The Making of a Conceptual Design for a Balancing Tool

Av: Jonas Eriksson
Handledare: Inger Ekman
Abstract

Balancing is usually done in the later phases of creating a game to make sure everything comes together to an enjoyable experience. Most of the time balancing is done with a series of playthroughs by the designers or by outsourced play testers and the imbalances found are corrected followed by more playthroughs. This method occupies a lot of time and might therefore not find everything. In this study I use information gathered from interviews with experienced designers and designer texts along with features from methods frequently used for aiding the designers to make a conceptual design of a tool that is aimed towards simplifying the process of balancing and reducing the amount of work hours having to be spent on this phase.

Keywords

Balancing, Interviews, Conceptual Design, Game Development, External Tool
**Sammanfattning**

Balansering görs framförallt i de senare faserna när man skapar ett spel för att se till att alla delar tillsammans skapar en bra upplevelse. För det mesta utförs balanseringen i form av upprepade speltester genomförda av antingen utvecklaren eller inhyrda testpersoner. Obalanserade saker som upptäcks korrigeras och följs sedan av ytterligare tester. Denna metod tar väldigt lång tid att utföra och på grund av detta är det inte säkert att alla fel upptäcks innan spelet lanseras. I den här studien använder jag information som insamlats från intervjuer med erfarna designers och designextexer sida vid sida med funktioner från metoder som ofta används för att underlätta för utvecklarna. Detta för att göra en konceptuell design av ett verktyg som är riktat mot att simplifiera balanseringsprocessen och reducera mängden arbetstimmar som behövs läggas på denna fasen i utvecklingen.

**Nyckelord**

Balansering, Intervjuer, Konceptuell Design, Spelutveckling, Externt Verktyg
Table of Contents

Contents

ABSTRACT ................................................................................................................................................. 2

SAMMANFATTNING ................................................................................................................................. 3

TABLE OF CONTENTS ............................................................................................................................ 4

1 INTRODUCTION ......................................................................................................................................... 6

1.1 BACKGROUND ....................................................................................................................................... 6

1.2 PREVIOUS AND RELATED RESEARCH ............................................................................................. 6

1.3 RESEARCH GOALS .............................................................................................................................. 7

1.4 ESSAY STRUCTURE ............................................................................................................................. 8

2 METHOD .................................................................................................................................................... 9

3 RESULTS .................................................................................................................................................. 11

3.1 INTERVIEW RESULTS ......................................................................................................................... 11

3.1.1 CHARACTER ABILITIES ............................................................................................................... 11

3.1.2 GAME AND CHARACTER BALANCE ........................................................................................... 12

3.1.3 OVERVIEW AND AVAILABILITY ............................................................................................... 13

3.2 TEXT RESULTS ..................................................................................................................................... 14

3.3 PROGRAM DESIGN ............................................................................................................................ 15

3.3.1 INCREASE THE ACCESSIBILITY OF VARIABLES ................................................................... 15

3.3.2 LINKING VARIABLES FOR AUTOMATIC BALANCE ................................................................. 16

3.3.3 IMPLEMENTING DESIGNER MADE CALCULATIONS ............................................................... 16

3.3.4 PERMANENTLY CHANGE VALUES WHILE THE GAME IS RUNNING .................................... 17

3.3.5 ENABLE WORK FROM DIFFERENT LOCATIONS ...................................................................... 17

4 DISCUSSION ........................................................................................................................................... 19

5 CONCLUSIONS ....................................................................................................................................... 20

6 REFERENCES .......................................................................................................................................... 22
7 APPENDIXES .................................................................................................................. 24
7.1 INTERVIEW STRUCTURE ...................................................................................... 24
1 Introduction

1.1 Background

Balancing is a key aspect when developing almost any kind of game. Without it, the game might lose a big portion of its appeal. Balancing is done to make sure the progression goes smoothly and that all the different values are in sync with each other to create emergence (Salen & Zimmerman, 2004, p. 159-171) that is so important in games. If a game has been poorly balanced it might result in overpowered opponents or the player itself being too strong, or the drop rates and amount of gold that is found is either too high or too low. If it is off in any direction it will have a negative effect on the game's impression.

