Harri Heinisuo

GACHA MONETIZATION MECHANICS

Customizable simulator for random draws

Faculty of Information Technology and Communication Sciences
Master's Thesis
June 2022

ABSTRACT

Harri Heinisuo: Gacha monetization mechanics: Customizable simulator for random draws
Master's Thesis
Tampere University
Master's Degree Programme in Computer Science
June 2022

Mobile and video games are offered increasingly often as a free-to-play or freemium service. A gacha game is a Japanese term used for a subset of such games that implement a variety of random draw mechanics. These include a wide variety of random draw systems embedded into the gameplay. Most of them require or allow the spending of a real-world currency along with virtual in-game currencies accumulated via actual gameplay.

This study looks at some of the mechanisms and features present in these systems by examining several popular or noteworthy gacha games and related literature. Specific attention is given to the implementation of random draw. Gacha systems are usually found in games made in Japan and the rest of Asia but similar systems from elsewhere in the world are also looked at briefly.

This study also aims to create a customizable simulator that simulates the random draws of a gacha reward system and outputs data about the process. A number of presets are included to simulate some existing gacha systems as accurately as possible while customization options allow for fine-tuning and experimentation. The customizable simulator expands on the idea of popular community-made summon simulators by providing more insight into the inner workings of gacha and loot box mechanics. The customizable simulator can be used by players to help plan their purchases and by developers to plan, create or evaluate their own monetization systems. It can also be used for research by anyone interested in random draw monetization systems or random draw in general.

Keywords: Gacha, Random Draw, Monetization, Micropayment, Loot Box, Free-to-play

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TABLE OF CONTENTS

1 Introduction

Mobile and video games are offered increasingly often as a free-to-play or freemium service [Nieborg 2016]. One of the more popular types of such games in Japan are called *gacha (ガチャ)* games. This name originates from *gashapon (ガシャポン)*, a type of coin-operated vending machine that dispenses random capsule prizes [Shibuya et al. 2015]. Similarly, gacha games implement systems that provide random in-game rewards against virtual and real-life currencies [Yamakami 2012]. Since the early 2010s, these games have always ranked among the most profitable mobile games in Japan and the rest of Asia [Shibuya et al. 2015]. More recently, even some high-profile computer and console games have adopted this type of monetization [MiHoYo 2020].

Games utilizing these mechanisms are also increasingly subject to public and legal scrutiny, with many critics drawing links to gambling and even organized crime. In Japan, politicians and some members of the press have accused gacha game developers of exploiting teenagers and children. Many countries require full or partial transparency about the odds and some types of gacha have been outright banned in Japan [McCaffrey 2019].

In this study, I look at different types of monetization mechanisms within gacha games. The focus of this study is the main feature of gacha games: the *random draw*. It is often the primary or even the sole monetization feature but usually it is supplemented by several others.

Many gacha games have community-built simulators for testing their mechanics before investing real-world or in-game currencies into their monetization systems. I have expanded on this idea by creating a customizable browser-based simulator that performs random draws. Instead of a specific game, the customizable simulator can simulate multiple gacha and loot box systems. The basic functionalities include creating your own categories with their own chance percentages. The simulator can adjust individual probabilities at fixed intervals or introduce modifiers that change the probabilities after specific conditions are met. It can utilize very large sample sizes and display detailed charts and information on reward distributions. Additional information such as the cost of draws and verbose logs are also displayed on demand.

I have included some presets with pre-filled reward categories and modifiers inside the simulator that simulate the random draw systems of different games. I have selected some noteworthy existing gacha games as well as an original scenario for testing with the simulator. I evaluate these results, consider the benefits of the simulator, and assess its usefulness for three user groups: players, developers, and researchers. For players, the often time-limited nature of gacha systems requires a monetary investment under a time pressure. The simulator could offer a way to assess risk and the available rewards without spending real-world currencies. For developers the simulator could offer a way to test and design their own reward systems and in-game economies. For researchers it could offer a customizable way to study the percentages and values of random draw systems.

This study aims to answer the following two research questions:

1. What are the benefits of using a customizable simulator for random draws in testing, creating and studying gacha systems?
2. How can the customizable simulator for random draws help players, designers and researchers of gacha systems?

The structure of this thesis is as follows. In chapter 2 I give some brief background information on the origins, status, and notability of the gacha mechanic.

Chapter 3 is dedicated to gacha game monetization mechanics. I explain virtual currencies, rewards, gacha variations and the gacha interface. I also go over some terminology and monetization methods of gacha games.

Chapter 4 is dedicated to the customizable simulator and its specific requirements are listed here. I also cover the inspirations, features, and development of the simulator.

Chapter 5 introduces the presets. The chapter includes the background information of each game selected for testing. I also go over the settings of their presets and why they were selected.

In chapter 6 I go over the results of each test. I explain and analyze the results given by the customizable simulator in each of the four scenarios.

In chapter 7 I evaluate if the research questions were answered and if the requirements of the simulator were met. I consider usefulness and benefits generally and for each of the three user groups: players, developers, and researchers. I also offer some final thoughts on how the simulator performed as well as some future development ideas.

2 Background

The Asia-Pacific region is by most estimates the largest mobile gaming market in the world today. The Japanese video- and mobile game market is the third largest in the world by many industry estimates, behind only China and The United States. In 2021 the Japanese gaming market had an estimated revenue of more than 22 billion euros. In mobile games specifically, yearly consumer spending in Japan amounted to 12.8 billion euros by some estimates [Kuzuhara 2021; Statista 2021].

Mobile games worldwide mostly utilize *free-to-play* monetization. Games are offered free of charge and funded by advertisements or micropayments [Nieborg 2016]. Traditionally, western mobile game monetization systems have focused on offering specific purchases where the buyer always knows what they are receiving. The Japanese monetization systems since the early 2000's have instead focused on offering a "lucky draw" instead of a specific, direct purchase [Yamakami 2012]. This *gacha* mechanic would generally become known as the *loot box* outside the Asia-Pacific region [Schwiddessen 2018]. In Japan this mechanic predates the introduction of smartphones and was common in Japanese mobile games from very early on [Koeder and Tanaka 2017].

Because gacha mechanisms became widespread in Japan earlier than anywhere else in the world, they were also the target of media and government scrutiny much earlier [Schwiddessen 2018]. Many incidents involving trading of virtual items on the real-world marketplaces and children spending large amounts of money on gacha mechanisms caused the Japanese Consumer Affairs Agency to take notice. In response to this, many Japanese game developers introduced self-regulative measures aimed to protect children. A council was formed by some of Japan's largest social game developers and networks in 2012 to self-regulate the monetization mechanisms used in their products [Toto, 2012b]. Their goals included introducing payment caps for anyone below the age of 20 and restrictions on in-game market trading. Companies willingly removed some of their own gacha mechanisms that were deemed most harmful by the Japanese press and public.

There have also been attempts by Japanese lawmakers to classify gacha mechanisms as gambling although studies suggest there are few similarities [DeCamp 2020]. Gambling also typically rewards the player with real-world currencies, which is not the case with gacha games [Yamakami 2012; Shibuya et al. 2015].

To address transparency concerns, companies are now required to disclose the probabilities of obtaining rewards in gacha systems [Toto 2012a]. Shibuya et al. [2019] found that the transparency of probability information is “important for the social acceptance of gacha”. This requirement has been adopted by many other countries as well as app stores on iOS and Android [Liu 2019]. For this study in particular, this transparency is very helpful when replicating the rules of a gacha system.

3 Gacha game monetization mechanics

The exact definitions of a gacha system vary, but they generally all follow at least one basic rule: give the user a randomized virtual item or *reward* after they invest a set amount of a specified virtual currency or currencies [Yamakami 2012]. Different variations, probabilities and rulesets can cause each separate gacha system to behave very differently. Each single use of a gacha system is called a *draw, pull* or a *spin,* referring to real-life gambling devices such as slot machines and the roulette wheel [Yamakami 2012; Kanerva 2016]. In this study I am using the term *draw*.

3.1 Virtual currencies

The purchases in a gacha system are usually done with a virtual currency [Koeder and Tanaka 2017]. Gacha games can offer multiple virtual currencies. For example, in Genshin Impact [MiHoYo 2020] the gacha mechanic works with a virtual currency called *primogems*. This currency can be earned by playing the game. Instead of offering an option to directly buy primogems with a real-life currency, the game offers another virtual currency instead called *genesis crystals*. This purchasable currency can then be converted into the earnable currency, primogems, with a 1 to 1 ratio. This additional layer of obfuscation may partly exist because the purchasable currency can also be used for other direct shop purchases such as cosmetics. The earnable primogems are therefore less versatile than the store-bought genesis crystals.

In World Flipper [Cygames 2019] there is a clearer distinction between the earnable currency and the shop-bought currency. While both are called with the same name, the shop-bought version is prefixed as *paid*. As seen in image 3.1, the paid virtual currency may also be used to do a daily discounted draw. In World Flipper's main gacha system the paid and earned currencies are valued equally. However, when the player uses the gacha system, the earnable currency is spent first.

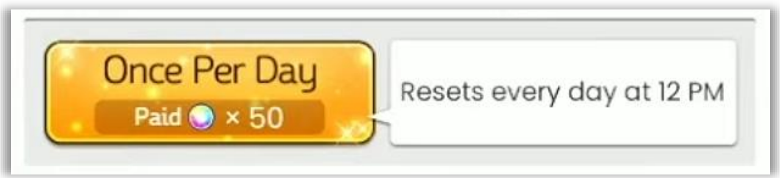


Image 3.1 An additional mechanic in the World Flipper gacha system.

There may also be multiple additional items that can be used to make draws in a gacha system. For example, in Arknights [Hypergryph 2019] there are additional complementary items that can be exchanged for draws. They are sometimes given to players as event and log-in prizes. Some games such as Genshin Impact [MiHoYo 2020] favor using the main currency as the complementary prize instead of an additional item.

The most straightforward way to obtain a virtual currency in a gacha game is to purchase it with a real-world currency [Koeder and Tanaka 2017]. Gacha games usually offer several tiers of *packs* containing different amounts of the virtual currency used for the gacha mechanic and other in-game purchases. The prices of these packs can range from 1 to 100 euros or more. Typically, the cost of each unit of currency is lower the larger the pack purchased is. Sometimes the ratio is not linear to make specific options more tempting than others. These packs can offer discounts or extra currency for first-time buyers. In image 3.2 is a view from the Arknights [Hypergryph 2019] in-game shop. Each pack of the virtual currency offers an extra amount when it is bought for the first time. Since each pack has a better real-world currency to virtual currency exchange ratio each time it is first bought, it can incentivize the player to purchase several of the different available options instead of just one.

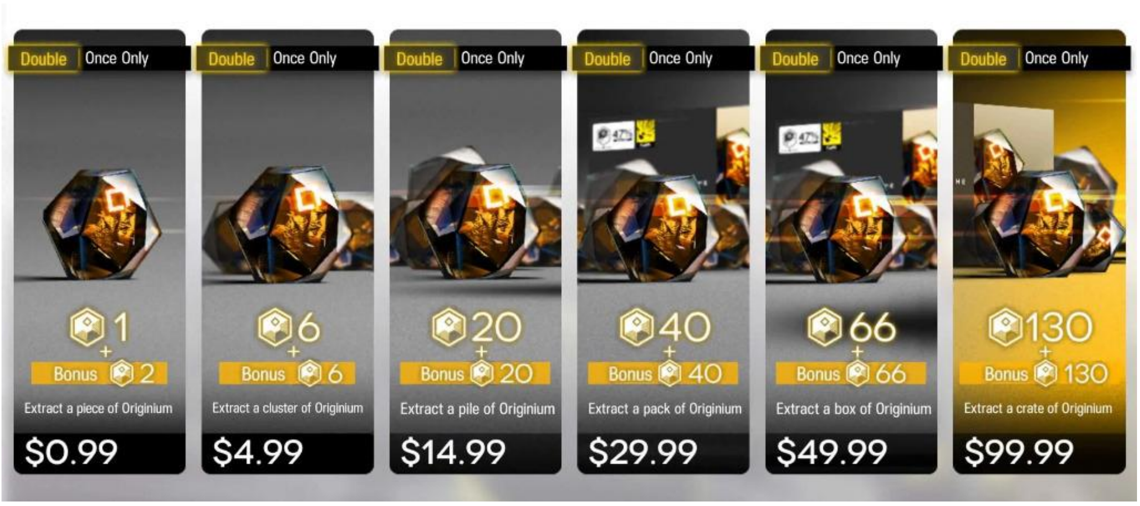


Image 3.2 A view from the Arknights in-game shop showing packs of virtual currency.

Commonly gacha games also offer subscription-based and *play-to-earn* monetization options [Yamakami 2012; Liu 2019]. Currencies are offered on discount but may not be immediately available for use. The players may have to log in daily to receive their currencies. They may have to play the game for a specified amount of time or perform specific tasks within the game.

In image 3.3 is an example of a subscription-style purchase from Genshin Impact [MiHoYo 2020]. It is a monthly subscription that gives the player a large amount of the virtual currencies used in the game. However, the player does not receive these currencies immediately. The purchase provides the player with 300 genesis crystals (purchasable currency) immediately and a total of 2700 primogems (earnable currency) staggered across 30 days. This requires the player to log in every day or their daily batch of 90 currency is wasted. Compared to the purchasable packs the ratio of real-life currency to virtual currency is more favorable to the customer. This is perhaps one of the least demanding play-to-earn monetization models for the player, but it still requires at least starting the game. This can lead to more participation and additional spending.

