Gate Me If You Can: The Impact of Gating Mechanics on Retention and Revenues in Jelly Splash

Thomas Debeauvais
UC Irvine
tdebeauv@uci.edu

Cristina V. Lopes
UC Irvine
lopes@uci.edu

ABSTRACT

Current free-to-play mobile games use various mechanics to control player progression. Through the game Jelly Splash, we study three such gating mechanics: occasional spikes in level difficulty, chapter gates every 20 levels, and a limited number of lives regenerating slowly over time. Using telemetry data from a quarter-million players, we explore how players approach each gating mechanic, focusing on retention and revenues. We find that difficulty spikes and chapter gates boost revenues at the expense of retention. The life mechanic seems to have no effect on retention and revenues.

Keywords

Retention; Gating; Jelly Splash; Mobile Games; Free-to-play; Purchases;

Categories and Subject Descriptors

K.8.0 [**Personal Computing: General**]: Games

1. INTRODUCTION

The last few years have seen the rise of mobile gaming in the West. There are now more mobile game players than traditional console or PC game players [17]. As of January 2015, Clash of Clans, one of the most played mobile games, is estimated to have 6 million daily active users[1]. The commercial success of mobile games may be attributed to their free-to-play (F2P) business model. F2P mobile games aim at attracting as many players as possible, and expect that enough of them will purchase some amount of virtual currency with real money. To this end, F2P game designers use certain monetization game mechanics to make players spend money.

Monetization mechanics oftentimes revolve around controlling player progression so players spend money to progress faster. For example, F2P designer Aki Jarvinen recommends to "design for pay to progress, but balance for grind to commit" [12]. A theoretical framework from Zagal et al. distinguishes three fundamental ways for designers to control and structure gameplay: 1) temporally, by limiting or coordinating in time what players can do, 2) spatially, by breaking the game world into zones, and 3) based on challenges, by making the player face a puzzle, boss fight, or bonus level [25]. As shown in table 1, most monetization mechanics in mobile games segment gameplay in one of these ways.

Figure 1: Screenshots of Jelly Splash. Level 13 (left) and world map showing the level-100 gate (right).

Controlling player progression is not a new idea in game design. In action-RPG and FPS level design, it is called gating. Gating means confining players in a certain area until they learn a new skill or complete a certain task, such as finding the key for a door or solving a puzzle [4, 10]. In this paper, we refer to gating mechanics broadly as mechanics controlling player progression.

Some F2P designers report that gating mechanics are very effective at generating revenues, but they also cause players to leave the game, a behavior also called churn [13]. Industry and academia have started focusing not just on revenues, but also on retention [2, 3, 13, 18, 21, 22]. In this paper, we try to answer the following questions: Which gating mechanics drive revenues? Which are most likely to make players churn? Are there trade-offs?

[1]See `http://thinkgaming.com/app-sales-data/1/clash-of-clans/`

Mechanic	Examples	Segmentation
Time-based	Speed-ups in Clash of Clans	Time
Energy points	Stamina in Puzzle and Dragons	Time
Luck-based	Crafting gear in Angry Birds: Epic	Time
Collecting/Gacha	Card Packs in Hearthstone; Egg machine in PaD	Time
Ads (voluntary or forced)	End of a round in Words With Friends	Time
Expansion packs/DLC	Naxxramas Wings in Hearthstone	Space
Chapter gate	Mystery quests in Candy Crush; Stars in PvZ 2	Space and Challenge
Continues/Mulligans	Arcade and endless runner games	Challenge
Power-ups	Rare gems in Bejeweled Blitz	Challenge
Customization	Vanity items; Skins in League of Legends	None

Table 1: Common F2P monetization mechanics and their type of segmentation.

2. STUDY CONTEXT AND METHODS

2.1 Jelly Splash

To answer these questions, we look at Jelly Splash, a F2P mobile game developed by Wooga, a German game company. Jelly Splash launched for iOS, which includes iPhones, iPads, and iPods, on August 22, 2013. The game quickly reached number 1 on Apple's AppStore in the US, Germany, and several other countries [19]. The game was installed 25 million times between August 2013 and April 2014, and had 8 million monthly active players in December 2013 [21].

At its core, Jelly Splash is a tile-matching game similar to Bejeweled or Candy Crush. As shown in figure 1, the game consists of connecting 3 or more same-colored tiles, called jellies, to remove them and make new ones drop from the top of the board. The core loop is designed to be 70% luck and 30% skill [21]. The game shipped with 140 levels, and 20 new levels are added roughly every other month. Each level is a puzzle where the player has to complete a certain goal in a limited number of moves to unlock the next level. Goals include reaching a certain score, making 2 to 6 diamonds fall to the bottom of the board, or clearing a certain number of gray cells.