Balancing is often very time-consuming as it is generally done by designers or outsourced testers and includes a very large number of playthroughs, then tweaking some variables followed by more playthroughs. In Unity for example, the variables that need changing will in most cases have been made public by the programmers to ease access and reduce the risk of accidentally changing the wrong parameter. This also takes valuable time that could have been spent on something else. It has many times been discussed that this way is far from optimal, but it is hard to do it differently since “there is no mathematical formulation of game balance that unifies many of its forms” (Jaffe, et al., 2012, p 26). The focus of this study will be to try to decide what features would be needed in a third party program or plugin that simplifies balancing all those different variables thus increasing productivity and quality.

1.2 Previous and Related Research

Apart from guidebooks to balancing and articles about the importance of proper balancing there was not much actual research on the topic. I managed to find a couple of concrete examples of previous work.

One is a study done by three Japanese students on how to solve the balancing problem of online role playing games through the use of evolutionary algorithms to maintain a good balance between the function of each character’s attributes
and abilities (Chen, et al., 2012). They use integration of the algorithm together with probabilistic software and construct a simple turn based game to do experiments on. The study is extremely heavy on math and very complicated due to the fact that they focus on RPGs with multiple races and classes that all have different abilities. The second thing I found was an already developed tool for balancing variables created through the tool while the game is running, called “Tuning Tool” developed by Clifford Roche (cliffordroche.ca, 2012). This is exclusively for Unity (Unity3D, 2014) and is only available through their “Asset Store”. The tool runs as a plugin in Unity and is used to create tuneable variables that can be changed in real time and applied onto the variables in the project that reset when the game is stopped. Since it is not available for a free trial I have not tested it myself and when looking for other users there was nothing. There are no reviews, no comments, and no tutorial videos available and Roche has not released the source code so there is no way to analyse how the tool is constructed or how well it works. However, the direction he took is similar to the one I plan on taking with a somewhat simplistic plugin that allows for overview and easy tweaking of multiple variables.

Lastly there were studies on dynamic game balancing. Dynamic game balancing is where the game constantly monitors the player’s current skill level and adjusts itself to an appropriate level to keep the player satisfied and interested. In this case, the study was done using Reinforced Learning to get a computer controlled opponent, an NPC, in a fighting game to adapt to the human player and fight at a similar level (Andrade, et al., 2005). Dynamic game balance is by its very nature different from classic balancing since the point of it is to eliminate the set difficulties that are usually part of most PvE games where the player just face more enemies with more health and more damage for example. The same amount of balancing must of course be done to make sure the ability of the dynamic NPC is scaling properly with the player’s increased skill as the game goes on instead.

1.3 Research Goals

This is a qualitative study using the informed grounded theory method, IGT, (Thornberg, 2012) with the goal to first establish and understand the problems
of balancing a “Player vs. Environment”- or PvE-game, through interviews and analysis of designer texts. The information gathered will then be used to make a conceptual design of a third party tool that is aimed at easing the process of balancing a large number of variables continuously through accessibility and increased overview. The focus will lie on trying to determine what functions and features are needed for the tool to be useful during balancing. Despite this paper being focused on tool development, no actual developing will be done beyond the conceptual design.

1.4 Essay structure

Following this introduction will be the Method section where the choice of methods used in the essay will be explained, where and how the data were gathered and how that information then was analysed. After that the Results section presents what was found during the interviews and in the designer texts in separate paragraphs, followed by the conceptual design of the tool based on these findings. Next is the Discussion and Conclusions section where problems experienced during the study are brought to light as well as what was unexpectedly useful. In this section, the different possibilities for future research are also presented along with what areas of use the finished tool might have. Then comes the References containing all the literature, websites and software used in the making of the study and lastly the Appendixes which is only the structured questions used for the interviews.
2 Method

The method chosen for gathering the needed information was a series of in-depth interviews (Cote & Raz, n.d.) with experienced designers and texts written about balancing. The interviews were prepared with some set questions (Appendix 6.1) to ensure that the conversation did not spread out to other areas that were of no importance to the study. The interviews were conducted in form of a call or chat over Skype (Skype, 2014) in both Swedish and English and were recorded for further analysis with Amolto Call Recorder (Amolto, 2014). To get in contact with designers that were willing to be interviewed, a short ad describing what kind of experience was preferred and what purpose the interview had was placed on Facebook (Facebook, 2014).