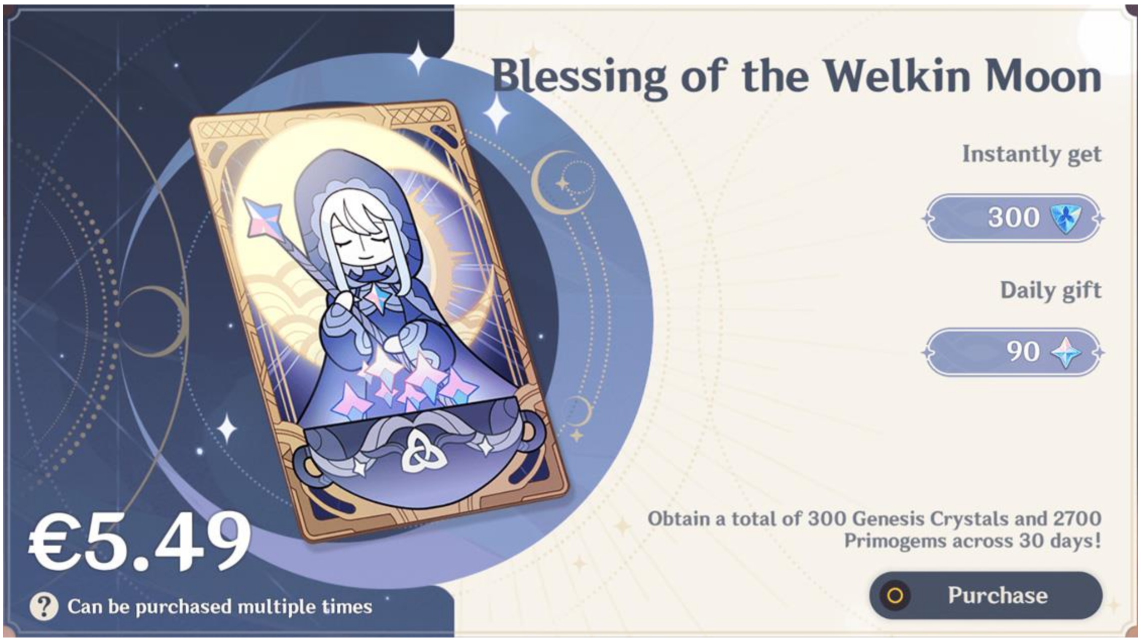


Image 3.3 A subscription-style in-game shop item from Genshin Impact.

3.2 Rewards

The virtual rewards that a gacha system offers can be new playable characters, weapons, or other in-game items. The types of these rewards on offer can vary greatly in each game. They can offer clear gameplay benefits that save the player time and effort, or they can offer new ways to play the game. They can also be purely cosmetic items such as clothing and skins that can be used for social differentiation from other players [Yamakami 2012].

The rewards can usually be divided into different rarities based on the probability of drawing them from the gacha. On average, an item belonging to a tier with a lower probability is thought of as being more desirable as it is more difficult to obtain. This is not always strictly true. In Arknights [Hypergryph 2019], 4-star rewards are thought to be more powerful and valuable than 3-star rewards. However, the probability of drawing a 4-star is slightly higher than that of drawing a 3-star. This may have to do with the fact that the 4-star rarity includes 49 different rewards, and the 3-star rarity only includes 21. This effectively means that each individual reward in the 4-star rarity is more difficult to obtain than each individual reward in the 3-star rarity.

3.3 Gacha variations and features

In its most basic form, gacha typically just means a system where some form of currency is inserted, and a random item is received [Yamakami 2012]. There are many more complicated variations of this system that can often be combined freely to form new kinds of gacha. While they are often thought of and introduced as their own isolated systems like in the study by Koeder et al. [2017], it can be easier to think of these variations as features, rules, and limitations of the same base system. For example, a box gacha can instead be thought of as a gacha that implements a box ruleset. Because a box gacha can also be a kompu gacha at the same time, we can think of these archetypes as variations or features of the base gacha system instead.

Kompu gacha (コンプガチャ), short for *complete gacha,* is a system that combines the rewards obtained from separate gachas into a set [Shibuya et al. 2015]. This set needs to be completed to obtain a premium reward or a grand prize. As a set nears completion, the odds of obtaining the missing items get smaller. This is sometimes called a multi-level loot box or a combination loot box [Schwiddessen 2018]. As illustrated in image 3.4, this system can potentially get very costly to the players.

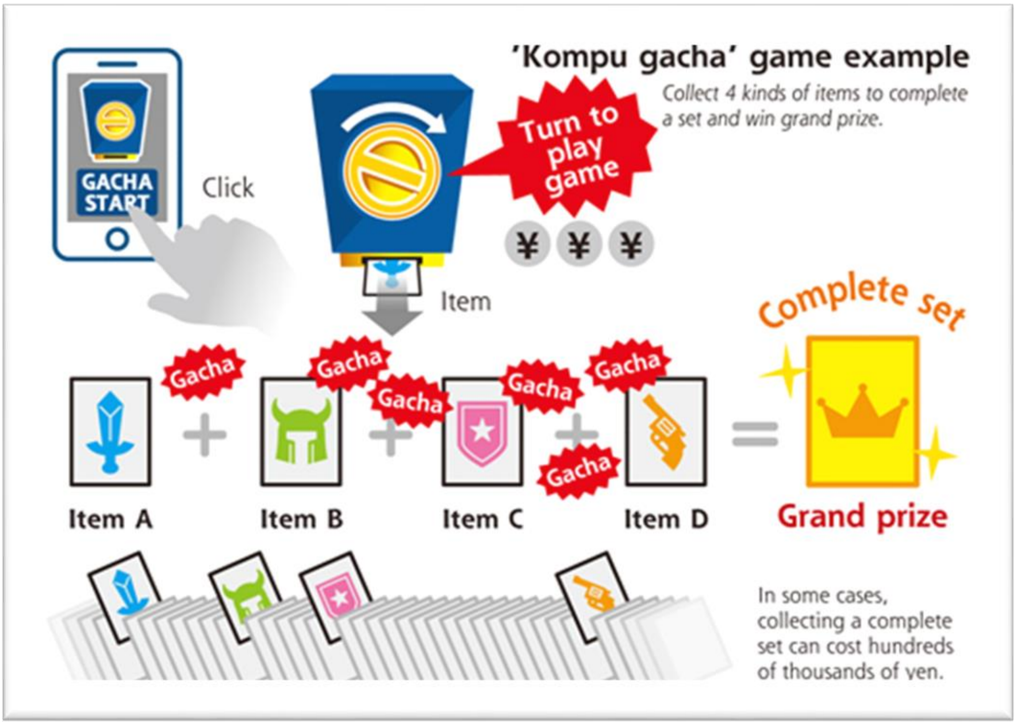


Image 3.4 An illustration of a Kompu gacha system from the Japanese newspaper Yomiuri Shimbun [2012].

After extensive media scrutiny and public backlash, the kompu gacha mechanic and multi-level loot boxes in general were effectively banned in Japan in 2012, first by the self-regulatory council and finally by the Japanese Consumer Affairs Agency [Schwid-dessen 2018]. All gacha systems that resembled kompu gacha or multi-level loot boxes were also included in these regulatory actions. A gacha mechanic very similar to kompu gacha called the *bingo gacha,* where the rewards are placed on a bingo sheet and won by completing columns or lines, was also banned [Toto 2012a]. While these gacha mechanics are not allowed when they charge real-life currencies, sometimes they can make an appearance in a game as a free or a play-to-earn mechanic. For example, in Girls' Frontline [MICA Team 2016] the bingo gacha is often used in seasonal events. The virtual currency used for this gacha is only attainable by playing the game.

In a *box gacha* the total number of available rewards is limited. As the name implies, rewards are drawn from a "box" and each successive draw has a smaller reward pool until the box is emptied. As the reward pool gets smaller, the probability of drawing a specific item increases. Once the box is emptied, it can no longer be drawn from [Koeder et al. 2017].

Some gacha systems may also offer *re-rolls* or *redraws* in case the player is not happy with the rewards they receive. The player may receive a certain number of attempts with the same amount of currency spent. Many gacha games offer free draws for new players and people may try to manipulate these systems by making new game accounts until they are happy with their initial draws [Koeder and Tanaka 2017].

Many gacha systems include different guaranteed rewards. Instead of relying on strict probabilities, they may provide a guaranteed higher tier reward after a certain number of draws. The probabilities may also change based on the player's actions and already received rewards. They may be one-time offers or permanent features of the system. They can be used to reduce the variance and to guarantee that each player can win more desirable rewards if they just keep using the system. At the same time, they can incentivize more spending because the player is aware of this guarantee. A gacha system can include multiple guarantees working simultaneously. An example of a temporary guarantee can be found in Arknights [Hypergryph 2019], where it is guaranteed that the first ten draws of the gacha system will include a reward of at least 5-star rarity. After this reward is drawn, the guarantee is disabled. An example of a permanent guarantee can be found in the same game. If after 90 draws the player has not received a reward of the highest rarity, it is automatically given to them. This guarantee then resets and activates again after another 90 draws.

Some gacha systems implement another kind of guarantee in form of secondary rewards. For example, in Girls' Frontline [MICA Team 2016], each gacha draw will also give the player extra items that can then be exchanged for the gacha rewards directly. However, the player needs to accumulate many such items making it a slow and costly process. These kinds of supplementary rewards can be thought of as a type of a guarantee because they similarly reward the players who keep investing in the gacha.

3.4 Gacha interface

In this study, I will refer to the element for interacting with the gacha mechanic as a *gacha interface*. It can have many different implementations. It can be a simple virtual button in a mobile app or a complicated form on a web page. Often it is stylized as a *banner,* which is both an advertisement and an interface.

Depicted in Image 3.5 is a typical banner from Arknights [Hypergryph 2019] advertising some of the rewards. The information on display also includes the expiration date and time. The specific details are accessible but not immediately visible. Embedded into the graphic in the lower right corner are buttons for interacting with the actual gacha mechanic.

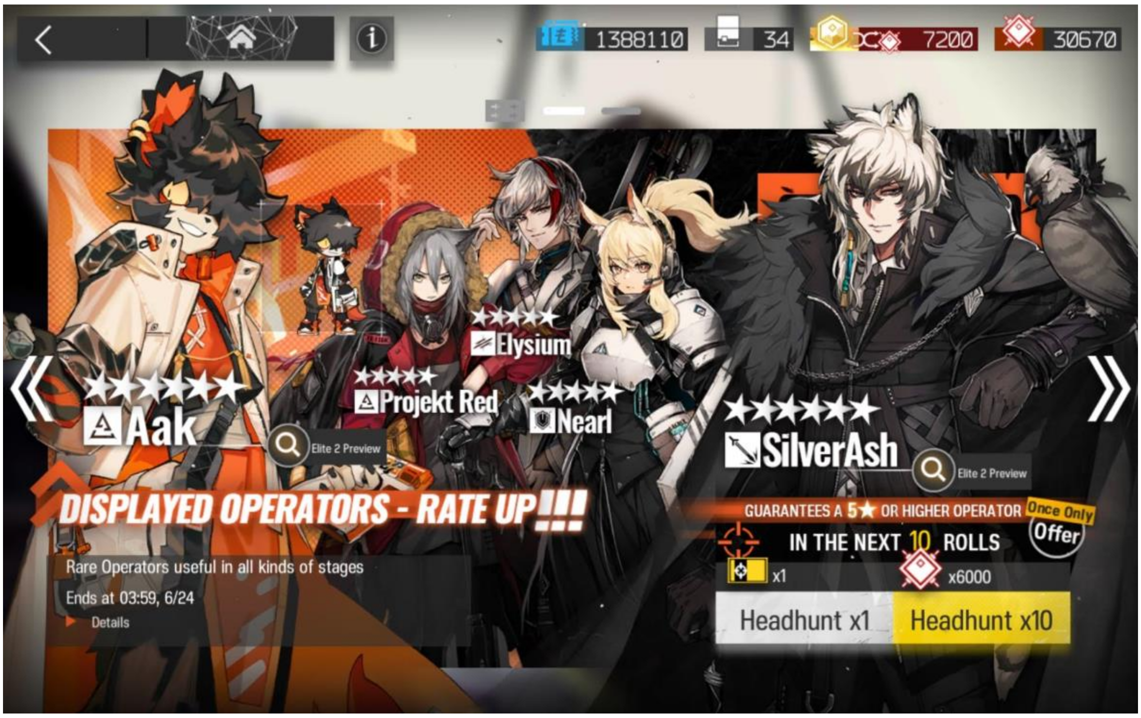


Image 3.5 A banner from Arknights, an example of a gacha interface.

A game may have multiple gacha interfaces that can be available concurrently. A common scenario is that a game comes with a fixed, permanent gacha interface and additional, time-limited gacha interfaces. The time-limited interfaces usually have their own reward pools and can have rules that are different from the permanent interface. They may offer discounts and unique rewards that aren't available anywhere else. They may be available for weeks or even just days at a time before rotating out in favor of a new, different time-limited gacha interface. This can create time pressure and cause the *FOMO*-effect (fear of missing out). Shibuya et al. [2015] found correlations between limited time gacha and increased spending. They suggest that this is "because players might assume that it is an ideal time to make in-game purchases!"

4 Customizable simulator for random draws

One of the broad goals of this thesis is to study the mechanics of gacha systems. After currency is inserted into such a system, what happens specifically? What is the actual likelihood of receiving a specific reward? How do the probabilities change during the drawing process? How can each of the specific user groups, players, designers and researchers, utilize this data?

To test these inner workings and mechanics of gacha systems, I have programmed an online web tool: a customizable simulator for random draws. If we were simply working with basic probabilities, the math would be relatively simple. However, because of many different additional rulesets and guarantees, a more complicated tool is useful. I have also tried to come up with ways to present and visualize the data on how the rewards received from the gacha are distributed. This information can be used to quickly identify how the amount of currency spent corresponds to the rewards received.

4.1 Requirements

To simulate different reward pools, we need to be able to customize our own reward categories. For implementing guarantees, we need to have modifiers that we can apply to those reward categories. We need to be able to use these custom categories and modifiers to replicate existing gacha systems. The simulator needs to display general statistics and individual statistics for each reward separately, to provide information on the entire simulated gacha system and its individual rewards. Statistical averages should be displayed for making conclusions about the average and the most common values in reward distributions. Visualization is helpful in interpreting data, so reward distributions should be visualized in some manner. The simulator needs to be able to handle very large sample sizes to provide statistically accurate info with less margin for error.

So, the specific requirements for the simulator are as follows:

- Customizable reward categories
- Modifiers that adjust probabilities during the draw process
- Display overall statistics and reward distribution (total price in virtual currency)
- Display per-reward statistics
- Display per-reward averages (mean, median, mode)
- Visualize reward distribution
- Allow very large sample sizes
- Replication of existing gacha systems

By the end of the thesis, I will have demonstrated how well the simulator has met each of these requirements. In the following sections I will go through the inspirations and features.

