CHAPTER

The Freemium Business Model

1

Commerce at a price point of $0

All business models are malleable thought structures, meant to be adapted and decisively employed to best achieve a specific product's or service's goals. This being understood, and for the purposes of this book, a broad and basic formal definition of the *freemium business model* is described as follows:

The freemium business model stipulates that a product's basic functionality be given away for free, in an environment of very low or no marginal distribution and production costs that provides the potential for massive scale, with advanced functionality, premium access, and other product-specific benefits available for a fee.

The freemium business model is an adaptation of a fairly common distribution and monetization scheme used in software since the 1980s: the *feature-limited* software paradigm was when consumers saw most of the fundamental core components of a product released for free, with the product's remaining functionality (such as saving progress or printing) becoming available only upon purchase, either in a one-time payment or through recurring subscription payments.

The most basic point of difference between the freemium business model—*freemium* being a portmanteau of *free* and *premium*—and the feature-limited model is distribution: feature-limited software products were generally distributed on physical discs, whereas freemium products are almost exclusively distributed via the Internet. So the distribution speed and ultimate reach of feature-limited products were a function of the firm's capacity to produce and ship tangible goods; no such restrictions limit the distribution of freemium products.

A second distinction between the freemium and feature-limited business models is the scope of functionality of each: whereas feature-limited products often merely showcased the look and feel of the full product and could not be used to fulfill their primary use cases at the free price tier, with freemium products, payment restrictions generally do not limit access to basic functionality. Rather, freemium products exist as fully featured, wholly useful entities even at the free price tier; payment generally unlocks advanced functionality that appeals to the most engaged users.

The freemium model represents a fundamental evolution from the feature-limited model, given a new set of circumstances under which software is distributed and consumed: mobile devices give users access to products at a moment's notice and throughout the day, cloud storage services and digital distribution channels

Freemium Economics. DOI: http://dx.doi.org/10.1016/10.1016/B978-0-12-416690-5.00001-4

allow products to be discovered and purchased without the need for physical discs, and digital payment mechanisms render purchases nearly frictionless.

The pervasiveness and connectedness of software, then, represents a heightened state of awareness with respect to the demands made upon software by users. And it also presents a massive opportunity to quickly and almost effortlessly reach millions, if not billions, of potential consumers upon product launch. This is the reality of the modern software economy, and it is the backdrop against which the freemium business model has emerged.

Components of the freemium business model

The ultimate logistical purpose of the freemium business model—and the source of the advantages it affords over other business models—is the frictionless distribution of a product to as large a group of potential users as possible. This potential for massive scale accommodates three realities of the freemium model:

1. A price point of $0 renders the product accessible to the largest number of people.
2. Some users will not engage with the product beyond the free tier of functionality.
3. If the product is extremely appealing to a group of users, and the product presents the opportunity to make large or repeat purchases, a portion of the user base may spend more money in the product than they would have if the product had cost a set fee. Thus, the revenue fulcrum, or the crux of a product manager's decision to develop a freemium product, is the potential to maximize scale, paid engagement, and appeal to the extent that the total revenue the product generates exceeds what could be expected if the product cost money.

While the freemium business model is not governed by a rigid set of physical bounds, some patterns hold true across a large enough swath of the commercial freemium landscape to be interpreted as intellectual thresholds. The first pattern that emerges is that the broader the appeal of a product, the more potential users it can reach and the more widely it will be adopted. A broadly appealing product has a widely applicable *use case*, or purpose. Generally speaking, products that address a universal need, pain point, or genre of entertainment appeal to more people than do products that serve a specific niche. Broad applicability obviously has a direct impact on the number of users who adopt a product.

The second pattern is that very few users of freemium products ever *monetize*, or spend money on them. The low proportion of users who monetize in freemium products contributes to the necessity of large potential scale: a low percentage of monetizing users within a very large total user base might represent a respectable absolute number of people. This concept is referred to in this book as the *5% rule*, or the understanding that no more than 5 percent of a freemium product's user base can be expected to monetize prior to product launch.

The third observable trend with the freemium model is that the spectrum of total monetization levels—that is, the total amount of money users spend within the product—spans a wide range of values, with a very small minority of users spending very large amounts of money. The larger the minority of highly engaged users, the more revenue the product generates in aggregate; when the spectrum of monetization levels is broad, more users tend to monetize than when the spectrum is limited to fewer values.

The confluence of these three trends establishes a freemium development directive: the broader the appeal, the higher the level of engagement; and the more numerous the opportunities to engage that the product offers, the more revenue it will generate. At some optimized point, these forces can contribute to a more advantageous revenue dynamic than could be expected under paid adoption circumstances. If frictionless distribution is the logistical purpose of the freemium model, then its commercial purpose is to establish the product attributes necessary to achieve greater monetization and absolute revenue than would be possible using a paid adoption model.

Establishing the aforementioned product attributes to fully leverage the advantages of the freemium model is not achieved through a trivial set of design decisions. The decision to apply the freemium business model to a product is one of the most formative choices made during the development process; it must be made while establishing the product's fundamental use case, and it must inform every aspect of the development process that follows.

The reality of the freemium model is that it can very easily be misapplied. Thus, the decision to employ the freemium model or another commercial framework is first and foremost a function of ability—not whether the *product* should be freemium, but whether the *team* possesses the expertise and experience to produce a successful freemium product. If the answer to this question is yes, but a qualified, conditional, or equivocal yes, then the team is not well positioned for success. The product that emerges from the development process is an external expression of the freemium decision point, while answering the questions, "Can this product succeed as a freemium product?" and, "Is this team capable of implementing the freemium business model?" is a more introspective exercise.

Scale

The potential for scale is essentially the conceptual foundation of the freemium business model. This isn't to say that freemium products must achieve massive scale to succeed; freemium products can be profitable and considered successful at any number of user base sizes. But the characteristics of a product that facilitates massive scale must be in place for a freemium product to achieve the level of adoption required to generate more revenue than it would if it was executed with another business model. These characteristics are low marginal distribution and production costs.

A product's marginal cost of distribution is the cost incurred in delivering an additional purchased unit to a customer. For physical products, these costs are often realized through shipping, storage, licensing, and retail fees, and they tend to decrease with increased volume through economics of scale; that is, the more products shipped, the lower the cost of marginal distribution.

Digital products are distributed through different channels and thus face different distribution cost structures. Often, the only costs associated with distributing a digital product are hosting expenses and platform fees. In aggregate, these costs can be substantial; at the marginal level, however, they are effectively $0.

Production costs are also structured differently for digital and physical goods. Physical goods are composed of materials that must be purchased before the product can be created; likewise, the production process represents an expense, as either a human or a machine must piece the product together from its source materials. Digital products incur no such per-unit production costs; they can be replicated for effectively no cost.

Low marginal distribution and production costs create the opportunity for a product to be adopted by a large number of people, quickly, at little to no expense on the developer's part. This is a prerequisite condition for the freemium model; because its revenue stream is not necessarily contributed to by the entirety of the user base—that is, product use and payment for the product are not mutually inclusive—the product must have the potential to reach and be adopted by a larger number of people than if each user contributed revenue.

Freemium distribution is achieved most often through platforms, or commercial outlets that aggregate products and allow for their discovery through digital storefronts. Platforms provide value through retail functionality such as the ability to comment on products, rate them, and search for them based on keywords. Platforms generally charge a percentage of the revenue generated by the product; a common fee is 30 percent, meaning the platform takes 30 percent of all pre-tax product revenues.