Jelly Splash has several monetization mechanics. When players run out of moves in a game, they can spend 70 in-game coins (approximately $1) to receive 3 extra moves, or they lose a life. Players have at most 5 lives, and one life regenerates every half hour. To refill all their lives immediately, players can spend 100 coins or login through Facebook, the only social platform supported by the game, and ask their friends for lives. Players can also spend 70 coins to unlock each of the gates stationed every 20 levels starting at level 40, or ask three of their friends for gate keys. So among the monetization mechanics detailed in table 1, Jelly Splash launched with:
1) Extra Move Requests (XMR), a continue mechanic tailored for the luck-based core loop of the game [21],
2) A limit of 5 lives, an energy mechanic,
3) Chapter gates every 20 levels, starting at level 40.

2.2 Methods

We select 5% of the Jelly Splash iOS players based on their device id, a number assigned by the hardware manufacturer and free of any sampling bias. This sub-sampling makes data analysis tasks more tractable, while retaining a very strong statistical validity [7]. An update in November 2013 significantly altered the chapter gates, so we restrict our data to the 10 weeks from launch on August 22 until October 31, 2013. The in-game telemetry tracks when users install the game, login, finish a level (whether won, lost, or abandoned), and purchase virtual currency. The telemetry also tracks when players log into the game with their Facebook account, and how many of their friends are playing. The telemetry does not track any data that can be used to identify a player's Facebook account.

Real-life telemetry data can be messy. We have to clean it up. First, we discard data that is erroneous (e.g. some players purchased a billion coins), inconsistent (e.g. some players finished a level before they installed the game), or even lost (e.g. some players won level 6, but have no data for level 5). Second, we searched Jelly-Splash-related forums for known exploits, cheats, and technical issues. An early exploit allowed some players to obtain a potentially infinite number of virtual coins by re-installing the game. Therefore we discard data from players who installed the game more than once. Some players manipulated their device clock to refill their lives faster. Thus we ignore data from players who have no Facebook friends playing the game and who managed to refill their lives without spending coins. After cleanup, we have data from 273,819 players. These players played 25 million games in 7 million sessions, and made 37,170 purchases.

This study is exploratory and descriptive. We want to find trends happening in the game, not predict when players are going to churn, or how much they will purchase. This is what is referred to as descriptive data mining [8]. To this end, we use basic statistical tools and graphs. Unless reported otherwise, all results are significant at the $p<.001$ level. For correlations, we often cannot report Pearson's linear correlation coefficient, because a lot of our exploratory plots show obviously non-linear relationships between variables. Thus we report Spearman's rank correlation coefficient ρ instead. We follow [5] when estimating effect sizes: when ρ is between .1 and .3 we consider the effect weak, between .3 and .5 medium, and above .5 strong.

2.3 Data overview

The average game session lasts 7 minutes (sd 14, median 4, 99% below 31). Users play 3 sessions per day on average (sd 3). The time between two sessions is very short: 20% of inter-session times are below a minute, and 99.2% below

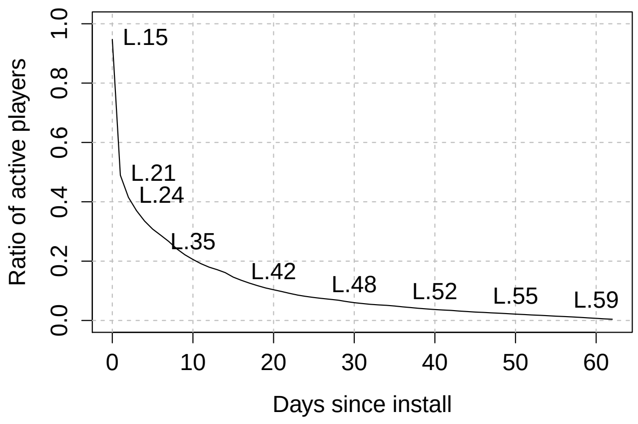

Figure 2: Ratio of active players X days after install. The average level reached by active players is indicated for days 0, 1, 2, and multiples of 10 since install.

a week. Previous work on Diamond Dash, another mobile game, found that 98% of players stay away from the game for less than 14 days, and thus defined churn as 14 days of inactivity [18]. We map this definition to our study, and consider that people who do not play for a week have taken a break from the game (vacation, tired of the game, and so on). With this definition and its limitations in mind, the average player leaves the game 13 days after install (sd 16, median 8). Figure 2 plots the ratio of users still playing the game against the number of days after they installed the game. For example, 21% of the player base is still playing the game 10 days after having installed it, and these players have reached level 35 on average. Around 5% of players install the game but never play. Half of players churn within a day after having installed the game, reaching only level 15.

3. DIFFICULTY SPIKES

Traditionally, game difficulty ramps up progressively as players learn new skills, and higher levels are generally harder than lower levels [1]. The designers of Jelly Splash took a different approach. In Jelly Splash, difficulty is not designed to increase monotonically, but rather to spike at certain levels [21].

3.1 Level difficulty

The Jelly Splash designers [21] measure the difficulty of a level by the average number of tries that players take to win it:

$$Difficulty = avg(number\ of\ tries\ until\ won)$$

We look at data aggregated by level. Each of the 140 rows is indexed by level number, and contains aggregates such as the number of players who ever reached the level, the number of players who leave at that level, the average number of tries for players to win the level, the total number of coins bought at that level, and the total number of extra move requests (XMR) made at that level. With this dataset, the average level takes 8 tries to pass, or 4 hours worth of lives. Level 97, the most difficult level in the game, takes 71 tries. The blue bars in figure 3 show the spikes in difficulty from levels 10 to 30. The spikes in churn coincide with the spikes in difficulty. Although the figure does not show levels below 10 or above 30 to remain readable, the spikes continue to visually coincide until level 140.