4.2 Summon simulators

The customizable simulator for random draws is heavily inspired by community-made *summon simulators*. These are tools often built into fan community websites by volunteer programmers. Their purpose is to simulate a gacha interface and provide a free testing environment for players of the game. In image 4.1 is a summon simulator for the game Arknights [Hypergryph 2019] from Gamepress [2020].

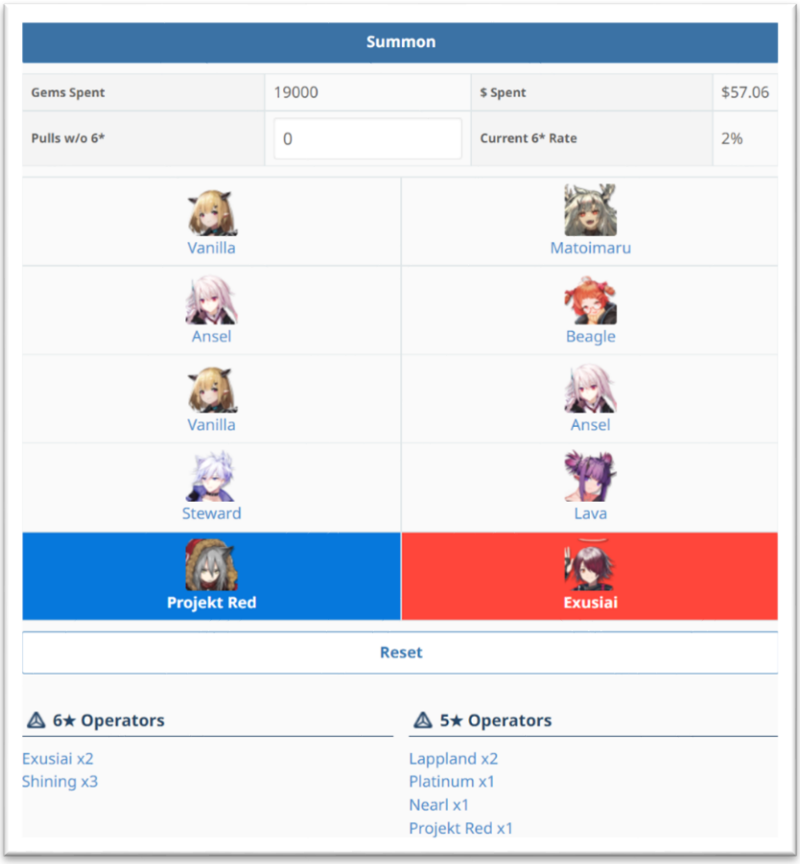


Image 4.1 A community-made summon simulator for Arknights.

While summon simulators usually provide a functionally exact replica of a specific gacha interface, the customizable simulator for random draws is fully customizable by the user. The purpose is to have a tool that can observe and analyze the process. Instead of specific rewards the focus is more on the reward categories. Two or more rewards with the exact same probabilities and modifiers can usually be combined into a single category. If necessary, this category can then be further divided into sub-categories. This way, individual rewards can be monitored as well if necessary.

Because the customizable simulator provides functionality to build and customize a gacha interface, it can essentially be used to construct summon simulators. This is implemented in the form of pre-filled forms that copy the reward categories, probabilities, and rules of existing gacha interfaces.

One of the benefits of the customizable simulator over existing summon simulators is that it supports very large sample sizes. While normally a gacha interface and summon simulators based on it perform one to ten draws at a time, the customizable simulator can perform up to hundreds of millions of draws at a time. This is very useful for obtaining statistical data of the process and results.

The customizable simulator provides transparent and detailed results with reward distribution graphs and verbose logging. It takes the form of a web page and works client-side in a browser window. It is intended to be open-source and freely available to everyone.

The customizable simulator can currently simulate gacha systems with unlimited reward pools. It can implement guarantees. Because gacha systems can implement many different rulesets it is very difficult to include them all. For example, kompu gacha and box gacha are not within the scope of this study. However, even in its current form the customizable simulator for random draws is capable of simulating gacha systems from several of the most prominent gacha games out today.

4.2 Functionality

The simulator's user interface consists of a simple HTML page in a browser window. The top portion of the page has the user-adjustable variables and settings laid out in a form. The results and data are displayed directly below it after the simulator is ran for the first time.

4.2.1 Settings and customization

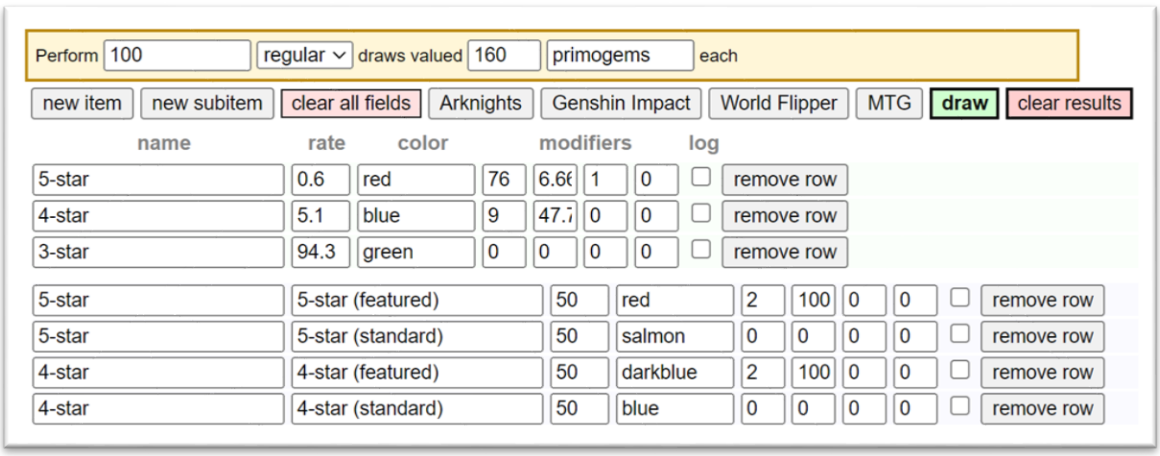


Image 4.2 The simulator interface with a Genshin Impact preset.

Image 4.2 displays a view of the simulator with a Genshin Impact [MiHoYo 2020] preset. The term *preset* refers to a full set of rewards and rules that have been applied to the form to simulate some gacha interface or a loot box system. This example utilizes most of the available options. The top row (in light brown) includes the basic user-adjustable variables in the form. They are the number of draws to perform, draw type, value of each draw and value type description.

The number of draws to perform sounds self-explanatory, but it can behave differently depending on the draw type selected. There are two options for a draw type, *regular* and *set*, as illustrated in image 4.3.

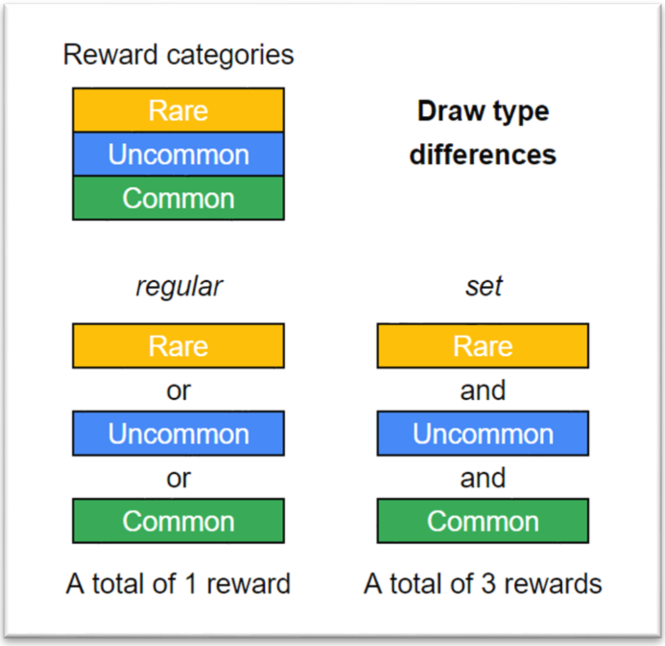


Image 4.3 Draw type differences.

If the draw type chosen is *regular,* a single draw from the entire list of categories is performed. The reward received is entirely random. With a *set* draw type, a draw is performed once for each category. So, the rewards received are always a fixed set of items, for example a rare, an uncommon and a common reward, multiplied by the number of draws requested by the user. The rewards belonging to the fixed set get their variance through the use of sub-categories.

The user may also set a numeric currency value for each draw and a specify a currency type in text form. The numeric value is used for calculating total costs and different types of average costs for each reward. The type is illustrative and can be anything, for example gems, coins, or even real-life currencies such as euros and dollars.

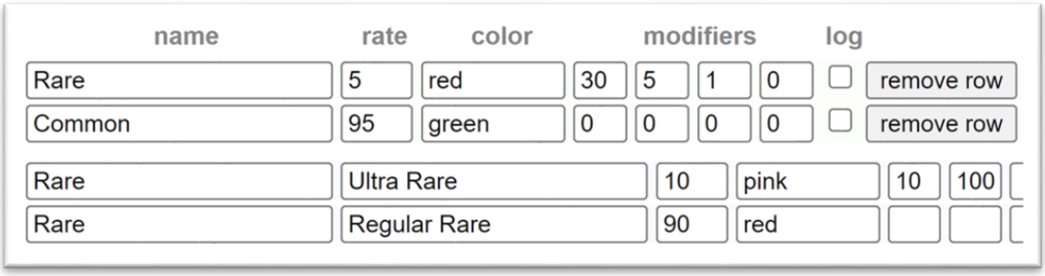


Image 4.4 The list of items and subitems.

There are two tiers of reward category lists as seen in image 4.4. The list of *items* is the higher tier of the reward hierarchy while the list of *subitems* is the lower tier. They illustrate reward categories and sub-categories, respectively. If a category has sub-categories, one of those sub-categories is always selected in place of the category when drawing rewards. Using image 4.4 as an example, when drawing a “Rare” reward, we then look at the sub-categories. Since “Rare” reward has two, “Ultra Rare” and “Regular Rare”, one of those would be selected and returned as the actual reward. The “Common” reward has no sub-categories, so when selected it would be received as it is.

Items and subitems are added and removed in a nearly identical way. The only difference is that subitems require a parent from the list of items. There is a button for adding an empty row for each in the utility bar. Each of these rows also comes with a button that removes it. The rows can be arranged by clicking and dragging with the mouse.

For the items, there are two required input fields in each item row: name and rate. The subitems additionally require the name of the parent item. The name of the item is used when displaying information and when assigning parents to subitems. The rate normally corresponds to the probability of obtaining the reward in the *regular* system. For example, we may specify that a rare reward has a base probability of 5 percent. For the first tier of items in the *set* system, rate instead corresponds to the amount of each reward belonging to a set. For example, instead of adding a percentage value, we can specify that a set always includes 1 rare reward, 3 uncommon rewards and 10 common rewards. A color for each item can bet set separately for chart visualization and logging purposes.

As outlined earlier, the draw type selected determines how items are received during the draw process. With a *regular* draw type, a random number is generated for each requested draw. This number is then used to pick a reward from the list of items according to their probabilities. If the picked item has subitems another random number is generated. One of the subitems is picked based on their probabilities and finally received as a reward. Otherwise, they behave just like items except that they do not have subitems of their own.

With a *set* draw type however, the item probabilities are replaced by fixed amounts. For every draw, an entire set of items is received as rewards. Each item is selected the requested number of times, as specified by the rate value, and then the subitems are handled the same way as in the regular system. As the name implies, this can be used to simulate fixed sets such as booster card packs where the rarity spread is identical every time. For example, a card pack could always include a rare card and two common cards. For each draw, we would receive all three of these cards. Sub-categories can then be used as necessary to customize the exact types of these cards further. It can also be used with gacha interfaces that provide different probabilities for rewards when bought in bulk. An example of this can be found in World Flipper [Cygames 2019], where purchasing a set of ten items provides an increased probability of obtaining a rare reward with one of the draws. This is different from the regular modifiers because the bonus is isolated within exactly these ten items.

Each item row also has several input fields for modifiers as seen in image 4.5.

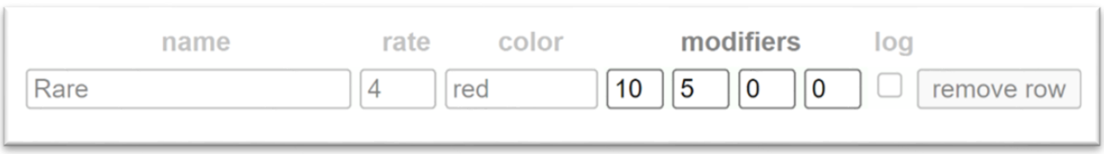


Image 4.5 An item row with modifiers highlighted, from left to right: starting point, probability change, reset for other modifiers and limit quantifier.

The first two fields are used for setting the starting point for a guarantee modifier and the modified probability amount itself. They are used to implement a guarantee. After a certain number of unsuccessful draws of the reward in question, the probability gets adjusted until a successful draw is made. For example, as seen in the image, a starting point of 10 and a probability value of 5 would add 5% to the base probability for every draw starting from the 10th. Since the base probability is 4%, this would make the probability of drawing the reward 9% on the 10th draw. If the reward is not drawn, the probability would be adjusted again for another 5% making it 14% for the following draw, and so on. This would go on until the reward is successfully drawn. After this the modifier is reset, the probability bonus is removed, and the count starts from zero again.

Next, we have an input field that can be set to a non-zero value to signify that obtaining that item resets modifiers of other items. This is normally used with guarantees that have multiple tiers. For example, a modifier can guarantee a reward of *at least* a certain rarity. In case the reward received is of higher rarity, the modifier for the *lesser* reward would be reset as well.

Last of the input fields takes a quantifier value. If set to more than zero, the modifier works a limited number of times, subtracting one each time the reward is received. When it reaches zero the modifier is disabled for the remaining draws.

