# Simulation Project: Diablo Immor(t)al

Dell'Acqua Matteo 09488A

June 20, 2023

# Summary

# 1 Introduction

The objective of this project is to provide a look into the *free-to-play* model for videogames. *Free-to-play* games usually offer a free entry but then heavily limit what the player can do without paying. Examples of this can be limited free lives, slow progress and entire features gated behind a paywall.

This particular project will analyze how long would it take for a player to complete a character on **Blizzard Entertainment©**'s **Diablo Immortal©**, infamous for being one of the most predatory games ever released, and show what a player can expect when accepting to play a *free-to-play* game. For transparency sake, the version of the game that is being simulated is the *day one* release, so successive updates will not be considered.

## 1.1 Gameplay loop

Diablo Immortal's gameplay loop is pretty simple. The main source of equiqment comes from completing "elder rifts" (a short dungeon that takes around 4-5 minutes to complete). Once completed, it gives as loot some equipment of different rarity. In order to unlock more powerful equipment, the player has to pay specific tokens (that the game gives in limited amount to a free-to-play player). In particular, each "elder rift" gives as loot:

- 8 "fading embers" (+1 per "rare/legendary token" used up to 3 per run) up to 300 per week (+120 using "rare" or "legendary tokens");
- 1 "gem" per "legendary token" used up to 3 per run.

### 1.1.1 Rare tokens

Players are given 3 rare tokens per day for free.

### 1.1.2 Legendary tokens

Players are given 1 legendary token a month for free.

### 1.1.3 Fading embers

Earned in "elder rifts". Used to craft "Fa runes".

### 1.1.4 Fa runes

Crafted with 18 "fading embers". Used to craft gems.

### 1.1.5 Gems

Obtained from "elder rifts" with "legendary tokens" or crafted with "Fa runes". Gems can have 1, 2 or ?/5 stars. This project ignores 1 and 2 star gems as they are not useful in completing a character. ?/5 gems can be 2/5, 3/5, 4/5 or 5/5 gems, with their relative drop chance. Running an "elder rift" with a "legendary token", each gem has the following odds:

- 75% chance of obtaining a 1 star gem;
- 20% chance of obtaining a 2 star gem;
- 3.75% chance of obtaining a 2/5 star gem;
- 1% chance of obtaining a 3/5 star gem;
- 0.20% chance of obtaining a 4/5 star gem;
- 0.05% chance of obtaining a 5/5 star gem.

After 50 legendary token spents without getting a ?/5 gem, a "pity system" kicks and forces a ?/5 gem to drop on the $50^{th}$ token spent.

In addition to this, a player can pay 22 "fa runes" to craft a random legendary gem with the following odds:

- 75.395% chance of crafting a 1 star gem;
- 20.105% chance of crafting a 2 star gem;
- 3.375% chance of crafting a 2/5 star gem;
- 0.90% chance of crafting a 3/5 star gem;
- 0.18% chance of crafting a 4/5 star gem;
- 0.045% chance of crafting a 5/5 star gem.

### 1.2 Completing a character

In order to continue with the modelling process, an important definition is needed: what does it mean to complete a character?

The first thing to exclude are features that are behind a paywall: "activating" the gems require a premium currency that can only be bought with real money, so a really complete character is actually impossible without paying. In order to continue the project, it is assumed that a complete character is one with all 6 gem sockets filled with 5/5 star gems (that in order to get with payed legendary tokens would correspond to around 24000$[2], a little less considering crafting).

## 2 Model

The simulation will be agent based.

In general, the player is modelled as an agent that decides whether to play or not (depending on its available time). If it decides to play, it starts by running an elder rifts using all the available tokens it has, then proceeds to craft fa runes and, if possible, gems.

## 2.1 Running an elder rift

An elder rift is modelled as a stand-alone agent. Once it receives from the player the signal to run, it simulates the time it takes to complete the dungeon through a Normal Random Variable with $\mu = 4$ and $\sigma = 0.5$. After the simulated time, it sends the player the loot for completing the dungeon.