Freemium products can also be distributed on the web with stand-alone websites. Obviously, such a distribution method incurs no platform fees—meaning the developer keeps all revenues it generates—but it also does not benefit from the infrastructure of a platform store (most notably, the ability to search). Web distribution may also be impractical or ineffective for some freemium products, especially mobile products for which installation from the web complicates the adoption process.

The freemium product development cycle may differ fundamentally from the development cycle for paid adoption products; freemium products are generally fluid and are therefore developed through an iterative method. Freemium products also often take the form of software-as-a-service or software-as-a-platform; as such, they evolve over time, based on user preferences and observable behaviors.

The costs incurred in continuous, post-launch development cycles do represent production costs: overhead expenses, such as employee salaries, are shouldered for as long as products are maintained and developed for, which may match the lifetime of the product. But, given low distribution costs and the freemium model's potential for very broad reach, these costs, when distributed over a very large user base, can normalize to a level approaching $0 per unit. In other words, while a given development cycle represents a real and potentially large cash expenditure for a freemium developer, the size of the potential user base that product is capable of being exposed to reduces the marginal cost of production to an immaterial amount.

This dynamic imposes requirements on the total cost of production, however. Freemium products with limited appeal yet high development costs may face more substantial obstacles in reaching profitability than do products with broad appeal. Thus, the scale and scope of production costs should be constrained by the total potential size of the user base; certain product use cases may have limited intrinsic appeal, and the higher their costs of development, the higher the marginal costs of production.

While a non-zero marginal production cost doesn't limit scale in the short term—products that can be distributed for free are still accessible by large numbers of people—it does limit scale in the long run by creating higher threshold requirements for product monetization. In other words, to fund its continued development, a product with niche appeal must capture higher levels of revenue per user than must a product with broad, universal appeal. If this revenue threshold can't be achieved, the product will be shuttered.

Note that massive scale itself is not a condition of the freemium business model; products need not be large to be successful, especially if success is measured in profit relative to what could be expected from the product under the conditions of a different business model. Niche freemium products experiencing high levels of monetization from passionate, dedicated users can achieve success with modestly sized user bases. Similarly, no law of physics restricts to 5 percent the proportion of a user base that monetizes in a freemium product; this number could theoretically reach 100 percent. In practice, the proportion of users monetizing in freemium products is low—often extremely low—and preparing for such an outcome during the development process allows a developer to construct a revenue strategy that more prudently accommodates the realities of freemium monetization.

Thus the 5% rule is not, in fact, a rule; it is a design decision through which the developer embraces the practicalities of the freemium business model, which suggest that a small, dedicated minority of users can monetize to greater aggregate effect than can a larger population of users that has monetized universally through paid access. This design decision is an outgrowth of the freemium model scale requirement: the larger the total user base, the more meaningful will be the minority proportion of users who monetize.

Insight

A second cardinal component of the freemium model is insight, or a methodical, quantitative understanding of user behavior within the context of the product. Insight is achieved through a battery of tools and procedures designed to track the ways users interact with the product, and it is implemented with the goal of optimizing the product's performance relative to some metric.

Insight is a broad term that roughly describes a freemium product's entire data supply chain, from collection to analysis. Freemium product usage is *instrumented* through the use of data collection mechanisms that track the interactions between users and the product; this collected data can be aggregated, audited, and parsed to glean a valuable understanding of what users like and what changes could be made to the product to better serve users' needs.

Insight is composed of two constituent parts that are equally integral to the entire process but require disparate skills to implement. The first part is data collection, or the means through which user interaction is tracked, stored, and made available to data consumers. (Data collection is typically done by the product's developer, but sometimes it is done by the product users themselves). The technical infrastructure—the software and hardware components that accommodate data retrieval, storage, and end-user access—is often encapsulated with the broad term *analytics*.

The second part of insight is the work undertaken to make sense of the data collected and stored in order to improve the product. This work might take the form of regular reporting of key metrics or an analysis of a specific process or product feature in an attempt to understand its performance. These report templates and processes are usually described with the term *business intelligence*.

An important and frequently recurring type of analysis undertaken on freemium products is *user segmentation*, which aims to draw fault lines between the naturally occurring archetypes in the user base and then map the commonalities among them. Such an analysis is especially important in the freemium model, as user archetype groups may exhibit similar payment behavior and thus bring to light meaningful opportunities to optimize revenues.

Because not every user of a freemium product can be expected to directly contribute revenue to the product, the needs of the users who will—the users for whom the product's core use case holds the most appeal—must be identified and catered to with the highest degree of expediency and zeal. Doing this requires understanding what those users pay for and some knowledge of when they pay for things and why. Insight is not mere record collection; it is the process of erecting a model of users' needs from user interaction histories in order to better meet those needs through further product development.

Freemium products encompass such broad and diverse user bases that mere developer intuition is not sufficient for tailoring the product to the desires of the most enthusiastic users. User feedback can take many forms, but perhaps the most salient of those forms is the monetary contribution. Relating that feedback to behavior requires a reasonably sophisticated framework for examining data both from a theoretical perspective and with practical granularity.

Insight into the user base—the ways users interact with the product, the impetuses on which users make purchases, and the factors that inspire users to abandon the product—is required in order to best leverage the advantages of the freemium model. The user base scale the freemium model provides has inherent value in the volume of data able to be mined for instructions on how to best serve the needs of the most enthusiastic users. Even when a user does not contribute monetarily to a product, the data artifacts that user produces can be used to draw conclusions about the product that may be useful in accommodating the users who can and do contribute monetarily.

Scale and insight are therefore complementary and mutually reinforcing: as a product scales in user base size, the data produced by the users and captured by the product confers on the developer additional knowledge about user behavior. This knowledge holds value even when it does not specifically relate to users who pay; all knowledge about user behavior can be used to optimize the product.

Data that is exclusively descriptive of users who don't contribute revenue—even if its only value is helping the developer to build better profiles of exactly that user archetype—establishes between the users and the developer a more robust, more refined channel of understanding, through which the users' needs and preferences can be communicated.

Monetization

Monetization is an obvious requirement of any business model. And despite the fact that the freemium model stipulates a price point of $0, monetization is not a secondary concern for freemium products; in fact, monetization is the nucleus of the freemium experience around which all product features are organized. The difference between monetization in the freemium business model and in the paid access business model is what a transaction delivers to the user. In the paid access model, a transaction delivers admission into a feature set environment that, depending on the amount of research the user conducts prior to purchase, may or may not capably fulfill user expectations. In the freemium model, a transaction delivers an enhanced experience.

The paid access model presents an inherent barrier to adoption with an upfront cost; the greater that cost, the smaller the potential user base. Under paid access conditions, eventual users of the product must meet two requirements: sufficient interest in the product generated by a perceived value equal to or higher than the product's cost, and sufficient disposable income equal to or higher than the product's cost. When these requirements are met for a potential user, that person purchases the product. But two other groups of potential users exist who will not purchase the product: the potential users who do not possess disposable income that matches the product's price, and the potential users who do not believe the product delivers value equal to its price.

The product may be significantly appealing to the first group of potential users; were the product priced lower, matching the lowest common denominator of what users in this group are equipped to pay for it, the product's use case would sufficiently meet those users' needs. To the second group, the product may hold limited appeal, and, likewise, if the price were decreased to match the lowest common denominator of what users in this group consider its value to be, the product's use case would sufficiently meet those users' needs. (See Figure 1.1.)

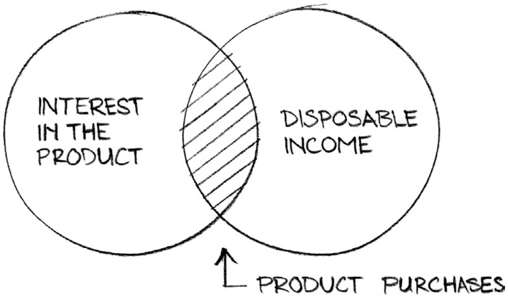


FIGURE 1.1

Purchases occur at the intersection between sufficient interest and sufficient disposable income.