There is also a checkbox for setting verbose logging on and off for each reward separately. This provides a line of text with detailed statistics in the log field each time the reward is drawn. The utility bar also provides buttons for emptying the item lists and results. There are also buttons that load existing presets and automatically fill all the required input fields. Presets can be further modified by the user after loading.

4.2.2 Data display and results

Once the draw button has been clicked and the calculations are complete, the simulator displays the results under the form. First, it displays a pie chart of the overall distribution of rewards as seen in image 4.6. On the right side of the pie chart, we get some additional information. The total value of draws is displayed here after a simple calculation that multiplies the amount of draws with the value of each draw. After that we have the distribution in text form. This includes the number of rewards received of each item and the percentage from the total amount of rewards. The main categories have been divided into sub-categories, if necessary, based on the reward types requested.

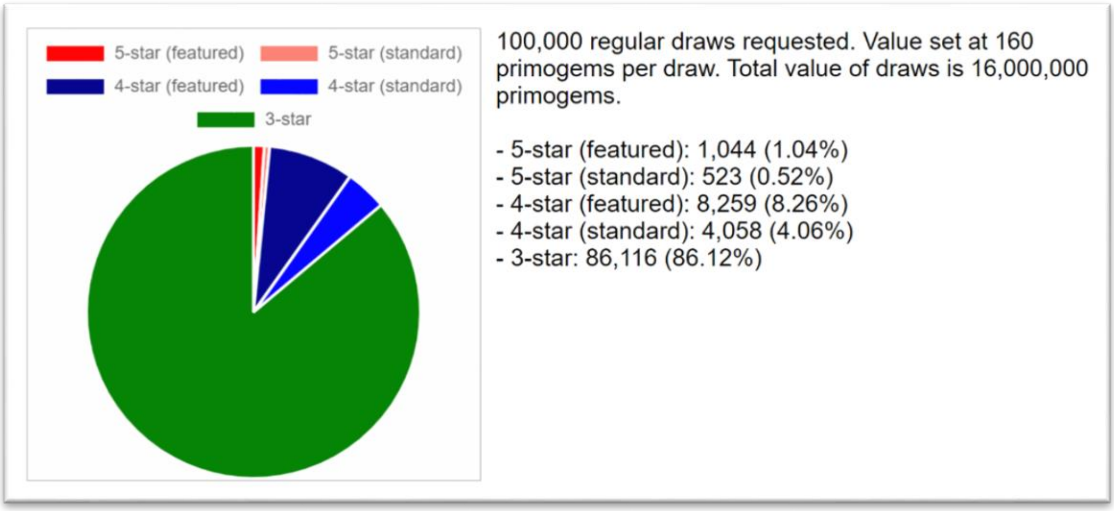


Image 4.6 An example of the overall distribution chart using a Genshin Impact preset.

After the overall distribution, the simulator constructs what I call a *reward distribution graph* for each reward as a bar chart. Each bar in the graph represents the number of rewards drawn after a certain amount of draws and this can be thought of as the reward distribution. In image 4.7 we can see an example of a reward distribution graph for a reward category called "6-star".

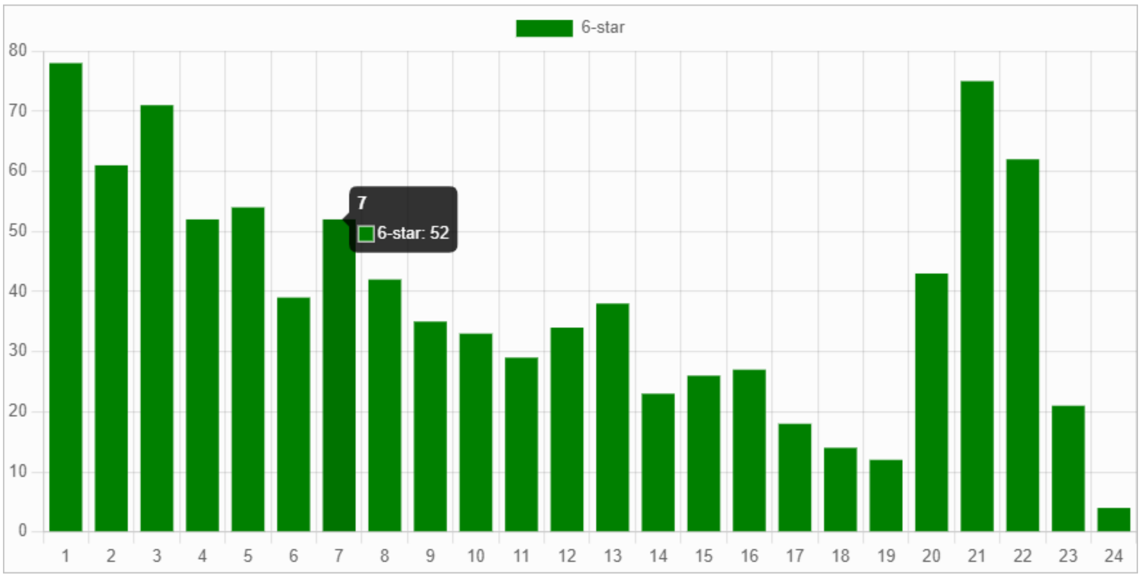


Image 4.7 An example of a reward distribution graph, sample size 10,000.

The horizontal axis displays the number of draws required to receive this reward. Here the highlighted value of 7 means that 6 draws were performed where some other reward was received, then at the 7th draw this reward was received. The vertical axis displays the number of such cases. The value of 52 means that this case happened 52 times. So, because the horizontal axis has a value of 7 and the vertical axis a value of 52, it means that this reward was received at the 7th draw 52 times during the test. This value can be useful in many ways. For example, multiplying this value with the amount of currency spent for each draw we get the amount of currency spent to get the reward. More notably, if we take the mean value of the reward distribution graph and multiply that with the currency spent for each draw, we get the *average* amount of currency spent to get this reward.

Increasing the sample size to 100 million increases the statistical accuracy and gives clearer graphs as seen in image 4.8. It also provides us with more useful and interesting information about the overall statistics of the gacha.

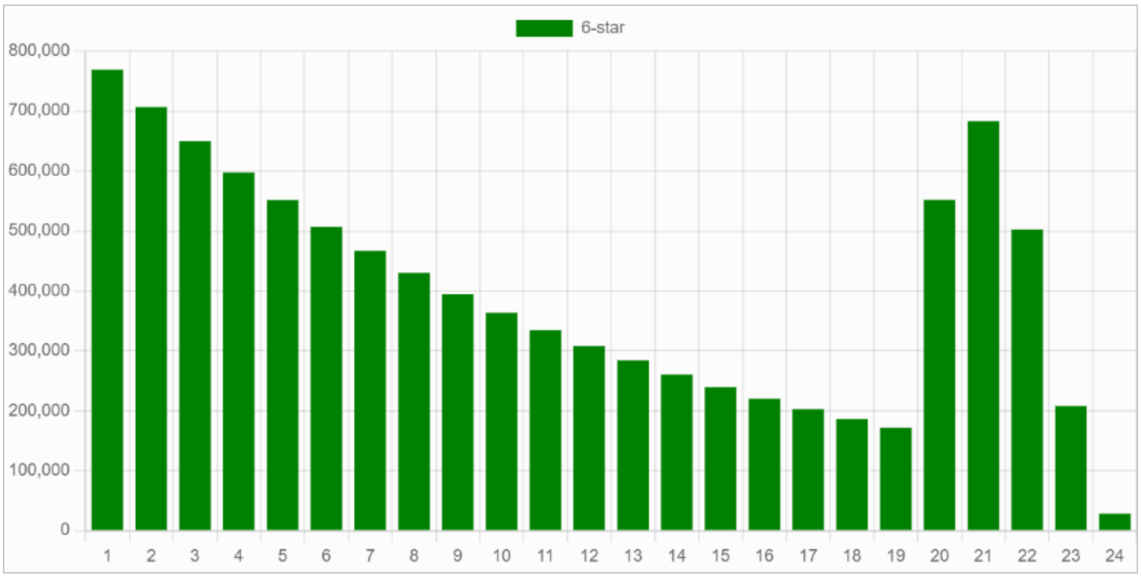


Image 4.8 An example of a reward distribution graph, sample size 100 million.

Next to each reward distribution graph is a box detailing some statistics in text form as seen in image 4.9.

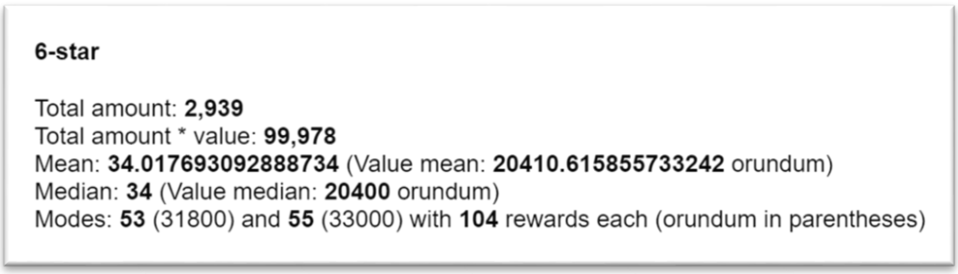


Image 4.9 Example of an individual reward statistics and averages display.

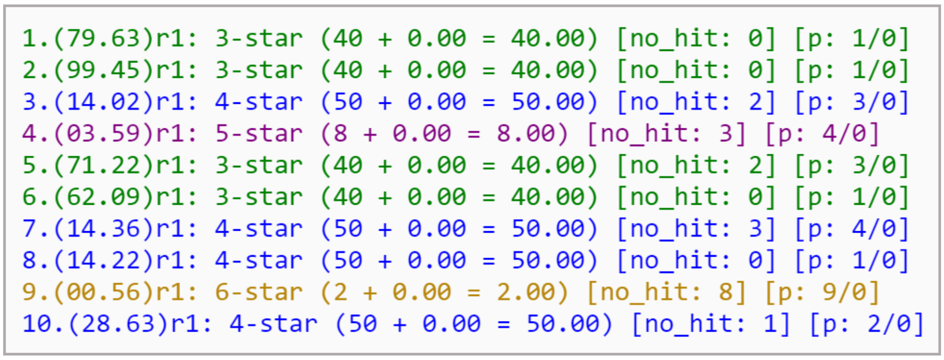
The box displays the total amount of received rewards of this type. The second line is the number of rewards multiplied with the draws required to receive them. This number is used for checking the data and it should be close to the sample size, in this case 100,000.

Most importantly, the box displays the statistical averages of the reward distribution data that I mentioned earlier. More accurately, it displays the statistical averages of the numbers of draws required, which are illustrated in the horizontal axis of the reward distribution graph as bars.

Mean shows the statistical mean or the average amount of times it takes to receive a reward. The value is also multiplied with the currency cost of each draw to illustrate average expenditure. The example in the image is from Arknights [Hypergryph 2019], where each draw is valued at 600 units of the virtual currency “orundum”. *Median* gives the value of a statistical median, meaning it shows the value of either the middle bar of the graph if the total number of bars is odd, or the average of the two bars in the middle if the total number of bars is even. *Mode* shows the bars with most cases (or simply the tallest bars), illustrating the most common amounts of draws it takes to receive the reward. The multiplied value can be thought of as the most common amount of currency spent to receive the reward.

After all the individual reward statistics the simulator optionally displays a text log. Each reward has a separate checkbox for setting logging on. Each of those rewards is then logged, and additional data is displayed in a dedicated area below all the other data. The logging focuses on telling as much information as possible in a small space so it can be a little difficult to decipher for a casual user. It is mainly used for debugging and double-checking but can be potentially useful in other situations.

In image 4.10 is an example of a log listing from Arknights [Hypergryph 2019]. The sample size is 10 draws, and logging has been turned on for every reward.



```
1.(79.63)r1: 3-star (40 + 0.00 = 40.00) [no_hit: 0] [p: 1/0]
2.(99.45)r1: 3-star (40 + 0.00 = 40.00) [no_hit: 0] [p: 1/0]
3.(14.02)r1: 4-star (50 + 0.00 = 50.00) [no_hit: 2] [p: 3/0]
4.(03.59)r1: 5-star (8 + 0.00 = 8.00) [no_hit: 3] [p: 4/0]
5.(71.22)r1: 3-star (40 + 0.00 = 40.00) [no_hit: 2] [p: 3/0]
6.(62.09)r1: 3-star (40 + 0.00 = 40.00) [no_hit: 0] [p: 1/0]
7.(14.36)r1: 4-star (50 + 0.00 = 50.00) [no_hit: 3] [p: 4/0]
8.(14.22)r1: 4-star (50 + 0.00 = 50.00) [no_hit: 0] [p: 1/0]
9.(00.56)r1: 6-star (2 + 0.00 = 2.00) [no_hit: 8] [p: 9/0]
10.(28.63)r1: 4-star (50 + 0.00 = 50.00) [no_hit: 1] [p: 2/0]
```

Image 4.10 An example of a log listing from the Arknights preset.

Each line provides the number of the draw in question, the random number generated for that draw and the name of the actual reward received. After that we have the reward’s base probability and any active modifiers. Finally, we have values that signify how many times it has been since the last time we received that reward. The lines are also color-coded according to the reward settings to make each category stand out from the others. This is useful especially when using large sample sizes as the logs can get lengthy.

4.3 Development

4.3.1 Overview

The simulator works in a web browser window and is programmed in HTML, CSS and JavaScript. Some JavaScript libraries are also utilized: jQuery [2022] and jQuery UI [2022] offer some useful interface features and Chart.JS [2022] provides all the charts used to visualize the data.

This means that the calculations are done client-side on the user's computer. A server-side solution consisting of PHP and MySQL was originally in consideration. However, the JavaScript code performed sufficiently well in testing and required less complicated programming.

4.3.2 Randomization

For selecting the rewards, I first tested an approach that used indexed tables. For example, the values would be stored into an array with 100 indexes, each index corresponding to an integer from 0 to 99. Then a random number from 0 to 99 was created and a value corresponding to the index was retrieved from the array. The weakness of this system was precision. For percentages consisting of full integers this approach is very fast, but it quickly becomes cumbersome as precision is added. Each additional decimal multiplies the number of required indexes by 10.

