random box. And remember, random boxes cannot be used to satisfy every need and attempting to force players to use random boxes to get what they want is going to leave them angry. We have seen over the past year several cases in which poorly designed random boxes have caused a lot of player backlash. And that backlash affects every game, even those that use random boxes well. Random boxes are really good at some things and they make some players very happy, but if we abuse them, if they become synonymous with cheap money grabs, we will lose them from our design tool box. Avoiding that fate really comes down to understanding what players want.

Player Motivations

Why do people want things?



Okay, we're gonna forget about making money for now and focus on doing great design based around value. But how do we know what is valuable to players? To start with let's zoom all the way out and look at why people want things at all.

Games satisfy higher band motivations



Lucky for us, there has been a lot of great research into human motivation. Since we're talking about video games we don't have to worry about basic motivations like food and shelter and can focus on models that deal in more high level human motivation. I often find it helpful to remember that when we sell microtransactions, we are selling something that no one truly *needs*. When people decide to buy the things we make they are free to evaluate those things using only the higher level motivations and consequently their quality bar is much higher.

Self Determination Theory

Intrinsic
Motivators



Mastery
Autonomy
Relatedness



There are many models available for understanding why people do things. I prefer to use Self Determination Theory because its examination of intrinsic motivation is well suited to game design work particularly reward and monetization work. Quickly, SDT tells us that many higher level human behaviors are driven by intrinsic motivators not extrinsic rewards or punishments. Further, intrinsic motivations exist to satisfy three high band human needs: Autonomy, Mastery, and Relatedness. Many microtransactions do fall cleanly into satisfying one of these needs, for example:

Mastery

- Weapons
- Armor
- Power-ups



Autonomy

- Content unlocks
- Accelerators
- Card or skill packs
- Fast Travel



Relatedness

- Cosmetics
- Customization
- VIP Systems
- Story



So, simple; if your microtransactions satisfy the human need for Autonomy, Mastery, or Relatedness it will be intrinsically valuable. But which needs are best suited to be met by microtransactions? Should you strive to meet them all or just one? The answers to those questions are largely game specific, but I can offer some guidelines:

It helps to know what needs the core gameplay of your game is aiming to satisfy. Aligning microtransactions and core gameplay needs can be great.

Motivation synergy



For example, collectable card games offer a lot of autonomy satisfaction as acquiring more cards give the player more combat choices. Selling card packs increasing that autonomy satisfaction. Complimenting core gameplay needs satisfaction can also be successful, if that same collectable card game doesn't offer much in the way of relatedness, microtransactions could fill in that need with things like custom backgrounds or avatars.

Watch out for cultural biases



There are some pitfalls to avoid too. If your core gameplay is aiming to satisfy mastery and your intended audience is largely western you may want to avoid selling items that satisfy the mastery need because traditionally western cultures are uncomfortable with the idea of money being used to gain mastery in contexts where mastery is seen as being earned.

Replacing earned mastery with microtransactions can feel unfair



For example, if you have a fighting game and as players win matches their fighter becomes more powerful also selling items that make the fighters more powerful is likely to feel unfair because you've already created a context where powerful fighters are a signifier for skill. You could argue that the microtransaction in this case is providing autonomy because you're allowing the player to trade time for money. If that is the case, then make that clear, optimize the design of the item around autonomy and find a way to signal that it isn't a replacement for earned mastery.

But microtransactions can satisfy
the mastery motivation



As a counter point, if you do want to sell items that tend toward the mastery satisfaction side there are ways to mitigate the feeling of unfairness. The MMO Blade and Soul has a system of weapon upgrades that requires the player to acquire lots of a specific resource, that resource can be acquired through microtransactions but once the resource is acquired the weapon is only upgraded when the player defeats a boss monster. So the process can be sped up with money but in order to see the benefits the player must still prove their mastery in the game.

Microtransactions and Rewards



You will also want to understand which needs your game's rewards are aiming to satisfy as attempting to satisfy the same needs through rewards and microtransactions can lead to these two systems fighting with each other. We suffer from this problem in Guild Wars 2.

Cosmetics in GW2

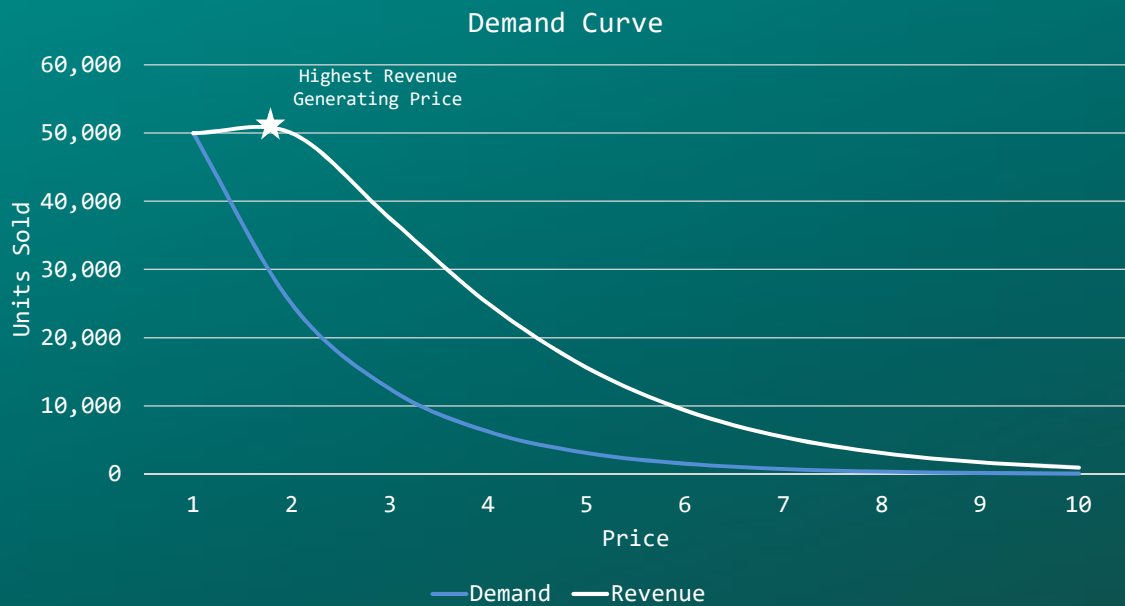


Because there are no stat increases on gear at the end game, most rewards for high level content are cosmetics, but we also sell a lot of cosmetics as microtransactions. This creates a situation where if our most dedicated players are motivated to complete high level content for cosmetic rewards they are inherently going to be unmotivated to purchase cosmetic microtransactions and vice versa, the motivation for rewards and the motivation for microtransactions are fundamentally fighting against each other. We've alleviated this over time by carving off types of cosmetics for rewards only and other types for microtransactions only but it isn't ideal.