When the interview started it was made certain that the designer in question agreed to being recorded for later use and also if he or she was willing to have their name stated in the essay or remain anonymous. After that the interview played out following the interview structure with follow up questions regarding some of the answers. Out of the four designers that were interviewed, two agreed to having their name published if needed and two wanted to stay anonymous. One of the anonymous designers also did not want to be recorded so the interview in that case was made using the Skype chat. When all interviews had been conducted, the recordings and chat log were analysed to better understand the information and how it could be used for the design of the program, for example what was more or less important to have due to the nature of challenges that were expected to occur. The designer texts were read through while looking for what aspects of balancing they considered being especially important and how to handle them. This information was then analysed further to see if anything would be beneficial to integrate in the program and how connections between them could be made.

As mentioned earlier I use “informed grounded theory”, IGT, as opposed to the normal grounded theory, GT. The main difference is the mentality considering the use of literature. In GT there is no use of literature while gathering data and while analyzing that date, this in order to keep the researcher free from
previously stated theories and concepts, since if this was to happen “the result would be a constructed theory, supporting what was already known, rather than emergent theory providing new insight” (Heath, 2006, p. 520). This was not an option considering that the very purpose of this study is to gather what is already known about balancing from both designers and texts and condense this information together with the data simultaneously gathered from the interviews into a conceptual design of a program. That is why IGT was chosen.

IGT encourages the use of literature in early stages of research because it can be a source of inspiration and a way to use existing theories and ideas as “lenses” to focus on certain elements and make new associations. This part is very much needed for this study since looking into how different people view balancing and placing all of these different experiences on top of each other to then create associations between them and make that into a function in a program is how results are achieved. If the literature had not been read in the data gathering phase of the research one can assume a lot of connections between different views on balance would have gone unnoticed due to lack of insight as to what could be done with the information at hand.
3 Results

The results below were gathered by going over the records from all the interviews, and from texts about different ways to addresses issues with balancing and tips on how to balance a game. The headlines following 3.1 are categories based on patterns in the answers and information that arose from the gathered data. As mentioned in the method section the information gathered from both interviews and texts were then used according to IGT to find connections and use these as the base of the design for the new program. There is a bullet list following each paragraph, pointing out key information to allow for easier overview. All of the results will then be taken into consideration when making the conceptual design in the last paragraph of the chapter.

3.1 Interview Results

3.1.1 Character Abilities

Every single designer interviewed had experienced at least a few instances of abilities needing an upgrade (a buff) or a downgrade (a nerf) to make the game more balanced. In most cases, one or two abilities available to the player were so much stronger than the rest that it was used almost all the time, defeating the purpose of multiple abilities. During the interviews, tweaking the power of abilities was also frequently given as an answer to simplest problem to solve since the damage done by abilities is generally based on three or four things. First is the base damage of the ability, meaning the numeric value of the skill alone. Second is damage increase through stats given from buffs or equipment. Third, although it is not used in smaller games many of the times, is enemy damage reduction such as resistances and abilities of its own. Lastly is the cooldown, or the time before the ability can be used again. This does not directly channel into the damage done by the ability but rather its damage-per-second, or DPS. If a high damage ability is on a too low cooldown, it deals way more DPS than originally intended and either need its damage reduced or cooldown increased. Since it is known which of these factors are used it is very simple to increase and decrease the damage output by tweaking one or more of these.
values, as Alexander Pereswetof-Morath states in his interview “…once we had identified the numbers we needed to change it was just a question of iterating the process an almost painful amount of times.”.

- Abilities often need some balancing before working as intended.
- There are many ways to tweak abilities due to the large number of factors (e.g. base damage, stat increase, cooldown).
- Keeping track of which affecting variables are being used to make balancing easier.