Second approach is creating an array of the rewards with the draw chance of each stored as a number. Then a random number, equal or larger to 0 and smaller than 100, is created. This number is then compared to the chance value of the first reward. If it is lower, the reward is drawn. If not, this chance value is added to the next chance value and the comparison is repeated. Eventually a reward is drawn. This approach allows a very high degree of precision in both random numbers and chance values.

While building the system, this basic method allowed for very quick drawing of tens of millions of rewards. A sample size of 100 million draws is currently possible to achieve in a reasonably fast time. After adding the rest of the features, the simulator still performed quickly and no further changes to the randomization were needed.

4.3.3 Reward distribution graph

The reward distribution graph is a clear and readable way to display the reward distribution of a gacha system. When thinking of ways to display the data, I felt it was important to have visual representation of the distribution instead of just raw numbers. The resulting graph is a simple representation of how many attempts at a random draw it requires to receive the desired prize. With smaller sample sizes the graphs can look very random and unclear but as the sample sizes rise the underlying statistics become more apparent.

When using a gacha system it is very common to stop immediately when the desired reward, usually the grand prize, is received. The reward distribution graph can help illustrate and observe this behavior. For example, if we assume that a player will stop once the prize is reached, each case can be thought of as a new individual player receiving that reward. It can also illustrate a single player continuing to draw repeatedly in the gacha system. These two ideas can also be combined as different players drawing an arbitrary number of rewards.

5 Presets

The simulator includes several presets. They are pre-filled forms included within the simulator that have all the required settings and reward lists required to simulate a gacha interface, loot box lottery system or even a collectible card game booster pack.

The presets selected for testing and demonstration include Arknights, Genshin Impact, World Flipper and Magic: The Gathering. The first three are notable gacha games while the last is a physical product and an original demonstration of the simulator's features. Furthermore, settings of each preset suitably differ from each other to give us a wider look into the simulator's capabilities.

Sample size in each test is a *total* of 100 million draws. Arknights and Genshin Impact use the *regular* draw type, so they will each draw the full amount of 100 million. In World Flipper's case this means 10 million draws of a *set* of ten, totaling up to 100 million. In the original test of Magic: The Gathering booster packs, 6.25 million draws of a *set* of 16 also total up to 100 million. This total number is selected because the simulator can still handle it without the browser freezing up completely. The number is suitably large, round and still easy to work with.

Guarantee modifiers describe a change in probability at a certain point of the draw process. A modifier amount of +5% means that the base probability is increased by five percent. If a modifier is set to begin at the 50th draw, this means that after there has been 49 unsuccessful draws of that reward, the modifier is implemented on the following draw. This process is repeated and another 5% is added to the probability each time until the reward is successfully drawn.

The tests are first conducted with the actual rules and modifiers in place. If applicable, another test is run with the modifiers removed. When using the *set* draw type, we may also do some additional tests on the subitems using regular draw type. All the tests can be repeated in the simulator by using or adjusting the included presets.

In the next sections, I will give some background on each game tested and go over the settings of their presets.

5.1 Arknights

Arknights [Hypergryph 2019] is a tactical role-playing tower defense game set in a post-apocalyptic science fiction setting. It was released in China in 2019 and globally in 2020. It is available for Android and iOS. Arknights is the first selected game for the testing because of the relative simplicity and transparency of its gacha mechanics. It has no hidden values that require reverse engineering. I'm also personally familiar with its gacha system and the community-made summon simulators based on it [Gamepress 2020]. It also provides a good baseline and comparison game for Genshin Impact.

item	rate	item modifier	subitem	rate
6-star	2%	+2% at 50th draw	6-star (featured)	50%
			6-star (standard)	50%
5-star	8%	100% at 10th draw, one time only, reset by a 6-star item		
4-star	50%			
3-star	40%			

Image 5.1 Arknights rewards and preset settings.

The settings for the Arknights preset can be seen in image 5.1. The highest tier of reward in Arknights is the 6-star and it has the base probability of 2%. It has a guarantee modifier of +2% starting at the 50th draw. This adds up to a probability of 100% at the 98th draw. The second tier of reward is the 5-star. It has a base chance of 8% and no permanent modifiers. However, each banner (gacha interface) typically has a one-time guarantee. After 10 draws a 5-star or a *better* reward is guaranteed. This is implemented as a +100% modifier at the 10th draw for the 5-star. Additionally, the 6-star reward is set to disable this modifier should it be drawn, as it is the sole qualifier for a *better* reward. The final reward tiers are the 4-star and the 3-star. They have base probabilities of 50% and 40%, respectively. They have no additional modifiers.

The subitems consist of limited and standard versions of the 6-star. They each have a probability of 50% and no additional modifiers. Sometimes the banners include multiple limited versions of 6-stars, and the probabilities can vary accordingly. However, they do not affect the probabilities of the main reward categories. This simplest version is used in the tests. Since the subitems have identical probabilities, they will always produce an identical spread of rewards and can be combined in the results. They are included here to showcase the capabilities and for some easier comparison.

5.2 Genshin Impact

Genshin Impact [MiHoYo 2020] is an action role-playing game taking inspiration from Japanese animation and the popular video game Legend of Zelda: Breath of the Wild. It was released globally in 2020 by the Chinese developer MiHoYo. It is available for Android, iOS, Microsoft Windows, PlayStation 4, PlayStation 5, and Nintendo Switch.

Genshin Impact is notable for being the first high budget multi-platform game that is almost entirely monetized by a gacha mechanic. Several financial data analysis services already rank Genshin Impact among the most successful mobile games of all time. During its first year, it generated an estimated 2 billion euros worth of revenue on mobile devices alone. This figure is thought to be nearly double when PC and console versions are included [Chapple 2021].

item	rate	item modifier	subitem	rate	subitem modifier
5-star	0.6%	+6.6666667% at 76th draw	5-star (featured)	50%	100% at 2nd draw
			5-star (standard)	50%	-
4-star	5.1%	+47.75% at 9th draw, reset by a 5-star item	4-star (featured)	50%	100% at 2nd draw
			4-star (standard)	50%	-
3-star	94.3%				

Image 5.2 Genshin Impact rewards and preset settings.

As seen in image 5.2, Genshin Impact's grand prize category is the 5-star. It has a base probability of 0.6%. The official documentation states that a grand prize is guaranteed at no later than 90 draws. However, reverse engineering and data analysis by the fan community [Paimon.moe 2021; sm0rky 2020] have revealed that the guarantee modifier is actually approximately +6.667% starting at the 76th draw. This eventually adds up to 100% at the 90th draw. This is a case of a developer providing required information while omitting details. Compared to Arknights, the base probability of the grand prize is much lower, but the guarantee modifier is higher.

The second tier of reward for Genshin Impact is the 4-star. It has a base probability of 5.1%. Officially, the developer MiHoYo states that a 4-star or a better reward is guaranteed at least every 10 draws. Again, the information provided by the developer is incomplete. Reverse engineering reveals that the guarantee modifier is actually approximately +47.75% starting at the 9th draw [paimon.moe 2021; Last hope 2020]. With the base probability of 5.1% the modifier adds up to more than 100% at the 10th draw. Additionally, because a 5-star reward also qualifies for this guarantee, a toggle is added for the 5-star reward to reset this modifier.

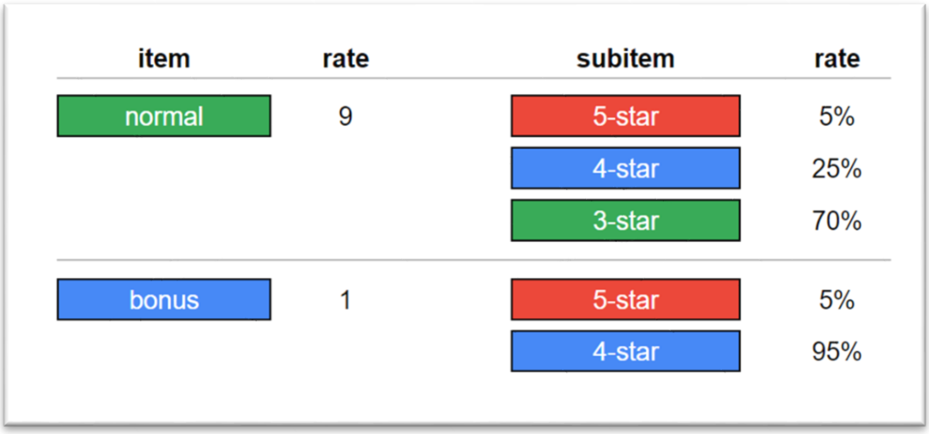
The third and final tier of reward is the 3-star. It has a very high base probability of 94,3% and no additional modifiers.

There are two subitem modifiers that behave identically. The featured 5-star rewards and featured 4-star rewards both have a guarantee modifier of +100% starting at the second draw. In practice this means that at least every second draw is a featured reward. With the base chance of 50% this makes the probability of drawing a featured reward as opposed to a standard one approximately 66.667%, or 2/3.

5.3 World Flipper

World Flipper [Cygames 2019] is a hybrid of a pinball game and a role-playing game by Japanese developer Cygames. It was released in Japan in 2019 and globally two years later in 2021. It is available for Android and iOS.

World Flipper is included mainly because of how its rules differ from the previous two games. In World Flipper, the player receives better rewards when purchasing ten rewards at a time instead of just one. For this purpose, the set feature of the simulator is used. A set of ten rewards is created as seen in image 5.3. The set includes nine rewards with normal probabilities and one bonus reward with one of the subitems omitted. This set is repeated the requested number of times.



item	rate	subitem	rate
normal	9	5-star	5%
		4-star	25%
		3-star	70%
bonus	1	5-star	5%
		4-star	95%

Image 5.3 World Flipper rewards and preset settings.

The grand prize of World Flipper is the 5-star, and it has a base probability of 5%. This base probability is relatively high compared to the other gacha interfaces tested. Arknights has a grand prize base probability of 2% and Genshin Impact only 0.6%. However, World Flipper has no guarantee modifiers in any of the individual reward categories, so the base probability never changes.

The second tier of reward is the 4-star. In single draws, the base probability of a 4-star is normally 25%. However, when drawing ten rewards at a time it is guaranteed that in one of those draws the 4-star's base probability is 95% and the 3-star reward is completely excluded.

The final tier of reward is the 3-star, and its base probability is 70%. As mentioned earlier, every ten draws this prize is omitted completely.

For the currency value in this *set* draw type, we must remember to enter the value of an entire set instead of just a single draw. In this case, for a set of 10 draws, the currency value is 1500 and value type is "lodestar beads".

5.4 Magic: The Gathering

Finally, I have added an original preset based on the collectible card game Magic: The Gathering designed by Richard Garfield and released by Wizards of the Coast [1993]. The simulator's *set* draw type mechanic was originally intended to simulate a collectible card game booster pack design. This preset shown in image 5.4 simulates a physical product, a Magic: The Gathering collectible card booster pack.

item	rate	subitem	rate
Rare	1	Mythic Rare	12.5%
		Rare	87.5%
Uncommon	3		
Common	10	Foil Card	2.22%
		Common	97.78%
Marketing Card	1		
Basic Land	1		

Image 5.4 Magic: The Gathering rewards and preset settings.

Each booster pack includes a total of 16 cards. These include 1 rare card, 3 uncommon cards, 10 common cards, 1 marketing card and 1 basic land card. These are added as main category items in the preset. The subitems add the only variance to the set. Each rare card has a 12.5% chance of being a mythic rare card and each common card has a 2.22222% chance of being a foil card. As the simulated entity is a physical product, we can also use a real-world currency as the value type. From 2006 until 2019 the MSRP (manufacturer's suggested retail price) of this type of a booster pack was 3.99 dollars [Wizards of the Coast 2006]. The company has since discontinued the practice of MSRP's and is now letting the retailers set their own prices instead. I have used this last known price as the variable in this preset.

6 Results

In this section I go over the results and data provided by running the simulator with each of the presets. I go over the overall statistics for each and examine some of the more interesting individual reward results. I also compare some of the results to each other. All the images in this chapter are screen captures of the simulator results.

6.1 Arknights

Looking at the overall statistics for the Arknights test as seen in image 6.1, we can see the effects of the guarantee modifier right away.

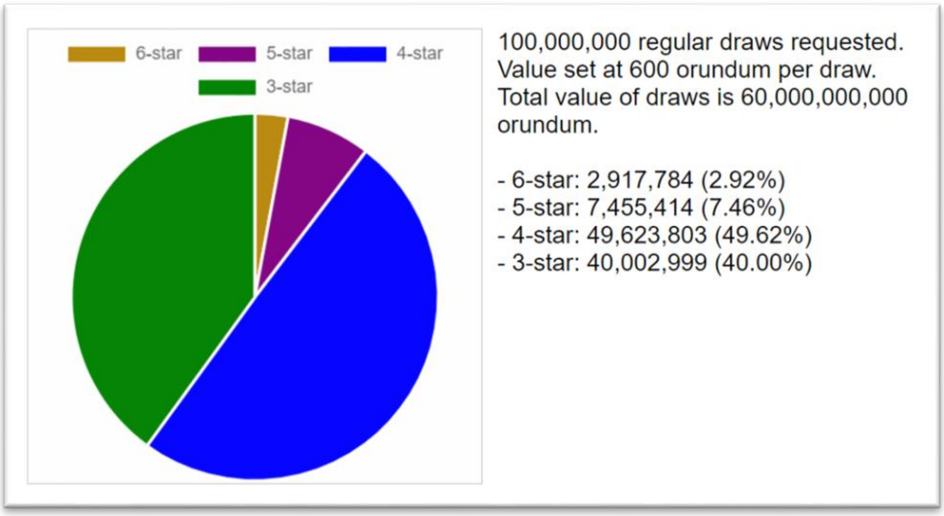
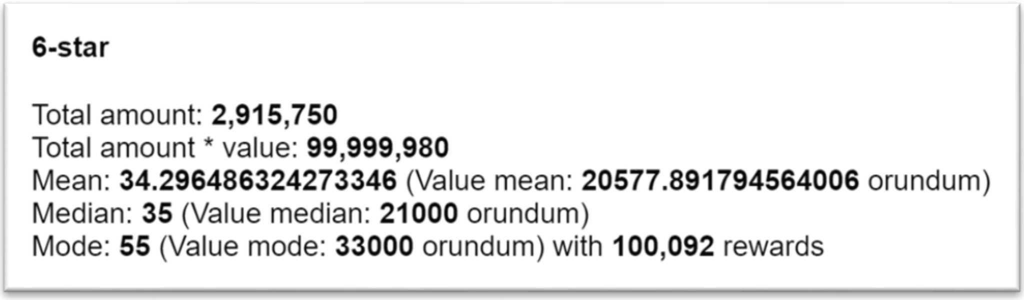


Image 6.1 Arknights overall statistics and reward distribution.