Despite the spikes, the overall difficulty increases as the levels go by ($\rho(138) = .41$). Therefore, we have to control the level number when assessing how player behavior relates to difficulty. To do so, we use partial correlations instead of regular correlations. Partial correlations measure the association between two variables while holding constant a third variable. In our case, controlling for level number, we observe very strong partial Spearman correlations between difficulty and churn ($\rho(138) = .71$), between difficulty and the number of coins purchased per user ($\rho(138) = .81$), and between difficulty and the number of XMR per user ($\rho(138) = .76$). This is intuitive: players are more likely to purchase coins and spend them in extra moves in more difficult levels, where they need them.

3.2 Level hopelessness

To differentiate "good" difficult levels from "bad" difficult levels, the Jelly Splash designers [21] measure a level's hopelessness as:

$$Hopelessness = avg(\frac{number\ of\ tries\ until\ won}{number\ of\ tries\ nearly\ won})$$

When players lose a difficult level with high hopelessness, they have usually completed only few of the level's goals. For example, they only bring one diamond down, out of the six required. When players lose a level with low hopelessness, they have usually completed most of the level's goals, such as five out of six diamonds. Levels with low hopelessness are "good" because they give players feedback that the goal is within their reach [21]. However, the designers' definition of "nearly won" is imprecise. Does it mean that players, without spending coins in XMRs, complete 50% of the level's goals? Or 90% of the level's goals? Determining this threshold is probably more art than science. For this study, we arbitrarily consider a try to be "nearly won" when the player completes at least 75% of the level's goals without using extra moves.

As a level metric, hopelessness has several limitations. First, hopelessness for a particular level is undefined for players who never "nearly win" that level. Second, the 75% threshold is arbitrary and difficult to justify. Third, hopelessness only makes sense for difficult levels. Players pass an easy level in so few tries that they probably do not have time to notice its hopelessness. A fourth limitation is that the hopelessness of some levels can not be calculated. Levels 1 to 4 are tutorial levels with infinite moves, so they can never be lost. Level 132 has only one recipe to make in 15 moves, which means it can be lost, but it can never be "nearly won", whatever threshold we pick.

With these limitations in mind, the average level hopelessness is 4, meaning that the average level makes players nearly win once every 4 tries. According to the Jelly Splash designers, a "good" difficult level has a hopelessness below 10 [21]. Hopelessness, much like difficulty, spikes occasionally, and increases with level ($\rho(132) = .47$). Hopelessness also increases with difficulty ($\rho(132) = .64$). When we con-

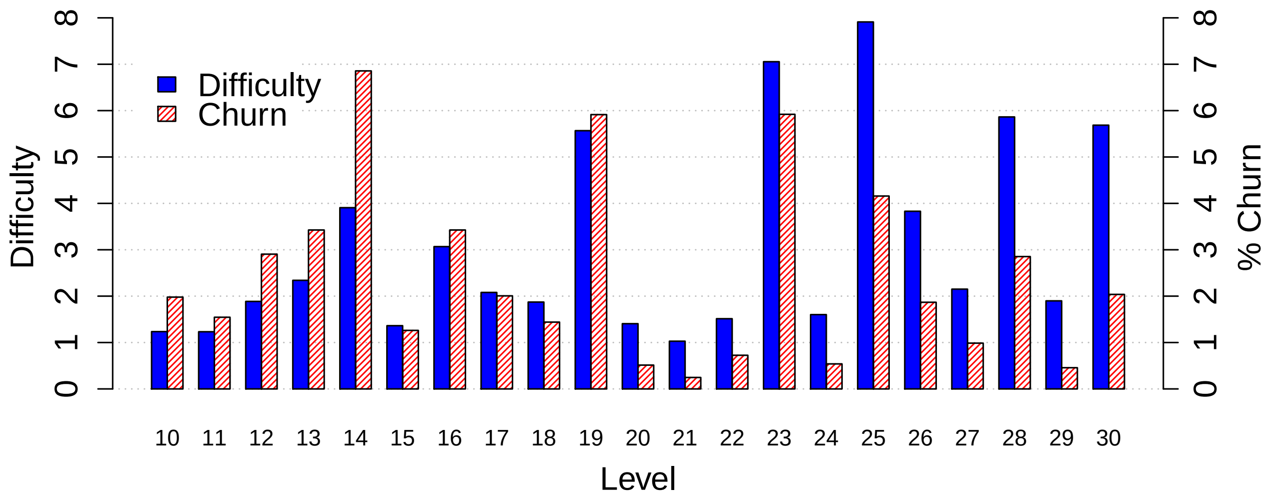

Figure 3: Difficulty and churn for levels 10-30. A level's difficulty is the average number of tries that players take to win that level. Churn is the number of players who stop at a level, divided by the number of players who have ever reached that level.

trol level number and difficulty on the relationship between hopelessness and churn, we find a very modest partial Spearman correlation ($\rho(132) = .13$, $p < .05$). Similarly, controlling level number and difficulty, hopelessness is strongly correlated with XMR per user ($\rho(132) = .54$) and coins bought per user ($\rho(132) = .55$). In other words, for two equally difficult levels, the most hopeless one has more purchases, more spending, and probably also more churn.