Pricing

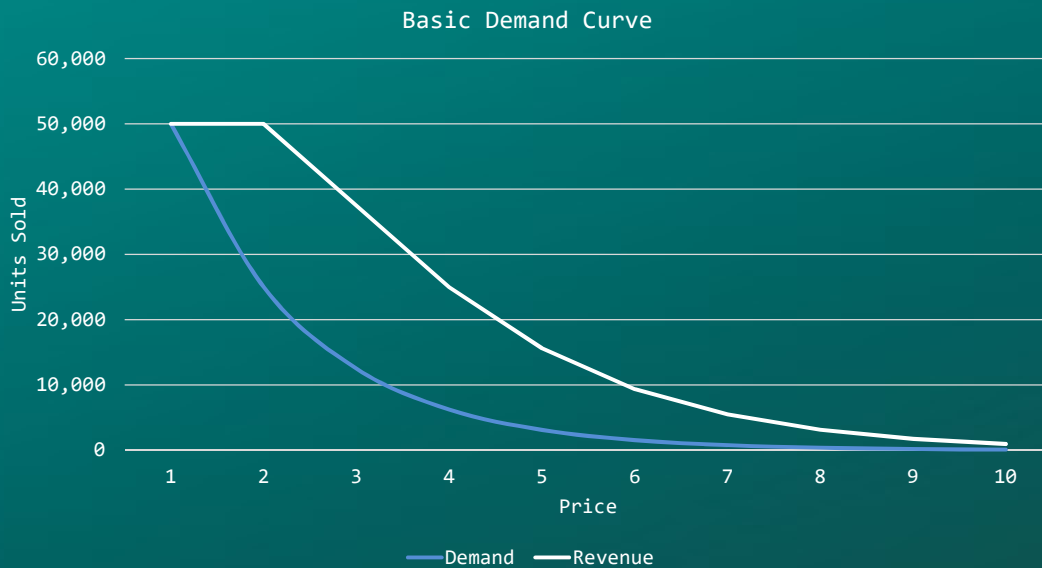
Once you've designed a great microtransaction it's time to start thinking about actually selling it which means moving value from an abstract concept to a concrete dollar amount.

Demand Curves



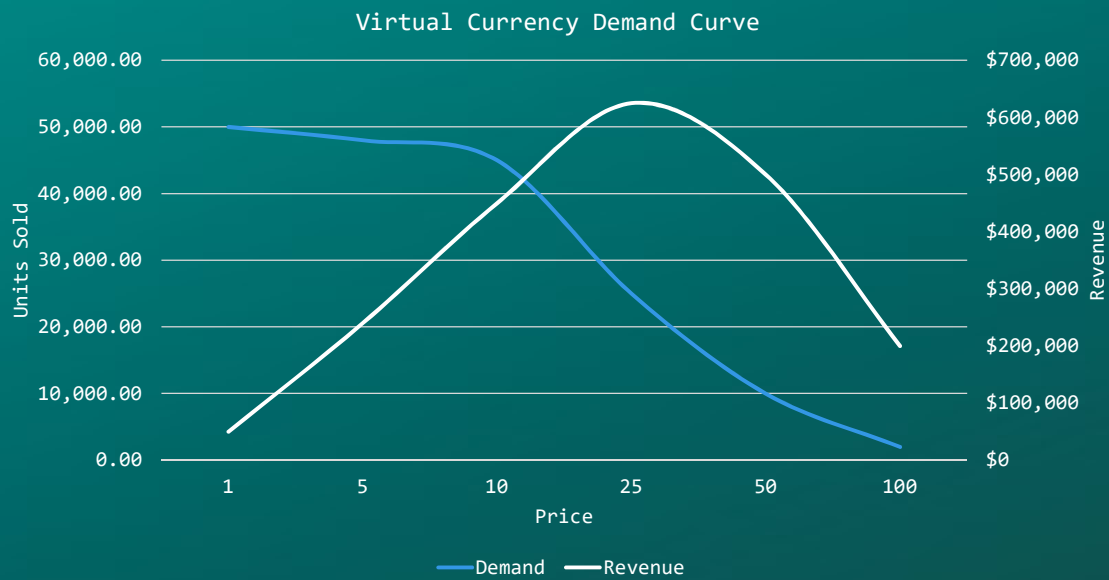
Micro economics tells us that pricing is very simple, just plot a demand curve against possible prices and bam! We have the optimal price. Unfortunately forecasting demand to this resolution is very difficult without historical data. And pricing can be unintuitive because humans do not always act rationally.

Virtual Currency Bundles



For example, let's look at virtual currency bundles. On paper virtual currency bundles should have a fairly linear demand curve; lower priced bundles should have a high number of purchasers and as the price increases the number of purchasers should drop off. This idea seems so obviously correct that many games do all sorts of things to entice purchasers to move up to higher price points while still keeping lower priced bundles around to increase the spending pool. But if we look at actual data around purchasing virtual currency bundles we do not see a linear demand curve.