The base probability of the 6-star was only 2%, yet almost 3% of the rewards received are 6-stars. The 5-star reward also had a guarantee modifier but since it was disabled after a single use, its effects are negligible on such a large sample size.



6-star

Total amount: **2,915,750**
Total amount * value: **99,999,980**
Mean: **34.296486324273346** (Value mean: **20577.891794564006** orundum)
Median: **35** (Value median: **21000** orundum)
Mode: **55** (Value mode: **33000** orundum) with **100,092** rewards

Image 6.2 Arknights 6-star reward statistics and averages.

Image 6.2 shows some statistics of the grand prize reward of Arknights, the 6-star. The currency value of each draw is set at 600 units of the virtual currency “orundum”. Multiplying this with the mean value of ~34.3 gives us a total of approximately 20,570 units of the currency. This can be thought of as the approximate average price of a grand prize draw in this Arknights gacha interface without considering the lesser rewards. Median at 35 is nearly the same with a currency value of 21,000. Mode, the highest number of observed cases, is 55 and gives a virtual currency value of 33,000. This could theoretically mean that 33,000 would be the most common amount of virtual currency spent in this gacha to attain this reward.

In the reward distribution graph in image 6.3 for the grand prize reward we can see the effects of the guarantee modifier clearly.

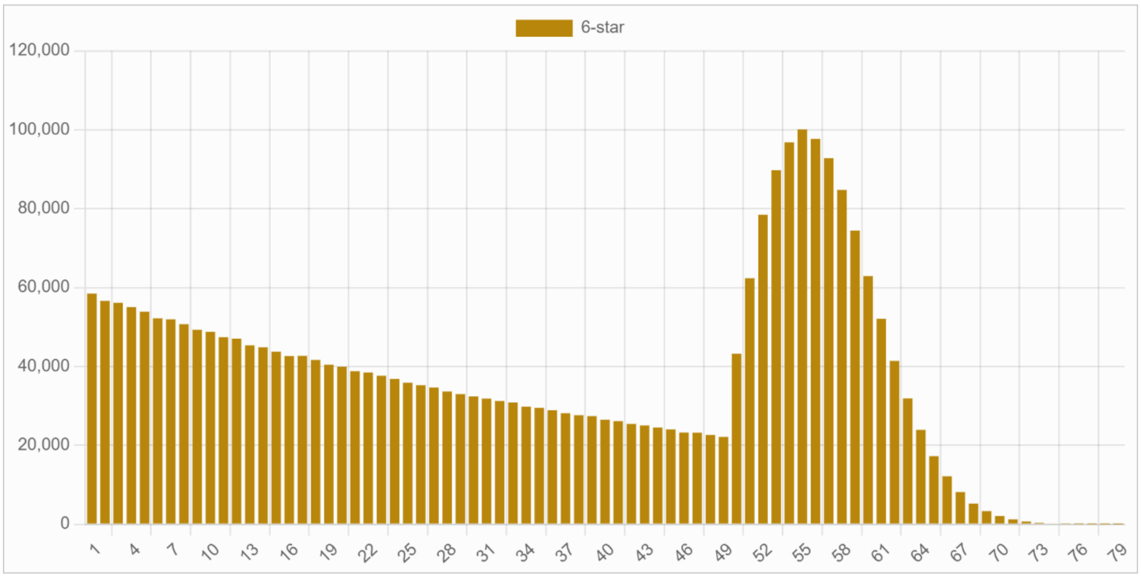


Image 6.3 Arknights 6-star reward distribution graph with modifiers.

At the 50th pull, the number of cases nearly doubles after the probability of a successful draw rises from 2% to 4% thanks to the guarantee modifier. At the mode of 55, where most of the rewards were drawn, the probability is 14%. The curve then decays exponentially as the probability rises. The last recorded case in this test is at the 79th pull, where the probability is 62%. Even when dealing with such a large sample size, draws past this point become very rare due to the high probability and small increments.

For comparison, running the test without the guarantee modifier in place gives a very different reward distribution as seen in image 6.4. Where the graph originally has a very clear spike and an approximate end point, without guarantees the curve simply decays exponentially. This means we can see up to almost 600 draws in a row without a single reward of this type, even though this happens very rarely. In this scenario the probability never changes from 2%.

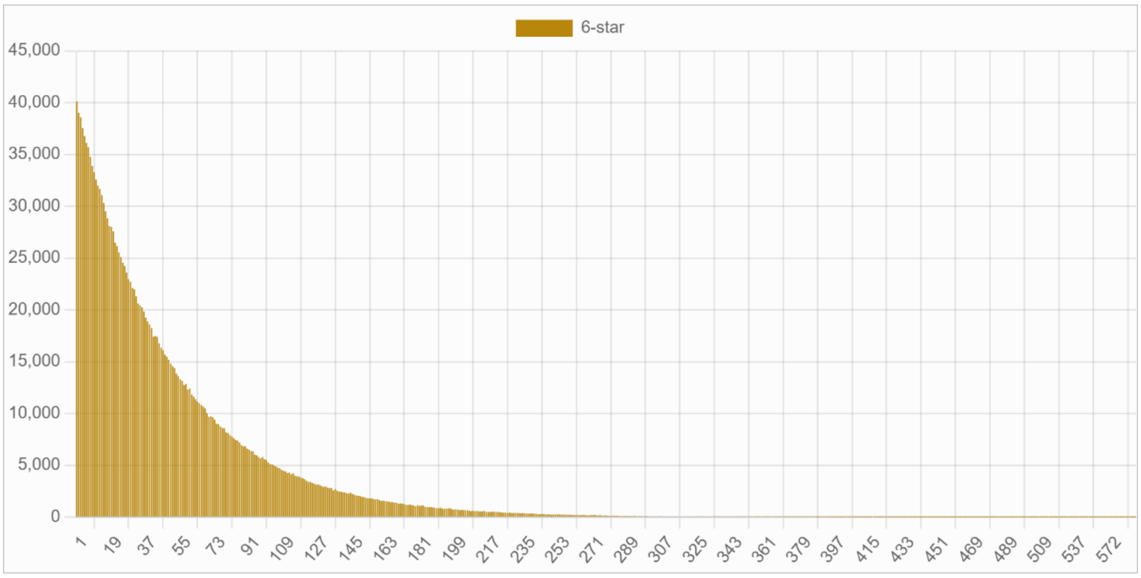


Image 6.4 Arknights 6-star reward distribution graph with modifiers removed.

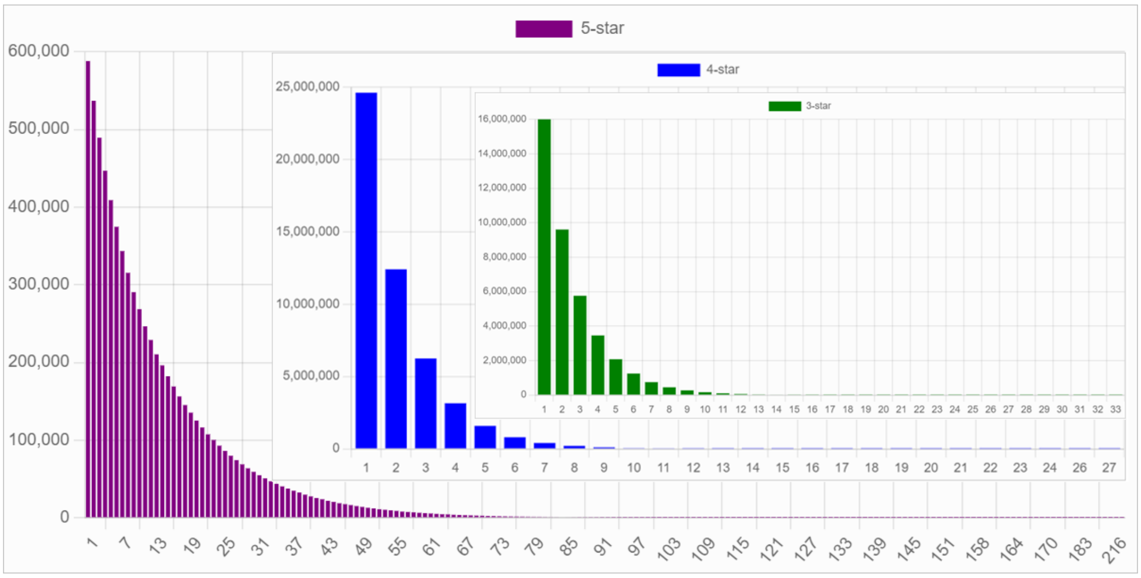


Image 6.5 5-star, 4-star and 3-star reward distribution graphs from Arknights.

As seen in image 6.5, the 5-star, 4-star and 3-star rewards in the actual test provide similarly uninteresting reward distribution graphs. However, the lack of guarantee modifiers is usually less of an issue because of their much higher base probabilities and sheer number of rewards.

6.2 Genshin Impact

The overall statistics of Genshin Impact as seen in image 6.5 are a bit more complicated than in Arknights.

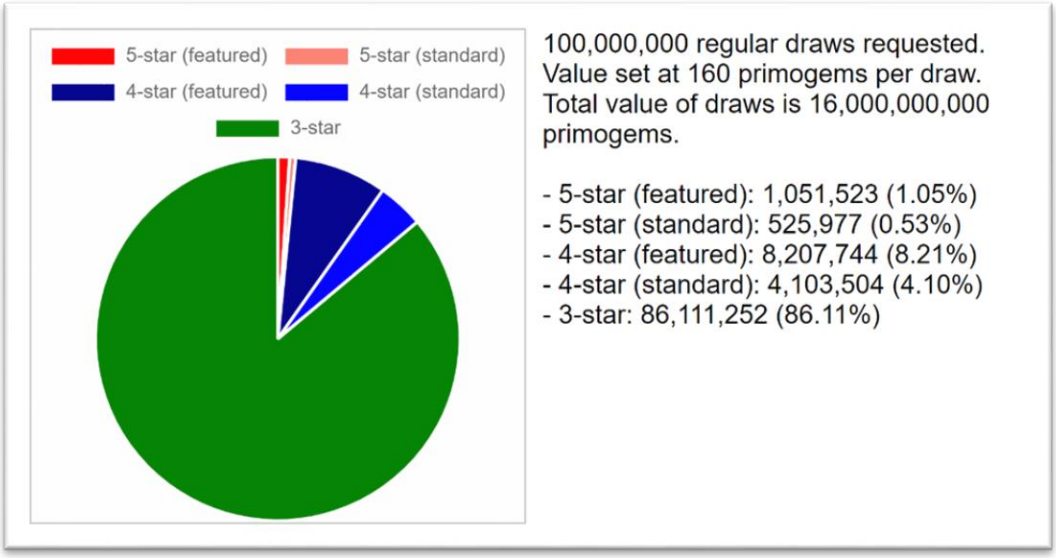


Image 6.6 Genshin Impact overall statistics and reward distribution.

The main reward categories of 5-stars and 4-stars are split between featured and standard versions. Adding up the values of featured and standard versions, the total value for 5-stars is 1.58%, up from the base probability of 0.6% thanks to the 5-star guarantee modifier. Similarly, the total value for 4-stars is 12.31%, up from 5.1%, adjusted by the 4-star modifier that guarantees a 4-star or better reward at least every 10 draws. A large majority of the rewards still consists of 3-star rewards. Even so, despite a large base probability of 94.3%, only 86.11% of the rewards are 3-stars thanks to the guarantee modifiers of the other rewards. As expected, approximately two thirds of both 5-stars and 4-stars are featured versions of the rewards.

In image 6.7 we can see statistics for Genshin Impact's grand prize, the 5-star, or more specifically its featured version. Both featured and standard versions have identical probabilities, and the results only differ in total amounts of rewards received. This means calculating average values offers nearly identical results. We only need to look at one of them.

5-star (featured)

Total amount: **1,052,135**
Total amount * value: **66,688,348**
Mean: **63.38383192270954** (Value mean: **10141.413107633527** primogems)
Median: **78** (Value median: **12480** primogems)
Mode: **79** (12640) with **115,981** rewards each (primogems in parentheses)

Image 6.7 Genshin Impact 6-star reward statistics and averages.

The virtual currency value for each draw is set at 160 primogems. The mean of ~63.38 multiplied with 160 gives us approximately 10,141 units of virtual currency. The high guarantee modifier pushes more of the rewards towards the later stage of the reward distribution. The median of 78 is, relatively speaking, noticeably higher than in Arknights, giving us a value median of 12,480 virtual currency. The mode is 79 and very close to the median, totaling up to 12,640 units of virtual currency.

The reward distribution for Genshin Impact's grand prize as pictured in image 6.8 looks similar to the distribution graph of the grand prize of Arknights. However, due to the low base probability and a relatively much higher guarantee modifier, the spike in successful rewards towards the end is more pronounced and abrupt.