In the abstract, the lowest common price denominator of both potential users groups—what users with low disposable incomes and what disinterested users are willing and prepared to pay for the product—is $0. In the paid access model, where the price of the product is paid up front, only the users with the greatest enthusiasm for the product and the means to purchase it are able to access its full feature set. But in the freemium model, all users in all three groups are given access to a limited feature set and invited to pay for additional functionality.

The advantages of the freemium business model are thus realized under two sets of circumstances. The first is when a subset of the group lacking disposable income, who otherwise would not have purchased the paid access product, experiences a fortuitous change in financial stature. These users, having adopted the freemium product upon discovery, are now able to spend money on additional functionality within a product they have already familiarized themselves with.

These same users, had they discovered only a paid access version of the same product, might not have purchased when their financial stature changed for any of several reasons: perhaps they forgot about the product, or perhaps they found a comparable freemium product, or perhaps they felt alienated by the product's price requirement. Whatever the case, and no matter how small this group of users might be, under these circumstances the freemium product would likely capture more total revenues from this group than would the paid access product.

The second, and more common, set of circumstances under which the freemium model boasts advantages over the paid access model is when some subset of users who can afford and want to purchase the paid access product are capable of extracting value from it in excess of its price—in other words, the product is worth more to the users than what the developer is charging for it. When a freemium product's revenue strategy provides a broad spectrum of levels across which users can potentially monetize, each user is given the agency to enjoy the product to whatever extent the user's disposable income provides for. A considerable amount of money may have been lost had the product been monetized exclusively through an upfront, singular price point.

The goal of the freemium model is to optimize for the second set of circumstances: to give those users who find the most value in the project the latitude to extract as much delight from it as possible. When this condition is met, the benefits of the freemium model are unlocked and can create a revenue dynamic that eclipses what would have been possible under paid access conditions. But the pricing model required to establish such a level of monetization is not inherent in every product with a price point of $0; rather, establishing such a pricing model requires significant effort and strategic consideration.

Optimization

As noted earlier, freemium products are generally developed under the auspices of the iterative method because of the aforementioned pillars of freemium design—scale, insight, and monetization—in that the developer must collect and parse data on the habits of its users before it can improve upon the product to an appreciable extent. The faster the developer implements these improvements, the faster the fruits of that labor are gained. Therefore, the iterative development mantra is to make decisions as fast as possible on sufficiently sized but not superfluous volumes of data.

Optimization is the process of converting data about user behavior into product improvements that increase some performance metric. These improvements are generally incremental and, individually, not wholly significant. The purpose of implementing them quickly, after studied and careful measurement, is that the improvements can compound each other and result in appreciable performance enhancements in a shorter amount of time than with the *waterfall model*, which consists of sequential, end-to-end development cycles punctuated by full product releases.

Optimization is a delicate process that can sometimes resemble a development tightrope walk. Too much emphasis on optimization can result in new product features or content being sidelined, and too little can leave potentially valuable performance improvements unexploited. Like any other element of software development, optimization incurs costs: direct costs in the form of developer time and focus, opportunity costs by not pursuing other development goals over the course of implementing optimizations, and hidden costs. The hidden costs of optimization are the hardest to measure, as they are usually the result of too narrow a focus on optimizing existing processes. In other words, when a developer's enthusiasm for optimization exceeds a certain threshold, the developer can lose sight of the product's long-term performance and instead pursue improvements to specific processes that render only short-term, incremental product gains.

The concept of product changes producing competing effects is predicated on the idea that optimization takes two forms: *local optimization* improves the performance of a process, while *global optimization* improves the overall performance of the product. Both forms of optimization are necessary, but global optimization is more abstruse and requires a broad understanding of the user experience and a definition of product success (usually related to revenue). Whereas local optimizations are easy to undertake and evaluate—a process either improves relative to its previous state or it doesn't—global optimization requires a much longer measurement scope and the organizational wherewithal to think about the product in the abstract. For these reasons, local optimizations are potentially easier to both execute and build design decisions around, but the gains made from local optimizations may be illusory if they haven't been thoughtfully considered as part of a global optimization strategy. In fact, local optimizations may produce negative long-term results when undertaken aggressively or without considering the secondary effects on other processes or performance metrics.

For instance, when monetization mechanics are rendered highly visible, many of them improve with respect to revenue performance, but they may also alienate users. When measuring the effect of the optimization on a specific process (local optimization), a product manager might consider the change a success; when measuring the effect of the optimization on the rate of satisfaction on the entire user base (global optimization), a product manager might consider the change a failure. Thus, a delicate balance must be struck between continually designing and implementing tests that bring real performance improvements to the product and ensuring that the net aggregate effects of these optimizations are positive at the product level. This dual focus can be achieved only through the aforementioned insight: systems that measure both short- and long-term impacts and can alert a product manager to negative trends as they emerge.

Freemium products are often designed as platforms that will evolve over years; short-term thinking with respect to process improvements is necessary, but it must also be tempered by the understanding that, to continue to serve the needs and tastes of its users, a product must evolve, not rapidly transform. Small, iterative process improvements over long time frames allow for slight but impactful changes to be quickly implemented (and potentially reversed) without taking for granted users' desires for stability and consistency.

Optimization allows the freemium model to flourish by adapting the product to the needs and tastes of its users, but it must be undertaken methodically. While it is important, optimization must be seen as a tool that can be utilized in the development process and not as a development strategy itself. As an application of data and quantitative methods, optimization should not be substituted for product development; rather, it should be used to heighten the effects of already-developed product features, with the goal of achieving long-term performance improvements.

Freemium economics

The freemium business model does not exist in an intellectual vacuum; a number of established, mainstream economic principles can be used to describe the dynamics of a freemium system at the levels of both the user and the product. Since the purpose of the freemium model is to allow for scale and optimize for monetization from the most engaged portion of the user base, the focal points of an academic analysis of the model are necessarily price and supply.

Academic models are generally drafted in the abstract; although the aim of this book is to establish a practical framework for the development of freemium products, an overview of the economic principles undergirding the business model is valuable because the freemium model is so broadly applicable. Freemium products exist across a wide range of software verticals, or categories of software products grouped around a specific purpose or theme; an overly specific prescription for freemium development would necessarily be restricted to a single vertical (or perhaps even a single product) and would therefore limit the applicability of the framework. Such an approach would do a disservice to the freemium model; as a flexible, dynamic, commercial structure, viable across a range of platforms and demographic profiles, it deserves the utmost conceptual consideration. A survey of the economic principles contributing to its expansive viability is therefore necessary in providing a complete yet pliable practical framework for developing freemium products.

Price elasticity of demand

In his book, *Principles of Economics*, originally published in 1890, the economist Alfred Marshall unified a number of the disparate and nascent economic theories of the time into a coherent intellectual tapestry oriented primarily around the price effects of supply and demand. One concept Marshall posited was that of the *price elasticity of demand*, which describes the degree to which changes in price affect the volume of demand for a good. Elasticity in this sense relates to consumer responsiveness to price changes. Marshall proposed that price changes generally correlate negatively with consumer demand; that is, as the price of a good increases, consumer demand for that good decreases. This is a direct application of the law of demand as Marshall defined, which describes the inverse relationship between the price of a good and the quantity demanded, all other characteristics of that product and external market forces remaining equal.