Virtual Currency Bundles



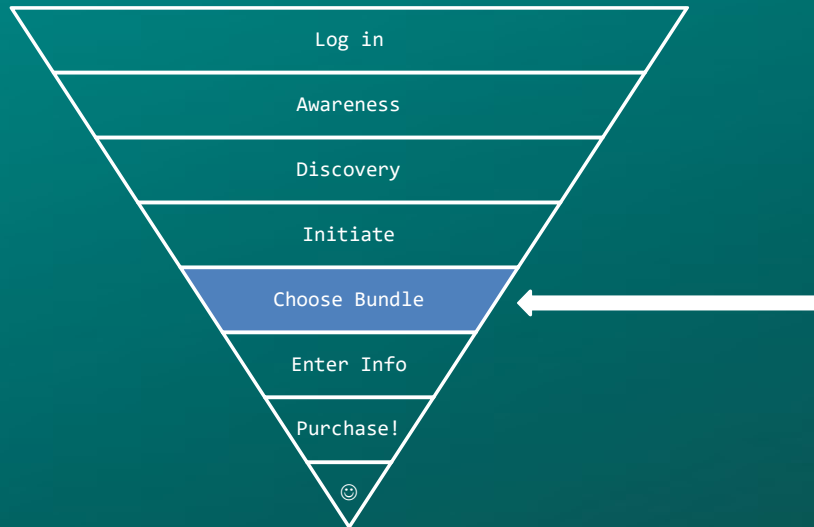
I am going to use some data from one of the games I have worked on to illustrate that demand curves for virtual currency actually look like this:
Demand is not significantly different between \$1 and \$10.

Virtual Currency Bundles First Time Purchaser Info

Offers	First Time Purchasers Week 1	First Time Purchasers Week 2	First Time Purchasers Week 3
\$1	2,435	Not Offered	Not Offered
\$5	1,122	2,145	Not Offered
\$10	1,034	1,207	2,032
\$20	884	903	1,576
\$35	704	704	906
\$50	429	429	429
\$100	166	166	166
Total Purchasers:	6,774	5,554	5,109
Total Revenue:	\$98,755	\$103,545	\$121,600

How do we know? Looking at purchasing behavior we see that over 70% of first time purchasers purchase the cheapest bundle available but when the cheapest bundle available was set to a higher price, number of first time purchases didn't change much at all. Purchasers would buy the \$1 option if it was available but seemed just as happy to buy the \$5 bundle if it was the cheapest option because demand itself was not higher for the \$1 bundle. In this case it doesn't make sense to offer the \$1 or \$5 bundles at all. In fact, Emily Greer of Kongregate presented purchasing data from over 150 of the games on their platform back in 2012 which suggests that this type of demand curve for virtual currency is the norm (<http://www.gdcvault.com/play/1016565/Maximizing-Monetization-in-Core-Games>). But why?

Virtual Currency Bundles Purchasing Funnel



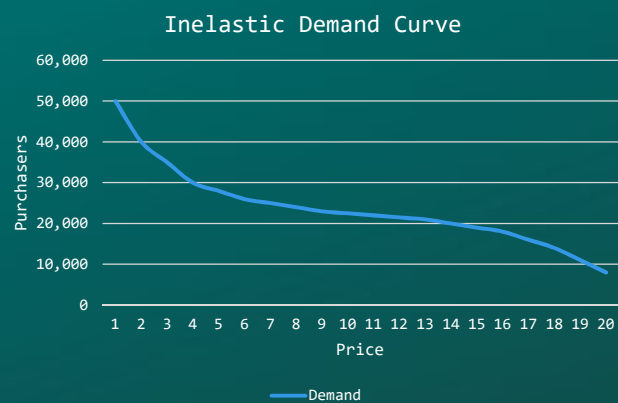
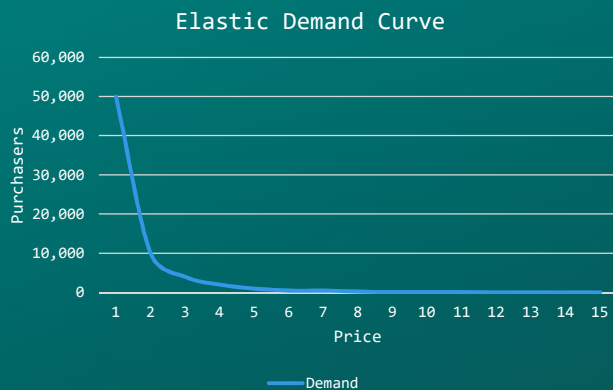
It has to do with the virtual currency purchasing funnel. Because the barrier to entry is the decision to purchase, not the price of entry demand isn't significantly influenced by price.

Pricing Walk Through



So demand curves can be tricky, if you are introducing an entirely new class of microtransaction to your game or launching your game for the first time, how do figure it out? Let's use a fictional item as an example and walk through the pricing process. Let's say you have a racing game that is heavily focused on players creating custom car paint jobs and you have a designed a microtransaction that allows players to store multiple paint jobs per car that they can switch between at will. First, we need to figure out what type of demand curve we are working with, elastic or inelastic?

Elastic and Inelastic Demand Curves



Elastic demands change considerably based on changes in price; the lower the price the more purchasers, the higher the price is the lower the number of purchasers.

Inelastic demands are effected very little by price changes; prices must change dramatically before number of purchasers change. One of the most inelastic categories of microtransactions is cosmetics. If I can purchase the ability to turn my head into a pumpkin in your game but I hate pumpkins that fact that the pumpkin head is cheap isn't really going to affect my purchasing decision. There is no such thing as a universally appealing cosmetic item so your market for any one cosmetic item is always limited but that market's demand is incredibly inelastic. Conversely, in games with number based progression systems, progression boosters are very elastic. Pretty much everyone would like to have them but whether or not they purchase them is largely based on if they feel the price of the booster is a good value, the lower the price the more people will feel that it is.

