UNIVERSITY OF TRENTO - Italy
Department of Information Engineering and Computer Science

Simulation and Performance Evaluation

Final Project report

AY 2019/2020

Student: Carlo Fanciulli – 198793

1.Introduction

This document contains an explanation of the project context and the problem analysed, followed by a brief illustration of the simulator implementation created also using a flowchart to get a general idea of it in a short time and to graphically represent every step of the process. The experiments and graphs created will be explained to give an overview and to show the results obtained.
The project is built with Python in order to exploit the benefits of using some common libraries for data manipulation such as Numpy and Matplolib for data visualization.

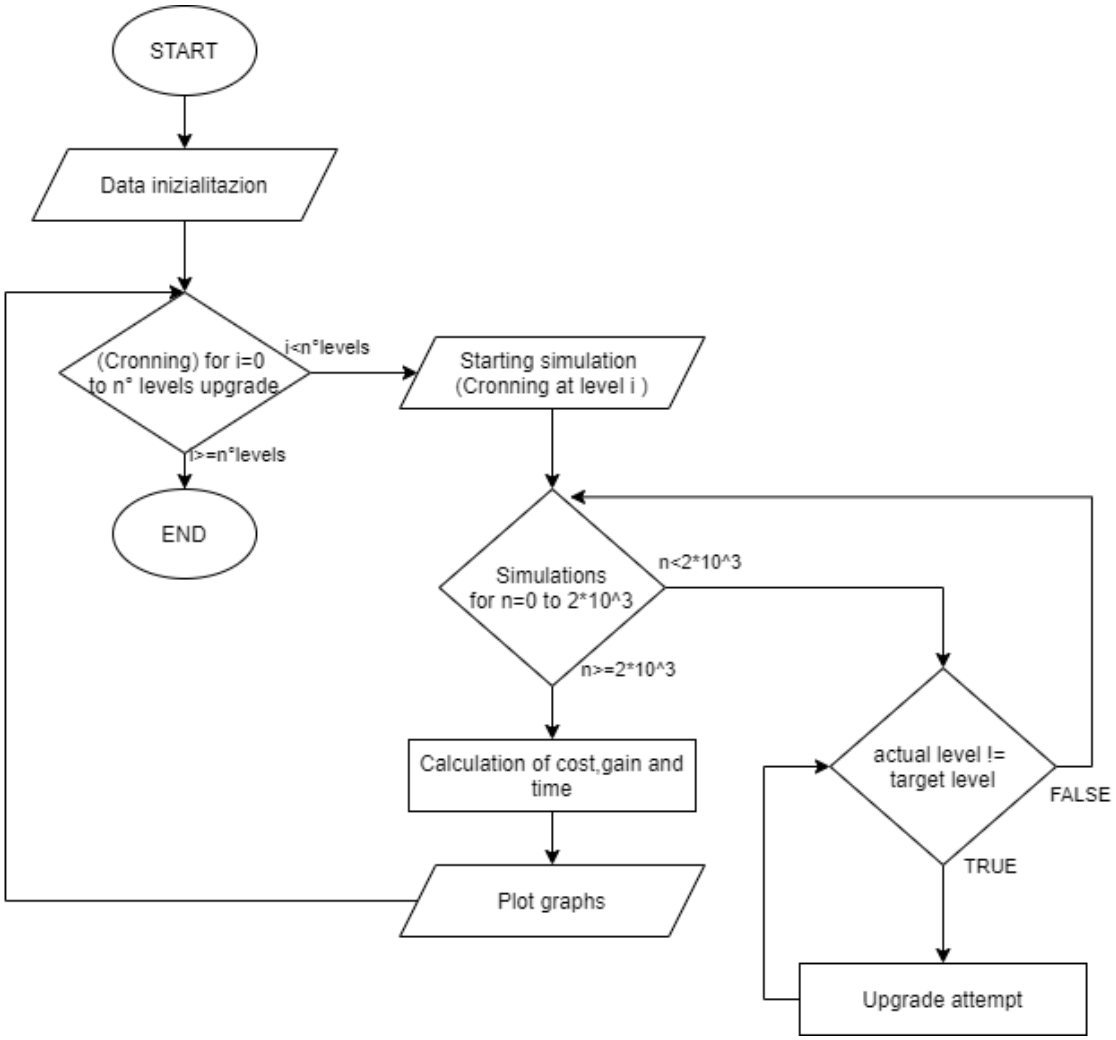
2.Context

In the videogame Black Desert Online (BDO), an enhanced item can be bought at auction from other players or enhanced personally. Due to the complexity of the enhancement system, the optimal decision is not always obvious, because the maximum price of each item is set to a predefined value and players do not always keep the items in their possession, sometimes selling them is a useful tool to recover funds for the next upgrade.
The order of upgrade levels are PRI, DUO, TRI, TET and PEN. As item reaches higher levels, it becomes stronger and more valuable. Of course, increasing the level also increases the difficulty of successfully obtaining the next upgrade.
Each upgrade has a chance of happening. Each time an upgrade fails, the chance of the next one succeeding goes up by a predefined amount. This system is called "fail stacking". Failstacking is a process through which the player can increase his or her chances of success. Every time he fails an enhancement attempt, he will be given extra enhancement chances. This is an indicator of the times the player has continuously failed at enhancement. Therefore, the more times one fails enhancement in a row, the more failstacks one obtains and the more failstacks one has, the greater the chances to succeed at the current upgrade. A failstack (a set of failures) can be saved (with its additional chance) and used later. If an item is improved with failstacks, they are reset and have to be accumulated again.
To make the process even more of a gamble, there are two ways to improve an object: one is what is called a "raw tap", or unsafe, and the other is called "cronned". When a “raw tap” fails, the value of the object decreases significantly as its quality drops. To return to the previous quality, the object must be upgraded twice. So, if the upgrade fails going from DUO to TRI, the item has a chance of dropping to PRI instead.