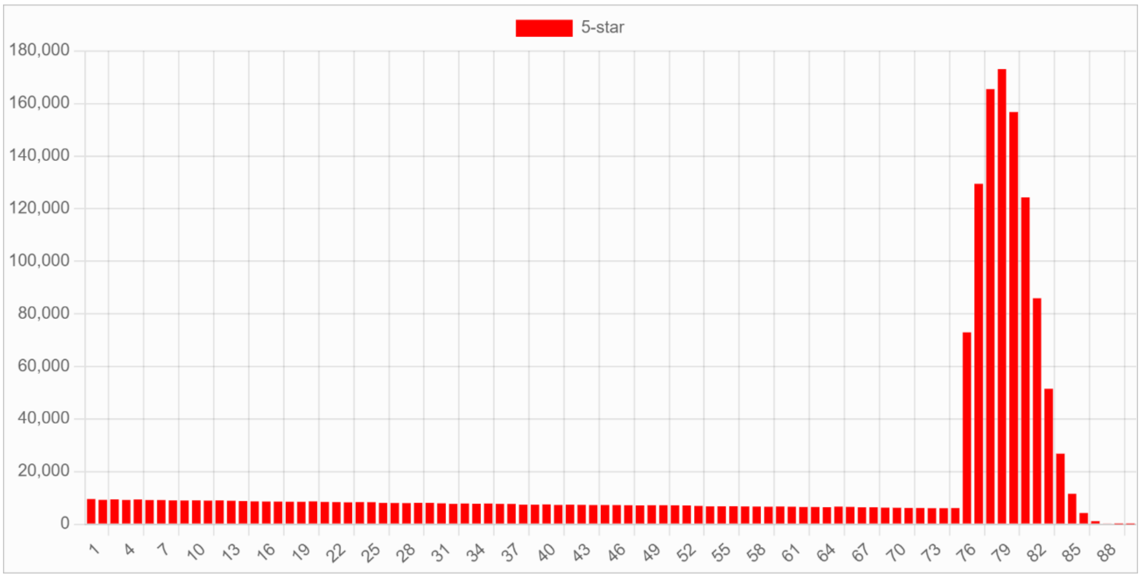


Image 6.8 Genshin Impact 5-star distribution graph with modifiers.

Rewards are drawn consistently often until the 76th draw, where the guarantee modifier adds approximately 6.667% to the base probability of 0.6%. While in Arknights the probability was doubled from 2% to 4%, in Genshin Impact the new probability is over ten times that of the original value. This causes the huge spike in reward distribution.

Similarly, when the test is run without the guarantee modifier, the variance is even more pronounced than in Arknights because of the much lower base probability. An unlucky player might have to draw thousands of times to receive their grand prize.

Genshin Impact's second most valuable reward is the 4-star. Its distribution has a notable spike at the 9th and 10th draw due to the guarantee modifier as shown in image 6.9.

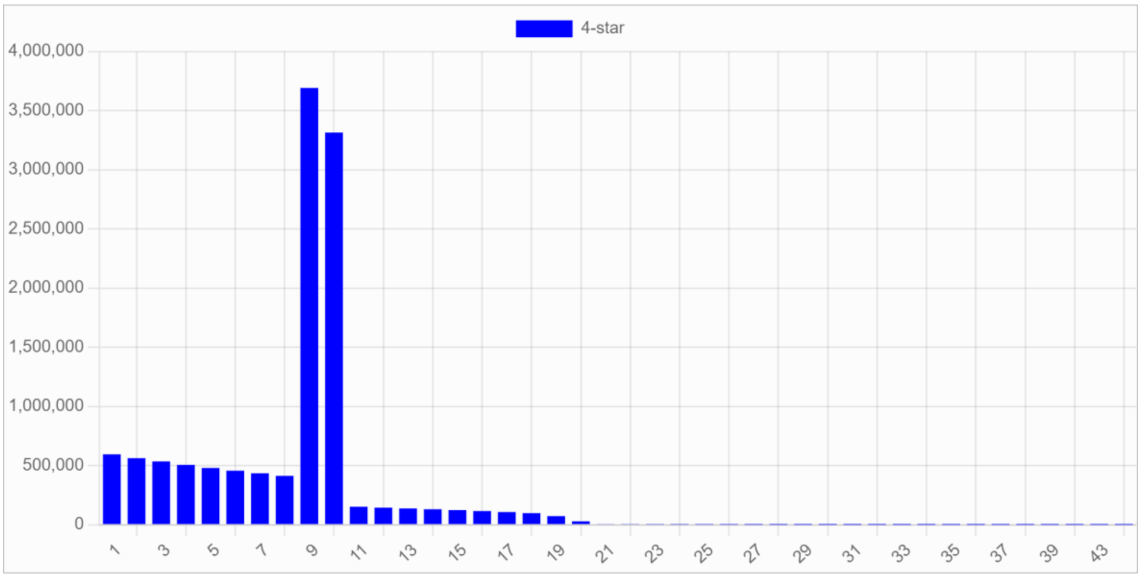


Image 6.9 Genshin Impact 4-star distribution graph with modifiers.

Most of the rewards are drawn at those two spots, but there is still a small chance that a higher tier reward resets the guarantee modifier. Every successful drawing of a 5-star starts the 4-star modifier from zero, which explains why there are cases of drawing a 4-star past the 10th draw where the probability would normally be 100%. A reset must always happen before the 10th draw. This means that a guarantee that is reset once will likely provide a reward during draws 11-20. Due to the very low chance of a grand prize causing a reset, multiple resets are unlikely but possible, as the chart illustrates.

6.3 World Flipper

As seen in image 6.10, the format for World Flipper's *set* type results differs only slightly from the *regular* draw type results seen in the previous two games. The set size and requested draws are both displayed, as well as a multiplication of the two to provide the total amount of actual draws.

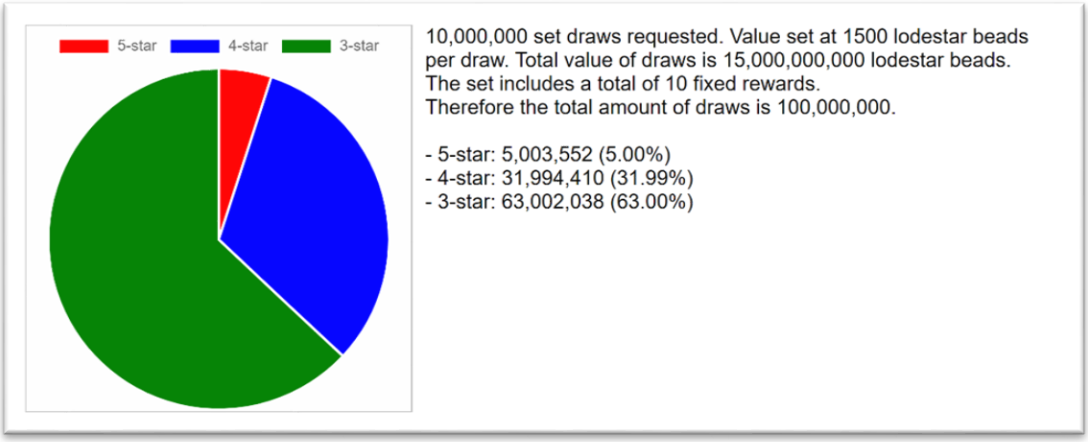


Image 6.10 World Flipper overall statistics and reward distribution.

While the list of rewards looks uniform, World Flipper's subitems are actually combined in the results. The nine normal probability draws have 5-star and 4-star subitems just like the bonus draw. These are separate entities but due to the same name, they are combined in the overall statistics. This way it is possible to draw sets of items that have different probabilities for the same reward. The 5-star rewards have the same probability at 5% but the 4-star rewards do not. For the 4-stars, the set of nine 25% probability rewards and one 95% probability reward end up taking a combined 32% share of the total rewards. The 3-star rewards have nine draws at a 70% probability and one draw at 0% probability, meaning one draw is omitted completely. This makes the 3-star's overall probability 63% for the set of 10.

Again, the lack of guarantee modifiers makes the individual reward distribution graphs less interesting. They consist solely of exponentially decaying graphs similar to image 6.4 in the Arknights tests.

6.4 Magic: The Gathering

The Overall distribution chart for Magic: The Gathering includes most of the relevant information in this test. Each set is mostly identical with variance in only two of main rewards. As we can see in image 6.11, three of the rewards, uncommon, marketing card and basic land follow their rates exactly. The only variance is in two of the main categories, rare and common. And as I have set a direct price for each set draw, the simulator displays the total value of the draws as a real-world currency. The sample size of 6.25 million booster packs at 3.99 dollars each totals up to nearly 25 million dollars.

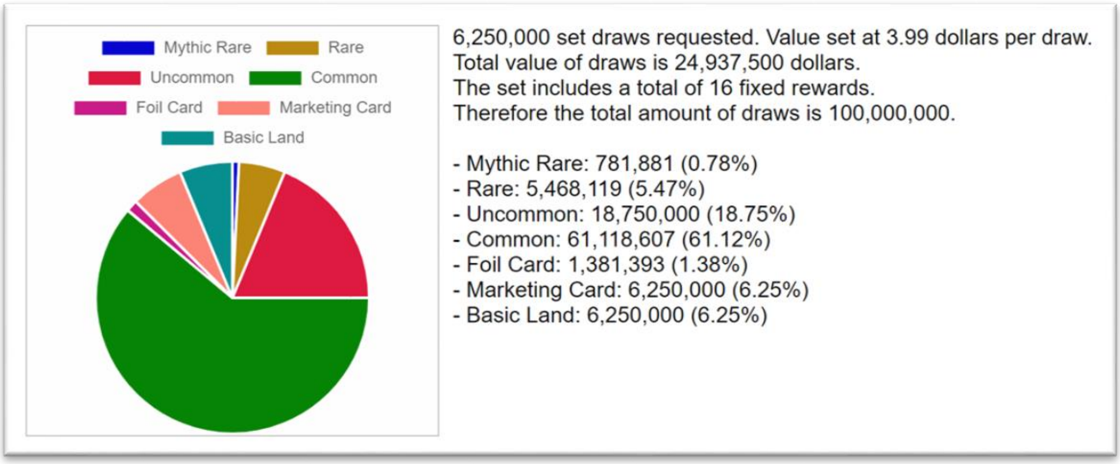


Image 6.11 Overall distribution pie chart for the Magic: The Gathering booster pack, sample size 6.25 million.

Like previously, since sets include a fixed selection of reward categories, individual reward distribution graphs aren’t quite as useful.

Isolating the desired rewards may also be more useful sometimes when examining fixed sets. In World Flipper’s case, we were combining reward categories, but in this case, we have separate sub-categories we can further investigate. For example, to simulate the rare and mythic rare distribution we can simply make a new regular preset. There are two categories, the mythic rare reward with a probability of 12.5% and a rare reward with a probability of 87.5%. The results of this new test can be seen in image 6.12.

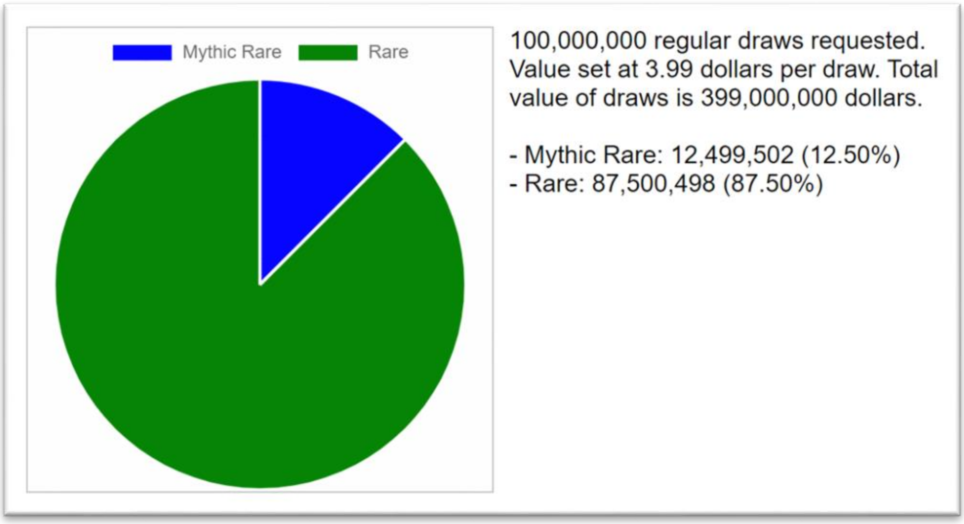


Image 6.12 Overall distribution chart for rare cards in Magic: The Gathering.

Because there are no modifiers, the results are straightforward and approximately correspond to the base probabilities set. While the sets have fixed amounts, the sub-categories introduce a small amount of variance even without any modifiers in place.

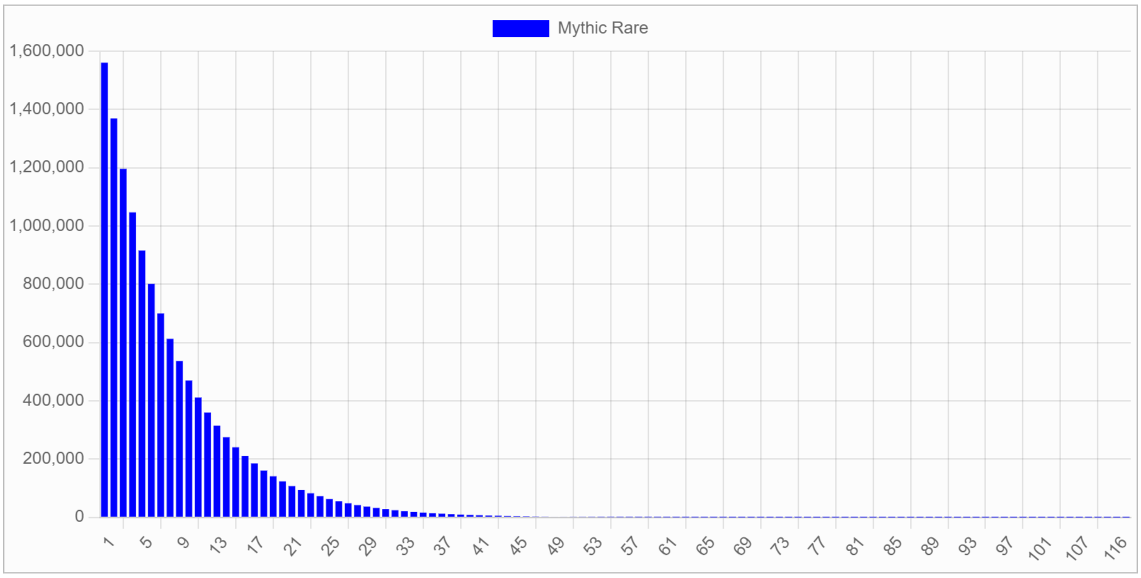


Image 6.13 Mythic rare reward distribution graph for Magic: The Gathering.