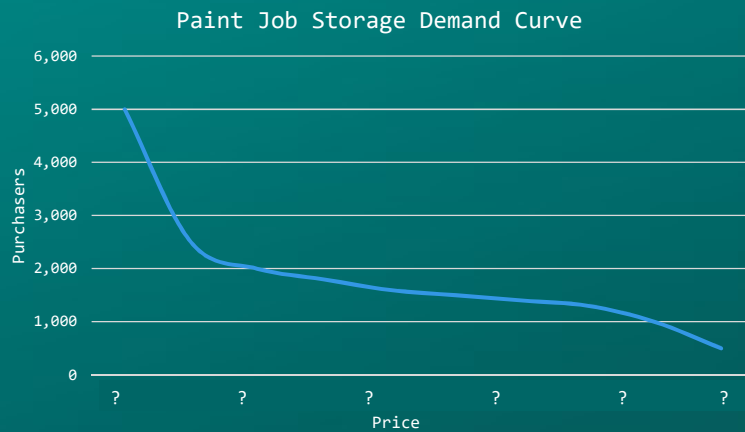
Our paint job storage item is somewhere in the middle, it's mostly a convenience item which are traditionally elastic because players often don't view time spent on in game activities as a cost, but this item in particular will be super valuable to players with a strong motivation of self expression which makes it somewhat inelastic.

How big is your market?



The next major consideration when it comes to pricing is market size. How many players are likely to be interested in this item? As I said before, inelastic items can often have very small markets and that's okay because the high value of the item to that market means you don't need to deal in bulk. Elastic items however count on having a large market because elastic items' prices need to be low to increase volume. Our paint job storage item has a market on the small side since we can only sell to people that want to store multiple paint jobs, that won't be everyone but the market will be reasonably size since we said customization is a core focus of the game.

Constructing a Demand Curve

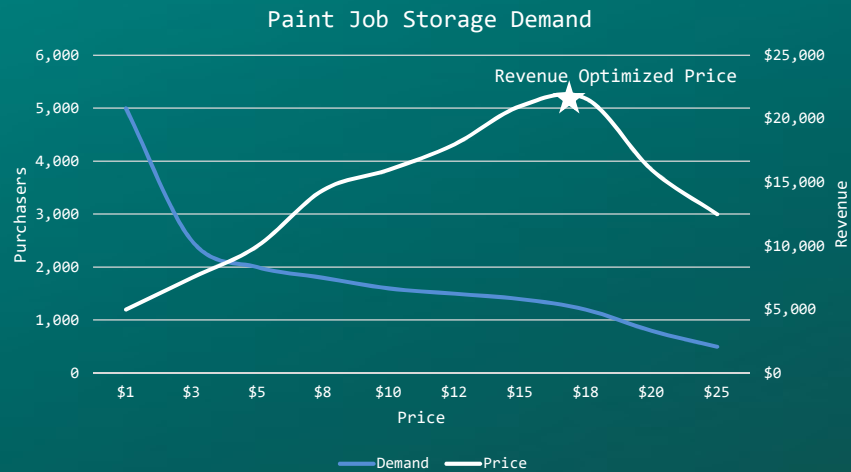


- Expected average revenue per paying user
- Comparable prices

Now that we know our item's elasticity and market size we can create a demand curve with the right shape and scale of potential sales but we still don't know the scale of possible prices. If you are starting from scratch you need to look at: The projected ARPPU (Average Revenue Per Paying User) of your players. Comparable titles and the prices of their microtransactions.

Finding Price Range

- Expected weekly average revenue per paying user
- Comparable prices (from your own game if possible, and from others)
- Now make a projected revenue graph:

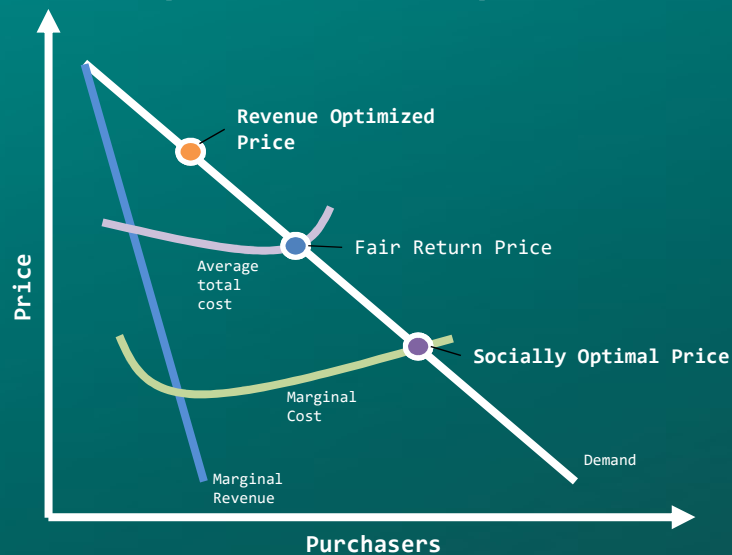


The projected ARPPU (Average Revenue Per Paying User) of your players.

Comparable titles and the prices of their microtransactions.

Then you can project revenue at each potential price point on a graph. The highest point on the revenue line corresponds to the revenue optimized price. But...

Should you charge the revenue optimized price?



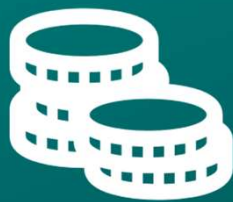
I want to briefly touch on a more complex economics concept; Socially Optimal Price. Micotransactions all exist as monopolies, if I don't like the price of boosters in your game, I can't go buy them in another one for cheaper. Monopolies push commodities to be more inelastic, without competition purchasers are more likely to continue buying something even when the price goes up, but that doesn't mean they are happy to pay that price. The concept of socially optimal price comes from what are known as necessary monopolies; monopolies that exist because competition is too inefficient for the market to bare but the commodity provided is essential. One of the best examples of this is pharmaceuticals. This graph shows how price ceilings are calculated for life saving drugs. We can't just look at the revenue optimized price because this cuts off a lot of people from getting the drug because they can't afford it. The Socially Optimal price is the price at which the cost to create the drug is just covered but the maximum number of people can get the drug, thus it is the price that is best for society. Then there is a compromise price in the middle where the company makes what is considered a "fair return" on its investment. I'm not going to get into how exactly that is calculated because it involves a lot of info that isn't really relevant to microtransactions but I believe the concept of socially optimal price brings up an important point even in our world of a very unnecessary monopoly.