A "cronned" upgrade, used to enchant items safely, involves the use of a predefined amount of an additional object called “Cron stone”, that actually prevents the level decrease when enhancing, but it’s fairly resource-intensive.
In any case, an item that is upgraded (successfully or unsuccessfully) loses durability. Each item has 100 durability and this statistic must be kept at the maximum because if it is less than 100, the item is less usable in the game and also cannot be sold at auction. Obviously, it’s possible to restore the maximum durability of items but this has a cost. In fact, durability is restored by buying and using items called “Memory”.
The goal of the simulator is to decide whether improving the equipment is an advantageous action compared to buying from other players at auction, considering that the item will have to be upgraded from basic to higher level. It will have to take into account also the costs of repairing durability and the possible cost of recovering failures if necessary.

3.Simulation

To get a statistically relevant results from the project many simulations are carried out by the software. The use of a pseudo-random number generator, is necessary to obtain a sequence of random values that is representative of the distribution to be modelled. For this reason, a different incremental seed was used for each simulation.
Parameters were initially set for the simulation and they will be used both to calculate the probability of a successful upgrade and to calculate the costs of cron and failstack.
In fact, the failstack class has been created and 4 failstacks have been initialised so that each time an upgrade passes, the failstack that is most convenient for the level of upgrade desired at that moment is used. So for a DUO a failstack between 20 and 30 will be used, for a TRI one between 30 and 40, for a TET between 40 and 90 and for a PEN between 90 and 190. Obviously, each of these failstacks has a different predefined cost that has been set.
Another parameter is the number of Crons needed to make a single upgrade attempt. Each level has a different number needed, in fact, only 1 Cron is used for a DUO attempt, for a TRI 38 Crons, for a TET 113 Crons, for a PEN 429 Crons. Each Cron has a cost of 1 million and each Memory, required to restore durability, has a cost of 2 million.
An important parameter is the probability of success for each upgrade level, consisting of the basic probability and the probability that increases each time an upgrade attempt fais, taking failstacks into account.
In order to calculate the cost and gain of improving an item, the cost of a basic item(PRI) worth 100 million and a maximum level item (PEN) worth 18 billion was established.