The loot contains both the earned "fading embers" (in deterministic number) and the gems (one gem per "legendary token" is generated with a Bernoulli Random Variable with $p = 0.05$ for a ?/5 star gem or with $p = 0.005$ for a 5/5 star gem).

## 2.2 Crafting fa runes

Crafting fa runes is not a stochastic process, but it is simulated as it takes some time to navigate menus and NPCs, so the crafting simply delays the result by 10 seconds (it is not negligible as some players only play the game for about 10 minutes a day, so 10 seconds actually make a difference).

## 2.3 Crafting gems

The process of using fa runes is modelled in a similar way to the elder rift: an agent with which the player interacts by sending a message to begin the interaction and waits for the results of the gem crafting. After the player initiates communication with the gem crafter, it simulates the time it takes to navigate the menus, skip dialogues, etc. Once the time is over, it sends back the crafted gem (generated with a Bernoulli Random Variable with $p = 0.045$ for a ?/5 star gem or with $p = 0.0045$ for a 5/5 star gem).

## 2.4 Simplifications

In the modelling process, the following simplifications will be used:

- gem crafting will ignore gems' uniqueness: every crafted gem is unique and duplicates are not taken into consideration;
- the delays for the crafting for fa runes and gems are simulated as being deterministic as they are not negligible enough to be ignored but at the same time simulating them as stochastic would add computational complexity to the simulation without impacting it in a noticeable way.

# 3 Implementation

The simulation is implemented in **Anylogic**[1]. The player is implemented as an FSM as seen in Figure 1. Inside a player, there are two bound agents: 1 elder rift and 1 gem crafter. The crafting of fa runes is handled directly by the player as it does not involve stochastic elements and it is not complex enough to justify implementing a stand-alone agent. Both the elder rift (Figure 2) and

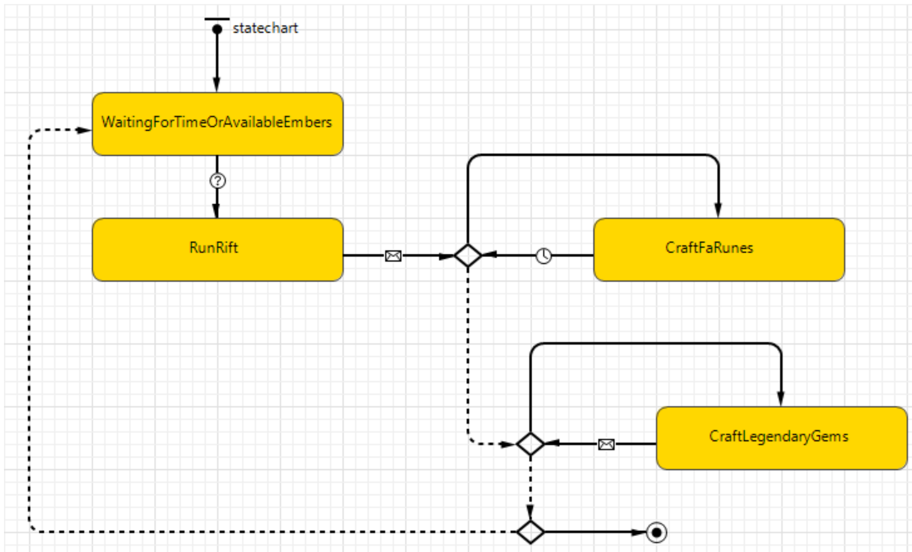


Figure 1: A player's Anylogic implementation

the gem crafter (Figure 3) are FSMs bound to the player that follow the simple pattern:

1. Idle state;
2. start signal;
3. simulate gameplay actions;
4. generate output;
5. send output signal to player.

### 3.1 Agent population

In order for the simulation to be relevant, a significant number of runs must be considered. In order to parallelize this simulation as much as possible, and to enable a visual evaluation at simulation-time, a population of 10000 players is simulated in a single run. While this allows to visualize aggregated data at runtime, these players are not to be considered as interacting: each player acts independently of every other player.