Social Costs in Games



Price can have a social cost in your game's community. It's not as obvious or severe as restricted access to life saving medicine but it's still there. Sometimes players will pay the revenue optimized price but feel taken advantage of. It's easy to say that if someone buys something they must have thought it was worth the price but in a monopoly that is not always true and charging prices that players perceive as too high can cause perception issues and community backlash. One example of this I see in games a lot is very high prices for things like name changes or server transfers. Are these services inelastic? Yes. Will people pay the high cost if they really want them? Yes. But they also have a tendency to create a situation where the player feels taken advantage of like their account or character is being held hostage. Social cost can also creep in when we look at your entire microtransaction catalog because...

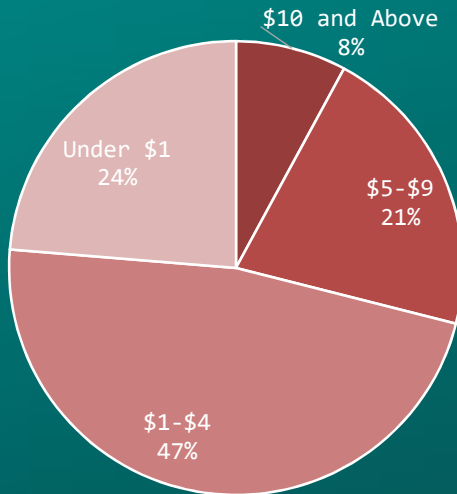
Players have different purchasing power



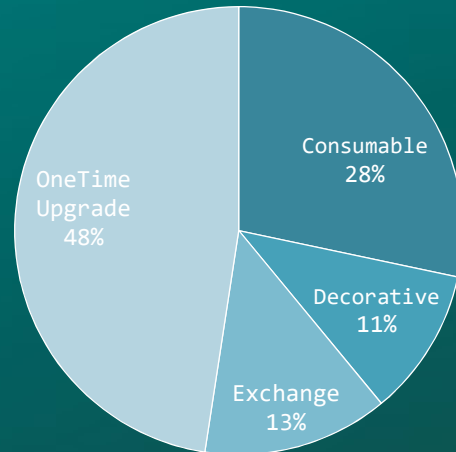
You need to understand that players have spending limits, at some point no matter how high their demand they will be out of disposable income. When you price something at the revenue optimizing price you will be cutting a lot of potential purchasers out, that's unavoidable especially for free to play titles, but you should also account for the fact that being priced out of items places a social cost on your game. If a potential customer cannot afford anything they want from your game they will likely leave, further if you attempt to squeeze players right to their purchasing limit over and over they will feel taken advantage of, you will have optimized revenue on a series of individual items but end up cutting your potential revenue pool significantly. And this is silly for microtransactions because they have such low continued production cost (pretty much all cost are upfront) you have incredible control over pacing and variety of items offered.

GW2 at launch

Price Range at Launch



Revenue by Category Month 1



For instance, when we launched Guild Wars 2 we were creating a premium game that cost \$60 but we planned to support the game through microtransactions so we wanted a steady, weekly ARPPU and a high paying rate. When we built the very first catalog of items for our Gem Store we were constantly asking ourselves “How will this store look to someone that just payed us \$60?” We needed to be aware of how those items’ value would be compared to that initial \$60 purchase and we wanted it to feel accessible and sustainable. We focused on having high value permeant unlocks to set a standard for value and a variety of low cost consumables to offer good value that was accessible. Think about your projected player base, not just the higher spenders but all of them, make sure your providing them items that make sense.

Price and value have a circular relationship



The prices you set for your items need to reflect that item's value but they also have a large impact on your player's perception of that value. First impressions are vitally important because once a number is associated with an item in a person's mind it is very hard to replace that number.

Never raise prices!



This can work in your favor when it comes to discounts (both sales and bulk discounts) but it makes it very hard to permanently change an item's price after the player base as already internalized that item's cost. Of course, lowering the price is generally better received (although you won't see the same results from a lowered price as you would have had the price premiered at the lower number.) In fact, on my teams it is a hard and fast rule that we can never raise prices on items that have already been released and we are very cautious about introducing items at a higher price than items of the same type we have released previously (there must be some very easy to understand added value for the player.)

Frequent price fluctuations are bad



Frequent price fluctuations, such as running sales events very frequently, will erode player's confidence when it comes to purchasing, you don't want your players to agonize over whether today is the right time to buy this item, so take setting your initial price seriously.

Indirect Pricing



We've looked at an example of the most standard form of pricing, but there are a couple of other techniques in which price is set indirectly.

Random Box Balance



I talked earlier about why people buy random boxes and I went over both jackpot and grab bag style random box designs. Both jackpots and grab bags use indirect pricing strategy but in different ways.

Jackpots and Rarity



Jackpots are usually built around items that derive their value from their rarity. Of course, this requires that your game supports the idea of rarity having value in the first place. Hats in a single-player first-person-shooter aren't going to be much more desirably just because there aren't many of them.

\$50 at \$1 a box = 1:50 drop rate
(or does it?)

~~1 item for every 1,000 players =~~
~~1:1000 drop rate~~

Jackpot random boxes allow you to price an item by calculating the amount of money a player will spend to get the jackpot. Don't fall into the trap of allowing the price to be determined by how many of the item you wish to have exist in the world. This point of view will not translate well to players who don't care about average drop rates or item per 1,000 users, they only care about their own experience and what they think is a fair price for the jackpot. Using total number of items in the world to determine price usually leads to jackpot items being over priced which then leads to players feeling like they are getting a bad deal.