Once the data had been initialised, it was decided to run simulations for one object using crons at a certain level. Therefore, for each object, with the cron at a different level, 2*10^3 simulations are carried out so as to have a very precise estimate of the results that will be obtained. In each simulation, the item is increased from the base level to the highest level, making thousands of attempts to improve it. The probabilty, given the current level and the failstack, is equal to the basic chance + (number of failstacks * increase of chance per failstack).
In case of success, the failstack is lost and the item is improved, in case of failure, failures are counted for the current level and, if cron was used, no level is lost and no failstack is gained. if no cron was used, level is lost but failstack is gained (+3). At the end of a series of simulations, the costs incurred, the profit gained and the time spent are counted. In addition, graphs are created for each upgrade. A flowchart explaining the functioning of the simulator is shown below.



4.Results

As said in the introduction, Cron stones are used to protect an item from downgrading a level after a failed enhancement attempt.
Each table below is a set of simulations with a series of upgrades aimed at taking the object from the lowest to the highest level.
What is taken into account in each table is where the Cron is used. In fact, As can be seen in each table, the title indicates at which level the Crone stones have been used (DUO, TRI, TET and PEN). For example, "Cron ALL" indicates that the Cron stones are used at all levels while in the table labelled "Cron TET" the Cron stones are only used when the item's level is in TET and so on.
The first parameter considered is the number of crons used to get from the first to the last upgrade level. In Cron NONE, as they have not been used, the number is 0.
The various costs in each table are alternatives and they are differentiated by considering failstacks and Cron stones with or without cost. In fact, the first case (pay both) is the most realistic and also the one that costs the most as both failstack and Cron stones are paid for in the game. The last case (free fs, free cron) is less realistic and less expensive because the cost of these items is not taken into account.
The gain is obtained from the value of the item that was upgraded to the maximum minus the cost of what was initially paid (at base level), the cost of failstacks, Cron stones used and Memory.
The time is formed by the click needed to make the upgrade attempt, made every 3 seconds, plus the time to create failstack, set at one minute and the time needed to obtain the lost durability, set at 5 seconds.

Cron ALL

Crons	14774.068
Cost (pay both)	15221.236
Cost (free fs, pay cron)	15221.236
Cost (pay fs, free cron)	447.168
Cost (free fs, free cron)	447.168
Gain	2778.764
Time (h:m:s.ms)	0:06:16.697

Cron TET

Crons	14362.705
Cost (pay both)	15168.678
Cost (free fs, pay cron)	15001.409
Cost (pay fs, free cron)	805.972
Cost (free fs, free cron)	638.704
Gain	2831.321
Time (h:m:s.ms)	0:16:19.034

Cron TRI

Crons	9642.911
Cost (pay both)	12405.427
Cost (free fs, pay cron)	10703.879
Cost (pay fs, free cron)	2762.516
Cost (free fs, free cron)	1060.968
Gain	5594.572
Time (h:m:s.ms)	0:34:38.481