### 3.2 Gem quality

The quality of the gems is actually simplified to two boolean attributes in the "gem" class:

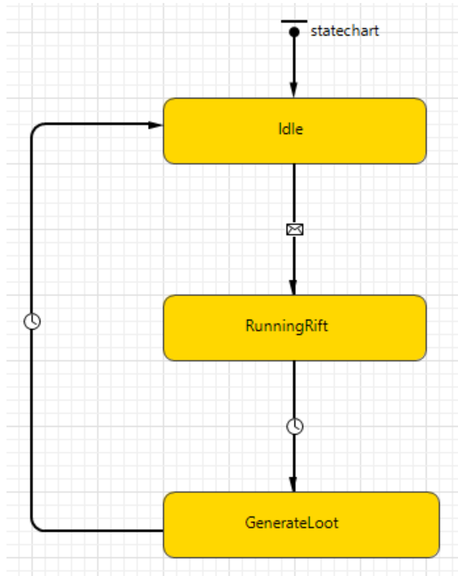


Figure 2: The elder rift implementation in Anylogic

- *isFiveStar* (*true* if the gem is of the ?/5, which includes the 5/5, type);
- *isFiveOutOfFiveStar* (*true* if the gem is of the 5/5 type).

# 4 Key System Parameters (KSP) and Key Performance Indicators (KPI)

## 4.1 Key System Parameters

In the first place, there are some evident KSPs: the loot and crafting odds are clearly relevant, but all they do is scale the time it takes, without actually changing the distribution shape. The pity system is also relevant, but the more important ones are more complex to identify.

In a free-to-play game, it is usual to try to hide how the system really work by putting useless mechanics and parameters in front of the real bottlenecks, but in order to understand how bad (for the player) this model of marketing a game actually is, two classes of KSP must be identified: the apparent KSPs and the real KSPs.

### 4.1.1 Apparent Key System Parameters

On a first glance, the real important KSP in this simulation is the time each player devotes to the game each day: the more a player plays, the faster he can

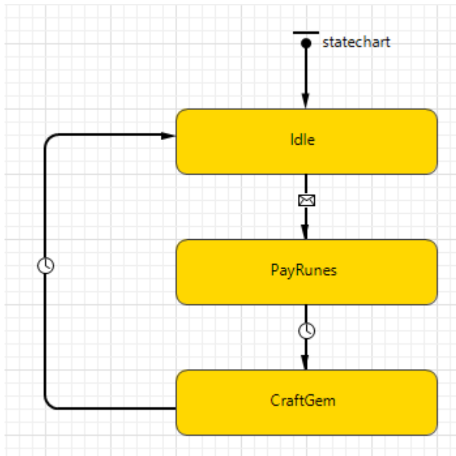


Figure 3: The gem crafter implementation in Anylogic

complete the game. This should be obvious and valid for all videogames.

Secondary KSPs could be identified in the number of rare and legendary tokens the players receive for free each day and each month, as the rare ones allow the player to earn more fading embers to craft more gems and the legendary tokens allow the player to loot more gems in elder rifts (in addition to also earning more fading embers).

#### 4.1.2 Real Key System Parameters

As seen in 5.2 and 5.3, reality is different:

- the number of free legendary tokens is a real KSP;
- there is an artificial cap for fading embers at 300 per week that effectively renders the time the player spends on the game after the cap is reached useless;
- the number of free rare tokens is useless after a certain threshold as there is an artificial cap for the fading embers that can be earned with tokens (fixed at 120 above the normal cap of 300 per week);
- given the previous two points, it is easy to notice how there is a threshold on the time spent on the game per day after which more time does not influence the result.