### 3.1.2 Game and Character Balance

This part refers to how the games different stages, enemies, characters base stats and equipment bonuses scale throughout the game if it is a role-playing-game (RPG), or draw percentages and such in a card game. As the game progresses, all of these values need to stay in sync to ensure that the game does not lose its appeal. New items and abilities needs to be introduced at the right times so the player has the proper tools to progress and has a goal in sight. Balancing an entire game is a lot harder than it seems. In order to make a game balanced a lot of smaller issues have to be handled, and it is the same with a character within the game. If the stats on equipment for example would scale at a higher rate than the enemies it would soon result in the player not feeling threatened by them and begin to lose interest in the game, of course this goes the other way around as well if all of a sudden a boss would be far to strong or an ability that is needed to move forward have not yet been learned. During the interview with Gemma Thomson a special case was brought up. A game using 30 levels had been created and the designer being interviewed had partaken in the level design and balancing phase. When the game was near completion it was decided that 10 more levels would be implemented. After adding everything associated with the new levels the balancing would have to be done again in order to adjust the scale to the new top values of everything. This resulted in having to completely rework some levels as well as adding additional content in order to keep the balance as intended.
• Balancing the equipment is just as important as the character.
• Drop percentages are important to balance so that drops stay relevant throughout the game.
• If a problem is hard to solve, consider remaking entire stages or abilities.

3.1.3 Overview and Availability

Every designer interviewed had, to some extent, used some kind of tool to aid them during balancing. Most of the time it was either Excel or a handmade version of the same kind of spreadsheet to help them keep track of numbers during repeated playthroughs so that they would be able to see the what results they got in the game when tweaking certain numbers without having to keep it all in their heads. There is no end to the amount of information you can store, one document with different tabs for easier separation is enough to hold every variable used. The layout on both paper and digital spreadsheets enables overview of several subjects at once, making it easier to draw parallels and understanding how they affect other values in the game. The very simple implementation of math is also a big factor. On paper the designer in question will have to do the math associated with the different cells which can be a bit time consuming but with the layout it is easy to know when, what values, and what formula would have to be used to get the correct answer. Excel also makes it possible to assign a function to a cell using values from other cells and continuously updating the result if one of the base values were to be changed. If the designer knows what end result he or she wants in relation to other variables, this feature can sometimes eliminate problems before they even surface. Since the work is done either in a digital file that can be e-mailed, saved online or on a USB-stick, or on paper that can be scanned or copied and taken anywhere, it is available to every member of the team no matter where the work is being performed thus drastically increasing the number of opportunities given to solve a problem.

• Spreadsheets are the preferred method of many to keep track of relevant information during balancing.
• Overview of the information is highly regarded when repeatedly changing values.
• It is important to be able to work in different places without losing access to information.

3.2 Text Results

The texts that were read were not limited to purely academic publications like those mentioned earlier. There were also articles about why balancing is important and guides about how and what to focus on when balancing games. For the most part the texts agreed with what the designers pointed out such as base stats and the importance of eliminating overly powerful strategies.

The benefit of the texts is the increased depth to specific information, such as giving examples of how to calculate damage taken and given by both parties in a duel with all different stats taken into consideration. One thing that is brought up in many of the texts is the importance of knowing your audience. Before digging in on all the balancing it must be made clear if the game is targeting for example casual players, hard-core players or maybe even competitive players. Depending on this, the course taken with the balancing is different (DellaFave, 2013).

It is also mentioned that over the last decade or so, making truly balanced games has grown even harder due to popular demand (Burgun, 2014). Nowadays the audiences want more and more all the time and quickly go from game to game without bothering to give feedback to the developers. Due to this, many games released seem to be balanced to the majority of the players that will not explore every possibility, but when a dedicated core of players starts exploring they find issues. Often this will result in patching but if the developers have moved on to new projects it just might be left imbalanced.

Even texts about dynamic balancing, which practically is the opposite of this subject, provides insight to where the designers balancing is flawed and in need of improvements. An example of this is the use of repeated patterns instead of true randomization. When using repeated patterns it can at first appear to be random due to very long sequences. Actual randomization is tough to manage and tough to balance, but provides truly random behaviour that cannot be learned, whereas a sequence, no matter how long can be learned and thus exploited by dedicated players.
• Know your audience. Do not waste time perfecting content that does not appeal the players.
• Do not take shortcuts, every approach is flawed in some way so prioritize the needs.