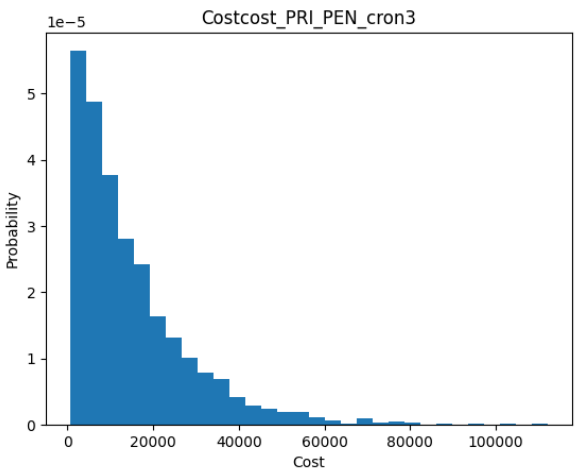
Cron DUO

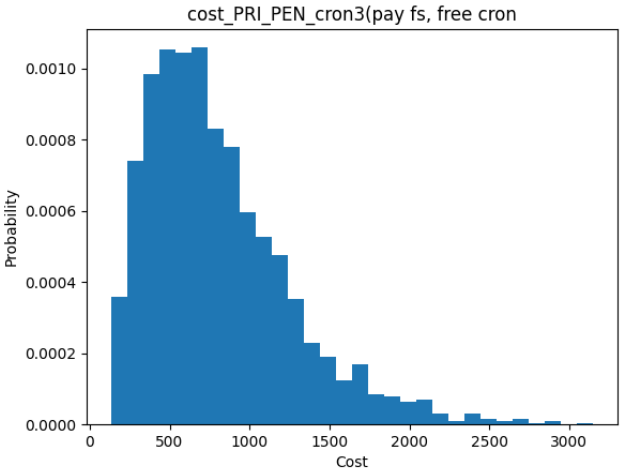
Crons	6764.361
Cost (pay both)	12287.429
Cost (free fs, pay cron)	9090.397
Cost (pay fs, free cron)	5523.068
Cost (free fs, free cron)	2326.036
Gain	5712.570
Time (h:m:s.ms)	1:31:59.721

Cron NONE

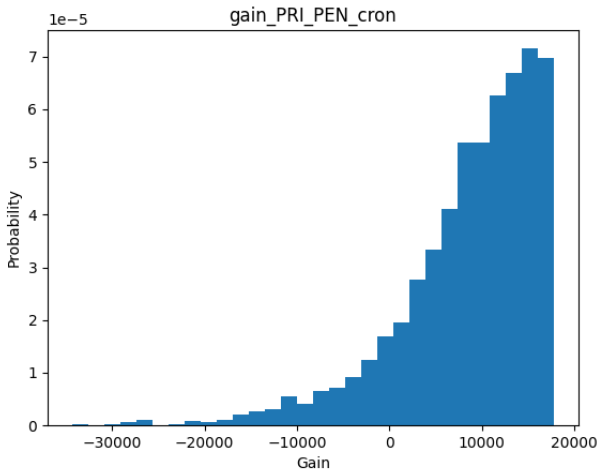
Crons	0.0
Cost (pay both)	9358.452
Cost (free fs, pay cron)	4549.644
Cost (pay fs, free cron)	9358.452
Cost (free fs, free cron)	4549.644
Gain	8641.547
Time (h:m:s.ms)	3:32:54.287

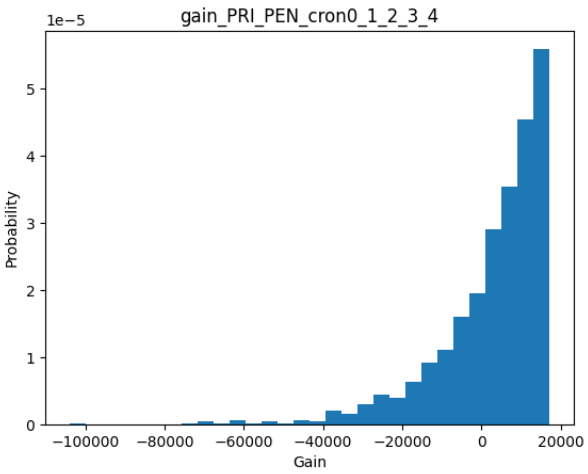
In the following diagrams, the results of the simulations are visually represented and are to be considered with empirical relevance. Further information on statistically important findings will be given below. All developed graphs can be found in the project folder.
In these graphs, the TET Cron is taken into account and the use of Crons is only considered during the upgrade from TET to PEN. In this case, there is a clear distinction between the two graphs as in the one on the left Crons have a cost, whereas in the one on the right Crons are free. Obviously, the most realistic case is the one with paid Crons, but it is possible to see that the cost of Crons for this upgrade is very high.