3.3 XMRs and board rerolls

Two game mechanics can help players mitigate luck in difficult levels: XMR and board rerolls. We just saw that XMR are more likely to happen in difficult and hopeless levels. To see how helpful XMR are to players, we look at the won or lost but not canceled games after the tutorial ends, at level 5. Our dataset comprises 23 million games. We find that games in which players do not request any extra moves are won 34% of the time. For games with one XMR, the win rate is 38%, two XMRs 69%, three XMRs 83%, and 10 XMRs 87%. This progression suggests two take-aways. First, players hardly increase their winning chance with only one XMR. Difficult levels may be designed to be reliably won with 2 or 3 XMRs, but not with one. Second, players seem to severely misjudge their winning chance when using extra moves. Why spend 70 coins (worth $1) in an XMR if it does not really increase the chances of winning?

Players can cancel a game without losing a life if they have made no move. We call this a board reroll. In a game designed to be 70% about chance [21], this mechanic should be very useful to players. Aggregating by players the dataset of 25 millions games, we find that 18% of players ever reroll a board. And among the players who reroll, 60% reroll only once, which may be by mistake. These results can be interpreted in two ways. The first assumes that the developers never intended to allow board rerolls. In the game by mistake, that mechanic is actually an exploit abused by nearly one every five players. This is a lot, and it should be fixed.

Dep. var.	β Level number	β Difficulty	β Is Gate	Adj. R^2	F
Churn (in %)	-.04	.10	11	.50	47
Num. purch.	.002	.009	.093*	.59	68
Coins purch.	1.0	3.5	45*	.52	51

Table 2: Results of three linear regressions. All p-values are below .0001 except for * where p is .092.

The other way to interpret the results is to assume that the reroll mechanic was intended, or at least considered innocuous, since the developers never removed it. In that case, many players seem unaware of a mechanic that could increase their odds of winning a level. This may be a user interface issue, since there is not really a button for it.

Summary: Hopelessness and difficulty are correlated. Both spike, increase with levels, and are positively correlated with churn and purchases. XMR and board rerolls mitigate luck, but players use them ineffectively, if at all.

4. CHAPTER GATES

Chapter gates happen every 20 levels, starting at level 40. Passing a gate requires 70 coins or asking 3 Facebook friends for keys. Thus we have to look at retention and purchases, and also Facebook logins. To drive our analysis, we try to confirm or reject the following hypotheses: 1) Gates have a negative influence on retention. 2) Gates are responsible for a large portion of all revenues. 3) Gates are effective at making players login with Facebook.

4.1 Retention

The average churn rate for non-gate levels is 1.4% (median 5%, max 9%). Gate levels have much higher churn: around 31% of players who reach level 40 churn at the first gate, and 28% who reach level 60 churn at the second gate. At first glance, gates are terrible for retention.

We saw in the previous section that level number and difficulty are correlated with churn. So if we want to measure the relationship between churn and gates, we need to control for level number and difficulty. We perform a linear regression using the level data with 140 rows. The dependent variable is churn, measured as a percentage of the player base that ever reached the level. The independent variables are the level number, the level difficulty, and whether the level is a gate. The results of this regression are shown in the first row of table 2 and can be interpreted as follows: controlling for level number and difficulty, churn in gate levels is 11 points higher than in non-gate levels.

Being stuck at a gate does not prevent players from replaying previous levels. One could argue that players can replay the levels below 40 and stay forever. The data suggest it is not the case. Among the players who churn at the level-40 gate, 12% do so right when they reach the gate (they have 0 replays), and 50% leave before 15 replays. Among the players who churn at the level-60 gate, 13% do so right when they reach the gate, and 50% leave before 14 replays. So the argument that players can replay previous levels is moot: players churn very quickly and do not replay much of the previous levels when they reach a gate.

4.2 Purchases and Spending

Similarly to churn, we perform two linear regressions using the 140-row level data. In the first, the dependent variable is the number of purchases made by the average player at a given level. In the second, the dependent variable is the number of coins purchased by the average player at a given level. In both, the independent variables are the level number, the level difficulty, and whether the level is a gate. The results are given in table 2. Controlling for level number and difficulty, gate levels increase the number of purchases by .093, and the number of coins per purchase by 45, compared to non-gate levels.

Gates are an effective conversion mechanic. In the e-commerce industry, a conversion is the moment when a website visitor makes a purchase [11]. The free-to-play game industry uses the metric as well, but only for the first purchase, not for the subsequent ones, if any. In Jelly Splash, 42% of all conversions happen at the level-40 gate, and 14% at the level-60 gate. These two gates are responsible for more conversions than all other levels combined.