1:50 drop rate means 36% of players
pay more than \$50 to get the
jackpot

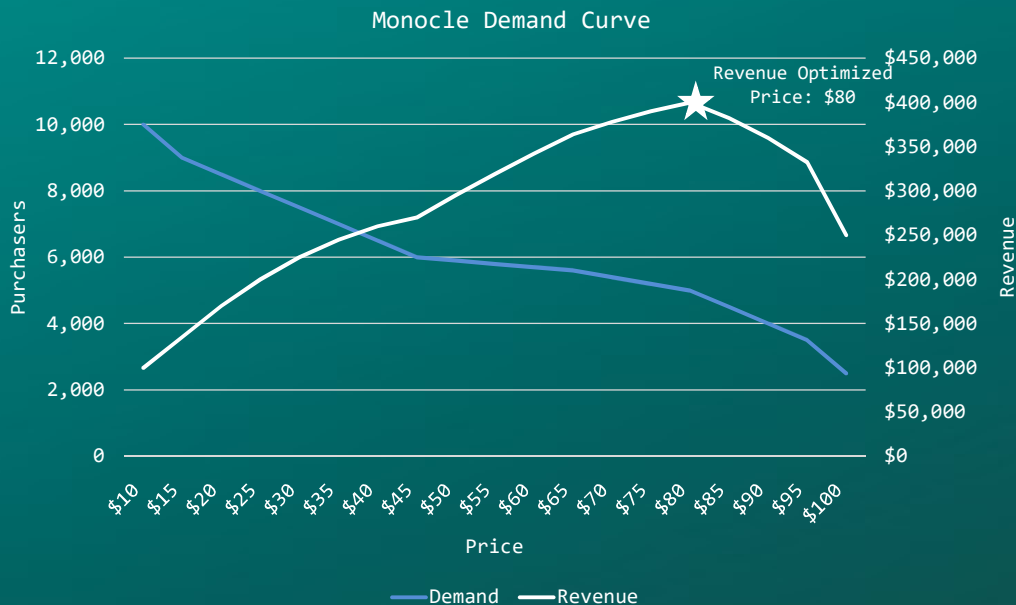
Getting your jackpot to come out at the right value can be hard if you're just using straight randomness. So consider not using randomness!. Random number generators are easy to implement but there are other options that allow designers more control over individual experiences, consider using a set schedule or token systems to increase value for individual players.

Grab Bags, how do they work?



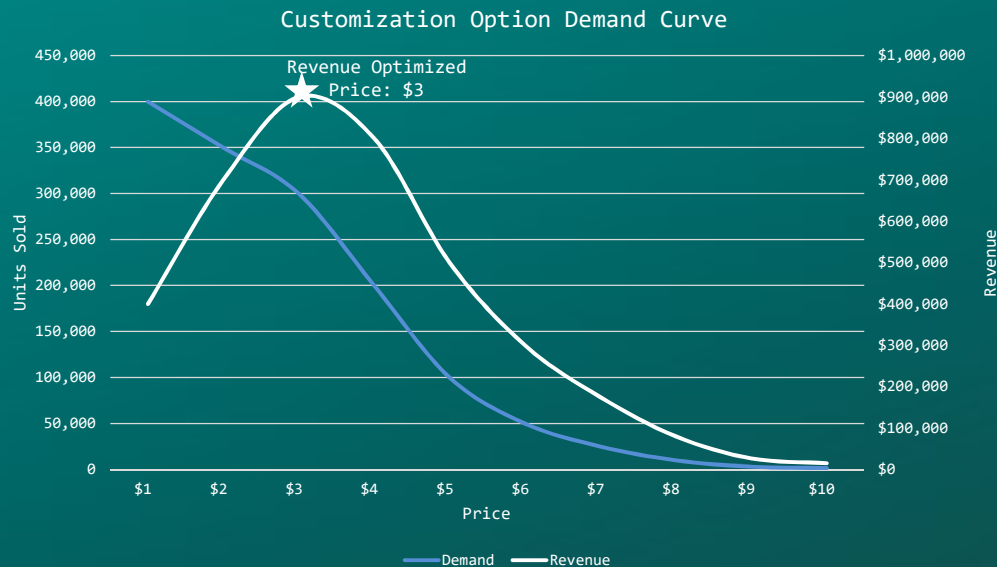
Grab bags aren't always built around rarity markets, and grab bags function best when all the items in them come out at close to the same rate (not actually the same rate though as we do want the pulls to feel like they have variety.) Even so, you may want to use grab bags as an alternate strategy for pricing items with niche demand curves.

Monocle Example



For example, let's say you've created a cosmetic item, for instance a monocle. Let's say you plotted your demand curve for this monocle, looked at your ARPPU and the potential market, and found that this monocle is very niche and very inelastic and so the optimal price to charge for the monocle is \$80. You may look at that number and realize that it is well out of the expected range for an item of that type and so player perception would be very poor at that price point. Even if we know that potential buyers for an item would value the item at this high price, it is difficult to release the item at this price without damaging your catalog.

Monocle Solution



One potential solution to this problem is to release the monocle in a grab bag including lots of other cosmetic items for a low price. A demand curve exists for customization options in general, it has a much lower price range but it is also a much larger market than the one for the monocle. Players who want to expand their customization options would be happy to have the monocle, not for \$80 but possibly for \$3 and you are likely to make more money off that item in the long run with this strategy. Games like Overwatch employ this strategy exclusively and do very well with a high level of player satisfaction.

Feedback and Data

Feedback and microtransactions are not friends



Feedback is the lifeblood of design. We are all full of cognitive biases, skewed perspectives, and personal heuristics. We rely on a diverse set of voices to give us the perspective we need to do great design that isn't made solely to please ourselves. But what do you do when you're designing something that is almost impossible for people to speak openly and honestly about? Well, traditionally the answer for microtransactions has been to turn to data.