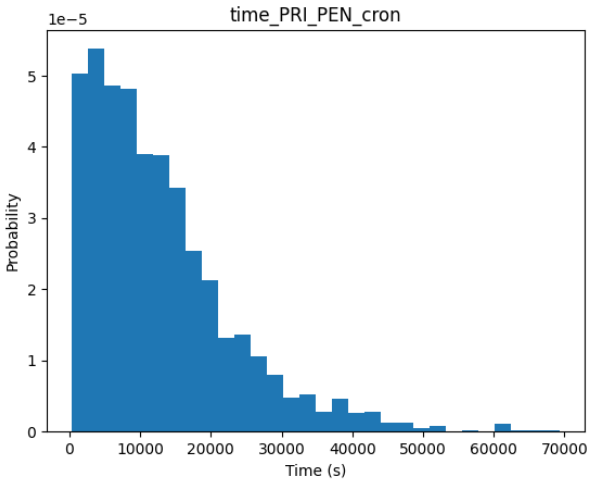


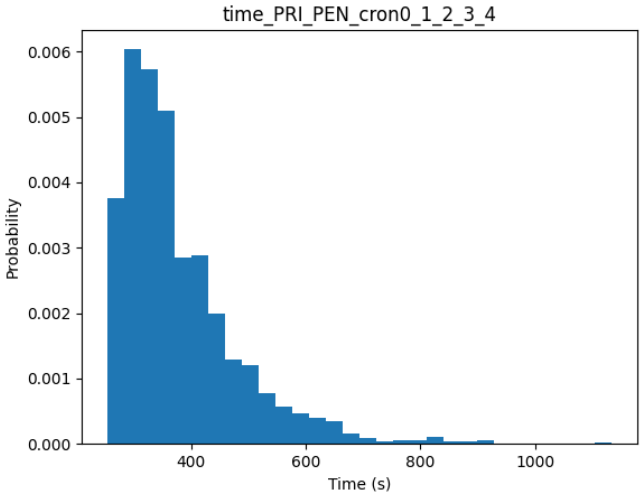
Gain is the most important metric to take into account since, as can be inferred by the graphs, it is better to improve the item than to buy it at auction. In fact, if the gain tends towards 0, it means that it is not convenient to invest resources in it. While if the gain is negative, it is better to buy the item. There is a significant difference between the gain obtained by using Crons at all levels and the gain obtained by not using them at all. In fact, the absence of an expense for Crons, which would be consistent, leads to a higher gain; therefore, it is apparently more convenient not to use Crons. However, in reality, as highlighted by the following graphs, the time needed to improve the item up to the last level without using Crons is much higher than if using them.
It can also be observed that in the case of the use of Cron, they also increase the possible maximum loss, as can be seen from the graph on the right where the gain also reaches 100 billion in the negative.





In fact, as you can see from the graphs below, the time needed to improve an object in the two cases is substantial. This highlights the fact that although it is more convenient in terms of gain to prefer not to use the Cron, the amount of time to finish the upgrade is very high and therefore in practice it is not worth considering this option as the best one. To make it clearer, the average time needed to upgrade an item using Cron at all levels is 6 minutes, not using Cron the average time needed is 3.5 hours. Both considering an upgrade attempt made every 3 seconds.





6.Conclusions

In this project, several parameters were taken into account in order to determine whether it is more cost-effective to improve an item on your own or to buy it at auction from other players.

Summarizing the results:

- It is always better to improve one's own items than to buy them. This means that when deciding whether to enhance an item or to buy it at an auction, it is always more convenient, in terms of gain, to enhance it. More specifically, the use of Crons for enhancement is always better than buying an item regardless if the Crons are used to upgrade all, none or just some levels

- Time is a key variable to consider because, although intuitively it may seem that it is clearly better not to use the Cron, then realistically the improvement is so time consuming that it is better to find a gain in other ways.

- Realistically the most convenient option is cronning TRI (Cron TRI table) because the gain and time needed are optimal compared to the other cases and it is the most feasible in practice.