The law of demand is generally visually represented with a *demand curve*, as depicted in Figure 1.2. A typical demand curve is constructed with price (*P*) on the

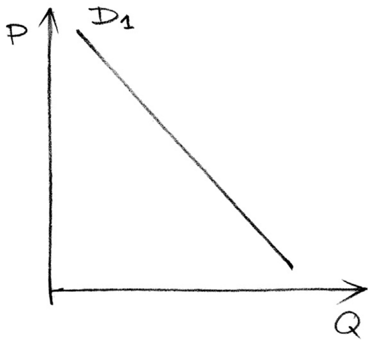


FIGURE 1.2

A linear demand curve.

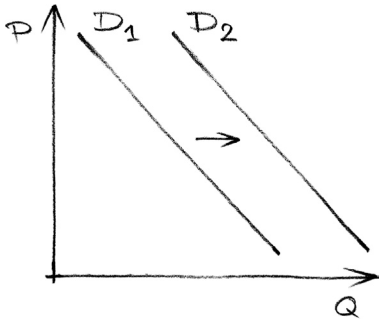


FIGURE 1.3

A demand curve shift to the right.

y-axis and quantity (*Q*) on the x-axis; the curve can be linear, indicating a constant relationship between the price of a good and quantity demanded, or nonlinear, indicating a changing relationship between price and quantity at various price points.

Movement along the curve happens when the price of a good changes but overall consumer sentiment and market conditions do not; a *demand curve shift* is said to take place when market dynamics change. A demand curve shift to the right is illustrated in Figure 1.3.

Broadly speaking, demand curve shifts for products are catalyzed by changes in consumers' abilities or desires to purchase a product, such as population migrations from rural to urban areas, worldwide economic recessions and recoveries, large-scale technological innovation, and other global or ultra-regional developments.

Within the context of market dynamics for software products, demand shifts are generally precipitated by either one or both of two factors:

- A change in the *total addressable market* for a product. In some cases this change is instituted through the increased adoption or deprecation of technology platforms (such as the increased demand for mobile applications brought about by the popularization of tablet computing), and in other cases is caused by changes in government regulations, decreased materials or production costs, etc.
- The evolution of tastes and expectations. As a product vertical matures, users become accustomed to a higher level of sophistication and functionality and develop increasingly demanding requirements on products in that vertical at a specific price point. This tends to gradually shift the demand curve for products in a vertical to the left, as rising levels of competition and consumer scrutiny reduce profit margins.

By definition, *normal goods* are those for which demand shifts in the same direction as changes to disposable income: as income increases, demand increases, and vice versa. This is at odds with the behavior of *inferior goods*, for which demand shifts in the opposite direction from that of changes to disposable income: as income increases, demand decreases, and vice versa. Inferior goods are more affordable

$$e_{(r)} = \frac{dQ}{dP}$$

FIGURE 1.4

The coefficient of price elasticity of demand.

relative to normal goods but are not necessarily of a lower quality; preference for normal goods relative to inferior goods could arise for any number of reasons.

Price elasticity of demand is measured in terms of movement along a demand curve. Marshall described this relationship as the coefficient of price elasticity of demand; this is illustrated in Figure 1.4, where *dQ* represents a percentage change in demand from one point on the demand curve to another, and *dP* represents a percentage change in price between those same two points.

When *e* is exactly −1, the product is considered to be *unit elastic*, meaning any change in price is met with an equal change in quantity demanded. When *e* falls between −1 and −∞, meaning the percentage change in quantity demanded is greater than the causal change in price, the product is considered to be *relatively elastic*: small changes in price can cause large swings in demand. This is true of goods for which many *substitute goods*, or products that are considered effective alternatives and between which consumers are relatively indifferent, exist. When*e* falls between −1 and 0, meaning the percentage change in quantity demanded is less than the causal change in price, the product is considered to be *relatively inelastic*: large changes in price are met with relatively small changes in quantity demanded. This is true of goods that are considered *necessity goods*, or products for which few alternatives exist and without which consumers' lives would be adversely affected.

Two theoretical scenarios also exist: those of *perfect inelasticity* and *perfect elasticity*. A perfectly inelastic product—a product for which $e = 0$—incurs no change in demand due to any change in price. In other words, demand is constant across all price levels and is completely insensitive to price; this represents the theoretical scenario where a supplier has a total monopoly on the production of an absolute necessity good, and demand exists at a constant level.

A perfectly elastic product—a product for which $e = -\infty$—incurs absolute change in demand due to any change in price. This means that demand is infinitely sensitive to price, representing the theoretical scenario where a product could immediately be replaced with an infinite number of perfect substitutes. That is, an infinitesimally small increase in price reduces demand to zero, and an infinitesimally small decrease in price increases demand infinitely. (See Figure 1.5.)

The coefficient of price elasticity is only positive (i.e., price and quantity demanded move in the same direction) for two types of goods: *Veblen goods* and *Giffen goods*. Veblen goods, named after economist Thorstein Veblen, are luxury goods for which quality is considered a direct function of price. These goods are generally purchased in pursuit of status and therefore exhibit reverse demand sensitivity; as the price of a Veblen good decreases, demand for that good decreases as a result of a perceived decrease in status.

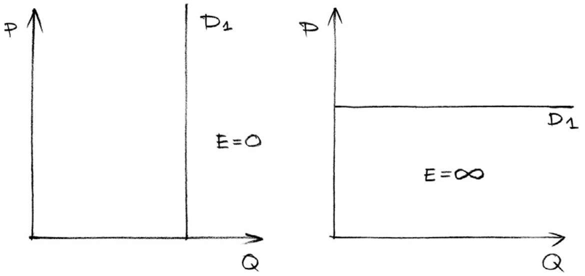


FIGURE 1.5

A perfectly inelastic product (*left*) and a perfectly elastic product (*right*).

Giffen goods, named after economist Sir Robert Giffen (although the concept was articulated by Alfred Marshall), are staple goods consumed as part of a basket of goods and for which no sufficient substitute exists. The quantity demanded of Giffen goods increases with price because, since no alternative for the Giffen good exists, the other goods in the basket are abandoned out of necessity. Giffen goods are generally inferior goods, and the phenomenon of Giffen goods usually takes place under circumstances of poverty. The prototypical example of a Giffen good is an inferior staple food item; as the price of that food item increases, consumers with constricted budgets are prevented from buying supplementary food items because they must apportion the remainder of their budgets to increased volumes of the inferior food item.

Price discrimination

The price elasticity of demand describes a dynamic where price sensitivity impacts the volume of goods sold at various price levels. But this concept is predicated on the assumption that only one price exists for any given product; that is, a supplier participating in a competitive marketplace uses only its marginal cost of production and the concept of price elasticity of demand to inform its pricing strategy and optimize total revenue.

In highly transparent markets with little barrier to entry, where a large number of suppliers produce near-perfect substitute goods, suppliers price their goods according to the intersection of the demand curve and their *supply curve*. A supply curve, like a demand curve, represents a series of points on the plane constructed with price (P) on the y-axis and quantity (Q) on the x-axis.

The supply curve slopes upward because, all other things equal, a supplier should be willing to produce more products at a higher sale price than at a lower sale price. The point at which the supply and demand curves meet is called the *equilibrium price*; this is the price P_1 at which the entire quantity Q_1 will be sold. Figure 1.6 illustrates a supply and demand curve meeting at equilibrium price EP.

While Q_1 units of the product are sold at the equilibrium price EP, the demand curve does not originate at this price point; it originates at the demand curve's intercept on the y-axis, P_0, which represents a price point at which no consumers would

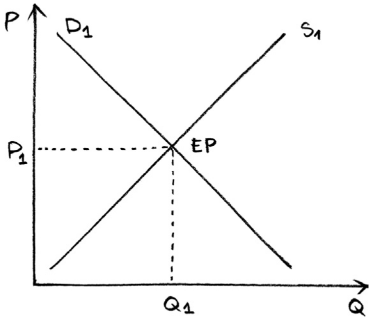


FIGURE 1.6

Consumer surplus is the area bounded by EP, P1, and the point at which the demand curve intercepts the y-axis (P0).