Data is not predictive



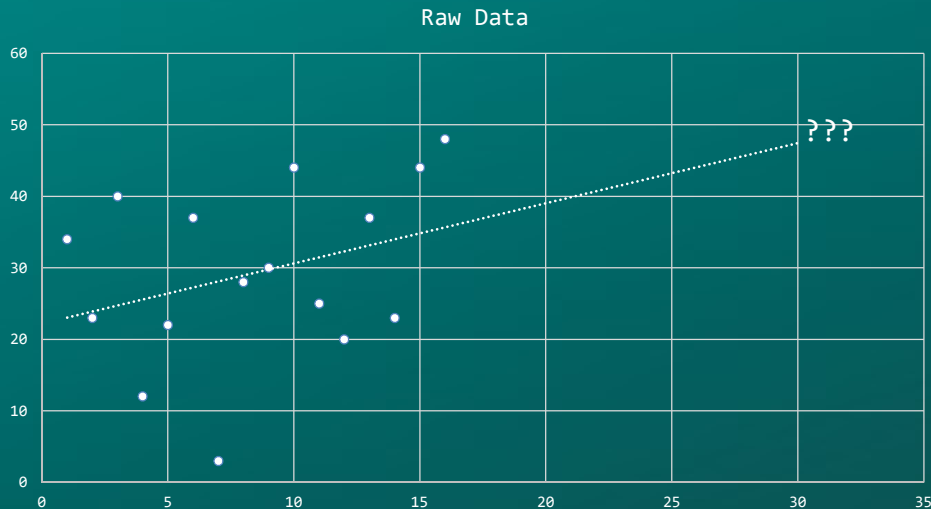
Data is great, the collection of data on a large scale has allowed us to gain amazing insight in the activities of our players. But data has one very big weakness when it comes to making design decisions; it is analytical not predictive. Data can be *used* to make predictions but it is not itself predictive.

How inconvenient



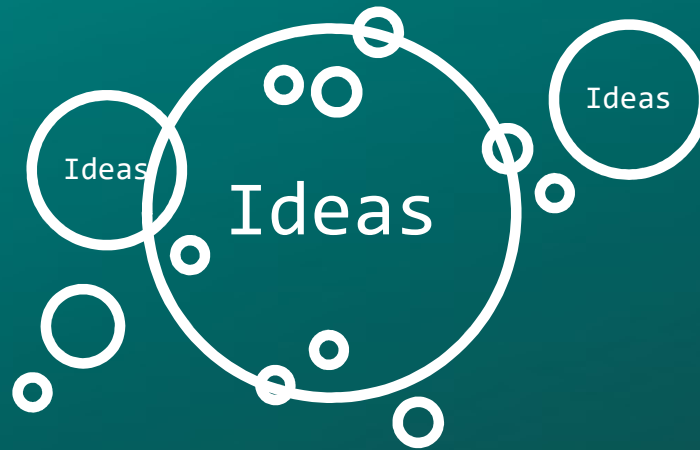
But we really need to make predictions, to understand how people will react to our designs, to understand where we should focus our efforts next, and we need to make predictions fast. Forecasting and predictive analysis is time consuming and complex so it makes a poor substitute for feedback. Data is fast and easily obtainable and so I see many developers simply decide that since they need data to be predictive, then it is.

Using data as if it were predictive



They create items, or features, or changes then look at the data and react to that data. This strategy is somewhat effective; data sometimes lines up with predictive analysis so you're going to get some hits and this strategy generates very fast iterations so it covers up a lot of misses by rolling over them on the way to a new action. But from a design perspective, this strategy is very dangerous.

Keep your design space wide



A vast majority of outcomes play out over a period of time, this strategy guarantees that only a very small subset of possible design decisions will ever be made because they are the only design decisions that create positive data in a short time period. This is the equivalent of making all design decisions using brainstorming only; only ideas that can be thought of, explained, and defended in a single sentence will surface.

Your relationship with your
players last longer than a
single data cycle

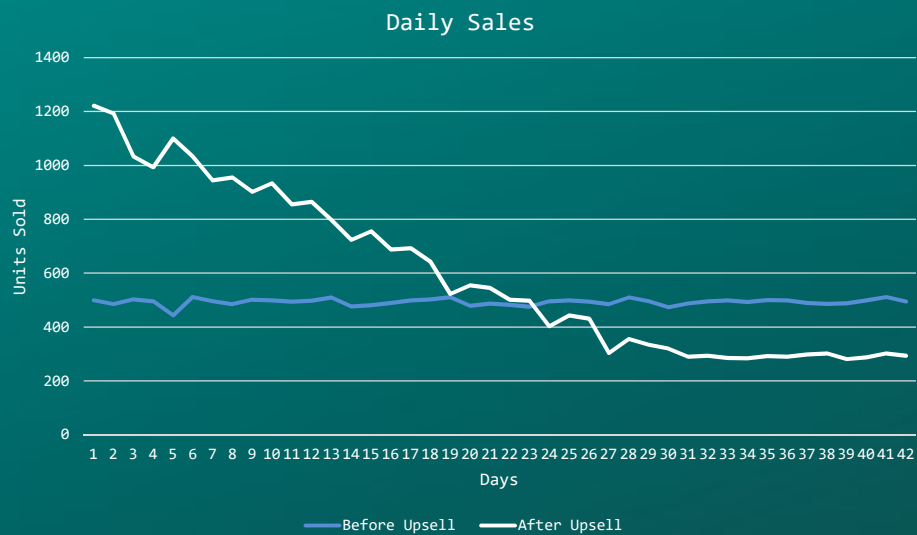


It should not be shocking to anyone that short term gain is often paid for with long term lose. If your relationship with your players last longer than a data cycle then making design decisions based on single data cycles is ludicrous. It is not good enough to know that players did something, you need to know why.



It should not be shocking to anyone that short term gain is often paid for with long term lose. If your relationship with your players last longer than a data cycle then making design decisions based on single data cycles is ludicrous. It is not good enough to know that players did something, you need to know why.