3.3 Program Design

With all the results analysed from both texts and interviews, in accordance with IGT, it is now time to apply everything that was learned into the conceptual design of the program, this means that the functionality of this program is purely hypothetical and serves as inspiration to continued development in the same way as concept art is used by artists. In this case the conceptual design is a collection of “guidelines” and suggested functions that derived from all the previous information and as was mentioned in the introduction, no developing was done so it is entirely possible that there are better solutions to be found. These guidelines are presented and explained in the separate paragraphs below. As it stands, the results of this study is meant to be used as a base or starting point for developers that are looking to produce a tool for game development using all or some of these suggested functions.

3.3.1 Increase the accessibility of variables

First and foremost the programmer chooses which variables from the project should be able to be changed from the program. Next the chosen variables will be assigned to a category tab that the programmer or designer can create any number of and rename at will to easily keep variables apart to avoid mistakes, for example the variable “player1HP”, the health points of player one, would be appropriate to put under a tab named “Player1 Stats” so it will not be confused with say “player2HP” if it is a multiplayer game. This procedure will, apart from making it perfectly clear which variables are subject to change by the designers and what parts of the game they affect, completely remove the possibility of changing the wrong variable which could result in breaking the game, since it will not even be visible to the designer.
3.3.2 Linking variables for automatic balance

To further simplify the process the designer can also link variables in the program to one another and set a condition, for example the base value “player1HP” and “player2HP” should always be the same. Now, if one of the two is changed, but the other one is forgotten, a notification of this will pop up when the change is trying to be saved. The same principle can be used to keep a balance throughout the whole project; “boss1HP” for example could, by following this principle, always be three times that of the players HP no matter what value that might be. Through this process of linking a lot of things that would normally have to be checked over and over again for the duration of the project can now be set up according to the will of the designer and then left alone, effectively decreasing the time spent revisiting variables and thus the overall amount of work.

3.3.3 Implementing designer made calculations

Taking inspiration from Excel’s function to calculate formulas, a feature using a chosen set of variables and continuously calculating the result is made available. For example, if we first take a base enemy and calculate its DPS using the base damage and attacks per second, and then calculate the player’s effective HP, EHP, meaning the actual amount of damage that needs to be taken before defeat, using the base HP, bonus HP from equipment, and defensive stats from equipment. Using the two results from these calculations and linking those values as described above it is possible to make sure that the players EHP does not exceed a level where there is no longer any threat from the monsters DPS, thus eliminating any need to play with caution or to even stop and fight it at all. If the designer links the two with the condition that the base enemy’s DPS should be no lower than 5% of the players EHP, the same notifications as pointed out in 3.3.2 would appear when trying to save values that compromise this balance. Applying this could properly could detect imbalances in the game before it has even been play tested. Calculations like these can be hard to get to work properly if made entirely inside the game, and therefore do not get used very often despite their usefulness. By including a number of pre-set formulas in the program such as DPS, EHP, damage reduction etc., which only needs the designer to assign the
variables that are to be used, and allowing the designer to create their own formulas without the need of occupying the time of programmers, the versatility and depth of the game grows much larger.

### 3.3.4 Permanently change values while the game is running

Since the program runs simultaneously as the engine, values changed in the program will not be reset when stopping the game, and can then be saved onto the variables that were subject to change during the test. By allowing changes to be made to variables while the game is running, it enables easier pin pointed balancing, especially when testing drop rates or boss fights that usually comes at a later stage in the game. If one is able to test these things while the game is running it saves all the trouble of having to stop the playthrough, tweak the variables, and restart the game again. For example, in every PvE game, by default, there has to be a large number of enemies since the player fights against the environment around him/her, often of different races, but also of the same race with different settings. If we take an enemy, say a goblin, there can easily be three or four different stat settings. The standard setting for the goblin one can encounter alone on the plains. Then we have another setting if there is a pack of goblins, they have to be individually weaker because one have to fight several at the same time. Then there might be equivalents of these types but inside a boss dungeon, so they have to be stronger and thus two more settings arise. By being able to fight multiple instances of the same type as well as all the different types in succession in a single playthrough and changing values in between each fight without having to restart the game or search the hierarchy for the correct type of goblin makes it much easier keeping the difference in power relevant.