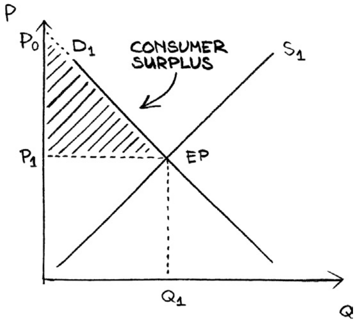


FIGURE 1.7

Consumer surplus, bounded by points *EP*, P_1 QUOTE and P_0.

be willing to purchase the product. Thus some units of the product could be sold at price levels between P_0 and Q_1; the total value of these units is the area of the polygon bounded by price points *EP*, P_1, and P_0.

This value is known as *consumer surplus*, and it represents a form of savings for consumers who were prepared to purchase the product at price points between P_0 and P_1 but were offered the product at the cheaper price of *EP*. Consumer surplus exists when a supplier imposes only one price point on its product, but the surplus can be captured by the supplier if the consumers willing to pay prices on the demand curve between P_0 and P_1 can be identified, segmented, and specifically marketed to. See Figure 1.7.

The process of offering different prices to different consumer segments, rather than pricing a product at its equilibrium price, is known as *price discrimination* (or *price differentiation*). The price point at which an individual consumer is willing to purchase a product is known as that consumer's *reserve price*; by charging consumers their individual reserve prices, down to the equilibrium price, a supplier maximizes its total potential revenue.

Price discrimination is generally considered to take three forms, as defined by economist Arthur Pigou in his book *Economics of Welfare*, first published in 1920. *First-degree price discrimination* occurs when a supplier charges consumers their individual reserve prices; this represents total consumer surplus capture by the supplier.

A few conditions must exist for first-degree price discrimination (sometimes called *perfect price discrimination*) to take place. First, the supplier must be a *price maker*; that is, the seller must exert sufficient control over the market of a good or service to be able to dictate prices. This is usually possible when the supplier is a monopoly or a member of an oligopoly participating in *collusion*, or organized price fixing. Under circumstances of perfect competition, first-degree price discrimination is difficult to achieve, as consumers being charged their higher relative reserve prices can migrate freely to suppliers charging equilibrium prices.

The second condition is the existence of a spectrum of reserve prices across a group of consumers. If all potential consumers have the same reserve price (which is usually the case in highly transparent markets, where consumers are aware of supplier profit margins and pricing schedules), then consumer surplus is 0.

The third condition is that consumer segments are isolated and cannot form secondary markets on which to sell the product. If secondary markets for the product emerge and consumers can resell the product between segments, then arbitrage opportunities will surface for consumers who were offered lower prices to sell to consumers who were offered higher prices, and the pricing systems will converge.

Second-degree price discrimination occurs when a supplier knows that multiple demand curves for its product exist but the supplier is not capable of identifying, prior to a sale, the level of each consumer's reserve price. In such a scenario, in order to increase total revenue, the supplier can establish different pricing tiers within its product without knowing about the consumers before they make purchases. A typical example of second-degree price discrimination is how tiered airline seating works: an airline knows that business travelers are willing to pay more for air travel than leisure travelers, but the airline cannot distinguish between the two groups before the travelers purchase tickets. By establishing both first class and economy seat tiers, the airline can capture additional revenues from business travelers based on their higher reserve prices by allowing them to self-select into the first class tier.

Third-degree price discrimination occurs when the supplier knows that multiple demand curves for its product exist across different groups of consumers and it can identify, prior to a sale, members of each group. In such a scenario, the supplier can establish price tiers valid only for specific groups and based on the principle of inverse elasticity, which posits that prices should be relatively high for consumers exhibiting demand curves with low price elasticity (i.e., consumers with little price sensitivity). This is known as *Ramsey pricing*, so named after economist Frank Ramsey, and is usually manifest when suppliers sell products to multiple markets, each with different levels of price elasticity. An oft-cited example of third-degree price discrimination is the relatively high price of food in airports; because airports represent a market where consumers' price elasticity is low (given the consumers' captive nature), food outlets can charge more for equivalent products inside airports than they can outside of airports.

Pareto efficiency

Pareto efficiency describes a state of resource allocation where no participant's situation can be improved upon without another participant's situation worsening. The concept is part of a broad body of work produced by economist Vilfredo Pareto in the late nineteenth and early twentieth centuries. Pareto efficiency represents a specific state of allocation that may not be equitable; that is, while no individual's situation can be improved upon without another's being impaired, a large disparity can exist between individuals' situations within the state of Pareto efficiency. The pursuit of Pareto efficiency—that is, making changes to the allocation of resources that improve the situation of one individual without worsening the situation of any other—are called *Pareto improvements*.

Pareto efficiency is measured in terms of *utility*, which is an abstract quantification of consumer satisfaction. Under conditions of finite resources, the best allocation of resources at the individual consumer level is that which maximizes that consumer's utility. This notion of maximized utility is often expressed as a curve with many points, with each point representing different combinations of goods, all of which produce the same value of utility. Such a curve is called an *indifference curve* (illustrated in Figure 1.8), so named because a consumer is indifferent to the various combinations of goods, given that they all produce the same amount of utility.

Because indifference curves describe a quantity of physical goods, they can be rendered only in the positive quadrant of a graph (i.e., the top right portion of a standard Cartesian plane, where both axes have positive values). Indifference curves

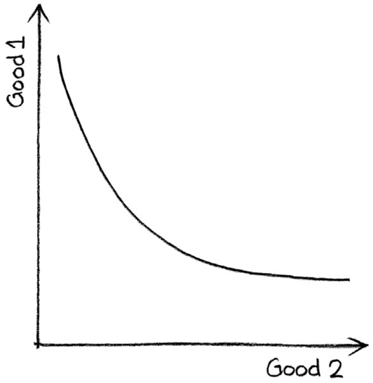


FIGURE 1.8

An indifference curve.

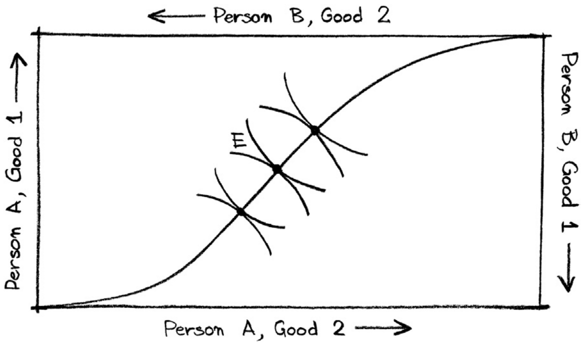


FIGURE 1.9

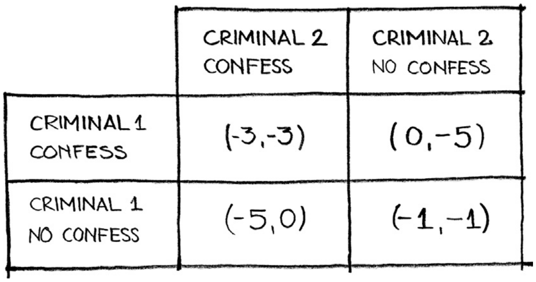
A contract curve.

are negatively sloped because resource constraints are considered concrete: the total frontier of allocation cannot extend beyond the real limitations of the consumer's ability to procure more goods. The bowed shape, or convexity toward the origin, of an indifference curve is a result of a concept called *diminishing marginal utility* (or Gossen's first law). Diminishing marginal utility describes the decreasing per-unit utility of a good that a consumer already has; because each additional unit of a good produces less utility than the unit acquired before it, the consumer is less willing to displace a unit of another good for it. The curvature of the indifference curve reflects this.