We can also look at spending patterns through figure 4. We call non-buyers the players who do not purchase coins, and small buyers those who ever make a single $2-purchase of 140 coins. Both types of players start with 70 "free" coins. Only 11% of non-buyers spend the 70 coins they start with to unlock the level-40 gate. Maybe they do not anticipate that there is going to be a gate to unlock. Or maybe most churn before reaching the gate. Small buyers are very different. First, 28% of all the coins they spend go into gate unlocks. Second, 44% of small buyers pass exactly 1 gate using 70 coins, and 22% pass exactly 2 gates using 140 coins. Therefore gates are effective at making small buyers spend their coins.

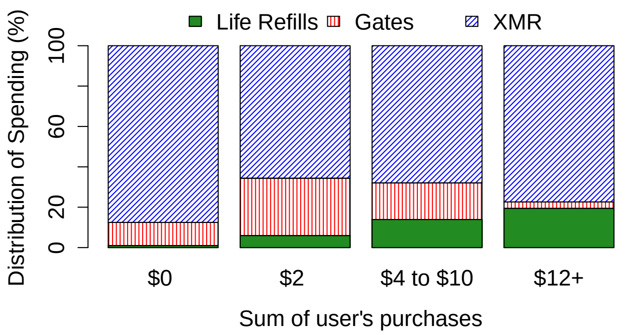

Figure 4: Distribution of in-game spending for non, small, medium, and big buyers. XMR stands for extra move requests.

Group	Ask Friends	"Free" Coins	Purchase Coins	Never Pass
FB Login	30%	34%	9%	27%
No FB Login	–	49%	5%	46%

Table 3: Behavior at the level-40 gate for players who are logged-in with Facebook versus those who are not: pass by asking friends for gate keys, pass by spending the 70 coins offered at the beginning of the game (no purchase required), pass by spending purchased coins, and do not pass.

4.3 Facebook logins

Gates are a critical milestone in the game in terms of retention and purchases. But the designers may have introduced them in the game to foster virality as well. To analyze the whole impact of gates on players, we look at Facebook logins, since gates are the first time the game explicitly asks players to login with Facebook and use their Facebook friends.

To ask for keys at level 40, players need to login or be already logged-in with Facebook. Nearly a quarter of players login with Facebook at some point during the game. Are gates responsible for these Facebook logins? Three quarters of Facebook logins happen before the level-40 gate, and only 12% of all Facebook logins happen at the level-40 gate itself. Players who login with Facebook do so on average 3 days after installation (median 5 hours). So at first glance, it seems that players login early or never. But much fewer players reach level 40 compared to level 5. If we control for the number of players who reach a level, figure 5 shows that the level-40 gate triggers 9% of the players who reach it to login with Facebook. Facebook logins at the level-60 gate also spike. Hypothesis 3 is confirmed.

If the level-40 gate encourages players to login with Facebook, does it also make logged-in players ask their friends for keys? Table 3 shows that only 30% of the players logged-in with Facebook manage to unlock the gate with friend keys. This could be explained by the fact that a quarter of the players who login with Facebook have no friends playing the game. This itself could be explained by the fact that we only look at the first 10 weeks after the game launched.

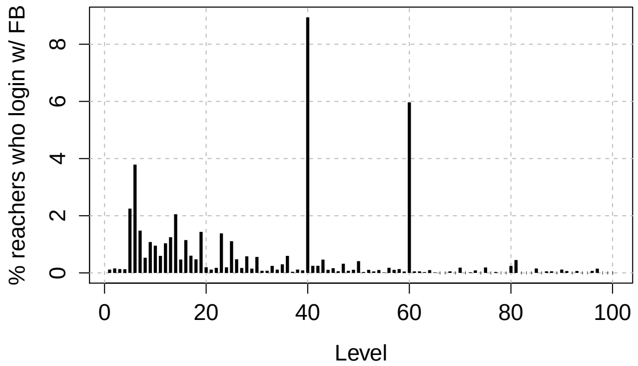

Figure 5: Percentage of players who login with Facebook at a given level among the players who ever reached that level.

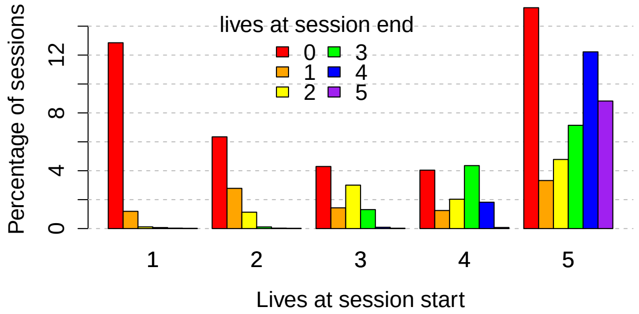

Figure 6: Histogram of sessions based on the number of lives at the start and end of the session.

Summary: Gates at levels 40 and 60 are effective at converting free players into spenders, and at making players who reach them login with Facebook. However, they represent a very small fraction of big buyers' spending, and have high churn rates.