In the end, the parameters that have the largest impact on the simulation are the number of legendary tokens given to the player for free and the caps on fading embers.

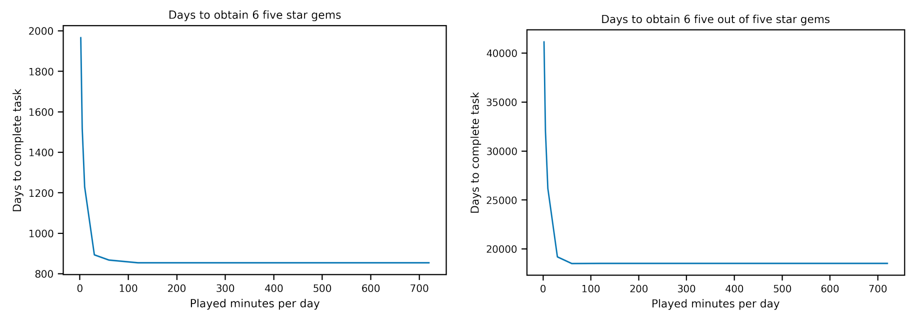


Figure 4: Average time to acquire 6 ?/5 star gems (left) and 6 5/5 star gems(right) using the original model.

### 4.2 Key Performance Indicators

For this analysis, there is only one main key performance indicator: the time it takes for a player to complete the task. A secondary indicator could be the remaining time the player has to play the game but is already capped (so it should not bother playing).

## 5 Experimental analysis

The experimental campaign for this model focused on 3 experiment batteries. In order to highlight how these KSP influence the time it takes a player to complete a character. For comparison sake, each battery is with two different definitions of completing a character:

- obtaining 6 5/5 star gems;
- obtaining 6 ?/5 star gems.

### 5.1 Original model

Figure 4 shows the results of the simulation on the original model. As can be seen from the charts, the threshold for the played minutes per day to be relevant is really low: after 60 minutes it no longer has any impact on the simulation. In addition to this, the charts show that in order to obtain 6 ?/5 star gems you need to play for at least around 850-900 days (∼2.5 years) and to obtain 6 5/5 star gems you need to play for at least around 19000 days (∼52 years).

### 5.2 What-If scenario 1: 2 legendary tokens per day (5$/day)

Figure 5 shows the results of the simulation if the player decided to buy 2 legendary tokens a day (an expense of 5$/day). Once again, the relevance threshold for the played minutes per day is still low (60 minutes) but the bounds

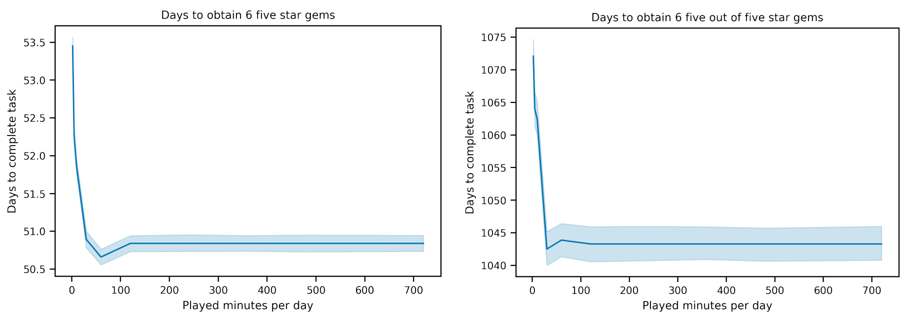


Figure 5: Average time to acquire 6 ?/5 star gems (left) and 6 5/5 star gems(right) spending 5$/day.