A *contract curve* is a set of points containing the intersections of all indifference curves between two parties, each trading one good, and produces a Pareto efficient outcome. A contract curve is constructed in what is known as an *Edgeworth box*: a four-axis graph on which individual parties are represented by origins at the bottom left and top right corners, with each commodity being put on an axis and running in the opposing direction to the axis across from it. Figure 1.9 depicts a contract curve between two individuals, A and B, who are trading in two goods, 1 and 2, with an equilibrium trade point of *E*.

Given that both individuals start with a discrete allocation of resources and are free to trade with each other, the contract curve evolves to represent the points at which both parties would stop trading, having reached Pareto efficiency. In game theory, the concept of *Pareto domination* is used to compare a set of points that are Pareto efficient: when one point P_1 is at least as beneficial for every individual as another point P_2, but at least one individual prefers P_2 to P_1, then P_2 is said to Pareto dominate P_1. And when a point P_X exists that no other point Pareto dominates, P_X is said to be *Pareto optimal*.

In 1950, two employees of the RAND corporation, Melvin Dresher and Merrill Flood, developed a game to illustrate the fact that a *non-zero-sum game*—that is, a set of transactions between two parties where each party's loss is not necessarily accounted for by an attendant gain by the other party—could produce a unique equilibrium outcome which is not Pareto optimal. The game became known as the



	CRIMINAL 2 CONFESS	CRIMINAL 2 NO CONFESS
CRIMINAL 1 CONFESS	(-3,-3)	(0,-5)
CRIMINAL 1 NO CONFESS	(-5,0)	(-1,-1)

FIGURE 1.10

Potential payoff options for a prisoner's dilemma.

Prisoner's Dilemma, and it serves as an example of the conflict between one individual's dominant strategy, absent knowledge of the actions of other individuals, and group rationality under non-zero-sum conditions.

The premise of the Prisoner's Dilemma, as defined in narrative form by mathematician Albert Tucker, is that two criminals have been arrested for committing a crime together and are being interrogated separately, in different rooms, by one district attorney. The district attorney realizes that he does not have enough evidence against the pair to convict them for the crime and admits as much to each one, although he notes that he has enough evidence to convict them on a lesser charge. The district attorney offers each criminal the following bargain:

- If one confesses and the other does not, the person who confessed goes free (payoff 0) and the other receives a heavy sentence (payoff −5).
- If both confess, each receives a medium sentence (payoff −3).
- If neither confesses, each receives a light sentence for the lesser charge (payoff −1).

The choices and outcomes facing the criminals are represented in Figure 1.10. The dominant strategy for both criminals, individually, is to confess; each criminal is better off confessing no matter what the other criminal does. But this outcome is non-Pareto optimal, as both players would be better off not confessing; if both criminals confess, they're both worse off than if neither had confessed.

The conceptual foundation of the Prisoner's Dilemma lies at the heart of many social phenomena and informs modern digital pricing strategy, given low or zero marginal distribution and production costs. When deciding whether or not to reduce the price of its products, a developer may reason that a price cut could attract a rival developer's customers to its own products, compensating for the loss of per-unit income with sales volume. Likewise, if the developer believes its rival will cut prices, it reasons that it must also cut prices to remain competitive. In either scenario, the developer's dominant strategy is to cut its prices; when the rival engages in the same logic, it also cuts prices. The end result is a loss for both developers, having forced prices down without capturing additional market share.

An eventual price floor emerges in such a scenario, usually representing the developers' marginal costs of production and distribution, around which the market converges. Once this price floor has been established as a standard, ascending above it is difficult, given consumer expectations and the fear of market share capture by rival developers.

Freemium product case studies

There is perhaps no better means of understanding the fundamental principles of the freemium model than by examining them through the lens of highly successful freemium products. Certainly there is no shortage of successful freemium products from which to draw case studies; an entire book could be written consisting of nothing but case studies.

The three products presented in these case studies were selected based on a specific set of criteria. The first criterion was that the product achieved scale in a relatively short amount of time through viral features. In considering only products that achieved appreciable virality, each case study provides a context against which the discussion in chapter 7 can be considered. This requirement dismisses products with user bases developed primarily through paid advertising (which is outside the reach of most small freemium developers).

The second criterion was that the product was available on multiple platforms, if not at the time of initial launch, then shortly thereafter. The goal of this book is to provide a framework for developing freemium products, irrespective of the platform on which the product is designed to operate. By choosing products that work across a broad range of platforms (and, with some products, a broad range of devices), the implementation of the freemium model, and not the idiosyncrasies of the product's platform, are isolated for examination.

The third criterion was that the product achieved considerable scale. As has been discussed, while the potential for massive scale is a hallmark of the freemium model, many freemium products have achieved success with modest user bases composed of zealous, enthusiastic users. But these products tend to appeal to a narrow demographic scope: users with generous amounts of both disposable time and income. In order for a freemium case study to be broadly applicable to a large number of product verticals, it needs to appeal to a diverse cross-section of potential users.

The case studies are presented at the beginning of the book to frame what follows and to add depth to the topics explored later in the book by way of example. While they are not exhaustive, these case studies do cover a wide range of use cases, geographic points of initial development, and company sizes. The cases were selected with care to serve as a practical introduction to the freemium business model.

Skype

Skype is a software application that facilitates Voice-over-Internet-Protocol (VoIP) calling functionality. In its initial incarnation, VoIP allowed users to speak to each other through the Skype application over the Internet; over time, the application evolved to provide a rich feature set such as video calls, file transfer, group calls, and chat. Skype eventually grew into a platform used by more than 600 million people and, after a tumultuous period (which included multiple complicated transactions that changed the ownership structure of its parent company, as well as a formal filing by its parent company for an initial public offering [IPO] and the subsequent retraction of that filing), Microsoft acquired Skype's parent company in 2011 for $8.5 billion.

The origins of Skype are inextricably linked with the origins of Kazaa, a peer-to-peer file sharing service developed by three Estonians, Jaan Tallinn, Ahti Heinla, and Priit Kasesalu, using their proprietary peer-to-peer technology framework, FastTrack P2P. Kazaa operated without the use of a central routing server; instead, users of the service formed a network across which files were transferred, with the speed of transfer dependent on the number of users online. While Kazaa was eventually shuttered, having faced a battery of legal issues related to the trade of pirated and otherwise illegal materials made possible by the program, the core technology powering the product lived on as the functional core of Skype.

Kazaa and its FastTrack P2P technology platform were purchased by entrepreneurs Niklas Zennström and Janus Friis (from Sweden and Denmark, respectively), who, in turn, sold Kazaa to Sharman Networks in 2002. Zennström and Friis, along with the Estonian developers of FastTrack P2P, began developing Skype soon afterward. The first beta version of Skype was released in August 2003; it was a free download exclusively for the Windows operating system. Using the software, two users could call each other using their computers' speakers and microphones.

Like Kazaa, Skype's network consisted of no central servers; calls were routed through the network established between users. As a result, the larger the network grew, the more stable and robust it became; similarly, Skype calls between users in sparsely populated areas were capable of higher fidelity than some cellular telephone network operators were able to provide, given that those regions are generally underserved by cellular coverage. In June 2004, Skype launched a beta version of its product on the Linux operating system, and in August 2004, Skype launched a beta version for Macintosh computers.