5. ENERGY/LIVES

We have three hypotheses: 1) Lives do gate player activity. 2) Players are more likely to churn when they reach 0 lives than when they have 5 lives. 3) Players are more likely to buy coins when they reach 0 lives than at 5 lives.

According to figure 6, 43% of sessions end with 0 lives. This confirms hypothesis 1. Moreover, this number also shows that players very frequently run out of lives, and are often presented the possibility to refill their lives using coins. Looking at churn, lives do not seem to matter. When people churn, 20% churn with 0 lives, 55% with between 1 and 4 lives, and 25% with 5 lives. This rejects hypothesis 2. Regarding purchasing, 23% of purchases happen when players have 0 lives, 57% between 1 and 4 lives, and 23% at 5 lives. So we have to reject hypothesis 3 as well.

To better understand the impact of lives on players, we can look at the impact of lives on sessions. Having only 5 lives caps the amount of play per session. Players who lose five times in a row cannot play for half an hour. Much like arcade games of old, players have to spend money to refill their lives and continue playing. Thus in the rest of this section, we look for session factors related to churn and purchases.

5.1 Methods

We saw above that both chapter gates and spikes in difficulty influence churn and purchases. So we have to include level-specific data such as difficulty when analyzing sessions. The problem is that most sessions involve lower levels (because players churn way before they reach level 140). So if we want to include level variables, we need a way to control for level. Statistically speaking, this means we need a mixed-effects model. Moreover, our outcomes of interest are binary: will the player churn at the end of the session or not, and will the player purchase during the session or not.

Variable	Churn	Purch.
Lives at start of session	+	+
Lives at end of session	+	−
Level reached at session end	−	+
Ratio of wins during session	−	+
Levels unlocked during session		+
Session number	−	−
Games played during session		−
Difficulty of level reached	+	−
Hopelessness of level reached		−
Level reached is a gate	+	+

Table 4: Sign of the log-odds ratios from two mixed-effects logistic regressions. Session variables on top, level variables at the bottom. A +/- indicates that players are more/less likely to churn or purchase. Empty cells have p>.001.

We use one logistic regression for each of these two binary prediction tasks. Thus, our models are mixed-effects logistic regressions.

To be able to compute these models in a reasonable time, we use a sampling technique known as partial deep telemetry [7]: we decrease our sample size from 5% to .1% of the player base, but keep all information concerning these players. If we look at sessions from 5% of the player base, we have more than 7 million sessions, ie 800-MB worth of data. Shrinking our player set to .1%, ie 5,119 players[2], gives 139,017 sessions, ie 15-MB worth of data. This smaller dataset makes computations much more tractable while retaining a large-enough sample.

The two resulting models are shown in table 4. In the first regression, the dependent variable is churn likelihood at the end of the session. In the second, it is the purchase likelihood during the session. A +/− indicates that an independent variable is positively/negatively correlated with the dependent variable, controlling for all other variables. To run the mixed-effects regressions, we use the package lme4 in the statistical environment R.

[2]We used device id modulo 1000 = 0 to maintain an unbiased sample.

5.2 Churn

Table 4 shows that, controlling for all other dependent variables, players are more likely to churn at the end of a session the more lives they have at the end of the session. This rejects our second hypothesis again. It could be that players do not churn when they run out of lives, but simply when they get bored with the game, maybe after they replayed easier levels in which they did not lose any life. In any case, lives as a gating mechanic does not seem to harm retention.

Table 4 also presents several findings that are unrelated to lives, but confirm what we saw previously. For example, the likelihood to churn at the end of a session is higher in the first sessions than in later sessions, and it is also higher in earlier levels than later levels. Basically, players either leave the game right away or get hooked. This early-or-never pattern is not new; it was previously reported with respect to assist selection in Forza Motorsport 4 [6].

5.3 Purchases

As shown in table 4, purchases are more likely to happen during sessions that start with many lives and end with few lives. It could be that players who purchase coins use all their lives in one session, and eventually refill and use all their lives again. This reflects a more intense play style. Sessions in which players do not purchase coins may consume only one or two lives, which reflects a slower-paced play style.

As mentioned before, the purchase likelihood increases with the level number. Purchases are also more likely when the level reached is a gate. We saw earlier that purchasing and spending coins happen at the same time. The purchasing model confirms this: when players purchase during a session, they win and unlock more levels. Surprisingly, they also play fewer games, maybe because sessions without purchases involve more replays of easier levels.

Compared to extra move requests, life refills are a luxury item. First, they cost more: 100 coins compared to the 70 coins for an extra move request or to unlock a gate. Second, figure 4 shows that the proportion of spending in life refills increases with the amount of coins purchased. While small buyers spend 6% of their coins in life refills, big buyers spend 19% of their coins in life refills.

Summary: Lives do gate players, but do not seem to have an impact on churn or purchases. Life refills are a luxury item probably used by the more intense players.

6. DISCUSSION

Table 5 summarizes the impact of the gating mechanics covered in this paper. Based on these results, this section discusses community-building mechanics, trade-offs between retention and revenues, ways to tweak the life limit, and adaptive luck. We also list several limitations inherent to our study, and outline future research directions.