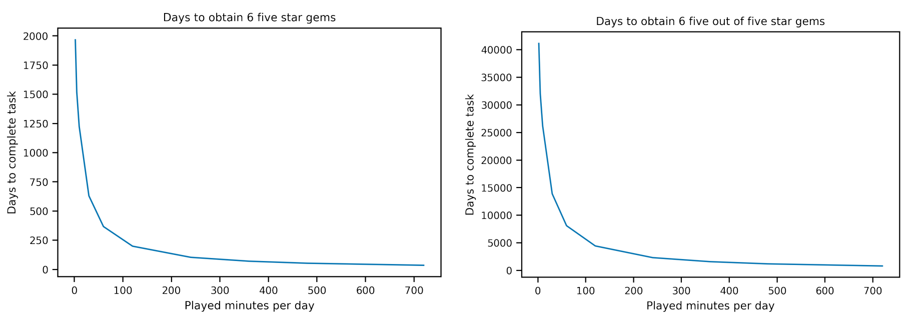


Figure 6: Average time to acquire 6 ?/5 star gems (left) and 6 5/5 star gems(right) raising the cap to fading embers to 200000/week.

of the simulation changed considerably. For the ?/5 case, the upper bound went from around 2000 days (∼5.5 years) to around 53 days (a little less than 2 months) and the lower bound from around 850 days (∼2.5 years) to around 51 days. For the 5/5 case, the upper bound went from around 41000 days (∼112 years) to around 1070 days (∼3 years) while the lower bound went from around 19000 days (∼52 years) to around 1045 days (∼3 years). It is then possible to see how the difference between upper bound and lower bound is drastically reduced: by going from 5 minutes/day to 60 minutes/day the player can reduce the time it takes to complete the task by at most 3%.

### 5.3 What-If scenario 2: fading embers cap raised to 200000/week

Figure 6 shows the results of the simulation if the cap to the fading embers was raised to 200000. In this case, the relevance threshold for the daily playtime is increased to all of the considered domain [5, 720], so increasing the playtime always translates to a decrease of the time required to complete the task. Of course the increase the player gets towards by passing from 360 minutes/day to

720 minutes/day is greatly inferior to the one it gets going from 5 minutes/day to 10 minutes/day.

Regarding the simulation bounds, the upper ones are unchanged, but the lower ones are no longer dictated by the artificial cap. For the ?/5 case, the lower bound went from around 850 days (∼2.5 years) to around 125 days (∼4 months). For the 5/5 case, the lower bound went from around 19000 days (∼52 years) to around 1250 days (∼3.5 years).

### 5.4 Experiment battery relevance

In order for these results to be relevant, the experiment battery needs to be relevant too. For these experiments each run simulates 10000 players and 10 runs are simulated for each parameter set, so for each parameter set a total of 100000 players are simulated.

## 6 Conclusions

As the experimental campaign presented in 5 shows, the free-to-play marketing model for videogames is actually based on deception and behavioral psychology:

- by writing on the price tag ”free” a lot of user will download and try it even if they are not fully convinced in the quality of the game;
- the free-to-play model often tries to convince players to spend in the game by artificially slowing down progress speed (in order to complete the game one has to play everyday for at least more than 50 years);
- the ingame shop is build using behavioral psychology in order to push players to spend money [3, 4];
- different fake currencies are used in order to confuse the players and make them lose track of how much they have to spend to get to a certain objective.

As a final thought, the author hopes that governments will regulate this predatory marketing by forcing companies to label the games as gambling and by putting every price as real currency and not as fake ingame currency.

## References

[1] Matteo Dell'Acqua. *Diablo Immoral*. 2022. URL: https://github.com/MatteoH2O1999/simulation-project.

[2] dimi777. *Diablo Immortal $25 Elder Rift Simulator*. 2022. URL: https://github.com/dimi777/diablo-rift-simulator.

[3] Josh Strife Hayes. *The Immoral Design of Diablo Immortal*. 2022. URL: https://youtu.be/o17lBUZgjTs?t=181.

[4] Torulf Jernström. *Let's go whaling: Tricks for monetising mobile game players with free-to-play*. 2016. URL: https://www.youtube.com/watch?v=xNjI03CGkb4.