Skype eventually added chat functionality and video conferencing to its client (all of which was available for free) and introduced its initial monetization mechanic with paid calls to physical phones via traditional phone networks in what it called SkypeOut. Paid calls could be made from a Skype client to any phone number around the world; because the international rates Skype charged were often cheaper than those offered by telecommunications companies, Skype, over time, became a major carrier of international telephone volume. From 2005 to 2012, Skype's share of worldwide international telephone call traffic grew from 2.9 percent to 34 percent.

From its earliest days, Skype's user base growth was meteoric. By 2006, the service had grown to 100 million users; by the third quarter of 2009, that number had ballooned to 500 million. But while revenues grew with the user base, profit was elusive: in its filing for an initial public offering in August 2010, Skype's parent company noted that only 6 percent of Skype users contributed revenue (although average revenues for that user group sat at $96 per year) and that its net income for the first half of 2010 was a mere $13 million on revenues of $406 million.

Throughout its history, Skype had experimented with various revenue streams, including Skype-branded phones, but by the time it filed paperwork for an IPO (the last point at which public financial records for the company are available), the bulk of its revenue stream was represented by SkypeOut calls. In 2010, Skype's board appointed a new CEO: Tony Bates, a veteran of Cisco. Bates' ambition was to increase Skype's ability to monetize by increasing the scope of Skype's brand appeal from a purely consumer platform to a fully capable business telecommunications solution. This was achieved through a number of product initiatives, the most prominent of which was probably the paid Skype Premium account upgrade, which allowed a user to organize video conference calls between more than two participants, among other things. Before its acquisition by Microsoft, Skype also introduced advertising products that users saw during phone calls.

Skype's success materialized despite the difficulties it faced in generating revenue. because its user base grew at an impressive, continuous clip from inception. From the very beginning, Skype was an inherently viral product; it couldn't be used without adding contacts, and as the service became synonymous with free telephone functionality, it spread virally, primarily through word of mouth. In fact, the word Skype eventually evolved into a verb in colloquial use; to "Skype" someone means to call them via Skype.

This rapid adoption likely contributed to the challenges Skype faced in monetizing users; as the product became more and more commonplace, fewer users needed to use its paid SkypeOut functionality. In other words, the growth of its free product essentially cannibalized revenues from its paid product, as both products essentially served the same use case: voice communication between people. The proliferation of smartphones in Western markets most likely exacerbated this dynamic, because not only did many users carry a Skype-capable device with them everywhere, but that device allowed for free video calling, whereas SkypeOut calls to physical phones did not.

By the time Microsoft acquired it, Skype boasted incredible breadth: 600 million global users (100 million of them active each month) across a variety of devices ranging from desktop computers to mobile phones and tablets. Users were using Skype to make telephone calls, host video conference calls, and send SMS messages. Skype had grown to form a truly integral layer of the modern communications amalgam when it was purchased and had accomplished it all with a fairly humble number of employees (500 as of the time of its IPO filing) distributed throughout multiple offices around the world.

Spotify

Spotify is an application that allows a user to stream music over the Internet. The application initially launched with two tiers of functionality: free and premium. The free tier, which launched as invite-only, imposed limits on usage, exposed the user to advertising, and was only available on the desktop; the premium tier offered ad-free, unlimited usage across multiple devices. Spotify's parent company had negotiated licensing agreements with a broad range of record labels, both large and small, upon its initial launch, which ensured that all usage of the service complied with copyright restrictions.

Spotify's parent company, Spotify AB, was founded in Sweden in 2006 by Daniel Ek and Martin Lorentzon, who had worked together previously at a Swedish marketing company called Tradedoubler. The genesis of Spotify was precipitated by a discussion between Ek and Ludvig Strigeus, the Swedish developer behind uTorrent, one of the most popular clients for BitTorrent, the peer-to-peer file sharing protocol. Ek, who had been interested in pursuing a music streaming project, realized that Strigeus' peer-to-peer expertise was the key component to developing such a service, and, together with Lorentzon, purchased uTorrent from Strigeus, only to sell it to the company behind BitTorrent. Ek and Lorentzon subsequently retained Strigeus to develop the peer-to-peer framework that would later evolve into Spotify.

Spotify initially experienced impressive but measured growth: by the end of 2009, a little more than a year after its initial launch, the service had garnered 6.5 million registered users. The service's growth was limited by a number of factors, most of which are the result of the dynamics of a two-sided marketplace. Spotify is essentially a medium through which content consumers (the application's users) connect with content providers (music labels); in order to facilitate the provision of content from providers to consumers, Spotify must manage relationships with the legal entities that control how music labels engage with intermediaries.

Because of the labyrinthine legal and procedural morass through which Spotify must continually navigate, the service has been rolled out on a country-by-country basis (usually in groups of countries). This natural impediment to large-scale, universal adoption was unavoidable: users in countries where agreements had not yet been struck between Spotify and the commercial associations that dictate the terms of streaming and downloading music were simply not permitted to install the application.

Music labels, too, had to be courted. To accomplish this, Spotify enlisted the help of perhaps one of the most highly visible luminaries in the technology sector at the time: Sean Parker, one of the original founders of Napster and the founding president of Facebook. Parker actually approached Spotify first, professing his admiration for the service in a 1,700-word missive to Daniel Ek. Shortly after, Parker invested $15 million into Spotify through the Founder's Fund, a venture capital firm Parker was a partner in. The investment led to Parker assuming a board seat; it was in this capacity that Parker negotiated key agreements between Spotify and the Warner and Universal music labels.

But perhaps the biggest coup Parker engineered was a partnership between Spotify and Facebook. From its initial launch, Spotify facilitated social sharing through linking: users could link to a specific song or a playlist they had created, linking by email, in chat, or on social networks like Facebook. The inclusion of these initial social features helped propel Spotify's growth, but the company's partnership with Facebook took that virality channel a step further: at Facebook's annual f8 conference in 2011, Sean Parker announced that Facebook and Spotify had come to an agreement to more deeply integrate Spotify into Facebook's "open graph," the data interface that facilitates sharing on Facebook.

Spotify's partnership with Facebook, which came mere months after the service launched in the United States, contributed to an intense surge in growth that year. From September 15th, 2011 (a few days before the Facebook partnership was launched) through December 2012, Spotify's user base doubled, from 10 million to 20 million users. Even more impressive was Spotify's rate of conversion; on December 6th, 2012, Spotify announced that 5 million of its 20 million active users, or 25 percent, were paying subscribers. A month earlier, Spotify raised approximately $100 million on a valuation of $3 billion from investors including Goldman Sachs and Coca-Cola.

Spotify's remarkable percentage of paying users was likely the result of its product strategy; in October 2009, Spotify offered an "offline mode" for its premium product tier, allowing users to download songs to their devices for non-streaming listening as long as their subscriptions were valid. And in May 2010, Spotify expanded its product portfolio to include an "unlimited" option, which provided the same feature set as the premium option but was limited to desktop clients, and an "open" option, which provided a reduced-functionality version of the free tier (with the free tier remaining invite-only). The free account types were combined into one in April 2011, with the amount of music accessible to free-tier users limited to 10 hours per month. New users, however, were not subject to this limitation for their first six months of product use.

Spotify's success as a freemium product can be attributed to the breadth and ingenuity of its product portfolio and the virality it experienced from its social features. Upon its initial launch, Spotify's product strategy involved engaging with the user on an unpaid basis, establishing regular use patterns, and then enticing the user to upgrade to a paid account tier. And once users had upgraded, they were incentivized to keep their subscriptions current through Spotify's offline mode, which only allows a user access to their downloaded songs while their subscription is active (the downloaded songs are deleted upon subscription lapse). By bundling cross-device functionality based on payment—paid accounts provide access through an unlimited number of the user's devices—the most engaged users are prompted to upgrade in order to access their music whenever they want. Aligning monetization with enthusiasm allows Spotify's premium users to feel that they are not only unlocking additional functionality with their payments, they are also unlocking omnipresence.