Community-building mechanics: Gates are the most prominently social mechanic in the game, and yet very few players end up interacting with each other through them. The Jelly Splash designers may have realized that, as they tweaked the gate mechanic in November 2013. Instead of friends, gates now require a certain number of stars obtained by winning the previous levels. Furthermore, Wooga has decided to build their community not through the gates, but rather around the most difficult levels. The Jelly Splash designers purposefully make the level difficulty spike because frustrating levels make players complain on social networks, and fosters virality [21]. Winning a particularly frustrating level probably gives players a feeling of personal triumph over adversity, also known as fiero, one of the four core emotional drivers in games [14]. So Wooga community managers run Facebook campaigns such as “The makers of level 75 have a special place in Hell” to foster community building around these levels.

	Retention	Revenues
Difficulty spikes	−	+
Level hopelessness	−	+
Gates	−	+
Lives limit	0	0

Table 5: Summary of the overall impact of gating mechanics on retention and revenues.

Retention vs revenues: For Wooga, retention matters more than monetization [22]. The designers of Diamond Dash, another match-3 mobile game by Wooga, report that focusing on revenues too aggressively is not sustainable [20]. Our findings suggest that there may be a trade-off between retention and revenues. In Jelly Splash, the spikes in difficulty generate revenues, but they also cause churn. The Candy Crush designers also noticed this phenomenon: level 65 used to be notoriously difficult, and churn was 40% at that level. When the designers reduced the level’s difficulty, churn was halved [15]. But revenues for that level were probably halved as well. So increasing revenues and retention at the same time seems difficult, but there may be solutions in game design. For example, the chapter gates covered in our study successfully convert many non-buyers. But the players who do not convert end up churning because 1) they have no other way to pass the gate, and 2) they have no incentive to replay previous levels. The tweak made to the gates in November 2013 solves both issues.

Saving lives: Energy-based economies are very popular in free-to-play mobile games [13]. In Jelly Splash, half of sessions end with 0 lives, but most of the virtual currency goes into extra move requests and gate unlocks, not life refills. Therefore, making sessions last longer may improve retention while preserving revenues. This could be done by making it easier to pass a level with one star, but keeping it difficult to pass with two or three stars. Players will be able to play more, and win more often, but they will not progress faster. Progress ultimately remains gated by the chapter gates tweaked in November 2013 to require a certain number of stars to unlock.

Adaptive luck: Matching-tile games like Bejeweled Blitz, Jelly Splash, and Candy Crush could benefit from research on adaptive difficulty [16, 23, 26], and especially adaptive difficulty focused on retention [9]. For example, if a player loses several lives on the same level in one session, the game could reduce the difficulty of a level by tweaking the color of the tiles falling from the top of the board. The game industry does not wholeheartedly agree. For example, the Candy Crush designers loudly defend that "all players are treated equally", and "the candies are added to the board randomly" [15]. While the designers of Jelly Splash openly say that their game is 70% about luck, so as to slow the more skilled players and help the less proficient ones [21], the gaming press has called their design philosophy "manipulative"[3]. Zagal et al. have also found that it is very easy for commercial game designers to use time-based, luck-based, and social game mechanics unethically [24]. Thus, even though tweaking luck to fit player skill is acceptable in academic game prototypes, it seems more controversial in commercial F2P games.

Limitations: First, the time frame of the study is the ten weeks immediately following launch. The players in our dataset are early adopters. Early adopters and late adopters may play differently. For example, early adopters have very few friends playing with them when the game launches. So they ask their friends for life refills less often than late adopters do, and may therefore be more likely to spend coins to refill their lives. A second limitation is the scope of the study: one F2P tile-matching mobile game. The results found in this paper may be directly applicable to Candy Crush or other F2P tile-matching mobile games. But they may not directly apply to other genres, platforms, and business models. A last limitation of our study is erroneous coin counting. The game sometimes awards players 40 coins as a promotional gift. Some players are awarded this gift multiple times. We cannot distinguish between a player who was awarded multiple gifts, and a player who purchased the same amount of coins but whose transaction failed to be recorded. These gifts are unpredictable. We discarded players who installed the game multiple times because they could get a potentially infinite number of coins while not losing any progress. But there may be other hacks that we are not aware of.

Future work: To complement this paper, and address some of the limitations mentioned above, future research could explore the following directions. First, are there significant differences between cohorts of players, such as early adopters compared to late adopters? Second, F2P game analytics distinguish between organic and paid users. Organic users land on the App Store page to install the game spontaneously, by themselves. Paid users land on the page through web or in-app advertising that the game company paid for. Do organic and paid users play differently, or respond to different mechanics? Third, this paper brushed on differences between players logged-in with Facebook and those who are not, especially when reaching chapter gates. Are there other significant differences between players logged-in with Facebook and those who are not? Finally, can tile-matching games benefit from adaptive luck in terms of retention, purchases, and overall player enjoyment?