Upsell example



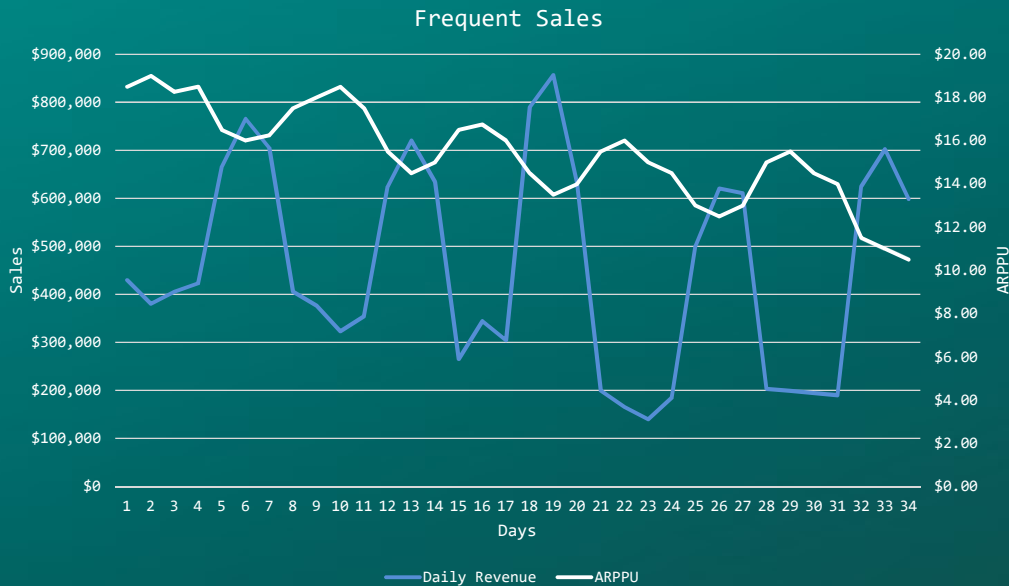
A classic example is upsell messaging. You might consider popping up upsell messages to your players at times when you think purchasing an item would be especially beneficial to them. If you do this you will likely see a big spike in sales for this item but that spike is destined to fall and likely fall well below the sales rate of the item before you implemented the upsell.

Attention seeking is not a long term strategy



Because people do respond to immediate attention seeking calls to action but they can't keep responding to them over the long term, we are wired to filter those messages out after we've seen them a few times. At this point either the item you are upselling's demand is entirely dependent on these attention-grabbing messages, meaning you will need to keep upping the attention seeking behavior in an inevitable diminishing returns cycle or worse, the item does have inherent demand but now you have trained your players to ignore it.

Constant sales = eroded ARPPU



This is exactly the same type of short-sighted number chasing that leads to games falling into the endless cycle of sales events which slowly eat away at your ARPPU over time.

Make data work for you



So how do you avoid short-term data being used as a substitute for feedback? First get data to work for you as a design tool. Proactively set data goals and benchmarks for your microtransactions. This could mean setting expectations on how long you will need to collect data before you make decisions or it could mean asking to be evaluated on metrics other than just total revenue generated. If you release an item you expect to be a consistent repurchase, you want to set goals around number of unique repeat purchasers or frequency of purchase. If you release an item with niche appeal, you want to look at item ARPPU as a measure of success.

Put on a hazmat suit and get the feedback!



Another way to avoid having data replace feedback is to get feedback. I know, earlier I said that people aren't capable of talking about microtransactions honestly and that is true. When you design microtransactions you need to be prepared for feedback, even feedback from other designers on your team, to be highly emotional. Hyperbole will be used, slipper-slopes will be invoked, and someone may even ask if you won't please think of the children. This is understandable.

Game designers are scary



Game designers spend a lot of time learning how to manipulate human nature and encourage specific behaviors. We have the capacity to do a lot of evil and once you add money to the mix the stakes are high. So, when people give feedback on microtransactions you need to remember that a lot of that feedback is going to be an expression of fears and worse-case-scenarios. Knowing this going in can arm you to sift through the fears to get at the heart of feedback.

Are you being evil?



But also keep an eye on the level of fear and anxiety coming from your players. Is it rising over time? Is it spiking around a particular system or item? Even if the content of this feedback is not valuable you shouldn't tune out its existence. Check yourself, are you doing something evil? Are your designs pushing or manipulating people to do things they regret or that make them unhappy? Remember your job is to entertain players and make fun, no one needs to play your game to live, if you are creating a situation that makes people unhappy they will leave. Preventing that from happening is part of your job in doing good microtransaction design.

I don't want to have to pay for things!



The majority of unstructured microtransaction feedback comes down to 'I don't want to have to pay for things.' And as much as I love to sympathize with the inherent unfairness of capitalism, engaging with that feedback won't help you make better design decisions.

Opened ended feedback is not good



If you are soliciting feedback on microtransactions you need to focus the feedback on things that are relevant to design, and you're not likely to get far by asking questions directly. Open ended, qualitative questions (like, would you buy this item?) make people feel pressured to answer in a way that makes them seem clever or virtuous, the contents on their answers end up having little to do with the actual question.

Ask good questions



Ask questions that focus on action instead like; how would you use this item? Or ask questions that focus on value like having responders rank items in order of desirability. Or better yet, don't ask questions, simply watch people use your item or system without comment. Just like other areas of design, this tends to produce the most honest feedback.

Microtransactions and Design

Microtransactions should belong to design



I strongly believe that the mechanisms by which your game generates money should be the prevue of game designers. Game designers should have the same vision, control, and implementation responsibilities when it comes to microtransactions as they do with any other feature that is part of a video game.

It's your responsibility



Furthermore, game designers should *enthusiastically* participate in vision, control, and implementation of microtransactions because they matter a great deal to your game and they are very difficult to get right. If your attitude around microtransactions is “I don’t want to have to think about that” or “I don’t like microtransactions so I don’t want to get involved” you are doing yourself a disservice and more importantly, you are doing your players a disservice.

Step Up!



Game designers are uniquely equipped to understand what their players will value and how that value should be presented. As microtransactions become more and more of a staple for all types of video games, game designers of all kinds must become comfortable designing microtransactions and being a part of microtransaction strategies. I implore designers to get involved and demand high quality standards be placed on microtransaction designs, stand up and confidently declare that you are the right person for this job, then deliver great design.

Thank you for attending!
Special thanks to:
Guild Wars 2 Commerce Team
Project Horseshoe

Contact me:
Crystin@arena.net

Appendex

Further info/reading on SDT:

GDC Talk: [Why Did Players Buy That?](#)

GDC Talk: [Intrinsic and Extrinsic Player Motivation Drive by Daniel Pink](#)

Further info/reading for economics:

If you just need basic microeconomic concepts, I like [Khan Academy](#)

[Elasticity](#)

[Monopoly](#)

[Socially Optimal Price](#)

[Virtual Economies by Ted Castronova and Vili Lehdonvirta](#)