In terms of distribution, Spotify's viral, social strategy served as the engine of its initial growth. The ability to link to objects within the application allows users to simultaneously share the efforts of their playlist creation and incite new users to adopt the product. And through its partnership with Facebook, Spotify allows its viral dispatches to be associated with the feel-good sentiment of new artist discovery: what are essentially advertisements for the service on Facebook can be interpreted by their recipients as a courtesy by Spotify. This passive viral strategy was extremely effective in instilling initial growth in Spotify's user base, and by incorporating social features into the product, Spotify ensured that its growth would experience compounding effects.

Candy Crush Saga

Candy Crush Saga is a puzzle game published by King, the London-based social game developer. The game was released on Facebook in April 2012 and experienced considerable success on that platform, claiming the top position (in terms of daily active users, or DAU) from Zynga, which Zynga had held non-stop for years, in January 2013. The game soared on mobile, where it was released for both the iOS and Android platforms in November 2012; in December 2012 alone, the game was downloaded more than 10 million times. Candy Crush Saga breached the Top 10 Grossing list, overall, for the iPhone in the United States—a considerable feat—on December 4th, 2012, and reached the number 1 position on March 11th, 2013. Such positions on the Top 10 Grossing list, which sorts applications by revenue, are indicative of substantial daily revenue generation, usually in the range of hundreds of thousands to millions of dollars.

King was founded in 2003 as Midasplayer International Holding Co. with the intention of developing web-based games, primarily for Yahoo! but also for its own website and other portals. The company was founded by Riccardo Zacconi, Patrik Stymne, Lars Markgren, Toby Rowland, Sebastian Knutsson, and Thomas Hartwig, six former employees of a company called Spray. Spray had ambitions of launching an initial public offering in March 2000, but those plans were sidelined when the market for stock listings soured in the face of the burst of the dotcom bubble, and the company was sold. Three years after the sale, the six co-founders, with Zacconi at the helm as CEO, formed Midasplayer International Holdings Co., and in 2005, the company raised nearly $50 million from investors Apax Partners and Index Ventures. Shortly after the investment, the company rebranded as King.com.

King.com published a wide variety of games on its own website and across a number of partner websites and portals. The company specialized in what it called “skill games”—primarily puzzle, strategy, and board games—and built a competitive social platform on its site where members could participate in tournaments and compare their performances against each other through leaderboards. The impetus for the change in brand name was a player ranking system that bestows titles onto users, ranging from “Peasant” to “King,” based on the users’ achievements in the company’s portfolio of casual games.

The rise of Facebook as a gaming platform provided King.com with an additional medium on which to market and distribute its games, and the company quickly emerged as one of the social network’s top gaming providers. In July 2012, King.com boasted 11 million DAU on Facebook, behind Zynga (at 48 million) but ahead of Electronic Arts (at 9.5 million). At that time, King.com’s largest game, measured by DAU, was Bubble Witch Saga, a witch-themed bubble shooter puzzle game first launched on Facebook in September 2011. Bubble Witch Saga was King.com’s first foray into mobile: the game was released for iOS in July 2012.

Candy Crush Saga's release on Facebook in April 2012 catalyzed a sea change on Facebook's gaming platform; less than one year after release, in March 2013, King.com's Candy Crush Saga took the number 1 position from Zynga's FarmVille 2. And a month later, in April 2013, the company, which had rebranded itself simply as King on its tenth anniversary in late March 2013, announced that it had surpassed Zynga in terms of DAU on Facebook, with 66 million daily users to Zynga's 52 million.

Candy Crush Saga was a major contributor to King's ascension up the Facebook charts. The game belongs to the "match 3" puzzle family, which describes a game board filled with shapes of varying colors; the positions of any two of the shapes can be swapped to create either a vertical or horizontal line of three or more similarly colored shapes. When those shapes are matched, they disappear from the board, and the shapes above the void created by their absence drop, with new shapes appearing at the top of the board. Each match earns the player points, and various combinations, special shape matches, and lengths of matches can engage point multipliers that increase the player's total score.

In Candy Crush Saga, the shapes are candies, and the game progresses through levels separated into episodes. Levels are completed by satisfying either a points requirement or a predetermined objective (such as clearing all candies of a specific type). Players in Candy Crush Saga are given a set of five lives, which regenerate every 30 minutes; a life is consumed each time a player attempts a level but either does not meet the objective or does not meet the minimum score required to obtain at least one star. (Based on performance, a player can earn up to three stars on any given level.) When a player's lives are depleted, three options are presented to the player: (1) wait for a new life to regenerate, (2) ask an in-game friend to provide the player with a life, or (3) purchase a life from the in-game store.

This set of options represents one of two "choice gates" at the heart of Candy Crush Saga's success. Because the game was initially developed for Facebook, it is inherently social; players can view and connect with their friends on the network by default. As the difficulty of the game increases (and, despite its youthful aesthetic, the game becomes challenging quite early on), the importance of being connected to friends in-game becomes paramount. In fact, borrowing lives from friends (and gifting them in turn) is a core component of the game; rather than detracting from the experience, the interactions between friends enhance it. This core social communication mechanism is an effortless and organic source of significant virality for Candy Crush Saga.

The second "choice gate" in Candy Crush Saga occurs at the end of certain episodes, when players are required to either invite friends to the game or pay money to progress to the next set of levels. This gate sets a hard, decisive contribution condition on each player reaching that point in the game, through either virality or direct monetization. These choice gates and gifting opportunities dovetail with a competitive element to the game; players are shown their performance on a per-level basis relative to their friends' performances, and players can see their friends' progressions through the game.

Candy Crush Saga is uniquely monetized compared to other popular freemium social games; the product catalogue is limited and is priced directly in platform currency as opposed to being priced in an in-game hard currency. With a product catalogue consisting only of one-time-use gate advancements and boosts—which improve the player's performance in a single level—and a small number of permanent performance-enhancing accessories called charms, Candy Crush Saga focuses on long-term retention and user base scale, as opposed to product catalogue depth, to deliver revenue.

Retention and scale certainly haven't escaped the game. In May 2013, Zacconi announced that Candy Crush Saga was generating 500 million gameplay sessions per day on mobile devices alone. That level of engagement is likely due in part to the massive amount of content available in Candy Crush Saga; since its launch, King has released a new episode for Candy Crush Saga every few weeks, each one consisting of 15 levels. Upon launch of the game, the player is shown the vastness of the map over which levels are dispersed; the volume of levels and potential for gameplay hours is obvious from the first session of play.

As a game, Candy Crush Saga's combination of a proven entertainment archetype with social and competitive features contributes significantly to retention, and its gifting and gate mechanics inspire substantial virality. But in terms of engagement, King's decision to develop the game as truly persistent across all platforms—meaning a user's progress is tracked across all devices the game is played on—probably had the most profound effect on its staying power with highly engaged users. Because game progress is never lost from one device to another, the most engaged players can advance in the game on their phones, tablets, or desktop computers whenever they have the time to play it.

Candy Crush Saga is perhaps one of the most iconic games of the mobile era, given its inherent social functionality and its cross-platform support. The game proved that casual games can generate massive volumes of revenue, given a large enough user base and a compelling enough social infrastructure. And the game also proved that a developer with an employee count in the hundreds—not thousands—can develop a hit mobile franchise and maintain a competitive chart position for months on end.