From the reward distribution graph in image 6.13, we can then conclude that opening booster packs in hopes of receiving mythic rare cards is just like drawing gacha rewards without guarantee modifiers. The high base probability keeps most of the rewards towards the beginning of the graph but some extreme cases of not receiving the reward may still happen for some unlucky players. Applying guarantee rules to physical booster packs would fix this problem but because each booster pack is a separate isolated purchase, a solution like this would only work for containers of multiple booster packs. This way, for example, each container could contain a fixed amount of booster packs with mythic rare cards and a fixed amount of booster packs with regular rare cards.

7 Conclusions

In the following sections I will attempt to answer the research questions and offer some final thoughts on the results. First, I will try to evaluate if the requirements set for the simulator in chapter 4.1 were met. Then I'll try to assess the usefulness of the simulator for three distinct potential user groups: players, designers, and researchers. Finally, I'll address the simulator more generally and consider its shortcomings as well as its future development. I will consider the various benefits of the customizable simulator throughout this chapter.

7.1 Meeting the requirements

7.1.1 Customizable reward categories

As demonstrated in each of the test cases, this requirement was met. Creation of custom reward categories and dividing them into sub-categories allowed enough precision for each of the gacha systems to be simulated. Two tiers of rewards are usually enough to simulate current gacha systems, but an option to create additional tiers recursively would offer even more possibilities. Additional tiers would allow dividing categories into individual rewards much easier. This wasn't specified as a requirement but is one of the more obvious future development ideas.

7.1.2 Modifiers that adjust probabilities during the draw process

This requirement was met. First two test cases demonstrated the usage of guarantee modifiers in both rewards and sub-rewards. These modifiers can have drastic effects on the overall and individual reward distributions. The tests of Arknights and Genshin Impact both show that the removal of a guarantee modifier could cause some "unlucky" users to never attain a grand prize in the gacha. A reward or its sub-reward can have a single modifier which was enough for all the test cases. Further development could focus on making modifiers completely modular so several modifiers could be attached to one reward or one modifier to several rewards, for example.

7.1.3 Display overall statistics and reward distribution (total price in virtual currency)

This requirement was met. Overall statistics are displayed for every use of the simulator as demonstrated in each of the test cases. The pie chart is a simple way to visualize the overall distribution of rewards. Overall statistics include necessary information such as the total number of draws and calculated overall costs. Overall statistics include total amounts and percentages of each reward from the total amount of rewards.

One idea to improve this view would be to make a distinction between categories and sub-categories. In the current system sub-categories always replace their parent category in these views. In some cases, such as with World Flipper's test case this is suitable since we are combining separate sub-rewards into a single reward. In other cases, such as with Arknights and Genshin Impact, displaying both rewards *and* sub-rewards in some way might be interesting and even helpful.

7.1.4 Display per-reward statistics

This requirement was met as seen with each of the tests. Each reward is handled separately after the overall statistics are displayed. A reward distribution graph, total statistics and averages are all displayed. Additional per-reward statistics should be considered, but the graph and averages are already very illustrative of the results.

7.1.5 Display per-reward averages (mean, median, mode)

This requirement was met. Statistics for each individual reward distribution include the three average values, mean, median and mode. Additionally, these values multiplied with the user-assigned draw currency cost provided examples of average draw costs in the gacha system. Even more statistical variables could be considered for this view. Some of the data displayed might also be unnecessary, so further testing and brainstorming of ideas would be necessary.

7.1.6 Visualize reward distribution

This requirement was met. The pie chart visualizes the overall distribution sufficiently well in each of the test cases. Reward distribution graphs provide a more detailed visual representation for each reward. In the future, the calculated averages could also be added into the reward distribution graph view. As discussed earlier, some distinction between rewards and sub-rewards could also be added into the pie chart visualization.

7.1.7 Allow very large sample sizes

All the tests used a sample size of 100 million draws. Especially compared to summon simulators that generally simulate anywhere from one to ten draws at a time, this requirement can be considered met. The benefits of a large sample size are clear. Average values are easier to determine and the margin of error on any assumptions based on the statistical data is smaller.

7.1.8 Replication of existing gacha systems

This requirement was met as illustrated by the replication of three existing gacha systems. Additionally, a booster card pack was simulated. For each case, reward customization was sufficient, as was the support for guarantee modifiers. The ability to draw a fixed set allowed to connect sub-rewards with different probabilities in World Flipper and to simulate a physical Magic: The Gathering booster card pack. These cases should be sufficient to show that a good variety of gacha systems and loot box mechanics can be simulated. Future development should still focus on bringing new and different types of gacha systems within reach of the simulator's customization options.

7.2 Players

There seems to be a demand for summon simulators in the fan communities. A quick search on Google turns up dozens if not hundreds of community-made creations. Looking at Arknights alone, there are hundreds of simulators for its specific gacha interfaces at Gamepress [2022]. Paimon.moe, a popular Genshin Impact community site, has more than 239,000 users submitting data on their gacha draws from the game [Paimon.moe 2021].

The customizable simulator does however have some clear advantages over any specific summon simulator. The customizable nature of the tool allows the players to simulate gacha systems for games that do not have a dedicated summon simulator of their own. The ability to handle large sample sizes and provide statistical data will give the player a clearer picture of how the gacha probabilities work. It will provide them with information on average cost of receiving a reward, the most common cost of receiving a reward and other potentially useful data. This information can be used to make better purchasing decisions with in-game and real-world currencies or decisions not to invest at all.

The open-source nature of the simulator will also allow anyone to make updates and modifications to the code. This means that most current gacha interfaces, systems and variations could eventually be added to the simulator.

7.3 Designers

The simulator may be helpful in designing gacha systems. The customizable nature of the simulator is naturally useful for a designer. New gacha systems can be built from the ground-up with custom reward lists and modifiers. Since it can simulate existing systems and modify them, designers can take an existing game, replicate its gacha system and adjust it to their own specifications. Tests can be run after adjustments to see how each change affects the distributions and averages. The reward distribution graph is a readable, interactive graph that provides an overview of each reward in the system. This along with the calculated average and total costs can help design and adjust a video game's virtual economy. The virtual currency values can be freely adjusted or even set as real-life currencies. Programming-savvy designers will also have an easier time working with the source code of the simulator than casual players.

7.4 Researchers

The third group mentioned in the research questions, researchers, is a bit harder to evaluate. The simulator can simulate gacha systems and other loot box systems. As the test on the collectible card game booster packs indicates, simulating similar physical products is possible as well. It provides plenty of detailed data on the process and results that anyone researching random draw or lottery systems might find interesting. The large sample sizes allow obtaining better statistical data with a smaller margin for error. Further research on gacha mechanics might be easier with the help of a dedicated simulator. Again, the open-source nature would allow it to be modified to the requirements of the research.

7.5 Simulator evaluation and final thoughts

The simulator is a prototype that I worked on constantly while still working on the study and running the tests. In many cases I had to start over and fix something unexpected in the code. It is likely that it will never reach a "final" state as gacha systems are complex and too many different variations already exist. It is intended to be an ever-evolving tool that anyone with programming skills can modify to their requirements. Some additional thoughts and shortcomings are outlined in the following paragraphs.

The modifiers support simple incremental or decremental changes in the probability. Originally the idea was for each modifier to implement a customizable mathematical function. For example, multiplication, division, or more complicated mathematical formulas could be supported. After reviewing the requirements of the test cases and some other gacha interfaces this idea was abandoned as none of them required anything beyond simple addition and subtraction. It would still be a relatively simple process to implement at least basic formulas that use the probabilities and other variables set in the reward objects.

In the prototype, all the presets were hardcoded in the index.html file. There should be an easier option to save and enter presets. One possibility would be to do this with saving information inside browser cookies. Another would be to create unique encoded URL addresses for each preset. A server-side solution would be to store information into a relation database of some sort. This sort of approach would require a more extensive code rehaul, however.

The prototype also doesn't allow to seamlessly continue a set with the same modifiers. As everything is reset upon starting a new draw, every use of the simulator provides an isolated set of data. It might be useful to see how a draw process proceeds bit by bit. Temporary data could be displayed, and changes observed.

The user interface is relatively simple but some sort of documentation beyond simple tooltips would most likely be helpful to new users. This wasn't a high priority during testing due to the prototype nature of the simulator but would be important in case of a wider release. Similarly, the logging could be clearer and displayed neatly in columns. Currently it is somewhat difficult to browse at a quick glance.

As the primary focus of the study was gacha systems using the regular draw type, the set-based functionality has some notable weaknesses and inconsistencies. If subitems include rewards that share the name and type of another reward in a different subitem, some of the individual reward results can be inaccurate. None of this information was required or used in the tests, however. Guarantee modifiers are also unsupported in the main categories of *set*-type draws.

With all that said, the simulator still worked sufficiently well for each of the planned tests of the presets. Some adjustment was required during the process but most of the goals were met. Benefits over existing options have been considered in previous sections in some detail but two features still stand out to me: The customizability and ability to use very large sample sizes. These features combined with detailed statistics and informative graphs could prove useful especially with further development. I hope to release the source code on GitHub or an equivalent website in the near future.

8 References

Chapple, Craig. 2021. Genshin Impact generates $2 billion on mobile in first year. Sensor Tower, website. Available (accessed May 26, 2022): *https://sensortower.com/blog/genshin-impact-mobile-two-billion-revenue*.

Chart.js. 2022. Chart.js, Chart.js Team and contributors, website. Available (accessed May 26, 2022): *https://www.chartjs.org/.*

Cygames. 2019. World Flipper, Kakao Games.

Gamepress. 2020. Summon Simulator, Gamepress, website. Available (accessed May 28, 2022): *https://gamepress.gg/arknights/tools/summon-simulator.*

Hypergryph. 2019. Arknights, Yostar.

jQuery. 2022. jQuery, The jQuery Foundation, website. Available (accessed May 28, 2022): *https://jquery.com/*.

jQuery UI. 2022. jQuery UI, jQuery UI Team, website. Available (accessed May 28, 2022): *https://jqueryui.com/*.

Kanerva, Taija. 2016. Cultures combined: Japanese gachas are sweeping F2P mobile games in the west, GameRefinery, website. Available (accessed May 28, 2022): *https://www.gamerefinery.com/japanese-gachas-sweeping-f2p-games-west/*.

Koeder, Marco J. and Ema Tanaka. 2017. Game of chance elements in free-to-play mobile games. A freemium business model monetization tool in need of self-regulation? In: *28th European Regional Conference of the International Telecommunications Society (ITS): "Competition and Regulation in the Information Age"*

Koeder, Marco J., Ema Tanaka and Philip Sugai. 2017. Mobile game price discrimination effect on users of freemium services - an initial outline of game of chance elements in Japanese F2P mobile games. In: *14th Asia-Pacific Regional Conference of the International Telecommunications Society (ITS): "Mapping ICT into Transformation for the Next Information Society"*

Kuzuhara, Tomofumi. 2021. The gamers powering Japan's $22.1 billion games market: consumer motivations, behavior & data. Newzoo, website. Available (accessed May 26, 2022): *https://newzoo.com/insights/articles/the-gamers-powering-japans-22-1-billion-games-market-consumer-motivations-behavior-data.*

Last hope. 2020. 中奖概率计算(附代码) [Winning probability calculation (with code)]. Mihoyo.com, website. Available (accessed June 27, 2022): *https://bbs.mihoyo.com/ys/article/2102609*.

Liu, Kevin. 2019. A global analysis into loot boxes: Is it "virtually" gambling? *Washington International Law Journal, Vol.28 (3)*.

McCaffrey. 2019. The macro problem of microtransactions: The self-regulatory challenges of video game loot boxes. *Business Horizons, 2019-07, Vol.62 (4)*, 483-495.

MICA Team. 2016. Girls' Frontline, Dark Winter.

MiHoYo. 2020. Genshin Impact, miHoYo.

Nieborg, David. 2016. From premium to freemium: The political economy of the app. *Social, casual and mobile games: The changing gaming landscape*, 226-227.

Paimon.moe. 2021. Paimon.moe, website. Available (accessed May 23, 2022): *https://paimon.moe/.*

Schwiddessen, Sebastian. 2018. Loot boxes in Japan: Legal analysis and kompu gacha explained. *Interactive Entertainment Law Review.* Baker McKenzie.

Shibuya, Akiko, Mizuha Teramoto and Akiyo Shoun. 2015. Systematic analysis of in-game purchases and social features of mobile social games in Japan. *DiGRA 2015: Diversity of Play*, 1–16.

Shibuya, Akiko, Mizuha Teramoto, Akiyo Shoun, and Kumiko Akiyama. 2019. Long-term effects of in-game purchases and event game mechanics on young mobile social game players in Japan. *Simulation & Gaming 2019, Vol. 50(1)*, 76–92.

sm0rky. 2020. Gacha rates listed in details are wrong/misleading (a computer simulation probability perspective). Reddit.com, website. Available (accessed June 6, 2022): *https://www.reddit.com/r/Genshin_Impact/comments/ja7c0b/.*

Statista. 2022. Statista, website. Available (accessed May 23, 2022): *https://www.statista.com/markets/417/topic/478/video-gaming-esports/.*

Toto, Serkan. 2012a. Japan's social gaming industry announces new set of self-regulatory measures. Serkantoto.com, website. Available (Accessed May 27, 2022): *https://www.serkantoto.com/2012/06/27/social-gaming-japan-self-regulation/.*

Toto, Serkan. 2012b. Six Japanese social gaming companies form council to self-regulate monetization. Serkantoto.com, website. Available (Accessed May 26, 2022): *https://www.serkantoto.com/2012/03/24/japan-social-game-makers-self-regulation-council/.*

Wizards of the Coast. 1993. Magic: The Gathering, Wizards of the Coast.

Wizards of the Coast. 2006. Wizards.com, website. Archived (accessed June 5, 2022): *https://magic.wizards.com/en/articles/archive/feature/wizards-announces-price-change-2006-08-22.*

Yamakami, Toshihiko. 2012. Revenue-generation pattern analysis of mobile social games in Japan." *14th International Conference on Advanced Communication Technology (ICACT). IEEE, 2012,* 1232–1236.

Yomiuri Shimbun. 2012. Yomiuri.co.jp, website. Archived (accessed May 31, 2022): *http://www.yomiuri.co.jp/dy/national/T120505002978.htm.*