### 3.3.5 Enable work from different locations

The function to save the work currently being done in the program will create a file containing all the information about links, values, variables and tabs that can be mailed or stored on a USB-device. If one of the designers have been home sick for a couple of days and therefore does not have the latest information, simply send the current save-file that can then be loaded into his/her program and from there overwrite the values in the project just as one would do after a
playthrough. Since there are normally more than one designer balancing a project, and the work might be done on different locations, being able to move the information about changes without compromising anything is essential.

By using different combinations of all the tools available many of the problems encountered while balancing a game through play testing can be handled quicker and with a reduced risk of accidentally adding another issue.
4 Discussion

While working on this essay I came to understand that writing a study based on interviews was harder than previously anticipated. When reaching out on Facebook to gather designers I posted in a group consisting of more than 2000 members so I was pretty certain that I would be able to pick and choose the five or six that was most suitable. That was not the case. I only got four interested and had to interview them all to get sufficient material even though some had less experience or was working in a different field than what would have been ideal to the study. When compiling the results from the interviews I became aware that the questions I had asked were in some cases not open enough to get the answers I wanted, no doubt this was because of my inexperience with interviews, and that made the quality of the answers differentiate between the designers. Had this not been the case the results from the interviews would ideally have had more details to certain aspects of, for example, in what cases different solutions are suitable or not, and whether there is some predetermined course of action when balancing.

When reading the different texts I found that the diversity really helped the overall impression I got. Since the origins of the information were so different from one another it, strangely enough, made it easier to merge it into a design for the program that would be able to handle many of the problems from all those different points of view. Because of that the texts were in quite a few used more to complement the results from the interviews rather than providing their own solutions. When compiling the program design I would generally start thinking of a function based on the information from the interviews and then finish and polish it with information from the texts. As I explain in the method, if I had used traditional GT instead of IGT this would not have been possible.
5 Conclusions

The purpose of this study was to first establish what functions and features were necessary to have in a balancing tool using information from interviews and designer texts and to then merge this together to a design such a tool. The result was mostly focused around increased accessibility and overview through tabs and custom naming as well as a save/overwrite function, but new functions such as continuous calculations of formulas and linking of variables for almost automatic balancing of values, were also put in place to reduce time spent on iterations. Finally, the ability to change values while the game is running and apply these changes to the project afterwards can heavily reduce the number of playthroughs needed to achieve balance.

Since the result of this study is a conceptual design for a program and not a functioning program, that alone presents a good opportunity for future research. With the right knowledge this could be made into reality and tested for further improvements or if this is even a viable solution. Besides actually putting this information into practice one can always continue to come up with additional features that would be nice to have and more intricate ways of using math and linking variables and such. As explained in the research question segment, this study was focused on PvE games, so another possible direction for further research would be to try and implement the complex nature of “Player versus Player”, PvP. This is exponentially harder since both parties involved possess the creative mind of a human rather than one of them being programmed and predictable, thus eliminating a lot of possibilities.

By its very nature I feel like this program would be of most use to small or medium sized games rather than AAA-titles because that is where the amount of work hours needed for good balancing cannot always be put in due to lack of staff or even economical means as well as the possibility that AAA-teams in many cases would rather develop their own tool that is tailored for their game. At the same time this would be a third-party software and thus having to be bought for a certain sum of money, making it maybe not as appealing as it could have been.
to the smaller games that might need it the most as it would be an additional spending.
6 References


7 Appendixes

7.1 Interview Structure

For how long have you been working with game development?

What is your experience with game balancing?

Do you use anything to aid you during balancing (i.e. Excel, spreadsheets etc.)?
   How do you use this?

What was the latest issue you worked on?
   How did you solve this issue?

What is, according to you, the hardest and the easiest problem in balancing?
   What solutions do you use for these?

Which problems do you consider being the most common and the rarest?
   How would you solve these?

If the same problem occurs repeatedly, do you follow the same course to solve it again?

Have you ever used programs/plugins/tools specifically made for balancing?
   Which one/ones?
   What did you think was good respectively bad about this?

If such a tool was available, would you consider using it?

What would such a tool need to be of use to use during balancing?