[3]See `http://gamebreakingnews.com/jelly-splash-developer-woogas-design-philosophy-might-be-manipulative-but-at-least-its-honest/`

7. CONCLUSION

This quantitative paper contributes an exploratory study of retention and revenues in Jelly Splash, a free-to-play mobile game. The game uses three main mechanics to gate player progression: spikes in level difficulty, chapter gates every 20 levels, and a limit of 5 lives, regenerating slowly over time. We show that 1) chapter gates are the main mechanic making players purchase coins for the first time, 2) level difficulty is the main driver for player spending, and 3) lives do gate players, but seem to have little impact on revenues and retention. Overall, our findings suggest an inherent trade-off between revenues and retention.

Acknowledgments

We thank Markus Steinkamp and Jesper Richter-Reichhelm from Wooga for making their data available to us.

8. REFERENCES

[1] E. Adams and A. Rollings. *Fundamentals of Game Design*. Prentice-Hall, Inc., 2006.

[2] C. Bauckhage, K. Kersting, R. Sifa, C. Thurau, A. Drachen, and A. Canossa. How players lose interest in playing a game: An empirical study based on distributions of total playing times. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 139–146. IEEE, 2012.

[3] Z. H. Borbora and J. Srivastava. User behavior modelling approach for churn prediction in online games. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 51–60. IEEE, 2012.

[4] J. Bycer. A look at negative game mechanics. `http://www.gamasutra.com/blogs/JoshBycer/20120903/177027/A_Look_At_Negative_Game_Mechanics.php`, September 2012.

[5] J. Cohen. A power primer. *Psychological Bulletin*, 112, 1992.

[6] T. Debeauvais, T. Zimmermann, N. Nagappan, K. Carter, R. Cooper, D. Greenawalt, and T. Soldberg. Off with their assists: An empirical study of driving skill in Forza Motorsports 4. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '14, 2014.

[7] A. Drachen, A. Gagné, and M. S. El-Nasr. Sampling for game user research. In *Game Analytics*, pages 143–167. Springer, 2013.

[8] A. Drachen, C. Thurau, J. Togelius, G. N. Yannakakis, and C. Bauckhage. Game data mining. In *Game Analytics*, pages 205–253. Springer, 2013.

[9] B. Harrison and D. L. Roberts. Analytics-driven dynamic game adaption for player retention in Scrabble. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pages 1–8. IEEE, 2013.

[10] C. Hauteville. Gating. `http://technicalgamedesign.blogspot.com/2011/04/gating.html`, April 2011.

[11] B. Heathman. *Conversion Marketing*. 2012.

[12] A. Jarvinen. Free to play, tricky to design (presentation). In *Northern Game Summit*, September 2012.

[13] J. Jordan, B. Cousins, E. Levy, M. Katkoff, N. Lovell, M. Faurholm, J. Stolzenberg, and M. Robinson. Is there any place in the F2P designer's toolbox for energy-based economies? `http://www.pocketgamer.biz/mobile-mavens/59247/is-there-any-place-in-the-f2p-designers-toolbox-for-energy-based-systems/`, June 2014.

[14] N. Lazzaro. The 4 keys 2 fun (presentation). In *Game Developers Conference*, GDC, 2004.

[15] T. Palm. Candy Crush Postmortem: Luck in the Right Places (Presentation). In *GDC*, 2013.

[16] C. Pedersen, J. Togelius, and G. N. Yannakakis. Modeling player experience for content creation. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(1):54–67, 2010.

[17] Renatus. Mobile and social gaming industry: 2014 highlights. `http://renatus.com/mobile-social-gaming-industry-highlights-2014`, December 2014.

[18] J. Runge, P. Gao, F. Garcin, and B. Faltings. Churn prediction for high-value players in casual social games. In *Computational Intelligence and Games*, CIG, 2014.

[19] E. Seufert. Profitably Launching Jelly Splash to #1: A Marketing Postmortem (Presentation). In *Game Developers Conference*, GDC, 2014.

[20] A. Sinoy. Game mechanics that keep players playing (presentation). In *Games Monetization Europe*, May 2013.

[21] F. Steinhoff. Jelly Splash: Puzzling your way to the top of the AppStore (presentation). In *Game Developers Conference*, GDC, 2014.

[22] D. Takahashi. Wooga's Jens Begemann on making the transition from social to mobile games (interview). `http://venturebeat.com/2014/03/22/woogas-jens-begemann-on-making-the-transition-from-social-to-mobile-games-interview/`, March 2014.

[23] J. Togelius, R. De Nardi, and S. M. Lucas. Towards automatic personalised content creation for racing games. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 252–259. IEEE, 2007.

[24] J. Zagal, S. Bjork, and C. Lewis. Dark patterns in the design of games. In *Proceedings of Foundations of Digital Games*, FDG, 2013.

[25] J. P. Zagal, C. F. Vara, and M. Mateas. Rounds, levels, and waves: The early evolution of gameplay segmentation. *Games and Culture*, 3, April 2008.

[26] A. Zook and M. O. Riedl. A temporal data-driven player model for dynamic difficulty adjustment. In *AIIDE*, 2012.